



無料電子ブック

学習

R Language

Free unaffiliated eBook created from
Stack Overflow contributors.

#r

.....	1
1: R	2
.....	2
R Docs	2
R	2
Examples	2
R	2
Windows	2
Windows	2
OSX / macOS	2
1	3
2	3
DebianUbuntu	3
Red HatFedora	3
Archlinux	4
""	4
.....	4
R	4
.....	4
R	4
.....	6
R	7
2: *	9
.....	9
*apply	9
Examples	9
.....	9
.....	11
`data.frames` `lapply` `mapply`	11
functionals	12

lapplysapplymapply	12
lapply	13
.....	13
mapply	13
.....	13
.....	13
3: .Rprofile	16
.....	16
Examples	16
.Rprofile -	16
R	16
.....	16
.....	16
.....	16
.....	16
CRAN	16
.....	17
.....	17
.....	17
.....	17
.....	17
.Rprofile	17
.....	17
.....	18
.....	18
4: ANOVA	19
Examples	19
aov	19
Anova	19
5: data.table/	21
.....	21
.....	21

.....	21
Examples.....	21
data.table/ - I.....	21
data.table - II.....	23
6: Date.....	25
.....	25
.....	25
.....	25
.....	25
Examples.....	25
.....	25
.....	26
.....	28
7: devtools.....	29
.....	29
.....	29
Examples.....	29
.....	29
.....	29
.....	30
.....	30
1.....	30
2.....	31
.....	31
.....	31
Github.....	31
CRAN.....	31
.....	32
.....	32
.....	32
8: dplyr.....	33

.....	33
Examples.....	33
dplyr.....	33
.....	33
.....	34
.....	35
.....	36
.....	37
.....	38
group_by.....	38
together.....	38
.....	39
.....	41
dplyr::filter() -	41
dplyr::distinct() -	42
>.....	43
dplyrNSE.....	43
9: ggplot2.....	45
.....	45
Examples.....	45
.....	45
.....	46
.....	50
.....	52
1.....	52
1.....	52
.....	52
.....	52
.....	52
.....	52
qplot.....	56

10: GPU	58
.....	58
Examples	58
gpuR gpuMatrix	58
gpuR vclMatrix	58
11: hclust	60
.....	60
.....	60
Examples	60
1 - hclust	60
2 - hclust	64
12: igraph	67
Examples	67
.....	67
13: JSON	69
Examples	69
JSONR/	69
14: LaTeXR with knitr	71
.....	71
.....	71
.....	71
Examples	72
R	72
KnitrR	72
KnitrLaTeXR	73
15: R MarkdownRStudio	74
.....	74
Examples	74
.....	74
.....	75
.....	76
.....	

.....	77
.....	77
.....	78
.....	79
16: Rcpp.....	81
Examples.....	81
.....	81
Rcpp.....	81
Rcpp.....	82
.....	83
17: RESTfulR.....	84
.....	84
Examples.....	84
opencpu.....	84
18: RMarkdownknitr.....	85
.....	85
.....	85
.....	85
.....	85
Examples.....	89
Rstudio.....	89
ioslides.....	90
19: RMarkdown.....	93
Examples.....	93
.....	93
LaTeX.....	95
.....	96
R.....	97
R.....	97
R-markdown.....	97
R.....	98

20: RMD	100
.....	100
.....	100
Examples	101
.....	101
.....	102
.....	103
21: RODBC	106
Examples	106
RODBCExcel	106
SQL Server	106
.....	106
22: R	107
Examples	107
.....	107
23: RODE	111
.....	111
.....	111
.....	111
Examples	111
.....	111
.....	113
ODE - R	114
ODE - C	115
ODE - Fortran	116
ODE -	118
24: R	120
.....	120
Examples	120
.....	120
R	120
.....	121

25: R	122
Examples	122
Windows	122
26: R	124
	124
Examples	124
S3	124
27: R	126
Examples	126
	126
	127
RStudio	127
	128
28: RWeb	130
Examples	130
RCurl	130
29: RI / O	131
Examples	131
RdsRDataRda	131
	131
30: R -	133
Examples	133
	133
NA	133
	133
31: R	135
	135
Examples	135
`grep`	135
32: R	137
	137

Examples.....	137
R.....	137
installR.....	137
.....	138
.....	141
R.....	142
33: R.....	143
Examples.....	143
PDFPMFR.....	143
34: S4.....	144
.....	144
Examples.....	144
.....	144
35: sqldf.....	157
Examples.....	157
.....	157
36: stringi.....	159
.....	159
Examples.....	159
.....	159
.....	160
.....	160
.....	160
37: strsplit.....	162
.....	162
Examples.....	162
.....	162
38: texreg.....	164
.....	164
.....	164
.....	164

Examples.....	164
.....	164
39: Web.....	166
.....	166
.....	166
Examples.....	166
rvest.....	166
rvest.....	167
40: xgboost.....	169
Examples.....	169
xgboost.....	169
41:	172
Examples.....	172
.....	172
42:	174
.....	174
Examples.....	174
.....	174
43:	175
.....	175
.....	175
Examples.....	175
.....	175
.....	175
.....	176
44:	178
Examples.....	178
viridis -	178
RColorBrewer.....	181
.....	183
colorspace -	184

R.....	185
.....	186
45:	189
Examples.....	189
.....	189
proc.time.....	189
.....	190
.....	191
.....	192
46:	194
Examples.....	194
.....	194
.....	194
.....	194
.....	195
.....	195
.....	195
47:	196
.....	196
.....	196
Examples.....	197
.....	197
.....	198
.....	200
.....	201
.....	201
.....	203
.....	204
.....	204
.....	205
48:	207
Examples.....	207

.....	207
1.....	207
2.....	207
ui.R.....	207
server.R.....	208
.....	208
.....	208
.....	209
.....	210
1. 2app.....	210
2. 1.....	211
.....	211
.....	213
.....	213
.....	213
49: R.....	215
.....	215
.....	215
Examples.....	215
R.....	215
R.....	215
R.....	216
Linux / Mac.....	216
Windows.....	216
littlerR.....	217
littler.....	217
R.....	217
apt-getDebianUbuntu.....	217
litr.r.....	217
littler.....	218
50: APISparkR.....	219
.....	

Examples.....	219
.....	219
RSpark.....	219
.....	219
.....	219
RDD.....	220
.....	220
csv.....	220
51:	221
.....	221
Examples.....	221
.....	221
.....	221
52:	223
.....	223
Examples.....	223
data.frame.....	223
.....	224
[[\$.....	224
data[rows, columns].....	225
.....	225
.....	225
.....	225
.....	226
.....	226
data[columns].....	226
data[[one_column]].....	227
\$.....	227
\$.....	227
.....	228
.....	228

.....	228
data.frames.....	228
.....	228
.....	229
.....	229
.....	229
do.call.....	230
data.frame.....	231
.....	232
53:	233
.....	233
Examples.....	233
R.....	233
dplyr.....	234
data.table.....	235
54: I / O.....	237
.....	237
.....	237
Examples.....	237
MySQL.....	237
.....	237
.....	237
MongoDB.....	237
55:	239
.....	239
Examples.....	239
.....	239
.....	239
.....	240
Gapminder.....	241
2015 -	242
.....	

.....	243
.....	244
.....	244
56:	247
.....	247
.....	247
.....	248
.....	248
.....	249
.....	249
Examples.....	249
data.table	249
.....	249
.....	249
data.frame	250
data.table	250
.....	250
.....	251
.....	251
.....	252
data.table	252
SD	252
.SDcols	253
N	253
data.framedata.table	254
.....	254
data.framedata.table	255
data.table	256
57:	259
Examples.....	259

NWord.....	259
58:	263
Examples.....	263
.....	263
.....	264
59: >	265
.....	265
.....	265
.....	265
.....	265
%>%.....	265
.....	265
.....	266
.....	266
Examples.....	266
.....	266
.....	267
<>.....	268
\$.....	268
dplyrggplot2.....	269
T>.....	269
60:	271
.....	271
.....	271
.....	271
.....	271
.....	271
.....	271
Examples.....	271
.....	271
.....	272
.....	272
.....	272

.....	272
.....	272
.....	273
.....	273
.....	275
61:	276
.....	276
.....	276
.....	276
.....	276
Examples.....	276
.....	276
CRAN	276
Bioconductor	277
.....	277
GitHub.....	278
CLI -	279
.....	280
62:	282
.....	282
.....	282
Examples.....	282
.....	282
.....	282
.....	282
.....	282
.....	283
63:	284
Examples.....	284
.....	284
.....	284
.....	

.....	285
.....	285
.....	286
.....	287
.....	288
listenv	289
64: .2	290
Examples	290
.....	290
::	290
1	290
2	291
3	291
4reorder	292
5 NO reorder	293
6	294
7	295
8cor	296
2	298
65:	304
.....	304
Examples	304
.....	304
66: /	312
.....	312
.....	312
tryCatch	312
.....	312
.....	312
Examples	313

tryCatch.....	313
tryCatch.....	313
.....	314
.....	314
67:	316
.....	316
.....	316
Examples.....	316
.....	316
Matplot.....	319
.....	325
.....	327
par().....	327
layout().....	328
.....	329
.....	331
R_Plots.....	332
68:	334
Examples.....	334
.....	334
seq.....	334
.....	335
.....	337
rep.....	338
.....	339
69:	341
.....	341
.....	341
Examples.....	341
boxplot{graphics}box-and-whisker.....	341
boxplotSepal.Length.....	342
.....	342

.....	343
.....	344
.....	345
.....	345
.....	346
boxplotsplot=FALSE	346
.....	347
.....	347
.....	347
.....	347
.....	347
.....	348
.....	348
70:	350
.....	350
Examples.....	350
.....	350
.....	350
.....	350
.....	350
.....	350
.....	351
.....	351
71:	352
.....	352
Examples.....	352
R.....	352
72: I / O	353
.....	353
Examples.....	353
.....	

73:	355
.....	355
Examples.....	355
GLCM.....	355
.....	357
74:	359
.....	359
Examples.....	359
-	359
75:	361
.....	361
.....	361
Examples.....	361
.....	361
.....	361
.....	362
.....	362
.....	362
.....	364
76:	365
.....	365
.....	365
Examples.....	365
`rle`.....	365
R.....	365
data.table.....	366
.....	367
77:	369
.....	369
Examples.....	369
.....	

78:	371
Examples	371
.....	371
.....	373
.....	373
.....	374
.....	375
79: 2	377
.....	377
Examples	377
roxygen2	377
roxygen2	377
.....	378
80:	379
Examples	379
.....	379
81:	382
.....	382
Examples	382
foreach	382
.....	383
.....	384
mcpipelineDo	385
.....	385
.....	385
82:	386
Examples	386
.....	386
.....	386
.....	387

010.....	387
01.....	387
100.5.....	387
0.2.....	387
3105.....	388
100.8.....	388
2.....	388
1.5.....	388
01.....	388
152.....	388
a = 1b = 0.5.....	388
3= 0.5.....	388
01.....	389
01.....	389
0.51.....	389
1020Wilcoxon.....	389
53.....	389
83:	390
.....	390
Examples.....	390
.....	390
.....	390
84: R	392
.....	392
Examples.....	392
.....	392
.....	392
.....	392
.....	392
.....	392
.....	392

.....	393
.....	393
85:	395
.....	395
.....	395
Examples	395
.....	395
NA	395
NAs	396
TRUE / FALSE/NA	396
.....	397
.....	397
.....	397
.....	398
86:	399
.....	399
Examples	399
.....	399
.....	399
.....	399
.....	400
87: R	401
.....	401
.....	401
.....	401
Examples	401
.....	401
dput()dget()	401
.....	401
88:	403
.....	403

.....	403
Examples.....	403
.....	403
.....	403
.....	403
.....	403
.....	404
.....	404
89:	405
Examples.....	405
.....	405
split-apply-combinesplit.....	407
90:	409
.....	409
.....	409
Examples.....	409
.....	409
.....	409
91:	414
Examples.....	414
.....	414
92:	416
.....	416
For	416
.....	417
Examples.....	417
.....	417
For.....	418
for.....	418
.....	419
vapply.....	419

colMeans.....	419
.....	419
whilerepeat.....	419
while.....	419
repeat.....	420
break.....	421
93:	423
Examples.....	423
Rcpp.....	423
.....	423
94:	426
Examples.....	426
.zip.....	426
.zip.....	426
.tar.....	426
.tar.....	426
.zip.....	427
95: I / O	428
.....	428
Examples.....	428
.....	428
96:	429
.....	429
Examples.....	429
map.....	429
Google Viz50.....	433
.....	434
HTML.....	436
Shiny.....	438
97:	440
Examples.....	440
.....	440

.....	441
.....	441
.....	441
.....	442
.....	442
.....	442
""	443
98:	444
.....	444
Examples	444
.....	444
.....	445
.....	445
.....	446
.....	446
.....	447
99: ExcelSASSPSSStatal / O	448
Examples	448
.....	448
Excel	448
xlsxExcel	449
XLconnectExcel	449
openxlsxExcel	450
Readingreadxl	450
ReadingRODBC	451
Readinggdata	452
StataSPSSSAS	452
.....	453
100:	455
Examples	455
.....	455

.....	456
101: + ...	458
.....	458
Examples.....	458
.....	458
102:	459
.....	459
Examples.....	459
.....	459
103:	460
.....	460
Examples.....	460
.....	460
.....	460
.....	460
.....	461
.....	461
.....	461
//.....	462
/.....	462
104:	464
Examples.....	464
.....	464
105:	466
Examples.....	466
tbl_df.....	466
tidyverse.....	466
tidyversetidyverse.....	466
.....	467
.....	467
106:	469

.....	469
Examples.....	469
spread.....	469
gather.....	470
h21.....	470
107:	471
.....	471
.....	471
.....	471
Examples.....	471
.....	471
108:	472
.....	472
Examples.....	472
.....	472
.....	474
.....	474
.....	474
.....	475
109:	476
.....	476
.....	476
.....	476
.....	476
.....	477
Examples.....	477
.....	477
.....	477
.....	478
.....	478
110: POSIXctPOSIXlt.....	480
.....	

.....	480
.....	480
.....	480
.....	480
Examples.....	480
.....	480
.....	481
.....	481
.....	481
.....	481
.....	482
111:	483
.....	483
Examples.....	483
.....	483
ts.....	484
112:	486
.....	486
Examples.....	486
ArimaAR1.....	486
113:	495
.....	495
Examples.....	495
barplot.....	495
114:	503
Examples.....	503
.....	503
115:	504
Examples.....	504
.....	504

116:	505
.....	505
.....	505
.....	505
.....	505
.....	505
.....	505
.....	505
Examples.....	506
.....	506
.....	506
.....	506
.....	507
.....	507
YYYYMMDD.....	507
.....	508
.....	508
R.....	509
PerlPOSIX.....	509
/.....	510
117:	511
.....	511
.....	511
Examples.....	511
.....	511
.....	512
Datetimes	512
.....	512
.....	513
lubridate.....	513
lubridate.....	514
.....	

.....	514
.....	516
.....	517
.....	517
118: <>	518
.....	518
Examples	518
.....	518
119:	519
Examples	519
randomForestSRC	519
-	520
Kaplan Meier	521
120:	524
Examples	524
XY	524
.shp	525
rgdal	525
.....	525
tmap	526
121:	527
.....	527
Examples	527
.....	527
.....	527
122:	530
.....	530
.....	530
Examples	531
mtcars	531

.....	532
.....	533
.....	535
.....	537
'predict'.....	539
123:	541
.....	541
Examples.....	541
.....	541
124:	543
.....	543
Examples.....	543
.....	543
125: CSVTSV	545
.....	545
.....	545
.....	545
Examples.....	546
.csv.....	546
R	546
.....	546
.....	546
data.table.....	547
.....	548
.tsvR.....	548
.csv.....	549
R	549
.....	549
csv.....	549
.....	549
R	549

readr	550
126:	551
.....	551
.....	551
.....	552
.....	552
Examples.....	553
.....	553
.....	554
factor factor_approach	554
ifelse ifelse_approach	555
list_approach	555
.....	555
.....	556
.....	557
.....	562
.....	562
.....	562
127:	564
.....	564
.....	564
.....	564
Examples.....	564
.....	564
.....	564
NA.....	565
128:	566
.....	566
.....	566
.....	566
Examples.....	566
.....	

.....566

.....567

.....567

R.....568

tidyr.....568

data.table.....569

129: 570

 Examples.....570

 570

130: 571

 571

 Examples.....571

 dplyr.....571

131: 573

 Examples.....573

 573

.....576

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [r-language](#)

It is an unofficial and free R Language ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official R Language.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: Rのい

スタックオーバーフローでR Docsをする

ドキュメントをするののドキュメントガイドラインをしてください。

のからしたRののいくつかは、しい

- のとはなり、Rのはをとしません。
- じには、にじて、なるでなるデータをりてることができます。
- ベクトルとリストのインデックスは、0ではなく1からまります。
- R arrays およびのなは、をたないRの「ベクトル」とはにdimをとっています。
- Rのリストをすると、1つのつまりリストのでさまざまなオブジェクトをどおりにできます。これらのオブジェクトは、 、ベクトル、データフレーム、さらにはのリストなどであってもよい。これらのオブジェクトがいにはしているはない。
- リサイクル
- がありません

Examples

Rのインストール

RStudioをインストールしたにRStudioをインストールすることもできます。RStudioはくのプログラミングをするRのです。

Windowsのみ

Visual Studio バージョン2015 Update 3には、ライブインタープリタ、IntelliSense、およびデバッグモジュールをむR Toolsというがされました。このをした、のセクションでされているようにRをインストールするはありません。

Windowsの

1. CRANの Webサイトにアクセスし、R for Windowsをダウンロードし、Rのバージョンをダウンロードします。
2. インストーラファイルをクリックし、としてします。
3. インストールにするをします。
4. インストールのにしてください。

OSX / macOSの

1

0. [XQuartz](#)がインストールされていることをしてください

1. [CRAN](#)のウェブサイトనికి、Rのバージョンをダウンロードしてください。
2. ディスクイメージをき、インストーラをします。
3. インストールのにってください。

これにより、RとMacGUIのがインストールされます。 GUIを/ Applications / FolderにR.appとしてします。ここで、ダブルクリックするか、Docにドラッグすることができます。しいバージョンがリリースされると、インストールプロセスはR.appをきしますが、のメジャーバージョンのRはされます。のRコードは/Library/Frameworks/R.Framework/Versions/ディレクトリにあります。RStudioでRをすることもで、のGUIでじRコードをします。

2

1. <https://brew.sh/>のにって、homebrewmacOSのパッケージマネージャーがないをインストールします。
2. `brew install R`

2のをするは、Macフォークのメンテナーがアドバイスし、R-SIG-Macメーリングリストのしさにするにはしないことにしてください。

Debianの、Ubuntuとその

`apt-get` ってあなたのディストリビューションにするRのバージョンをすることができます。しかし、このバージョンは、CRANでなバージョンよりもかなりれていることがよくあります。された「ソース」リストにCRANをすることができます。

```
sudo apt-get install r-base
```

ソースリストにCRANをすることで、CRANからのバージョンをすることができます。については、CRANのにってください。にこれをして `install.packages()` できるようにするがあることにしてください。Linuxパッケージは、ソースファイルとしてされ、コンパイルがです。

```
sudo apt-get install r-base-dev
```

Red HatとFedoraの

```
sudo dnf install R
```

Archlinuxの

Rは`Extra`パッケージレポでできます。

```
sudo pacman -S r
```

ArchlinuxでRをするのは、 [ArchWiki Rページ](#)をしてください。

"こんにちは"

```
"Hello World!"
```

また、 [この、の、の](#)についてのなもしてください。

ヘルプの

`help()` または `?ドキュメント` にアクセスし、Rのヘルプをします。さらになをうには、
`help.search()` または `??` 。

```
#For help on the help function of R
help()

#For help on the paste function
help(paste)      #OR
help("paste")    #OR
?paste           #OR
?"paste"
```

については、 <https://www.r-project.org/help.html> をしてください。

インタラクティブモードとRスクリプト

インタラクティブモード

Rをするもなはモードです。あなたはコマンドをし、すぐにをRからします。

Rをしてう

のようにしてRをRごのオペレーティング・システムのコマンド・プロンプトで、またはすることにより、 `RGui Windows` で。は、LinuxのインタラクティブなRセッションのスクリーンショットです。
。

```
user:~$ R

R version 3.3.2 (2016-10-31) -- "Sincere Pumpkin Patch"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

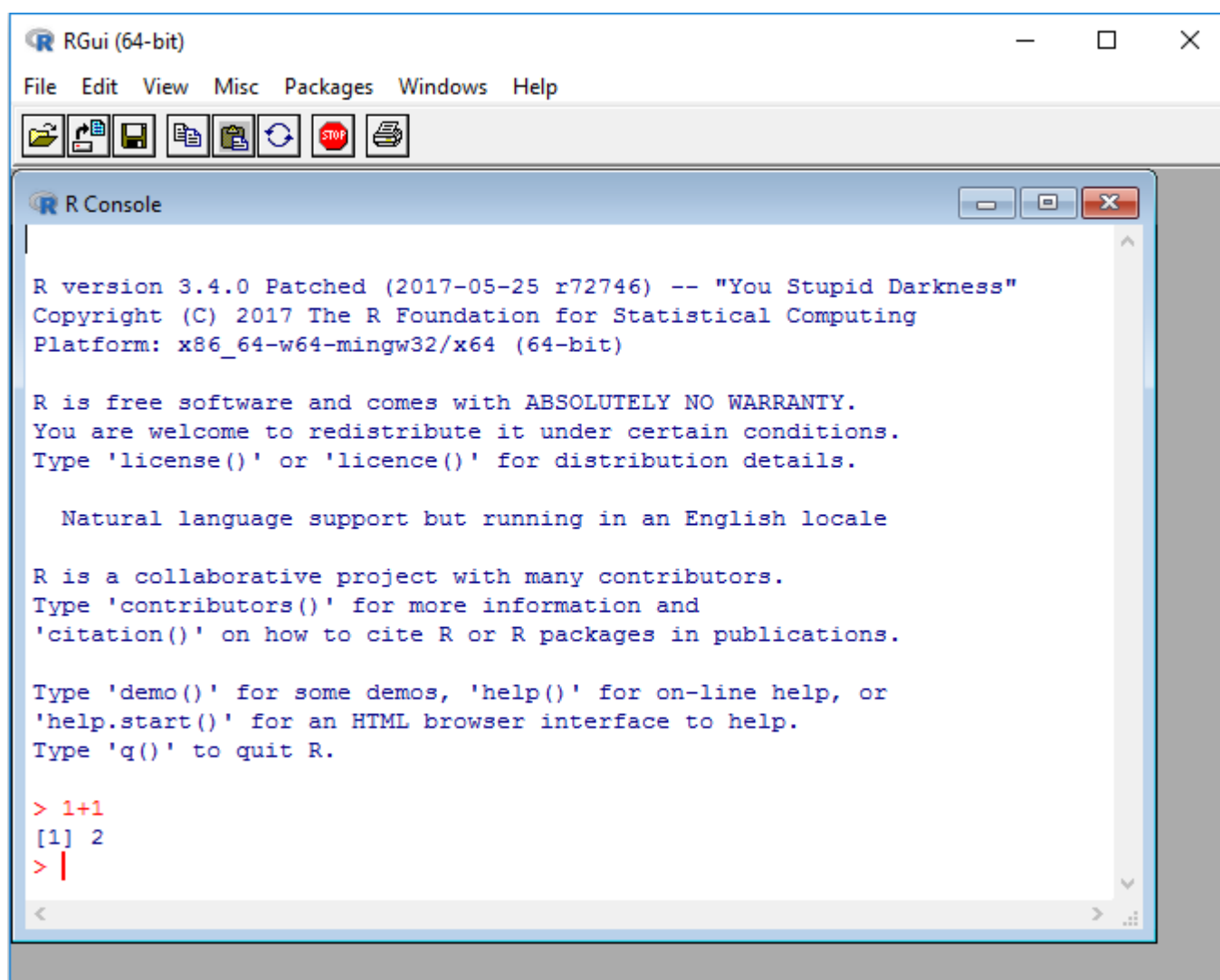
R ist freie Software und kommt OHNE JEGLICHE GARANTIE.
Sie sind eingeladen, es unter bestimmten Bedingungen weiter zu verbreiten.
Tippen Sie 'license()' or 'licence()' für Details dazu.

R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Information und 'citation()',
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.

Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.

> 1+1
[1] 2
> █
```

これは、WindowsのRのものであるWindowsのRGuiです。



のなにをすることができます。がされると、はRでされます。のスクリーンショットでは、Rがとしてされています。

```
1+1
```

をすぐにするには、2。の`[1]`は、Rがベクトルをすことをします。この、ベクトルは1つの2しかみません。

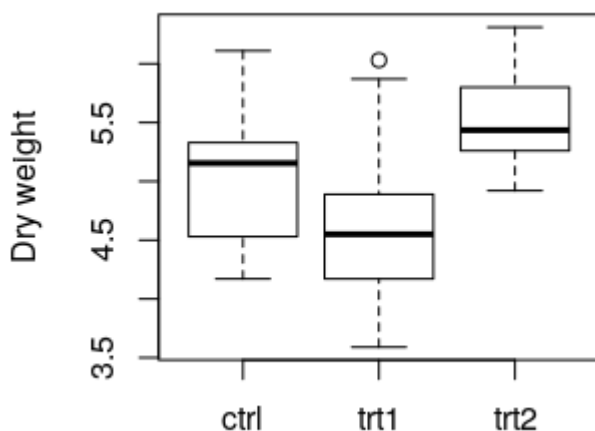
のプロット

Rをしてプロットをすることができます。のでは、データセット`PlantGrowth`しています。このデータセットは、R

##まっていないRプロンプトにのすべてのをしてください。##まるは、Rがすをするためのものです。

```
data(PlantGrowth)
str(PlantGrowth)
## 'data.frame':    30 obs. of  2 variables:
## $ weight: num  4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...
## $ group : Factor w/ 3 levels "ctrl","trt1",...: 1 1 1 1 1 1 1 1 1 1 ...
anova(lm(weight ~ group, data = PlantGrowth))
## Analysis of Variance Table
##
## Response: weight
##          Df Sum Sq Mean Sq F value    Pr(>F)
## group      2  3.7663   1.8832   4.8461 0.01591 *
## Residuals 27 10.4921   0.3886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
boxplot(weight ~ group, data = PlantGrowth, ylab = "Dry weight")
```

のプロットがされます。



`data(PlantGrowth)`は、2つのなるまたはくされていないのである、サンプルデータセット`PlantGrowth`ロードする。データセットは`PlantGrowth`というでできます。このようなはともばれます。

のデータをみむには、の2つのマニュアルページがちます。

- のデータをブレンテキストファイルCSV、TSVなどでみきする
- テーブルExcel、SAS、SPSS、StataのI/O

`str(PlantGrowth)`は、ロードされたデータセットにするをします。は、`PlantGrowth`がのRのである `data.frame`であることをします。 `data.frame`には2つのと30のがまれています。この、は1つのプラントにします。2つののは、\$まるにされています。のは `weight` とばれ、 `num`、それぞれのプラントのをみます。2の `group`には、プラントがけたがまれています。これはRの `factor` とばれるカテゴリデータです。 [データフレームのをしてください](#)。

3つのなるのをするために、 `anova(lm( ... ))`をいてANOVAをう。 `weight ~ group`「ののを `weight`、カラムのによってグループし、 `group`」。Rのとばれます。 `data data = ...`は、データがつかるテーブルのをします。

は、とりわけ、3つのうちのいくつかのに `Column Pr(>F)`、 `p = 0.01591` がすることをす。Tukey's Testのようなポストホックテストは、どのグループのがになるかをするためにするがあります。

`boxplot(...)`は、データのボックスプロットをします。プロットされるはどこからののか。 `weight ~ group`「ののにしてみのをプロット `group`。 `ylab = ... y`のラベルを [ベースは、プロット](#)

Rセッションをするには、 `q()`または `Ctrl - D`をします。

R スクリプト

をするには、にするコマンドをファイルにすることをおめします。そののために、 **R** スクリプトをすることができます。R スクリプトは、R コマンドをむシンプルなテキストファイルです。

`plants.R`というのテキストファイルをし、のコードブロックにれているのテキストをします。

```
data(PlantGrowth)

anova(lm(weight ~ group, data = PlantGrowth))

png("plant_boxplot.png", width = 400, height = 300)
boxplot(weight ~ group, data = PlantGrowth, ylab = "Dry weight")
dev.off()
```

あなたのターミナルにしてスクリプトをしてくださいのセクションのようなインタラクティブなRセッションではなく、オペレーティングシステムのターミナルです。

```
R --no-save <plant.R >plant_result.txt
```

`plant_result.txt` ファイルには、のRプロンプトにしたかのように、がまれています。これにより、がされます。

しいコマンド `png` と `dev.off` は、 `dev.off` をディスクにするためにされます。のにすように、2つのコマンドはプロットコマンドをむがあります。 `png("FILENAME", width = ..., height = ...)` は、されたファイル、さをピクセルでつしいPNGファイルをきます。 `dev.off()` はプロットをし、プロットをディスクにします。 `dev.off()` がびされるまで、はされません。

オンラインでRのいをむ <https://riptutorial.com/ja/r/topic/360/rのい>

2: *ファミリーをする

*apply ファミリーのは、forループの。forループとの*applyにはのがあります。

1. きむコードがなくてみます。
2. カウンタはありません。
3. をするためにをしません。

しかしforループはよりであり、*applyをすることはずしもではないなをするために、よりくのをえることが*applyます。

forループと*applyのについては、forループのドキュメントをfor。

*applyのメンバー

*applyファミリーには、にされるのについてなる、じののバリエーションがまれています。

apply	matrix、data.frame、またはarray	ベクトルまたはされるのさにじて
sapply	ベクトルまたはlist	ベクトルまたはされるのさにじて
lapply	ベクトルまたはlist	list
vapply	ベクトルまたは`list	ベクトルまたはされるのさにじてユーザーのクラス
mapply	のベクトル、listsまたはみわせ	list

これらののについては、「」をしてください。

Examples

にをする

applyは、やのマージンのおそらくのをするためにわれapply。

このアイデアをするためにirisデータセットをしましょう。irisデータセットは、3から150のをしています。このデータセットのをてみましょう

```
> head(iris)

Sepal.Length Sepal.Width Petal.Length Petal.Width Species
```

1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa

さて、あなたはこれらののそれぞれのをりたいことをしてください。これをする1つののは、`for`ループをすること `for` が、Rプログラマはしばしば `apply` を `apply` ことをみ `apply` はをしてください。

```
> apply(iris[1:4], 2, mean)

Sepal.Length Sepal.Width Petal.Length Petal.Width
    5.843333    3.057333    3.758000    1.199333
```

- のパラメータでは、`mean` はデータにしてのみするため、の4のみをめるように `iris` をサブセットしています。
- 2のパラメータ `2` 々はのCアレイ `x` Rののえカラムですことをしています。 `1` はをえる。

に、よりのあるをすることができます。

```
# standard deviation
apply(iris[1:4], 2, sd)
# variance
apply(iris[1:4], 2, var)
```

Rには、とのをするのにしたみみがいくつかあり `rowMeans`。 `colMeans`、 `rowMeans` と `rowMeans` をします。

ここでは、よりのあるのタスクをしましょう `0.5` よりきいにしてのみをしましょう。そのために、`mean` はの `mean` をします。

```
> our.mean.function <- function(x) { mean(x[x > 0.5]) }
> apply(iris[1:4], 2, our.mean.function)

Sepal.Length Sepal.Width Petal.Length Petal.Width
    5.843333    3.057333    3.758000    1.665347
```

`Petal.Width` ののいにしてください

しかし、コードのりののでこのをしたくないはどうすればよいでしょうかに、をして、のようなコードをくことができます

```
apply(iris[1:4], 2, function(x) { mean(x[x > 0.5]) })
```

これまでてきたように、`apply` をして、1だけをしてデータセットのまたはにしてじをすることができます。

`apply` はされたののさにじてになるをすので、にしていけないにはなではないかもしれません。の `*apply` のいくつかは、もうしです。

バルクファイル読み込み

のプロセスでにされたファイルでされるのあるのファイルについては、

まず、アクセスするファイルのベクトルを作るがあります。これにはのオプションがあります。

- `paste0()` でベクトルを作る

```
files <- paste0("file_", 1:100, ".rds")
```

- `list.files()` をファイルタイプのものである、じタイプののファイルがディレクトリにある、の
がです。

```
files <- list.files("./", pattern = "\\..rds$", full.names = TRUE)
```

`x`は、されるファイルののベクトルです。

`lapply`はをリストのとしてします。

`readRDS`は、`.rds` ファイルにであり、プロセスのにじてされます。

```
my_file_list <- lapply(files, readRDS)
```

これはテストからのforループよりずしもではありませんが、すべてのファイルをにりてずにリストのにすることができます。

に、のパッケージをにみむがあることがよくあります。このトリックは、たちがインポートしたいすべてのライブラリに`library()` をすることで、にうことができます

```
lapply(c("jsonlite", "stringr", "igraph"), library, character.only=TRUE)
```

の`**data.frames**` `lapply`、`mapply`をみわせると、

ここでは、4つのブートストラップモデルをし、これらのモデルのサマリーを1つのデータフレームにします。

```
library(broom)

#* Create the bootstrap data sets
BootData <- lapply(1:4,
  function(i) mtcars[sample(1:nrow(mtcars),
                             size = nrow(mtcars),
                             replace = TRUE), ])

#* Fit the models
Models <- lapply(BootData,
  function(BD) lm(mpg ~ qsec + wt + factor(am),
                  data = BD))
```

```

#* Tidy the output into a data.frame
Tidied <- lapply(Models,
                 tidy)

#* Give each element in the Tidied list a name
Tidied <- setNames(Tidied, paste0("Boot", seq_along(Tidied)))

```

これで、をdata.frameにするには2つのがあります。

```

#* Insert the element name into the summary with `lapply`
#* Requires passing the names attribute to `lapply` and referencing `Tidied` within
#* the applied function.
Described_lapply <-
  lapply(names(Tidied),
         function(nm) cbind(nm, Tidied[[nm]]))

Combined_lapply <- do.call("rbind", Described_lapply)

#* Insert the element name into the summary with `mapply`
#* Allows us to pass the names and the elements as separate arguments.
Described_mapply <-
  mapply(
    function(nm, dframe) cbind(nm, dframe),
    names(Tidied),
    Tidied,
    SIMPLIFY = FALSE)

Combined_mapply <- do.call("rbind", Described_mapply)

```

あなたがmagrittrスタイルパイプのファンであれば、モデルオブジェクトのようなオブジェクトのどれかがなはそうするのはではありませんがタスクをのチェーンでできます。

```

library(magrittr)
library(broom)
Combined <- lapply(1:4,
                  function(i) mtcars[sample(1:nrow(mtcars),
                                             size = nrow(mtcars),
                                             replace = TRUE), ]) %>%
  lapply(function(BD) lm( mpg ~ qsec + wt + factor(am), data = BD)) %>%
  lapply(tidy) %>%
  setNames(paste0("Boot", seq_along(.))) %>%
  mapply(function(nm, dframe) cbind(nm, dframe),
         nm = names(.),
         dframe = .,
         SIMPLIFY = FALSE) %>%
  do.call("rbind", .)

```

みみfunctionalsの

みみlapply、sapply、mapply

Rにはがみまれています。そのでもっともよくられているものはのファミリーです。もなのいくつかについてにします。

- `lapply()` = としてリストをとり、されたをリストにします。
- `sapply()` と `lapply()` が、ベクトルまたはにしようとしています。
 - `vapply()` = オブジェクトのをするがある `sapply()`。
- `mapply()` = のような `lapply()` が、されたへのとしてのベクトルをすることができます。 `sapply()` ようにすることができます。
 - `Map()` は、 `SIMPLIFY = FALSE` `mapply()` へのエイリアスです。

lapply

`lapply()` は、2つのなるでできます。

- `lapply(variable, FUN)`
- `lapply(seq_along(variable), FUN)`

```
# Two ways of finding the mean of x
set.seed(1)
df <- data.frame(x = rnorm(25), y = rnorm(25))
lapply(df, mean)
lapply(seq_along(df), function(x) mean(df[[x]]))
```

サブリ

`sapply()` は、そのをベクトルまたはにしようとしています。

```
# Two examples to show the different outputs of sapply()
sapply(letters, print) ## produces a vector
x <- list(a = 1:10, beta = exp(-3:3), logic = c(TRUE, FALSE, FALSE, TRUE))
sapply(x, quantile) ## produces a matrix
```

mapply

`mapply()` じように `lapply()` しますが、としてのベクトルをすることができますのは `m`。

```
mapply(sum, 1:5, 10:6, 3) # 3 will be "recycled" by mapply
```

ユーザーファンクショナルの

ユーザーファンクショナル

ユーザーはさまざまなのをすることができます。のは、Hadley Wickhamの[Functionals](#)からのものです。

```
randomise <- function(f) f(runif(1e3))
```

```
lapply2 <- function(x, f, ...) {
  out <- vector("list", length(x))
  for (i in seq_along(x)) {
    out[[i]] <- f(x[[i]], ...)
  }
  out
}
```

のケースでは、`randomise`はの`f`けり、それをUniformランダムサンプルでびします。をするために、の`set.seed`をびします。

```
set.seed(123)
randomise(mean)
#[1] 0.4972778

set.seed(123)
mean(runif(1e3))
#[1] 0.4972778

set.seed(123)
randomise(max)
#[1] 0.9994045

set.seed(123)
max(runif(1e3))
#[1] 0.9994045
```

2のは、`functionals`をしてリスト `x` のに `f` をする`base::lapply`です。 `...`パラメータをすると、`mean`の`na.rm`オプションなど、`f`にのをすことができます。

```
lapply(list(c(1, 3, 5), c(2, NA, 6)), mean)
# [[1]]
# [1] 3
#
# [[2]]
# [1] NA

lapply2(list(c(1, 3, 5), c(2, NA, 6)), mean)
# [[1]]
# [1] 3
#
# [[2]]
# [1] NA

lapply(list(c(1, 3, 5), c(2, NA, 6)), mean, na.rm = TRUE)
# [[1]]
# [1] 3
#
# [[2]]
# [1] 4

lapply2(list(c(1, 3, 5), c(2, NA, 6)), mean, na.rm = TRUE)
# [[1]]
# [1] 3
#
# [[2]]
```

オンラインで*ファミリをするをむ <https://riptutorial.com/ja/r/topic/3567/-ファミリをする-->

3: .Rprofile

なRプログラミングのについてはらしいがあります

Examples

.Rprofile - されたコードののチャンク

.Rprofile あなたがっているディレクトリからRをしたときにされるRコードをむファイルである .Rprofile ファイルを。 Rのホームディレクトリにあるの `Rprofile.site`、 のディレクトリからRをロードするたびにデフォルトでされます。 `Rprofile.site` およびよりきな .Rprofile をすると、 .Rprofile がやしたさまざまなユーティリティーをしてRセッションをできます。

なRStudioをするは、すべてのRStudioプロジェクトディレクトリに 々の .Rprofile ことができます。

.Rprofile ファイルにめるコードのをいくつかします。

R ホームディレクトリの

```
# set R_home
Sys.setenv(R_USER="c:/R_home") # just an example directory
# but don't confuse this with the $R_HOME environment variable.
```

ページサイズオプションの

```
options(papersize="a4")
options(editor="notepad")
options(pager="internal")
```

デフォルトのヘルプタイプをする

```
options(help_type="html")
```

サイトライブラリをする

```
.Library.site <- file.path(chartr("\\", "/", R.home()), "site-library")
```


CRANミラーをする

```
local({r <- getOption("repos")
  r["CRAN"] <- "http://my.local.cran"
  options(repos=r)})
```

ライブラリのをする

これにより、Rバージョンのアップデートですべてのパッケージをインストールするがなくなります。

```
# library location
.libPaths("c:/R_home/Rpackages/win")
```

カスタムショートカットまたは

いRのショートカットをつとなことがあります。こののなは、`.Last.value`ことなくのトップレベルのにアクセスするためのアクティブなバインディング`.Last.value`

```
makeActiveBinding(".", function(){.Last.value}, .GlobalEnv)
```

`.Rprofile`はなるRファイルなので、のRコードをめることができます。

もなパッケージをあらかじめロードする

これはいであり、にされているスクリプトからパッケージみみコードをするため、にけるべきです。

すべてのなるスタートアップスクリプト、およびそののについては、`help(Startup)`をしてください。に、2つのシステムの`Profile`ファイルもロードすることができます。の`Rprofile`にはグローバルがまれ、のファイル`Profile.site`は、システムがすべてのユーザーにしようことができるローカルがまれているがあります。のファイルは、Rインストールの`${RHOME}/etc`ディレクトリにあります。このディレクトリには、ユーザーのホームディレクトリのローカルファイル`~/.Renviron`で`~/.Renviron`できるグローバルファイル`Renviron`および`Renviron.site`もまれています。

`.profile`の

```
# Load library setwidth on start - to set the width automatically.
.First <- function() {
```

```
library(setwidth)
# If 256 color terminal - use library colorout.
if (Sys.getenv("TERM") %in% c("xterm-256color", "screen-256color")) {
  library("colorout")
}
}
```

オプション

```
# Select default CRAN mirror for package installation.
options(repos=c(CRAN="https://cran.gis-lab.info/"))

# Print maximum 1000 elements.
options(max.print=1000)

# No scientific notation.
options(scipen=10)

# No graphics in menus.
options(menu.graphics=FALSE)

# Auto-completion for package names.
utils::rc.settings(ipck=TRUE)
```

カスタム

```
# Invisible environment to mask defined functions
.env = new.env()

# Quit R without asking to save.
.env$q <- function (save="no", ...) {
  quit(save=save, ...)
}

# Attach the environment to enable functions.
attach(.env, warn.conflicts=FALSE)
```

オンラインで.Rprofileをむ <https://riptutorial.com/ja/r/topic/4166/-rprofile>

4: ANOVA

Examples

aovのない

aovは、2つのグループのがいにしくなるかどうかをするためにされます。はおいにしているとされ、されグループで、グループのはしいとみなされます。

データをさせるためには、データはいでなければなりません [データのトピック](#)を。 aov() のラッパーである lm() ウィルキンソン・ロジャースいて、 y~f ここで y であり f グループメンバーシップをすカテゴリです。 f がではなくである、 aov() はANOVAののをし、のいユーザをかせることがあります。

aov() は、Type I Sum of Squaresをします。このタイプのSum of Squaresは、すべてのとをにテストします。その、テストされたのエフェクトには、モデルのエフェクトとのエフェクトとののがりてられます。そのようなモデルのをできるものにするためには、データのバランスをるがありますすべてのグループはじサイズです。

タイプIの2のがりたないは、タイプIIまたはタイプIIIの2がである。 Type II Sum of Squaresは、のすべてのメインエフェクトのにメインエフェクトをテストし、オーバーラップするをします。しかし、Type II Sum of Squaresはのをしていない。

に、タイプIIIののは、のすべてのおよびすべてのので、をテストします。これは、がする、タイプIIIのをとする。

タイプIIおよびタイプIIIののは、 Anova() でされています。

として mtcars データセットをする。

```
mtCarsAnovaModel <- aov(wt ~ factor(cyl), data=mtcars)
```

ANOVAモデルのをするには

```
summary(mtcarsAnovaModel)
```

となる lm() モデルのをすることもできます。

```
coefficients(mtcarsAnovaModel)
```

Anovaのない

なデザインやのコントラストをうは、Type IIまたはType III Sum of Squaresがです。 car パッケージ

ジの `Anova()` はこれらをしています。タイプIIのはのをしていない。インタラクションがされるは、タイプIIIのがです。

`Anova()` は `lm()` をラップします。

として `mtcars` データセットをして、がテストされたときのタイプIIとタイプIIIのいをしします。

```
> Anova(lm(wt ~ factor(cyl)*factor(am), data=mtcars), type = 2)
Anova Table (Type II tests)

Response: wt
```

	Sum Sq	Df	F value	Pr(>F)
factor(cyl)	7.2278	2	11.5266	0.0002606 ***
factor(am)	3.2845	1	10.4758	0.0032895 **
factor(cyl):factor(am)	0.0668	2	0.1065	0.8993714
Residuals	8.1517	26		

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> Anova(lm(wt ~ factor(cyl)*factor(am), data=mtcars), type = 3)
Anova Table (Type III tests)

Response: wt
```

	Sum Sq	Df	F value	Pr(>F)
(Intercept)	25.8427	1	82.4254	1.524e-09 ***
factor(cyl)	4.0124	2	6.3988	0.005498 **
factor(am)	1.7389	1	5.5463	0.026346 *
factor(cyl):factor(am)	0.0668	2	0.1065	0.899371
Residuals	8.1517	26		

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

オンラインでANOVAをむ <https://riptutorial.com/ja/r/topic/3610/anova>

5: data.tableでピボット/アンピボット

- `melt(DT, id.vars=c(...), variable.name="CategoryLabel", value.name="Value")`
- `dcast(DT, LHS ~ RHS, value.var="Value", fun.aggregate=sum)` **キャスト** `dcast(DT, LHS ~ RHS, value.var="Value", fun.aggregate=sum)`

パラメーター

パラメータ	
id.vars	するを <code>melt</code>
variable.name	えて <code>melt</code> カテゴリラベルとをびすためにどのような
value.name	カテゴリラベルにけられたをつをにびすかを <code>melt</code> える
value.var	にキャストするのを <code>dcast</code> にします
	のレコードLHSをするためにどのをするか、カテゴリラベルRHSをするを <code>dcast</code> する
しい	キャストがセルののリストをするときにするをする

モデルやビジュアライゼーションをするためのデータには、`data.table` することができます。
のオプションとして、`data.table` はとのをします。

Examples

data.tableでのデータをピボット/アンピボットする - I

ワイドフォームからロングフォームへの

`data USArrests datasets` から `data USArrests` をロードし `data USArrests` 。

```
data("USArrests")
head(USArrests)
```

	Murder	Assault	UrbanPop	Rape
Alabama	13.2	236	58	21.2
Alaska	10.0	263	48	44.5
Arizona	8.1	294	80	31.0
Arkansas	8.8	190	50	19.5
California	9.0	276	91	40.6
Colorado	7.9	204	78	38.7

?USArrests、 ?USArrests を ?USArrests してください。まず、`data.table` にし `data.table`。 のはの

data.frame data.frame。

```
library(data.table)
DT <- as.data.table(USArrests, keep.rownames=TRUE)
```

これはこのデータです。のがあります。データはをうことなくいでもあります。いフォームには、ををする1つのがあります。に、のがあります。いのUSArrestsはそうそうです。

	State	Crime	Rate
1:	Alabama	Murder	13.2
2:	Alaska	Murder	10.0
3:	Arizona	Murder	8.1
4:	Arkansas	Murder	8.8
5:	California	Murder	9.0

196:	Virginia	Rape	20.7
197:	Washington	Rape	26.2
198:	West Virginia	Rape	9.3
199:	Wisconsin	Rape	10.8
200:	Wyoming	Rape	15.6

たちは、いフォームからいフォームにりえるためにmeltをします。

```
DTm <- melt(DT)
names(DTm) <- c("State", "Crime", "Rate")
```

デフォルトでは、meltはデータをつすべてののををつとしています。USArrestsでは、UrbanPopはののをします。それは、10あたりにされているであるの、Murder、Assault、Rapeとはなります。UrbanPopをしたいとします。id.varsをのようにして、これをします。

```
DTmu <- melt(DT, id.vars=c("rn", "UrbanPop" ),
              variable.name='Crime', value.name = "Rate")
names(DTmu)[1] <- "State"
```

カテゴリMurder、Assaultなどをむのをvariable.nameでし、をむをvalue.nameしていることにしてください。たちのデータはそうそうです。

	State	UrbanPop	Crime	Rate
1:	Alabama	58	Murder	13.2
2:	Alaska	48	Murder	10.0
3:	Arizona	80	Murder	8.1
4:	Arkansas	50	Murder	8.8
5:	California	91	Murder	9.0

split-apply-combineスタイルのアプローチでをするのはです。え、をごとにするとします。

```
DTmu[, .(ViolentCrime = sum(Rate)), by=State]
```

これはえる

	State	ViolentCrime
1:	Alabama	270.4
2:	Alaska	317.5
3:	Arizona	333.1
4:	Arkansas	218.3
5:	California	325.6
6:	Colorado	250.6

data.tableをしたデータのピボットとアンピボット - II

ロングフォームからワイドフォームへの

のからデータをするには、`dcast`にします。

```
DTc <- dcast(DTmu, State + UrbanPop ~ Crime)
```

これにより、のワイドフォームのデータがられます。

	State	UrbanPop	Murder	Assault	Rape
1:	Alabama	58	13.2	236	21.2
2:	Alaska	48	10.0	263	44.5
3:	Arizona	80	8.1	294	31.0
4:	Arkansas	50	8.8	190	19.5
5:	California	91	9.0	276	40.6

ここでは、をして、のレコードLHSをすると、しいRHSのカテゴリラベルをむをします。どのをにしますかデフォルトでは、`dcast`はのをして、のにつたをします。にするには、パラメータ`value.var`をとにします。

によってセルにのリストがされると、`dcast`は`fun.aggregate`メソッドをしてをします。をするに、はのがしているにが有るとします。はされたをつ`Decile`をします。

```
DTmu[, Decile := cut(UrbanPop, quantile(UrbanPop, probs = seq(0, 1, by=0.1)))]
levels(DTmu$Decile) <- paste0(1:10, "D")
```

、`Decile ~ Crime`をキャストイングすると、セルごとにのがされます。 `fun.aggregate`をつて、これらのをできます。テキストとのをこのようにうことができます。

```
dcast(DTmu, Decile ~ Crime, value.var="Rate", fun.aggregate=sum)
```

これはえる

```
dcast(DTmu, Decile ~ Crime, value.var="Rate", fun.aggregate=mean)
```

これはえる

	State	UrbanPop	Crime	Rate	Decile
1:	Alabama	58	Murder	13.2	4D
2:	Alaska	48	Murder	10.0	2D

3:	Arizona	80	Murder	8.1	8D
4:	Arkansas	50	Murder	8.8	2D
5:	California	91	Murder	9.0	10D

の10の1にはのがあります。 `fun.aggregate`をして、これらのをします。

```
dcast(DTmu, Decile ~ Crime, value.var="Rate", fun.aggregate=sum)
```

これは、ののデータをし、をえます。

	Decile	Murder	Assault	Rape
1:	1D	39.4	808	62.6
2:	2D	35.3	815	94.3
3:	3D	22.6	451	67.7
4:	4D	54.9	898	106.0
5:	5D	42.4	758	107.6

オンラインでdata.tableでピボット/アンピボットをむ <https://riptutorial.com/ja/r/topic/6934/data-table>でピボット-アンピボット

6: Date クラス

トピック

•

したメモ

- `Date` 1970-01-01 1970-01-01 UNIXエポックからのを1970-01-01ます。よりいのはのになります。
- これはでされますただし、ではされません
- カレンダーはまだされていませんでしたが、のグレゴリオののってにされます。
- タイムゾーンをしないので、`POSIXct` オブジェクトまたは`POSIXlt` オブジェクトのタイムアウトをりてるためにすべきではありません。
- `sys.Date()` は、`Date` クラスのオブジェクトをします。

もっとメモ

- `lubridate` の `ymd`、`mdy` などは、`Date` クラスをする `as.Date` です。 `lubridate` をしたからのののをしてください。
- `data.table` のな `IDate` クラスは、`Date` からしたもので、ほとんどは `Date` とがありますが、`double` ではなく `integer` としてされます。

Examples

の

`Dates` をフォーマットするには、`POSIXct as.POSIXct()` または `POSIXlt as.POSIXlt()` からえられるのいずれかをして `Dates` を `format(date, format="%Y-%m-%d")`

```
d = as.Date("2016-07-21") # Current Date Time Stamp

format(d, "%a")           # Abbreviated Weekday
## [1] "Thu"

format(d, "%A")           # Full Weekday
## [1] "Thursday"

format(d, "%b")           # Abbreviated Month
## [1] "Jul"

format(d, "%B")           # Full Month
```

```
## [1] "July"

format(d,"%m")           # 00-12 Month Format
## [1] "07"

format(d,"%d")           # 00-31 Day Format
## [1] "21"

format(d,"%e")           # 0-31 Day Format
## [1] "21"

format(d,"%y")           # 00-99 Year
## [1] "16"

format(d,"%Y")           # Year with Century
## [1] "2016"
```

については、`?strptime` してください。

をに `as.Date()` するには、`as.Date()` をします。

```
> x <- as.Date("2016-8-23")
> x
[1] "2016-08-23"
> class(x)
[1] "Date"
```

`as.Date()` をすると、をできます。デフォルトは `%Y-%m-%d`、- - です。

```
> as.Date("23-8-2016", format="%d-%m-%Y") # To read in an European-style date
[1] "2016-08-23"
```

は、またはのペアにすることができます。は、`"dm-yy"`または`"dm-YYYY"`または`"md-yy"`または`"md-YYYY"`または`"YYYY-md"`または`"YYYY-dm"`などのさまざまなでされます。これらは、`"-"`を`"/"`きえることによってできます。1986116、1986116、1986116、1986116などのように、もまたでされます。**as.Date**はこのようなをすべてけり、のなをすると、に`"YYYY-md"`でをします。

`"%d-%m-%Y"`"9-6-1962"があるとしします。

```
#
# It tries to interprets the string as YYYY-m-d
#
> as.Date("9-6-1962")
[1] "0009-06-19"           #interprets as "%Y-%m-%d"
>
as.Date("9/6/1962")
[1] "0009-06-19"           #again interprets as "%Y-%m-%d"
>
# It has no problem in understanding, if the date is in form YYYY-m-d or YYYY/m/d
#
> as.Date("1962-6-9")
[1] "1962-06-09"           # no problem
> as.Date("1962/6/9")
[1] "1962-06-09"           # no problem
>
```

のしいをすることにより、のをすることができます。のコードをして、**as.Date**のをします。

フォーマットコード	
%d	
%m	
%y	2の
%Y	4の
%b	3でされた
%B	のフルネーム

formatパラメータをするのをえてみましょう。

```
> as.Date("9-6-1962", format="%d-%m-%Y")
[1] "1962-06-09"
>
```

パラメータのはできます。

```
> as.Date("9-6-1962", "%d-%m-%Y")
[1] "1962-06-09"
>
```

には、の3にされたのは、をくにされます。この、フォーマット**%b**をします。

```
> as.Date("6Nov1962", "%d%b%Y")
[1] "1962-11-06"
>
```

のメンバーに、または、/、またはのいずれもないことにしてください。は、そのとにするがあります。のをえてみましょう。

```
> as.Date("6 Nov, 1962", "%d %b, %Y")
[1] "1962-11-06"
>
```

にカンマがあるため、フォーマットのコンマもあることにしてください。でコンマがされたは、**NA**ます。 **%B**のはのとおりです。

```
> as.Date("October 12, 2016", "%B %d, %Y")
[1] "2016-10-12"
>
> as.Date("12 October, 2016", "%d %B, %Y")
[1] "2016-10-12"
>
```

`%y`はシステムのもので、してするがあります。このでされるのパラメータは、**origin**および**tz**です。

をオブジェクトにする

Rには、またはのベクトルをとる`as.Date()`でされたDateクラスがまれています。がISO 8601のYYYY-MM-DDでないは、`strptime YYYY-MM-DD strptime`トークンの。

```
as.Date('2016-08-01')      # in ISO format, so does not require formatting string
## [1] "2016-08-01"

as.Date('05/23/16', format = '%m/%d/%y')
## [1] "2016-05-23"

as.Date('March 23rd, 2016', '%B %drd, %Y')      # add separators and literals to format
## [1] "2016-03-23"

as.Date(' 2016-08-01  foo')      # leading whitespace and all trailing characters are ignored
## [1] "2016-08-01"

as.Date(c('2016-01-01', '2016-01-02'))
# [1] "2016-01-01" "2016-01-02"
```

オンラインでDateクラスをむ <https://riptutorial.com/ja/r/topic/9015/date>クラス

7: devtools をってパッケージをる

き

このトピックでは、devtoolsパッケージをしてRパッケージをからするについてします。

1. パッケージをするためのRマニュアル
2. roxygen2 リファレンスマニュアル
3. devtools リファレンスマニュアル

Examples

パッケージのと

これは、コードからRパッケージをすばやくするについてのコンパクトなガイドです。したはなにはリンクされ、のながなは、まれるべきです。については、「 」をしてください。

コードがするディレクトリは、.とされ、すべてのコマンドはこのフォルダのRプロンプトからされます。

ドキュメンテーションの

あなたのコードのドキュメントは、LaTeXににたでなければなりません。

ただし、プロセスをするためにroxygenというツールをします。

```
install.packages("devtools")
library("devtools")
install.packages("roxygen2")
library("roxygen2")
```

roxygenのマnpページは[こちらから](#)できます。これはdoxygenとによくています。

roxygenをってをするにするなサンプルをにします。

```
## Increment a variable.
##
## Note that the behavior of this function
## is undefined if `x` is not of class `numeric`.
##
## @export
## @author another guy
## @name Increment Function
## @title increment
```

```
#'
#' @param x    Variable to increment
#' @return     `x` incremented of 1
#'
#' @seealso    `other_function`
#'
#' @examples
#' increment(3)
#' > 4
increment <- function(x) {
  return (x+1)
}
```

そして[ここ](#)にがます。

また、パッケージにするなガイドであるビネットをすることもされますトピックのをしてください。

パッケージスケルトンの

あなたのコードがえは、`./script1.R`と`./script2.R`ファイルにかけられているとして、あなたのパッケージのファイルツリーを作るためのコマンドをしてください

```
package.skeleton(name="MyPackage", code_files=c("script1.R","script2.R"))
```

その、`../MyPackage/man/`すべてのファイルをしします。すぐドキュメントをコンパイルしてください

```
roxygenize("MyPackage")
```

また、`../`したコマンドプロンプトから、`R CMD Rd2pdf MyPackage`をしてマニュアルからリファレンスマニュアルをするがあります。

パッケージプロパティのエディション

1. パッケージの

にじて、`../MyPackage/DESCRIPTION`しします。 「`Package`」、「`Version`」、「`License`」、「`Description`」、「`Title`」、「`Author`」、および「`Maintainer`」のフィールドはであり、もう1つはオプションです。

パッケージがのパッケージにするは、`Depends R version <3.2.0`または`Imports R version > 3.2.0`というのフィールドにしします。

2. オプションのフォルダ

いったんスケルトンビルドを `./MyPackage/` すると、`./R/` `./MyPackage/R/` は `R/` と `man/` subfolders しかありませんでした。しかし、それはのものをつことができます

- `data/` ここでは、あなたのライブラリがとする、コードではないデータをくことができます
 - `.RData` をつデータセットとしてするがあります。また、に `data()` および `load()` メソッドをしてロードできます。
- `tests/` このフォルダのすべてのコードファイルは、インストールにされます。エラーがある、インストールはします。
- `src/` な `C / C ++ / Fortran` ソースファイル `Rcpp` をつて...
- `exec/` のファイル。
- `misc/` ほとんどすべてのために。

ファイナライズとビルド

`./MyPackage/Read-and-delete-me` をすることができます。

のように、パッケージはすぐにインストールできます。

`devtools::install("MyPackage")` できます。

パッケージをソースtarballとしてビルドするには、 コマンドプロンプトで `./R CMD build MyPackage` をするがあります

あなたのパッケージの

Githubをして

に `mypackage` とばれるしいリポジトリをし、にすべてのものをアップロード `MyPackage/` マスターブランチに。ここに [あります](#)。

そうすればでも `devtools` でパッケージを `github` からインストールできます

```
install_package("MyPackage", "your_github_username")
```

CRANをして

パッケージは [CRANリポジトリポリシー](#) にする [あります](#)。あなたのパッケージはクロスプラットフォームいくつかのになケースをくでなければなりません、 `R CMD check` テストにするがあり

ます。

ここに[があります](#)。ソースtarballをアップロードする[があります](#)。

ビネットをする

ビネットはあなたのパッケージのいのガイドです。のドキュメントはなのをっていれ
ばらしいですが、そうでなければにたないのです。ビネットは、のやのようなもので
す。あなたのパッケージがするようにされたをし、それをするをにすことができます
。

ビネットはにマークダウンでされます。

- Rmarkdown `install.packages("rmarkdown")`
- [Pandoc](#)

ビネット

```
devtools::use_vignette("MyVignette", "MyPackage")
```

`./vignettes/MyVignette.Rmd`あなたのビネットをできるようになり`./vignettes/MyVignette.Rmd`。

あなたのビネットのテキストは[Markdown](#)としてフォーマットされています。

のMarkdownにされたのは、Rコードをし、し、をキャプチャし、フォーマットされたMarkdown
にするタグです。

```
```${r}
Add two numbers together
add <- function(a, b) a + b
add(10, 20)
```
```

のようにされます

```
# Add two numbers together
add <- function(a, b) a + b
add(10, 20)
## [1] 30
```

したがって、あなたのビネットでするすべてのパッケージは、`../DESCRIPTION`としてリストされな
ければなりません。

オンラインでdevtoolsをってパッケージをるをむ <https://riptutorial.com/ja/r/topic/10884/devtools>を
ってパッケージをる

8: dplyr

dplyrは、Rのデータをするな「」ベースのをするplyrのです。dplyrのバージョンは、CRANからのコマンドをしてダウンロードできます。

```
install.packages("dplyr")
```

dplyrのキーオブジェクトは、のデータをすtblです。、dplyrバージョン0.5.0はをサポートしています

- データフレーム
- データテーブル
- SQLite
- PostgreSQL / Redshift
- MySQL / MariaDB
- ビッグクエリー
- MonetDB
- によるデータキューブ

Examples

dplyrの

dplyrはRでのデータのをしします。データがされている [data.frame](#)、[data.table](#)、またはdatabaseなく、データをうしたインターフェイスをしdatabase。dplyrしてかかれているRcppメモリのデータをうための、それはにになり、。

dplyrのは、1つのことをうまくやるさなをつことです。5つのシンプルな `filter`、`arrange`、`select`、`mutate`、`summarise` をして、データをするしいをらかにすることができます。 `group_by` とみわせると、これらのをしてグループワイズサマリーをできます。

の

これらののはすべてています

- これらすべてのの1はにデータフレームです
- のをしてをすることができますつまり、`$`をせずに
- これらの、のデータをしません。つまり、がありません。したがって、はにオブジェクトにするがあります。

みみのmtcarsデータセットをして、dplyrのテーブルをしします。タイプするmtcarsするtbl_df それはクリーナーをるため、々は、rownamesいたカラムとして、データセットのをrownames_to_columnからをtibbleパッケージ。

```
library(dplyr) # This documentation was written using version 0.5.0

mtcars_tbl <- as_data_frame(tibble::rownames_to_column(mtcars, "cars"))

# examine the structure of data
head(mtcars_tbl)

# A tibble: 6 x 12
#       cars      mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear  carb
#       <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#1   Mazda RX4   21.0     6   160   110   3.90  2.620  16.46     0    1     4     4
#2   Mazda RX4 Wag 21.0     6   160   110   3.90  2.875  17.02     0    1     4     4
#3   Datsun 710   22.8     4   108    93   3.85  2.320  18.61     1    1     4     1
#4  Hornet 4 Drive 21.4     6   258   110   3.08  3.215  19.44     1    0     3     1
#5 Hornet Sportabout 18.7     8   360   175   3.15  3.440  17.02     0    0     3     2
#6    Valiant   18.1     6   225   105   2.76  3.460  20.22     1    0     3     1
```

フィルタ

`filter` は、のにするのサブセットにちます。のは `data.frame` ので、2のはデータをフィルタリングするですこれらののは `TRUE` または `FALSE` いずれかにするがあり `FALSE`

4つのシリンダーをつすべてのをサブセットする - `cyl`

```
filter(mtcars_tbl, cyl == 4)

# A tibble: 11 x 12
#       cars      mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear  carb
#       <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#1   Datsun 710   22.8     4 108.0    93   3.85  2.320  18.61     1    1     4     1
#2    Merc 240D   24.4     4 146.7    62   3.69  3.190  20.00     1    0     4     2
#3    Merc 230   22.8     4 140.8    95   3.92  3.150  22.90     1    0     4     2
#4    Fiat 128   32.4     4  78.7    66   4.08  2.200  19.47     1    1     4     1
#5   Honda Civic  30.4     4  75.7    52   4.93  1.615  18.52     1    1     4     2
# ... with 6 more rows
```

コンマでられたのをすことができます。 4または6 をするサブセットに- `cyl` 5つのギアをとっている

- `gear`

```
filter(mtcars_tbl, cyl == 4 | cyl == 6, gear == 5)

# A tibble: 3 x 12
#       cars      mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear  carb
#       <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#1 Porsche 914-2   26.0     4 120.3    91   4.43  2.140  16.7     0    1     5     2
#2  Lotus Europa   30.4     4  95.1   113   3.77  1.513  16.9     1    1     5     2
#3  Ferrari Dino   19.7     6 145.0   175   3.62  2.770  15.5     0    1     5     6
```

`filter` はにづいてをし、ごとにをし、 `slice` し `slice` 。 `slice` は2つのしからないのものは `data.frame` あり、2のものはのである。

69をするには

```
slice(mtcars_tbl, 6:9)

# A tibble: 4 x 12
#   cars      mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear  carb
#   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#1  Valiant  18.1     6  225.0   105  2.76  3.46  20.22     1   0     3     1
#2 Duster 360  14.3     8  360.0   245  3.21  3.57  15.84     0   0     3     4
#3 Merc 240D  24.4     4  146.7    62  3.69  3.19  20.00     1   0     4     2
#4 Merc 230  22.8     4  140.8    95  3.92  3.15  22.90     1   0     4     2
```

または

```
slice(mtcars_tbl, -c(1:5, 10:n()))
```

この、`slice(mtcars_tbl, 6:9)`と同じがられ`slice(mtcars_tbl, 6:9)`

`n()`は、のグループのをす

アレンジ

`arrange`は、されたでデータをソートするためにされます。のおよび`dplyr`のすべてのとに、のは`data.frame`であり、のはデータのソートにされます。のがされた、データはにのでソートされ、に2でソートされます。

でデータをするには - `hp`

```
arrange(mtcars_tbl, hp)

# A tibble: 32 x 12
#   cars      mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear  carb
#   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#1  Honda Civic  30.4     4   75.7    52  4.93  1.615  18.52     1   1     4     2
#2  Merc 240D   24.4     4  146.7    62  3.69  3.190  20.00     1   0     4     2
#3 Toyota Corolla 33.9     4   71.1    65  4.22  1.835  19.90     1   1     4     1
#4  Fiat 128    32.4     4   78.7    66  4.08  2.200  19.47     1   1     4     1
#5  Fiat X1-9   27.3     4   79.0    66  4.08  1.935  18.90     1   1     4     1
#6  Porsche 914-2 26.0     4  120.3    91  4.43  2.140  16.70     0   1     5     2
# ... with 26 more rows
```

するに`arrange` - ガロンたりマイルによるデータ`mpg` シリンダにいて、で- `cyl`

```
arrange(mtcars_tbl, desc(mpg), cyl)

# A tibble: 32 x 12
#   cars      mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear  carb
#   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#1 Toyota Corolla 33.9     4   71.1    65  4.22  1.835  19.90     1   1     4     1
#2  Fiat 128    32.4     4   78.7    66  4.08  2.200  19.47     1   1     4     1
#3  Honda Civic  30.4     4   75.7    52  4.93  1.615  18.52     1   1     4     2
#4  Lotus Europa 30.4     4   95.1   113  3.77  1.513  16.90     1   1     5     2
#5  Fiat X1-9   27.3     4   79.0    66  4.08  1.935  18.90     1   1     4     1
#6  Porsche 914-2 26.0     4  120.3    91  4.43  2.140  16.70     0   1     5     2
```

```
# ... with 26 more rows
```

する

`select` は、のサブセットのみをするためにされます。のみをするには `mpg`、`disp`、`wt`、`qsec`、および `vs` から `mtcars_tbl`

```
select(mtcars_tbl, mpg, disp, wt, qsec, vs)

# A tibble: 32 x 5
#   mpg   disp  wt   qsec    vs
#   <dbl> <dbl> <dbl> <dbl> <dbl>
#1  21.0  160.0  2.620  16.46     0
#2  21.0  160.0  2.875  17.02     0
#3  22.8  108.0  2.320  18.61     1
#4  21.4  258.0  3.215  19.44     1
#5  18.7  360.0  3.440  17.02     0
#6  18.1  225.0  3.460  20.22     1
# ... with 26 more rows
```

:をしてするをできます。 `cars` から `disp`、`vs` through `carb` をするには

```
select(mtcars_tbl, cars:disp, vs:carb)

# A tibble: 32 x 8
#   cars      mpg   cyl  disp    vs  am  gear carb
#   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#1 Mazda RX4    21.0     6  160.0     0     1     4     4
#2 Mazda RX4 Wag  21.0     6  160.0     0     1     4     4
#3 Datsun 710    22.8     4  108.0     1     1     4     1
#4 Hornet 4 Drive  21.4     6  258.0     1     0     3     1
#5 Hornet Sportabout  18.7     8  360.0     0     0     3     2
#6 Valiant      18.1     6  225.0     1     0     3     1
# ... with 26 more rows
```

または `select(mtcars_tbl, -(hp:qsec))`

のをむデータセットの、のをでするのはです。をにするために、ヘルパーがあるえば `starts_with()` `ends_with()` `contains()`、`matches()` `num_range()` `one_of()`、および `everything()` であることができる `select`。それらをするのについては、「`?select_helpers` and `?select`」をしてください。

`select()` はしていますが、のをしていますが、ヘルパーのをするときはをするがあります。

にのをするには

```
select(mtcars_tbl, cylinders = cyl, displacement = disp)

# A tibble: 32 x 2
#   cylinders displacement
#   <dbl>         <dbl>
```

```
#1      6      160.0
#2      6      160.0
#3      4      108.0
#4      6      258.0
#5      8      360.0
#6      6      225.0
# ... with 26 more rows
```

どおり、これはのすべてのをします。

のをせずにのをするには、`rename`

```
rename(mtcars_tbl, cylinders = cyl, displacement = disp)

# A tibble: 32 x 12
#       cars      mpg cylinders displacement      hp  drat      wt  qsec      vs
#       <chr> <dbl>      <dbl>      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#1   Mazda RX4  21.0          6      160.0   110  3.90  2.620  16.46     0
#2   Mazda RX4 Wag 21.0          6      160.0   110  3.90  2.875  17.02     0
#3   Datsun 710  22.8          4      108.0    93  3.85  2.320  18.61     1
#4   Hornet 4 Drive 21.4          6      258.0   110  3.08  3.215  19.44     1
#5   Hornet Sportabout 18.7          8      360.0   175  3.15  3.440  17.02     0
#6     Valiant  18.1          6      225.0   105  2.76  3.460  20.22     1
# ... with 26 more rows, and 3 more variables: am <dbl>, gear <dbl>, carb <dbl>
```

`mutate`をしてしいをデータにすることができます。 `dplyr`のすべてのとに、 `dplyr`はしくされたをのデータにしません。は `data.frame`のにされます。

```
mutate(mtcars_tbl, weight_ton = wt/2, weight_pounds = weight_ton * 2000)

# A tibble: 32 x 14
#       cars      mpg  cyl  disp      hp  drat      wt  qsec      vs  am  gear  carb
weight_ton weight_pounds
#       <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
<dbl>      <dbl>
#1   Mazda RX4  21.0      6 160.0   110  3.90  2.620  16.46     0     1     4     4
1.3100      2620
#2   Mazda RX4 Wag 21.0      6 160.0   110  3.90  2.875  17.02     0     1     4     4
1.4375      2875
#3   Datsun 710  22.8      4 108.0    93  3.85  2.320  18.61     1     1     4     1
1.1600      2320
#4   Hornet 4 Drive 21.4      6 258.0   110  3.08  3.215  19.44     1     0     3     1
1.6075      3215
#5   Hornet Sportabout 18.7      8 360.0   175  3.15  3.440  17.02     0     0     3     2
1.7200      3440
#6     Valiant  18.1      6 225.0   105  2.76  3.460  20.22     1     0     3     1
1.7300      3460
# ... with 26 more rows
```

にしてください `weight_tonweight_pounds`。 `R`とはなり、 `mutate`では、ですのためにしたばかりのをできます。

し、のみしくされたをするには `transmute`のわり `mutate`

```
transmute(mtcars_tbl, weight_ton = wt/2, weight_pounds = weight_ton * 2000)

# A tibble: 32 x 2
#   weight_ton weight_pounds
#   <dbl>      <dbl>
#1     1.3100         2620
#2     1.4375         2875
#3     1.1600         2320
#4     1.6075         3215
#5     1.7200         3440
#6     1.7300         3460
# ... with 26 more rows
```

する

`summarise`は、のをのにみむことによって、のをします。のをすることができ、じステートメントでこれらのサマリーにをけることができます。

データセットのすべてのの`mpg`と`disp`のとをするには

```
summarise(mtcars_tbl, mean_mpg = mean(mpg), sd_mpg = sd(mpg),
          mean_disp = mean(disp), sd_disp = sd(disp))

# A tibble: 1 x 4
#   mean_mpg sd_mpg mean_disp sd_disp
#   <dbl>   <dbl>   <dbl>   <dbl>
#1  20.09062 6.026948  230.7219 123.9387
```

group_by

`group_by`は、データにしてグループワイズをするためにできます。でしたがこのグループされたデータにされると、にグループにされます。

`cyl`による`mpg mean`と`sd`をつけるに`mean`

```
by_cyl <- group_by(mtcars_tbl, cyl)
summarise(by_cyl, mean_mpg = mean(mpg), sd_mpg = sd(mpg))

# A tibble: 3 x 3
#   cyl mean_mpg sd_mpg
#   <dbl>   <dbl>   <dbl>
#1     4  26.66364  4.509828
#2     6  19.74286  1.453567
#3     8  15.10000  2.560048
```

それをすべてtogetherにする

私たちはcarsからhpとgearまでのをし、cylでをべ、からまでmpg、gearでデータをグループし、にmpg > 20、hp > 75ののみをサブセットします

```
selected <- select(mtcars_tbl, cars:hp, gear)
ordered <- arrange(selected, cyl, desc(mpg))
by_cyl <- group_by(ordered, gear)
filter(by_cyl, mpg > 20, hp > 75)

Source: local data frame [9 x 6]
Groups: gear [3]

#       cars    mpg   cyl  disp    hp  gear
#   <chr> <dbl> <dbl> <dbl> <dbl> <dbl>
#1 Lotus Europa  30.4     4  95.1   113     5
#2 Porsche 914-2  26.0     4 120.3    91     5
#3 Datsun 710    22.8     4 108.0    93     4
#4 Merc 230     22.8     4 140.8    95     4
#5 Toyota Corona 21.5     4 120.1    97     3
# ... with 4 more rows
```

にはがないかもしれないので、びしをラップすることでとじをすることができます

```
filter(
  group_by(
    arrange(
      select(
        mtcars_tbl, cars:hp
      ), cyl, desc(mpg)
    ), cyl
  ), mpg > 20, hp > 75
)
```

これはむのがしいかもしれません。したがって、dplyrはpipe %>%をしてさせることができます。のコードはのようになります

```
mtcars_tbl %>%
  select(cars:hp) %>%
  arrange(cyl, desc(mpg)) %>%
  group_by(cyl) %>%
  filter(mpg > 20, hp > 75)
```

のをする

dplyrは、すべてのグループされていないにをするsummarise_all()をします。

のののべるには

```
mtcars_tbl %>%
  summarise_all(n_distinct)

# A tibble: 1 x 12
#   cars    mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
```

```
# <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
#1      32      25       3      27      22      22      29      30       2       2       3       6
```

cylによってごとなるのをめるには

```
mtcars_tbl %>%
  group_by(cyl) %>%
  summarise_all(n_distinct)

# A tibble: 3 x 12
#   cyl cars mpg disp hp drat wt  qsec vs  am gear carb
#   <dbl> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
#1     4    11     9    11    10    10    11    11     2     2     3     2
#2     6     7     6     5     4     5     6     7     2     2     3     3
#3     8    14    12    11     9    11    13    14     1     2     2     4
```

group_byをするだけで、りのコードはじであることにしてください。は、3つのからります.1つは、 cylのごとに1つです。

のを summarise するに summarise、 summarise_at

```
mtcars_tbl %>%
  group_by(cyl) %>%
  summarise_at(c("mpg", "disp", "hp"), mean)

# A tibble: 3 x 4
#   cyl mpg disp hp
#   <dbl> <dbl> <dbl> <dbl>
#1     4 26.66364 105.1364 82.63636
#2     6 19.74286 183.3143 122.28571
#3     8 15.10000 353.1000 209.21429
```

のわりに helper ?select_helpers をしてのをすることができます

のをするには、をベクトルとしてすか、

```
mtcars_tbl %>%
  group_by(cyl) %>%
  summarise_at(c("mpg", "disp", "hp"),
               c("mean", "sd"))
```

またはそれらを funs もののにみます

```
mtcars_tbl %>%
  group_by(cyl) %>%
  summarise_at(c("mpg", "disp", "hp"),
               funs(mean, sd))

# A tibble: 3 x 7
#   cyl mpg_mean disp_mean hp_mean mpg_sd disp_sd hp_sd
#   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#1     4 26.66364 105.1364 82.63636 4.509828 26.87159 20.93453
#2     6 19.74286 183.3143 122.28571 1.453567 41.56246 24.26049
#3     8 15.10000 353.1000 209.21429 2.560048 67.77132 50.97689
```


にがされ、それらをできるようになりました。これをするには、にするをします。

```
mtcars_tbl %>%
  group_by(cyl) %>%
  summarise_at(c("mpg", "disp", "hp"),
               c(Mean = "mean", SD = "sd"))

mtcars_tbl %>%
  group_by(cyl) %>%
  summarise_at(c("mpg", "disp", "hp"),
               funs(Mean = mean, SD = sd))

# A tibble: 3 x 7
#   cyl mpg_Mean disp_Mean hp_Mean mpg_SD disp_SD hp_SD
#   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
#1     4 26.66364 105.1364  82.63636  4.509828 26.87159 20.93453
#2     6 19.74286 183.3143 122.28571  1.453567 41.56246 24.26049
#3     8 15.10000 353.1000 209.21429  2.560048 67.77132 50.97689
```

きでをするには、 `summarise_if` し `summarise_if`。

`cyl`によってグループされた `numeric` であるすべてのの `mean` をる

```
mtcars_tbl %>%
  group_by(cyl) %>%
  summarise_if(is.numeric, mean)

# A tibble: 3 x 11
#   cyl   mpg   disp  hp  drat    wt  qsec
#   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#1     4 26.66364 105.1364  82.63636 4.070909 2.285727 19.13727
#2     6 19.74286 183.3143 122.28571 3.585714 3.117143 17.97714
#3     8 15.10000 353.1000 209.21429 3.229286 3.999214 16.77214
# ... with 4 more variables: vs <dbl>, am <dbl>, gear <dbl>,
#   carb <dbl>
```

しかし、いくつかのはであり、これらのの `mean` はをなさない。

`cyl` によるのの `mean` をとるには

```
mtcars_tbl %>%
  group_by(cyl) %>%
  summarise_if(function(x) is.numeric(x) & n_distinct(x) > 6, mean)

# A tibble: 3 x 7
#   cyl   mpg   disp  hp  drat    wt  qsec
#   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#1     4 26.66364 105.1364  82.63636 4.070909 2.285727 19.13727
#2     6 19.74286 183.3143 122.28571 3.585714 3.117143 17.97714
#3     8 15.10000 353.1000 209.21429 3.229286 3.999214 16.77214
```

サブセットの

`dplyr::filter()`

- をたすデータフレームののサブセットをします。

```
dplyr::filter(iris, Sepal.Length>7)
#   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
# 1         7.1         3.0         5.9         2.1 virginica
# 2         7.6         3.0         6.6         2.1 virginica
# 3         7.3         2.9         6.3         1.8 virginica
# 4         7.2         3.6         6.1         2.5 virginica
# 5         7.7         3.8         6.7         2.2 virginica
# 6         7.7         2.6         6.9         2.3 virginica
# 7         7.7         2.8         6.7         2.0 virginica
# 8         7.2         3.2         6.0         1.8 virginica
# 9         7.2         3.0         5.8         1.6 virginica
# 10        7.4         2.8         6.1         1.9 virginica
# 11        7.9         3.8         6.4         2.0 virginica
# 12        7.7         3.0         6.1         2.3 virginica
```

- `dplyr::distinct()` をする

```
distinct(iris, Sepal.Length, .keep_all = TRUE)
#   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
# 1         5.1         3.5         1.4         0.2   setosa
# 2         4.9         3.0         1.4         0.2   setosa
# 3         4.7         3.2         1.3         0.2   setosa
# 4         4.6         3.1         1.5         0.2   setosa
# 5         5.0         3.6         1.4         0.2   setosa
# 6         5.4         3.9         1.7         0.4   setosa
# 7         4.4         2.9         1.4         0.2   setosa
# 8         4.8         3.4         1.6         0.2   setosa
# 9         4.3         3.0         1.1         0.1   setosa
# 10        5.8         4.0         1.2         0.2   setosa
# 11        5.7         4.4         1.5         0.4   setosa
# 12        5.2         3.5         1.5         0.2   setosa
# 13        5.5         4.2         1.4         0.2   setosa
# 14        4.5         2.3         1.3         0.3   setosa
# 15        5.3         3.7         1.5         0.2   setosa
# 16        7.0         3.2         4.7         1.4 versicolor
# 17        6.4         3.2         4.5         1.5 versicolor
# 18        6.9         3.1         4.9         1.5 versicolor
# 19        6.5         2.8         4.6         1.5 versicolor
# 20        6.3         3.3         4.7         1.6 versicolor
# 21        6.6         2.9         4.6         1.3 versicolor
# 22        5.9         3.0         4.2         1.5 versicolor
# 23        6.0         2.2         4.0         1.0 versicolor
# 24        6.1         2.9         4.7         1.4 versicolor
# 25        5.6         2.9         3.6         1.3 versicolor
# 26        6.7         3.1         4.4         1.4 versicolor
# 27        6.2         2.2         4.5         1.5 versicolor
# 28        6.8         2.8         4.8         1.4 versicolor
# 29        7.1         3.0         5.9         2.1 virginica
# 30        7.6         3.0         6.6         2.1 virginica
# 31        7.3         2.9         6.3         1.8 virginica
# 32        7.2         3.6         6.1         2.5 virginica
# 33        7.7         3.8         6.7         2.2 virginica
# 34        7.4         2.8         6.1         1.9 virginica
# 35        7.9         3.8         6.4         2.0 virginica
```

>パイプでする

`pipe>` は `dplyr` とみわけてできます。このでは、 `mtcars` データセットは `help("mtcars")` ををして、データフレームを `summarize` するをし、のをしてデータにをします。

```
library(dplyr)
library(magrittr)
df <- mtcars
df$cars <- rownames(df) #just add the cars names to the df
df <- df[,c(ncol(df),1:(ncol(df)-1))] # and place the names in the first column
```

1. データの Sumarize

をするには、 `summarize` となをします。この、 `n()` はケースのをカウントするためにされます。

```
df %>%
  summarize(count=n(),mean_mpg = mean(mpg, na.rm = TRUE),
            min_weight = min(wt),max_weight = max(wt))

#   count mean_mpg min_weight max_weight
#1     32 20.09062     1.513     5.424
```

2. グループにをする

データのグループにをすることはです。この、シリンダ とギヤ

```
df %>%
  group_by(cyl, gear) %>%
  summarize(count=n(),mean_mpg = mean(mpg, na.rm = TRUE),
            min_weight = min(wt),max_weight = max(wt))

# Source: local data frame [8 x 6]
# Groups: cyl [?]
#
#   cyl  gear count mean_mpg min_weight max_weight
#   <dbl> <dbl> <int>    <dbl>    <dbl>    <dbl>
#1     4     3     1    21.500     2.465     2.465
#2     4     4     8    26.925     1.615     3.190
#3     4     5     2    28.200     1.513     2.140
#4     6     3     2    19.750     3.215     3.460
#5     6     4     4    19.750     2.620     3.440
#6     6     5     1    19.700     2.770     2.770
#7     8     3    12    15.050     3.435     5.424
#8     8     5     2    15.400     3.170     3.570
```

`dplyr` の NSE との

`dplyr` はNSEをしています。そのため、はなしでをできます。しかし、にはデータパイプラインに、Shinyのボックスなどのソースからをするがあります。 `select` のようなの、 `select_` をってをってすることが出来ます

```
variable1 <- "Sepal.Length"
```

```
variable2 <- "Sepal.Width"
iris %>%
  select_(variable1, variable2) %>%
  head(n=5)
#   Sepal.Length Sepal.Width
# 1             5.1         3.5
# 2             4.9         3.0
# 3             4.7         3.2
# 4             4.6         3.1
# 5             5.0         3.6
```

しかし、やフィルタリングなどののをするは、 `lazyeval` パッケージの `interp` をするがあります

```
variable1 <- "Sepal.Length"
variable2 <- "Sepal.Width"
variable3 <- "Species"
iris %>%
  select_(variable1, variable2, variable3) %>%
  group_by_(variable3) %>%
  summarize_(mean1 = lazyeval::interp(~mean(var), var = as.name(variable1)), mean2 =
    lazyeval::interp(~mean(var), var = as.name(variable2)))
#   Species mean1 mean2
#   <fctr> <dbl> <dbl>
# 1   setosa 5.006 3.428
# 2 versicolor 5.936 2.770
# 3  virginica 6.588 2.974
```

オンラインでdplyrをむ <https://riptutorial.com/ja/r/topic/4250/dplyr>

9: ggplot2

ggplot2はナリファレンスWebサイト<http://ggplot2.tidyverse.org/>があります。

たいていの、プロットされたデータのや `data.frame` は、 `data.frame` プロットですよりも `data.frame` です。

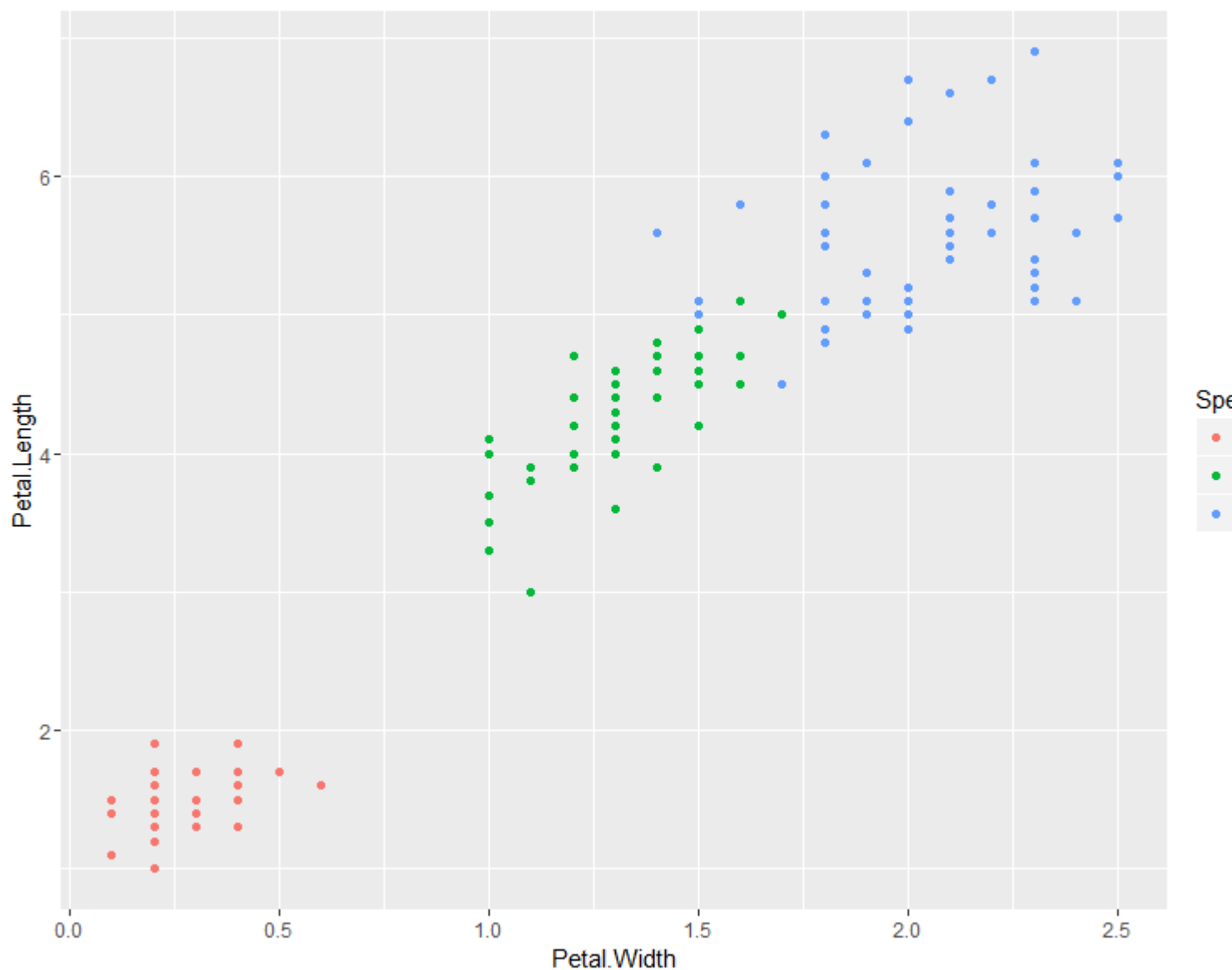
RStudioは、 [ここ](#)でつけることができるにな "ggplot2を使ったデータビジュアライゼーション"のシートシートをしています。

Examples

のように、みみデータセットをしてなをプロットします。

```
library(ggplot2)
ggplot(iris, aes(x = Petal.Width, y = Petal.Length, color = Species)) +
  geom_point()
```

これはえる

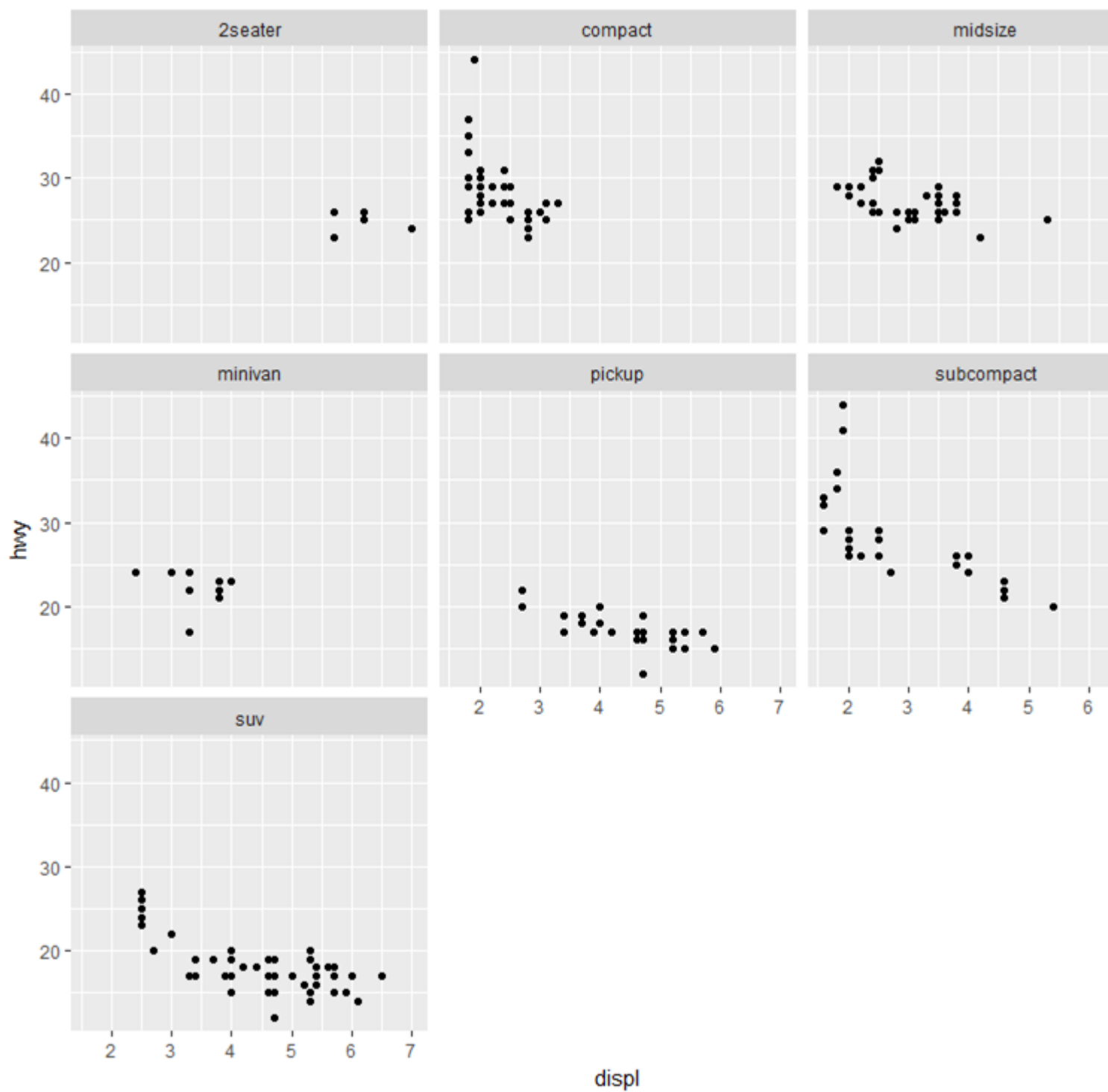


のプロットをする

なる `facet` をして1つののにプロットをする。こののは、すべてのがチャートでじスケールをするため、でそれらをする事がになります。 `ggplot2` まれる `mpg` データセットをします。

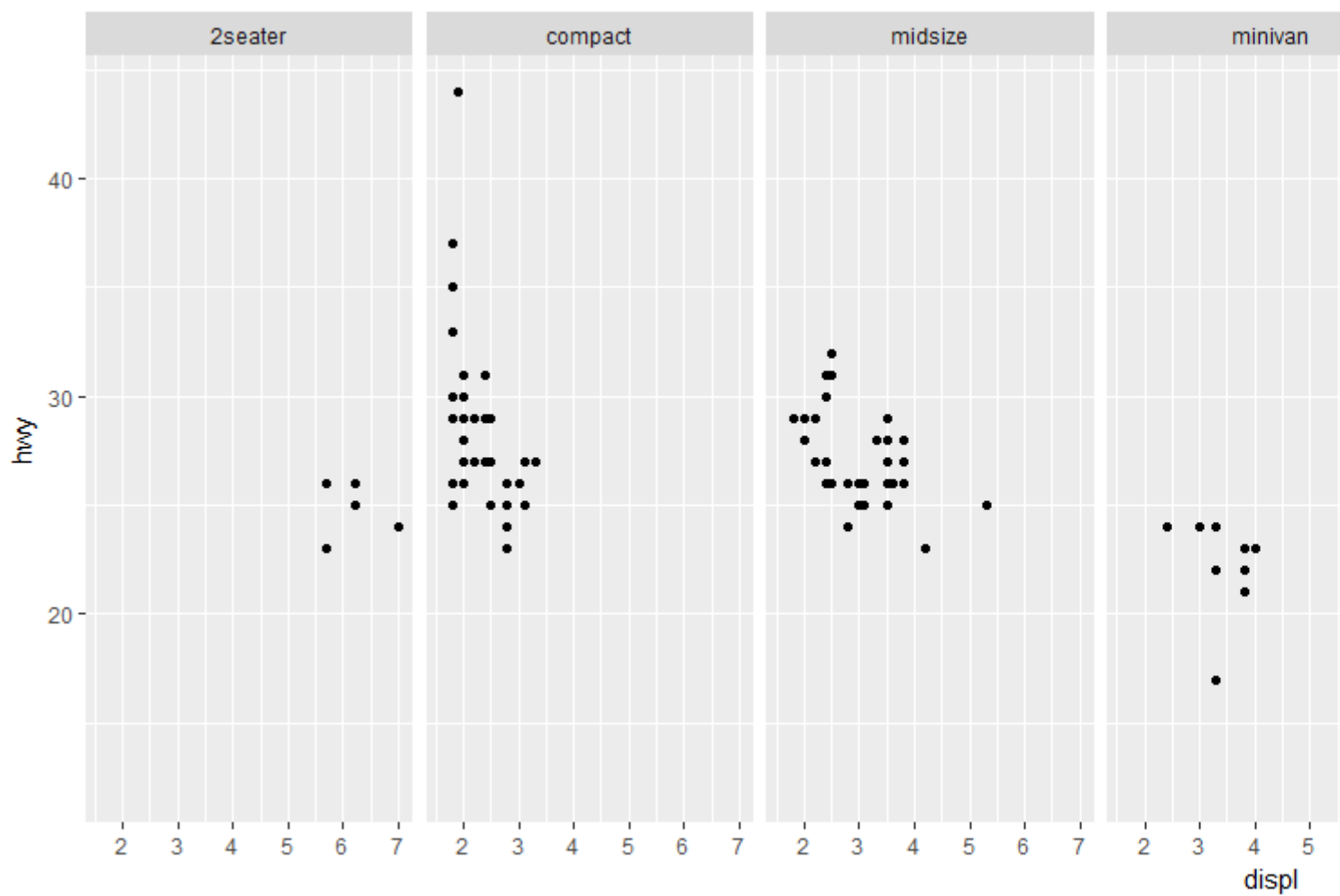
グラフを1ずつりますのレイアウトをしようします。

```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point() +  
  facet_wrap(~class)
```



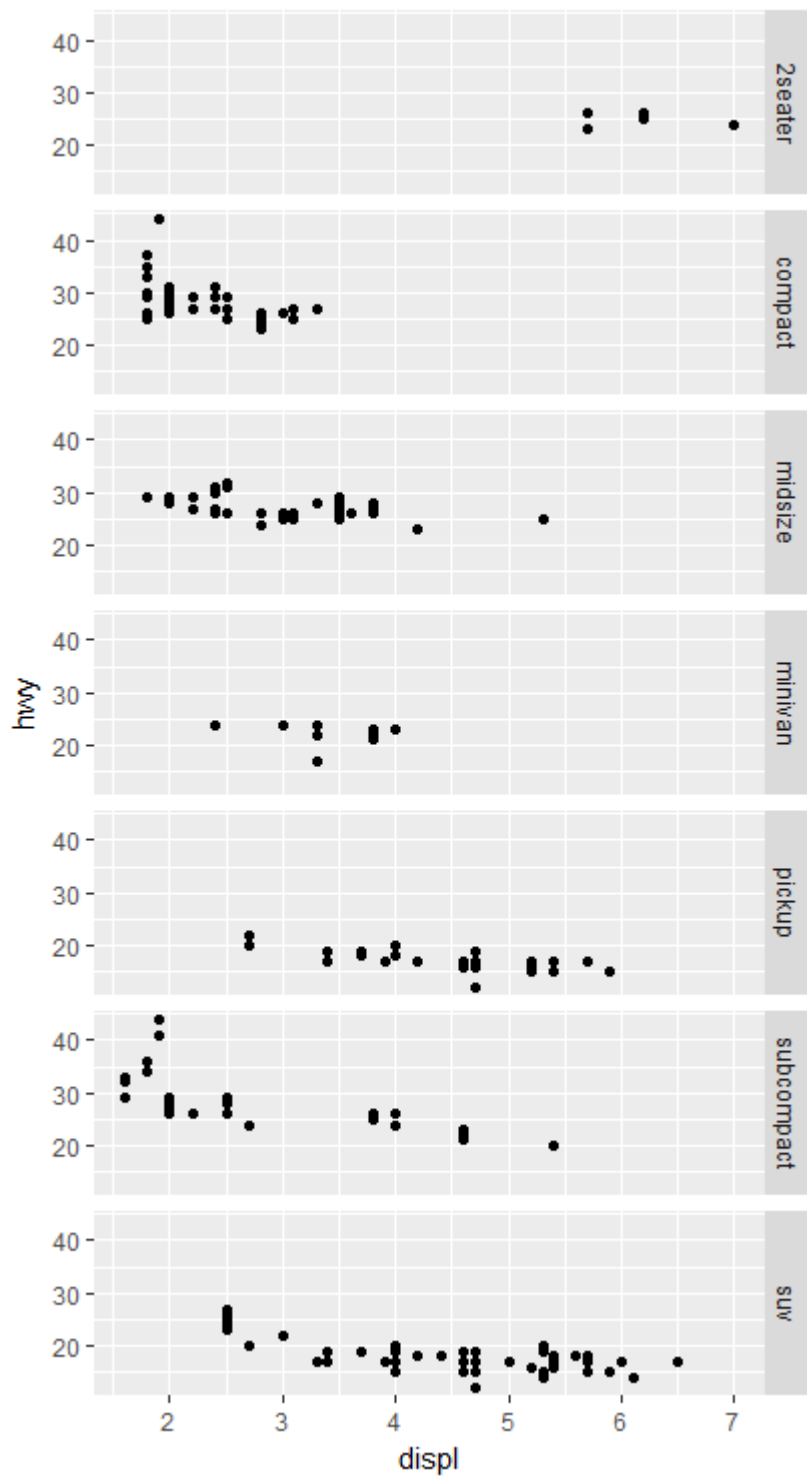
1つの、のにのグラフをする

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  facet_grid(.~class)
```



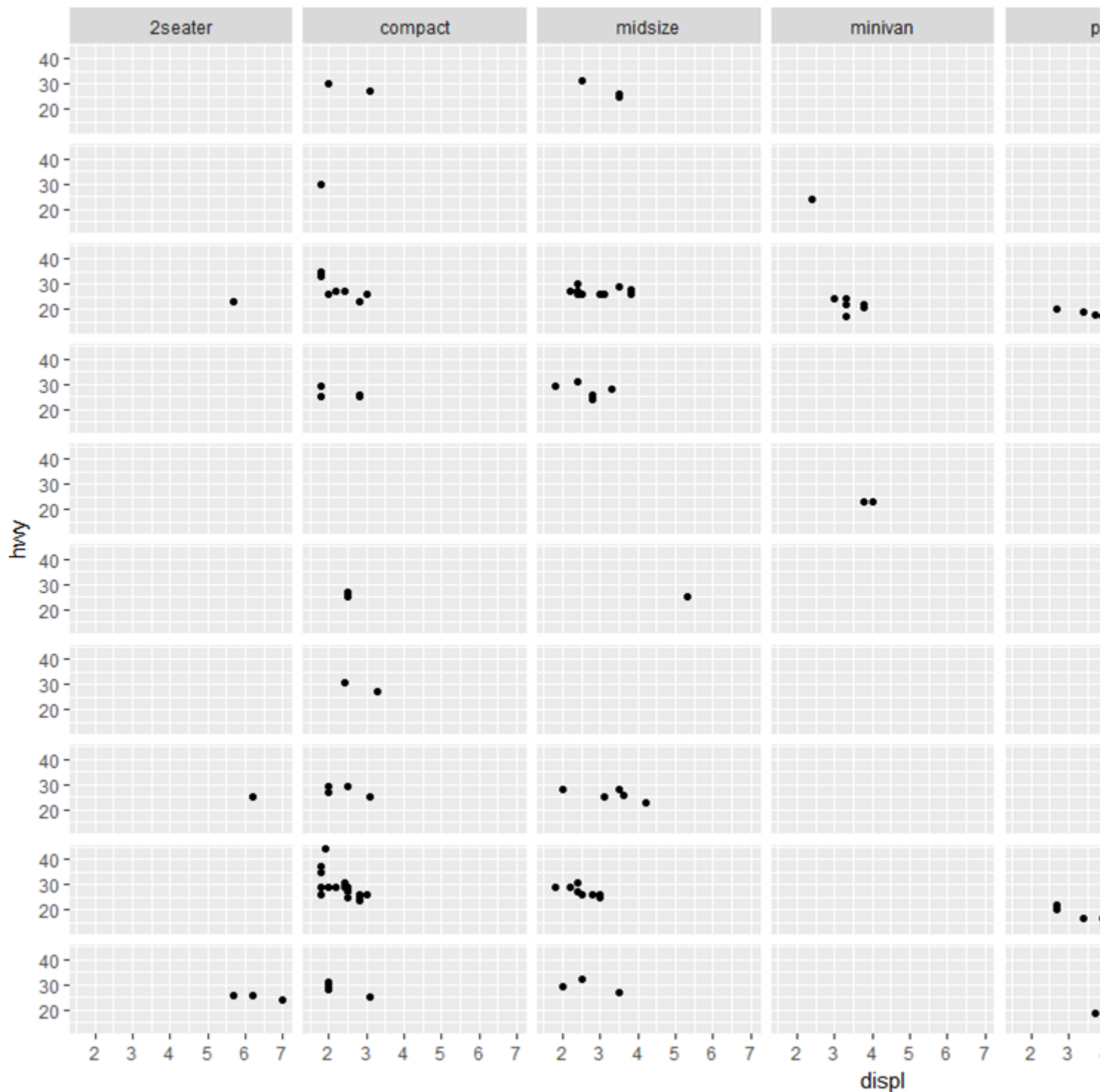
1つの、のにのグラフをする

```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point() +  
  facet_grid(class~.)
```

グリッドののグラフを2つので

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  facet_grid(trans~class) # "row" parameter, then "column" parameter
```



プロットにデータをする

ggplot2は、いデータフレームにです。のサンプルデータは、カテゴリにがあるため、くされたで20ののをします。

```
set.seed(47)
sweetsWide <- data.frame(date      = 1:20,
                          chocolate = runif(20, min = 2, max = 4),
                          iceCream  = runif(20, min = 0.5, max = 1),
                          candy     = runif(20, min = 1, max = 3))

head(sweetsWide)
```

```
##   date chocolate  iceCream   candy
## 1    1  3.953924  0.5890727 1.117311
## 2    2  2.747832  0.7783982 1.740851
## 3    3  3.523004  0.7578975 2.196754
## 4    4  3.644983  0.5667152 2.875028
## 5    5  3.147089  0.8446417 1.733543
## 6    6  3.382825  0.6900125 1.405674
```

する `sweetsWide` いとにするためのフォーマットに `ggplot2`、Rからのいくつかのな、およびパッケージ `reshape2`、`data.table` び `tidyr` ですることができます。

```
# reshape from base R
sweetsLong <- reshape(sweetsWide, idvar = 'date', direction = 'long',
                      varying = list(2:4), new.row.names = NULL, times = names(sweetsWide)[-1])

# melt from 'reshape2'
library(reshape2)
sweetsLong <- melt(sweetsWide, id.vars = 'date')

# melt from 'data.table'
# which is an optimized & extended version of 'melt' from 'reshape2'
library(data.table)
sweetsLong <- melt(setDT(sweetsWide), id.vars = 'date')

# gather from 'tidyr'
library(tidyr)
sweetsLong <- gather(sweetsWide, sweet, price, chocolate:candy)
```

すべてがのをします

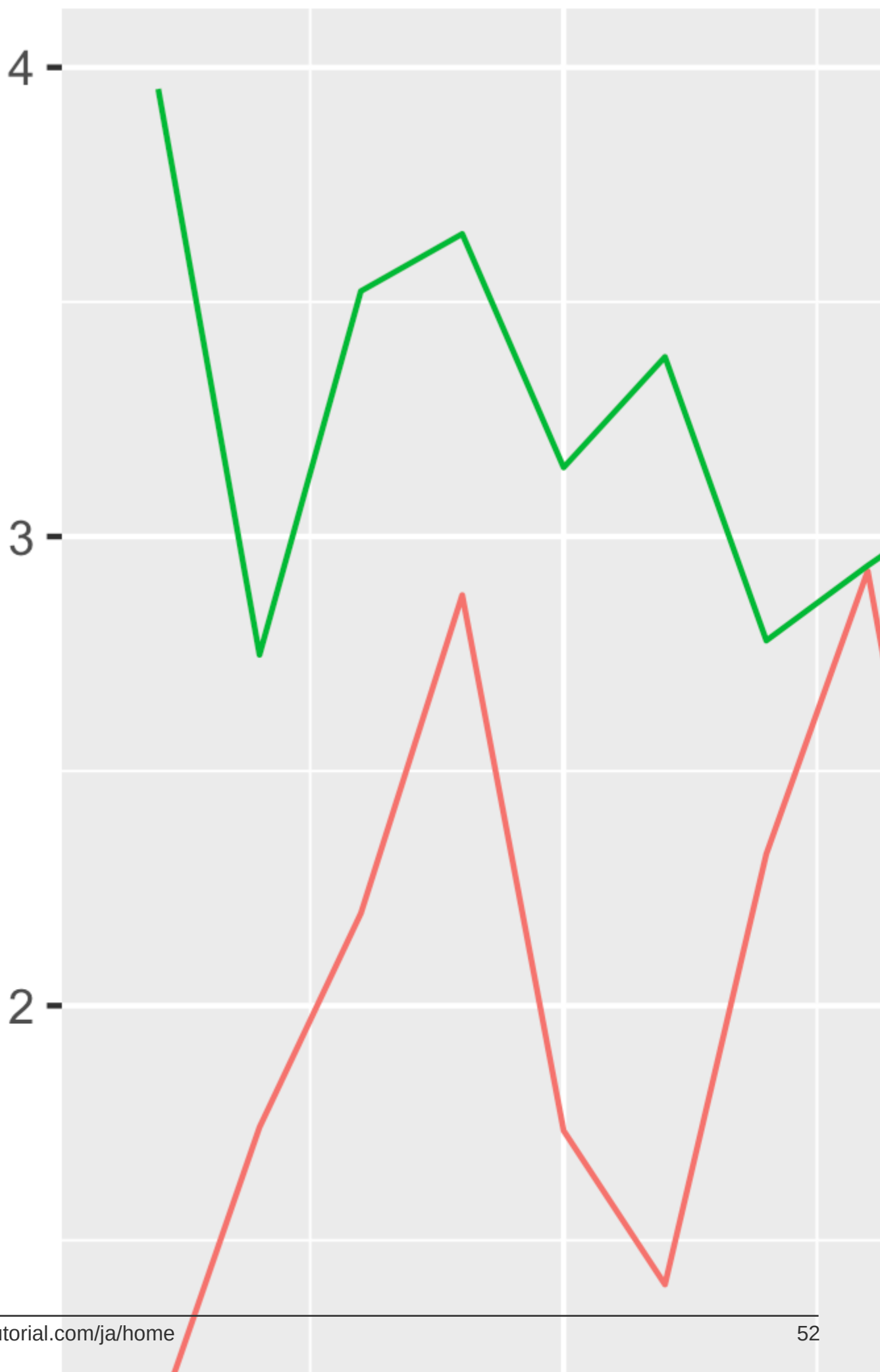
```
head(sweetsLong)
##   date    sweet   price
## 1    1 chocolate 3.953924
## 2    2 chocolate 2.747832
## 3    3 chocolate 3.523004
## 4    4 chocolate 3.644983
## 5    5 chocolate 3.147089
## 6    6 chocolate 3.382825
```

してください **いといフォームのデータリシェイプ** いとワイドフォーマットのでデータをのについては、を。

としてられる `sweetsLong` は、のとスイーツのをすがあります。プロットははるかにです

```
library(ggplot2)
ggplot(sweetsLong, aes(x = date, y = price, colour = sweet)) + geom_line()
```

price



ディストリビューションをべてするためにできます。ミラーリングは、いをするのにちます。

```
ggplot(diamonds, aes(cut, price)) +  
  geom_violin()
```

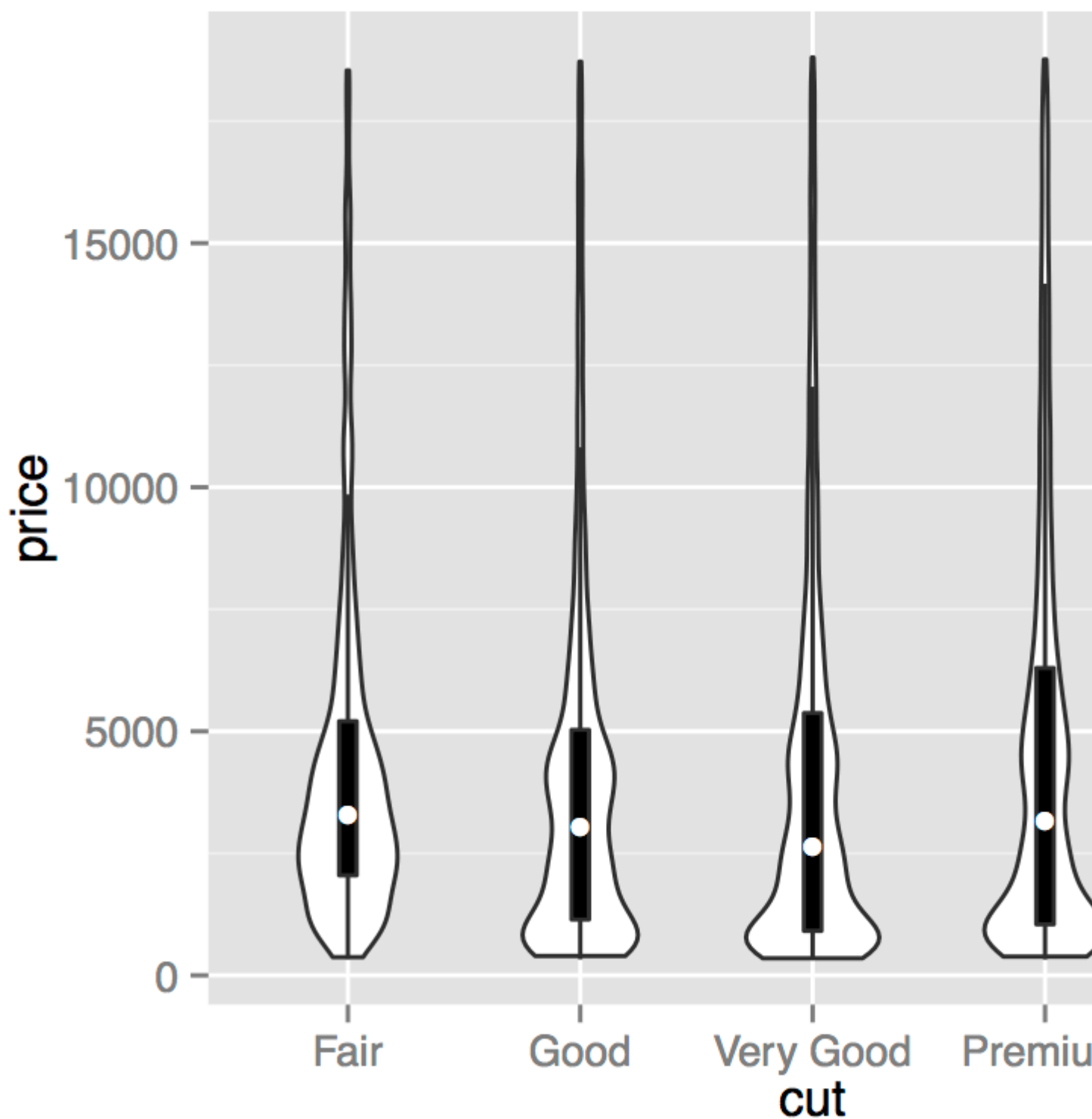
price

15000

10000

ボックスプロットとされたときににえます。これは、Tukeyの5つのボックスプロットとして
となのもりバイオリンののなをする。

```
ggplot(diamonds, aes(cut, price)) +  
  geom_violin() +  
  geom_boxplot(width = .1, fill = "black", outlier.shape = NA) +  
  stat_summary(fun.y = "median", geom = "point", col = "white")
```

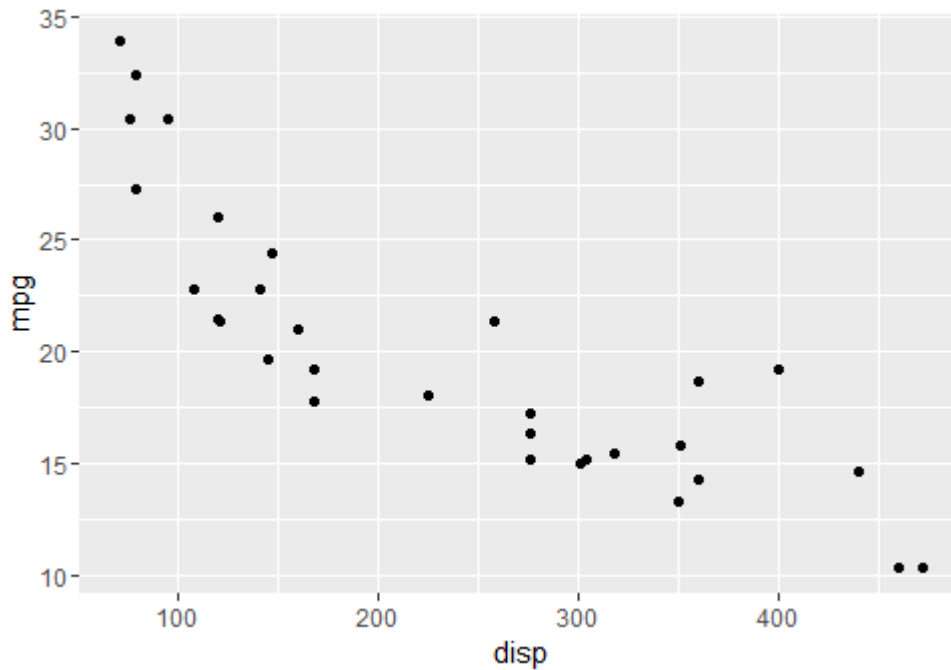


qplotを使ってプロットをする

qplotは、あまりくのをとせずにデータをにプロットしようとする、base `r_plot()` にています。

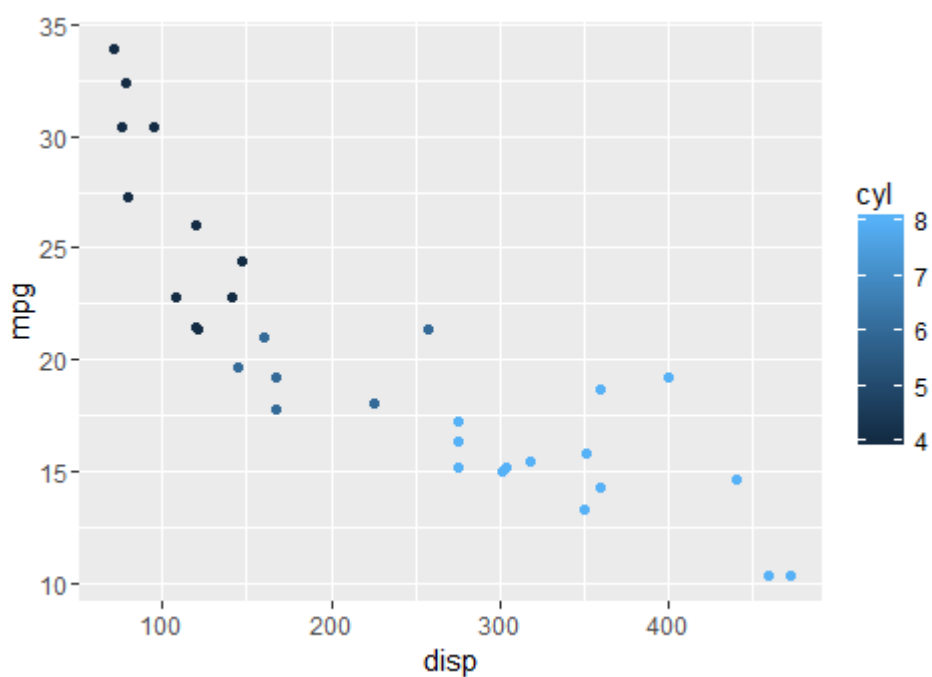
なqplot

```
qplot(x = disp, y = mpg, data = mtcars)
```



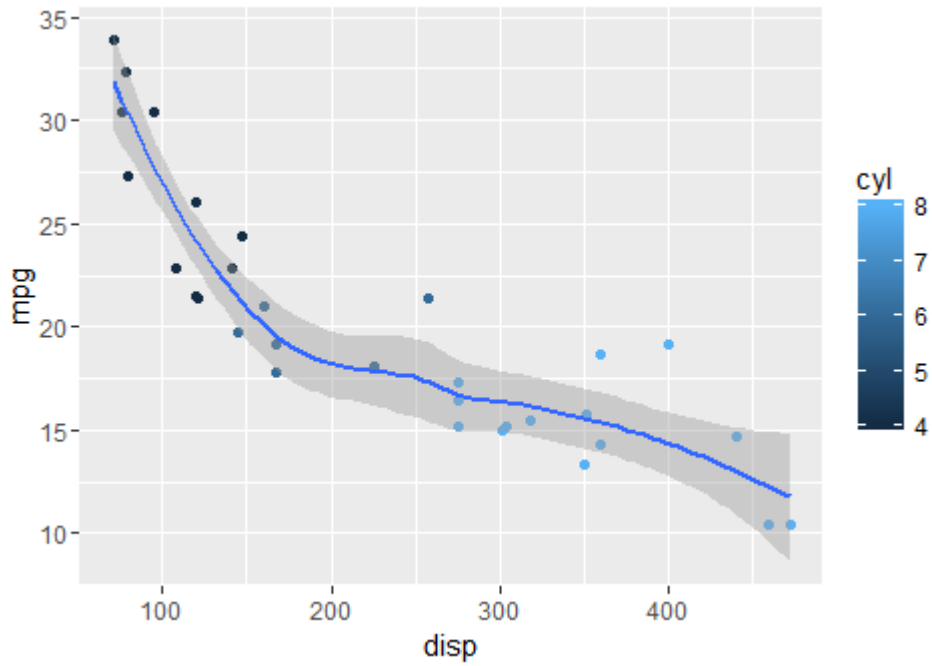
をする

```
qplot(x = disp, y = mpg, colour = cyl, data = mtcars)
```



よりスムーズに

```
ggplot(x = disp, y = mpg, geom = c("point", "smooth"), data = mtcars)
```



オンラインでggplot2をむ <https://riptutorial.com/ja/r/topic/1334/ggplot2>

10: GPUコンピューティング

GPUコンピューティングには、ハードウェアにしてできる「プラットフォーム」がです。これをする2つのなレベルは、CUDAとOpenCLです。はのNVIDIA CUDA Toolkitをインストールするがあり、NVIDIA GPUにのみされます。はNVIDIA、AMD、IntelなどとハードウェアにしないCPUまたはGPUのですが、SDKソフトウェアキットのインストールがです。RでGPUをするには、まずこれらのソフトウェアの1つをインストールするがあります。

CUDA ToolkitまたはOpenCL SDKがインストールされたら、なRパッケージをインストールできます。ほとんどすべてのR GPUパッケージはCUDAにし、NVIDIA GPUにされています。これらには、

1. [gputools](#)
2. [cudaBayesreg](#)
3. [HiPLARM](#)
4. [グマトリックス](#)

、2つのOpenCLパッケージしかありません

1. [OpenCL](#) - RからOpenCLへのインタフェース
2. [gpuR](#) - ライブラリ

- となるとGPUプラットフォームをつなげるオペレーティングシステムでは、インストールがしいがあります。

Examples

gpuR gpuMatrix オブジェクト

```
library(gpuR)

# gpuMatrix objects
X <- gpuMatrix(rnorm(100), 10, 10)
Y <- gpuMatrix(rnorm(100), 10, 10)

# transfer data to GPU when operation called
# automatically copied back to CPU
Z <- X %*% Y
```

gpuR vclMatrix オブジェクト

```
library(gpuR)

# vclMatrix objects
X <- vclMatrix(rnorm(100), 10, 10)
Y <- vclMatrix(rnorm(100), 10, 10)
```

```
# data always on GPU
# no data transfer
Z <- X %*% Y
```

オンラインでGPUコンピューティングをむ <https://riptutorial.com/ja/r/topic/4680/gpuコンピューティング>

11: hclustによるクラスタリング

き

statsパッケージはhclustをし、クラスタリングをします。

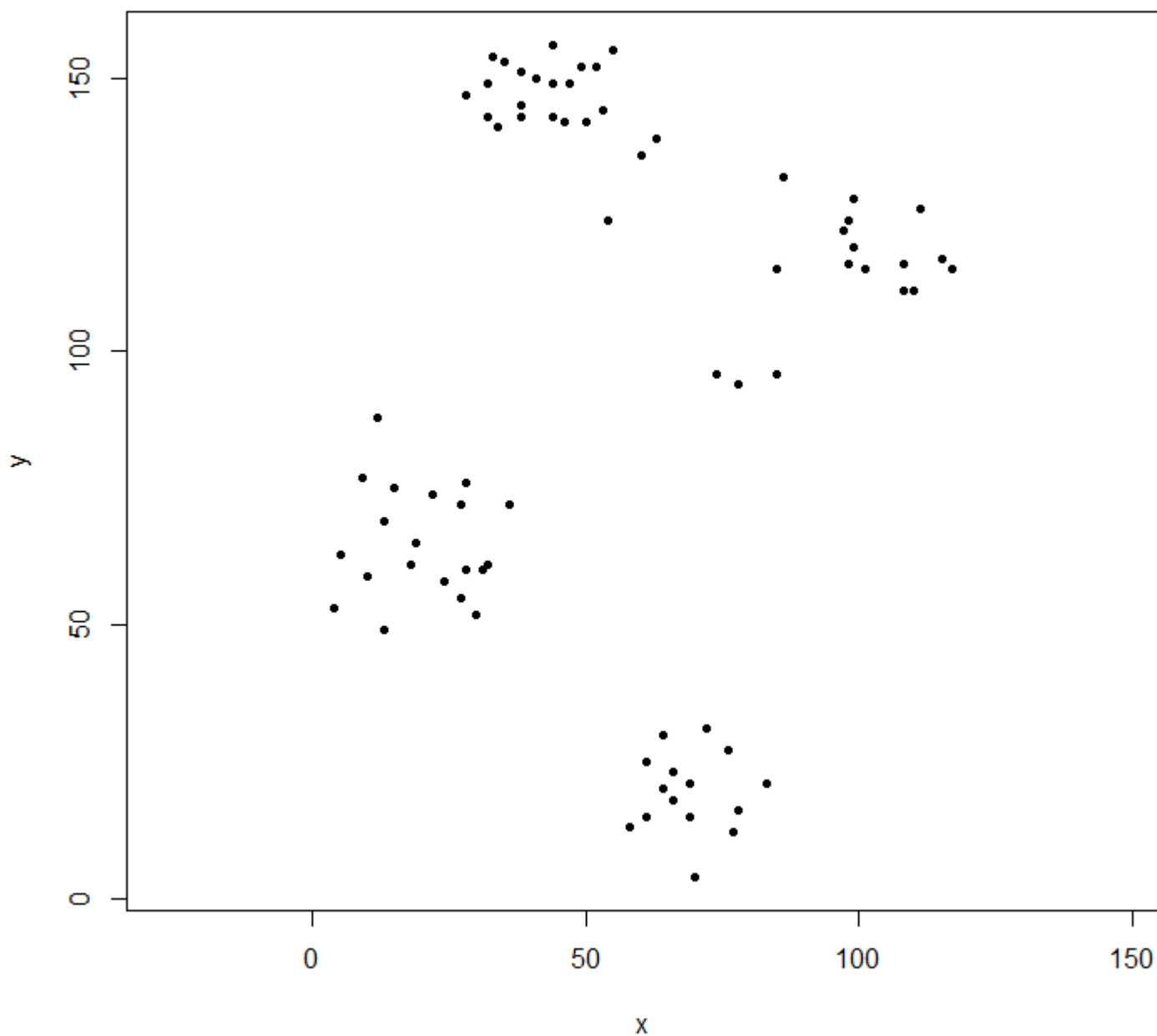
hclustにも、のがあります。「[クラスタリングのCRANパッケージの](#)」をしてください。

Examples

1 - hclustのない、デンドログラムの、プロットクラスタ

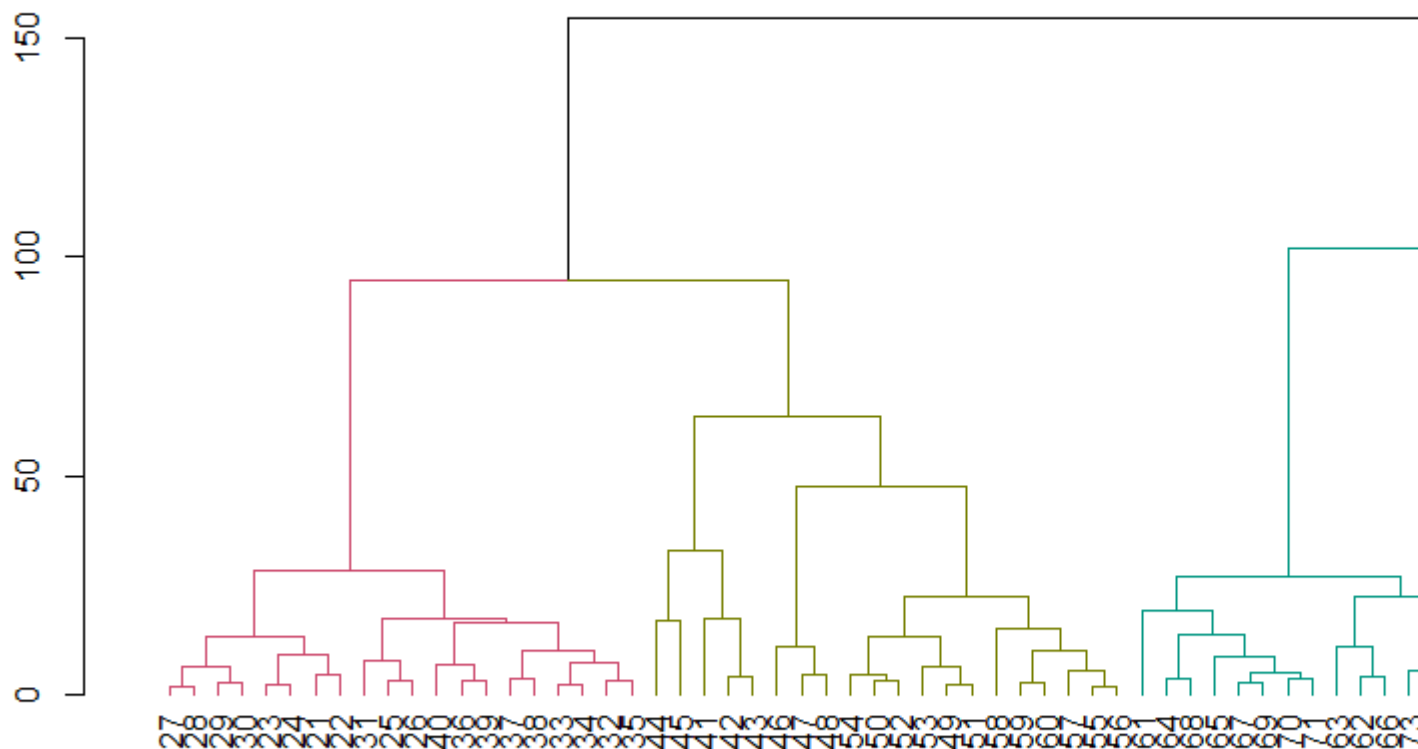
クラスタライブラリにはruspiniデータがまれています。これはクラスタをするためのデータセットです。

```
library(cluster)          ## to get the ruspini data
plot(ruspini, asp=1, pch=20) ## take a look at the data
```



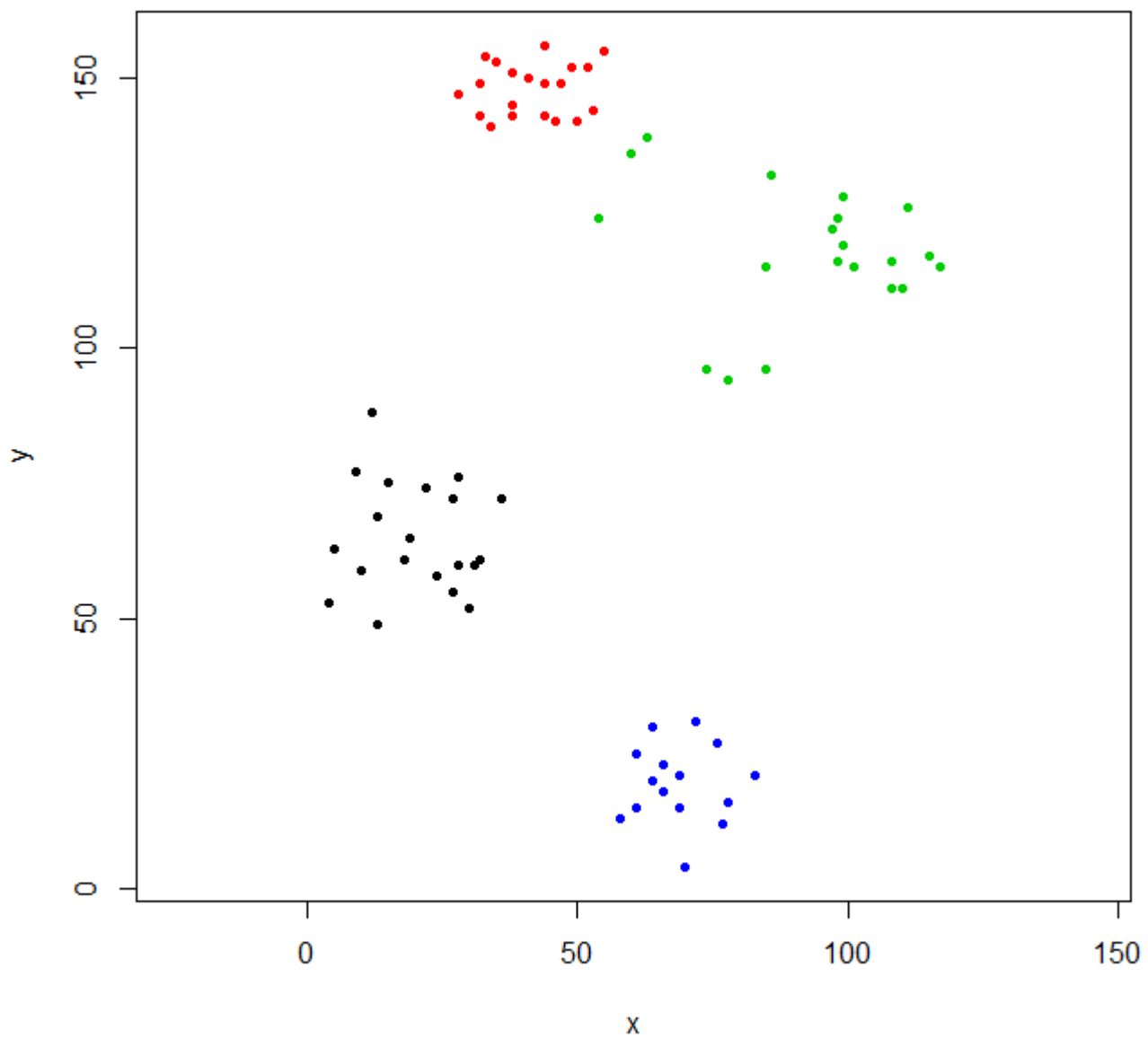
`hclust`はのデータではなく、をめます。デフォルトのパラメータをしてツリーをしてします。ハングパラメータは、ベースラインによってツリーのすべてののをべます。

```
ruspini_hc_defaults <- hclust(dist(ruspini))
dend <- as.dendrogram(ruspini_hc_defaults)
if(!require(dendextend)) install.packages("dendextend"); library(dendextend)
dend <- color_branches(dend, k = 4)
plot(dend)
```



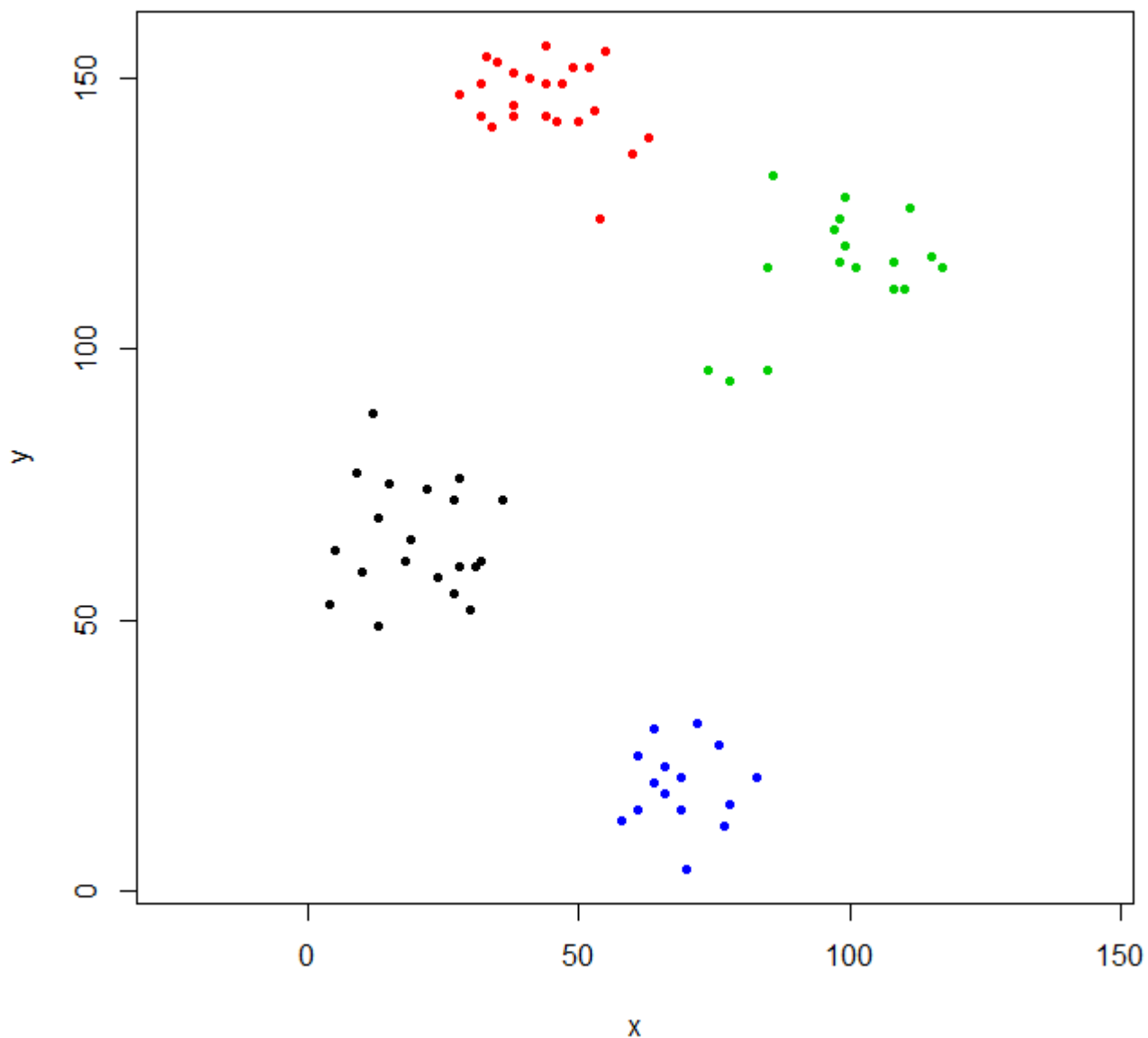
ツリーをカットして4つのクラスターをえ、クラスターごとにをりつぶすデータをきえます。kはのクラスターである。

```
rhc_def_4 = cutree(ruspini_hc_defaults,k=4)
plot(ruspini, pch=20, asp=1, col=rhc_def_4)
```



このクラスタリングはちょっとです。データをにスケーリングすることで、よりいいクラスタリングをすることができます。

```
scaled_ruspini_hc_defaults = hclust(dist(scale(ruspini)))  
srhc_def_4 = cutree(scaled_ruspini_hc_defaults,4)  
plot(ruspini, pch=20, asp=1, col=srhc_def_4)
```



クラスタをするためのデフォルトのは「」です。methodパラメータでのメジャーをできます。

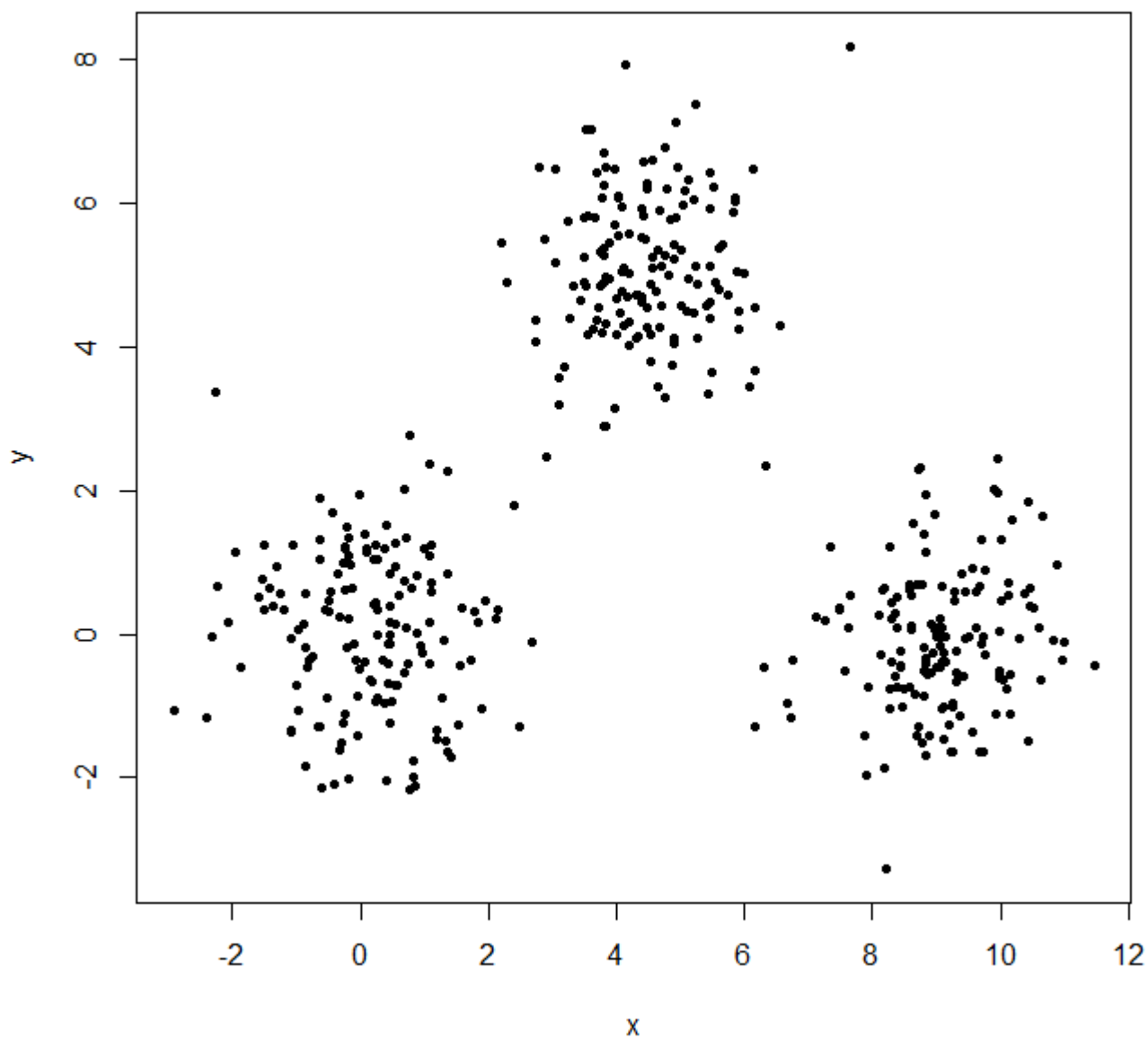
```
ruspini_hc_single = hclust(dist(ruspini), method="single")
```

2 - hclustとれ

クラスタリングでは、れは1クラスターとしてされることがよくあります。

のをするために、3つのガウスをします。

```
set.seed(656)
x = c(rnorm(150, 0, 1), rnorm(150, 9, 1), rnorm(150, 4.5, 1))
y = c(rnorm(150, 0, 1), rnorm(150, 0, 1), rnorm(150, 5, 1))
XYdf = data.frame(x, y)
plot(XYdf, pch=20)
```

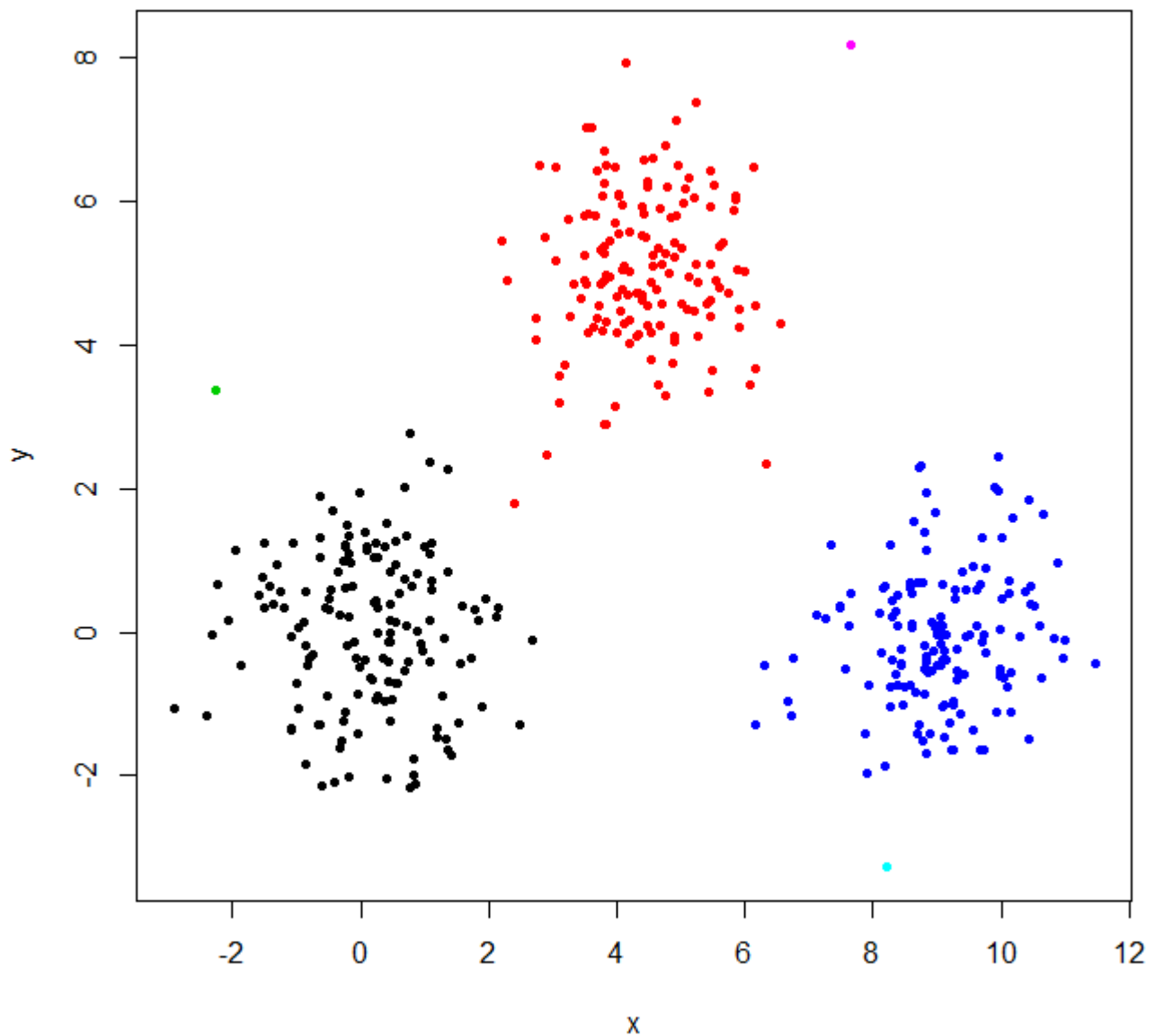
クラスタをし、3つのクラスタにします。

```
XY_sing = hclust(dist(XYdf), method="single")
XYs3 = cutree(XY_sing, k=3)
table(XYs3)
XYs3
  1   2   3
448  1   1
```

hclustは2つのをつけし、すべてをのきなクラスタにれました。「の」クラスタをるには、kをくするがあります。

```
XYs6 = cutree(XY_sing, k=6)
table(XYs6)
XYs6
  1   2   3   4   5   6
```

```
148 150 1 149 1 1
plot(XYdf, pch=20, col=XYs6)
```



この[StackOverflowポスト](#)には、クラスタのをするについてのガイダンスがありますが、クラスタリングではこのにしてください。

オンラインでhclustによるクラスタリングをむ <https://riptutorial.com/ja/r/topic/8084/hclustによるクラスタリング>

12: igraphパッケージによるネットワーク

Examples

シンプルなおよびネットワークのグラフ

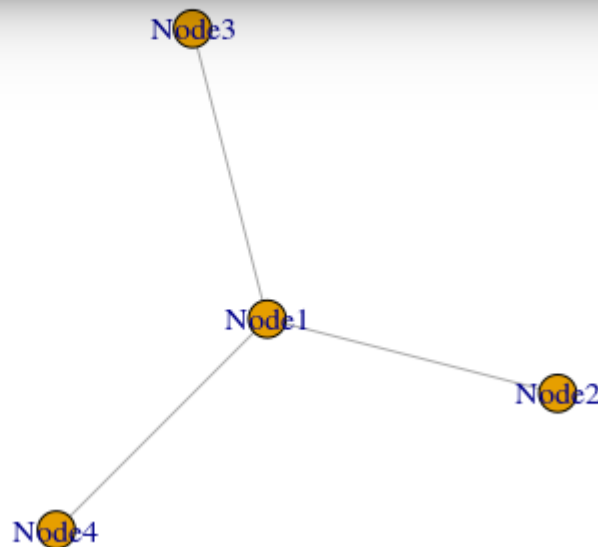
Rのためのigraphパッケージは、シンプルでどのようなネットワークをモデルするためにできるらしいツールです。これは、R v.3.2.3のigraphパッケージをして、2つのなネットワークグラフをするをしています。

ネットワーク

ネットワークはこのコードでされます

```
g<-graph.formula(Node1-Node2, Node1-Node3, Node4-Node1)
plot(g)
```

```
> g<-graph.formula(Node1-Node2, Node1-Node3, Node4-Node1)
> plot(g)
> 
```

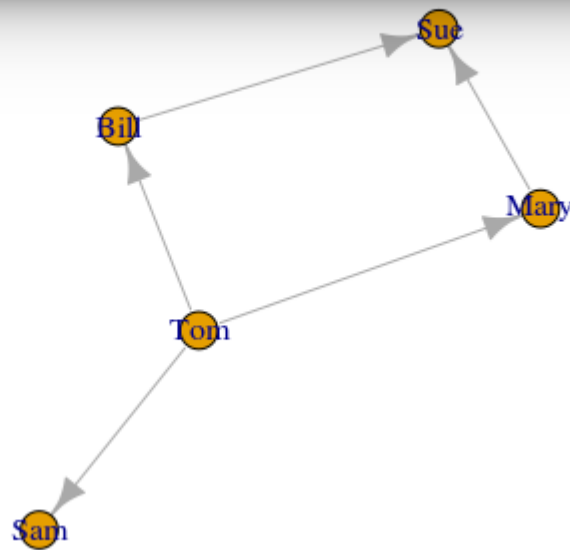


されたネットワーク

```
dg<-graph.formula(Tom->Mary, Tom->Bill, Tom->Sam, Sue->Mary, Bill->Sue)
plot(dg)
```

このコードは、きのネットワークをします。

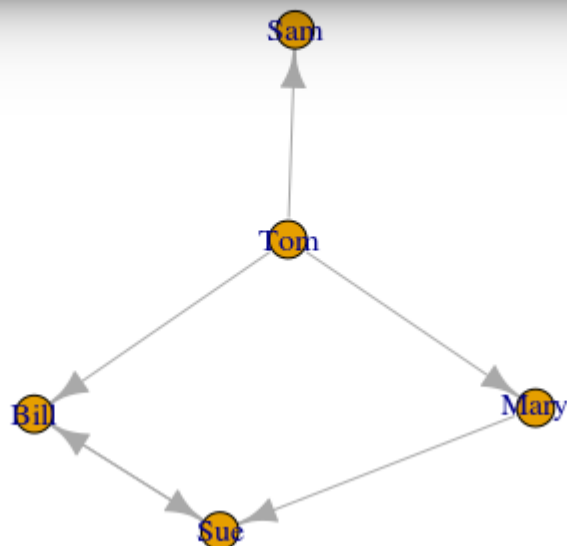
```
> dg<-graph.formula(Tom->Mary, Tom->Bill, Tom->Sam, Sue->Mary, Bill->Sue)
> plot(dg)
>
```



のコード

```
dg<-graph.formula(Tom->Mary, Tom->Bill, Tom->Sam, Sue->Mary, Bill->Sue)
plot(dg)
```

```
> dg<-graph.formula(Tom->Mary, Tom->Bill, Tom->Sam, Sue->Mary, Bill->Sue)
> plot(dg)
>
```



オンラインでigraphパッケージによるネットワークをむ

<https://riptutorial.com/ja/r/topic/4851/igraph>パッケージによるネットワーク

13: JSON

Examples

JSONからRオブジェクトへの/からの

[jsonlite](#)パッケージは、データとWebにされたのJSONパーサとジェネレータです。JSONのみきにされる2つのなは、JSON `fromJSON()` と `toJSON()`。Webからの `vectors`、`matrices`、`data.frames`、JSONのストリームを `vectors` ようにされています。

ベクターからJSONをするも

```
library(jsonlite)

## vector to JSON
toJSON(c(1,2,3))
# [1,2,3]

fromJSON('[1,2,3]')
# [1] 1 2 3
```

リストからきJSONをするも

```
toJSON(list(myVec = c(1,2,3)))
# {"myVec":[1,2,3]}

fromJSON('{"myVec":[1,2,3]}')
# $myVec
# [1] 1 2 3
```

よりなリスト

```
## list structures
lst <- list(a = c(1,2,3),
           b = list(letters[1:6]))

toJSON(lst)
# {"a":[1,2,3],"b":[["a","b","c","d","e","f"]]}

fromJSON('{"a":[1,2,3],"b":[["a","b","c","d","e","f"]]} ')
# $a
# [1] 1 2 3
#
# $b
# [,1] [,2] [,3] [,4] [,5] [,6]
# [1,] "a"  "b"  "c"  "d"  "e"  "f"
```

`data.frame` からJSONをし、そのもです

```
## converting a data.frame to JSON
df <- data.frame(id = seq_along(1:10),
                 val = letters[1:10])

toJSON(df)
#
[{"id":1,"val":"a"}, {"id":2,"val":"b"}, {"id":3,"val":"c"}, {"id":4,"val":"d"}, {"id":5,"val":"e"}, {"id":6,"val":"f"}, {"id":7,"val":"g"}, {"id":8,"val":"h"}, {"id":9,"val":"i"}, {"id":10,"val":"j"}]

## reading a JSON string
fromJSON(' [{"id":1,"val":"a"}, {"id":2,"val":"b"}, {"id":3,"val":"c"}, {"id":4,"val":"d"}, {"id":5,"val":"e"}, {"id":6,"val":"f"}, {"id":7,"val":"g"}, {"id":8,"val":"h"}, {"id":9,"val":"i"}, {"id":10,"val":"j"} ]')

#      id val
# 1     1  a
# 2     2  b
# 3     3  c
# 4     4  d
# 5     5  e
# 6     6  f
# 7     7  g
# 8     8  h
# 9     9  i
# 10    10 j
```

JSONをインターネットからみむ

```
## Reading JSON from URL
googleway_issues <- fromJSON("https://api.github.com/repos/SymbolixAU/googleway/issues")

googleway_issues$url
# [1] "https://api.github.com/repos/SymbolixAU/googleway/issues/20"
# [2] "https://api.github.com/repos/SymbolixAU/googleway/issues/19"
# [3] "https://api.github.com/repos/SymbolixAU/googleway/issues/14"
# [4] "https://api.github.com/repos/SymbolixAU/googleway/issues/11"
# [5] "https://api.github.com/repos/SymbolixAU/googleway/issues/9"
# [6] "https://api.github.com/repos/SymbolixAU/googleway/issues/5"
# [7] "https://api.github.com/repos/SymbolixAU/googleway/issues/2"
```

オンラインでJSONをむ <https://riptutorial.com/ja/r/topic/2460/json>

14: LaTeXのR with knitr

1. <<コードチャンク、オプション... >> =
Rコードここ
@
2. \Sexpr {ここで#Rコード}
3. <<読み-Rファイル>> =
read_chunk 'r-file.R'
@
<<コードチャンク、オプション... >> =
@

パラメーター

オプション	
エコー	TRUE / FALSE - Rソースコードをファイルにめるかどうか
メッセージ	TRUE / FALSE - Rソースのからのメッセージをファイルにめるかどうか
	TRUE / FALSE - Rソースのからのをファイルにめるかどうか
エラー	TRUE / FALSE - Rソースのからのエラーをファイルにめるかどうか
キャッシュ	TRUE / FALSE - Rソースのをキャッシュするかどうか
fig.width	- Rソースのによってされたプロットの
fig.height	- Rソースのによってされたプロットのさ

KnitrはLaTeXとソースコードRをりぜるためのツールです。に、とソースコードがしているというは、 [プログラミング](#)とばれています。 knitrファイルにはLaTeXに.texファイルにされているとRに.Rファイルにされているがしているため、R noweb.Rnwというしいファイルがです。 .RnwファイルにはLaTeXとRコードがしています。

Knitrは、PDFのレポートのをにし、 [のある](#)をするためのなツールです。

.RnwファイルをPDFにコンパイルすることは、2つのステップのプロセスです。まず、Rコードをどのようにし、LaTeXコンパイラができる「ニッティング」とばれるプロセスでをキャプチャするかをるがあります。これはknitrパッケージをっています。あなたが[knitrパッケージをインストールした](#)として、このコマンドをにします

```
Rscript -e "library(knitr); knit('r-noweb-file.Rnw')
```

これにより、の.texファイルこのではr-noweb.texとばれますがされます。このファイルは、をしてPDFファイルにできます。

```
pdflatex r-noweb-file.tex
```

Examples

ラテックスでのニットとコードのによるR

Knitrは、RコードとLaTeXコードのをにするRパッケージです。これをする1つののは、コードチャ
ンクです。コードチャンクをすると、RでRスクリプトを/テストし、そのをレポートにめること
ができます。これはなです。このアプローチをにします。

```
# r-noweb-file.Rnw
\documentclass{article}

<<echo=FALSE,cache=FALSE>>=
knitr::opts_chunk$set(echo=FALSE, cache=TRUE)
knitr::read_chunk('r-file.R')
@

\begin{document}
This is an Rnw file (R noweb). It contains a combination of LaTeX and R.

One we have called the read\_chunk command above we can reference sections of code in the r-
file.R script.

<<Chunk1>>=
@
\end{document}
```

このアプローチをするときは、にすようにコードをのRファイルにします。

```
## r-file.R
## note the specific comment style of a single pound sign followed by four dashes

# ---- Chunk1 ----

print("This is R Code in an external file")

x <- seq(1:10)
y <- rev(seq(1:10))
plot(x,y)
```

ラテックスのKnitrとインラインコードチャンクのR

Knitrは、RコードとLaTeXコードのをにするRパッケージです。これをする1つののは、インライン
コードチャンクです。このapproachはのとおりです。

```
# r-noweb-file.Rnw
\documentclass{article}
```



```

\begin{document}
This is an Rnw file (R noweb).  It contains a combination of LaTeX and R.

<<my-label>>=
print("This is an R Code Chunk")
x <- seq(1:10)
@

Above is an internal code chunk.
We can access data created in any code chunk inline with our LaTeX code like this.
The length of array x is \Sexpr{length(x)}.

\end{document}

```

KnitrとコードチャンクをつLaTeXのR

Knitrは、RコードとLaTeXコードのをにするRパッケージです。これをする1つののは、コードチャンクです。このapproachはのとおりです。

```

# r-noweb-file.Rnw
\documentclass{article}
\begin{document}
This is an Rnw file (R noweb).  It contains a combination of LaTeX and R.

<<code-chunk-label>>=
print("This is an R Code Chunk")
x <- seq(1:10)
y <- seq(1:10)
plot(x,y)  # Brownian motion
@

\end{document}

```

オンラインでLaTeXのR with knitrをむ <https://riptutorial.com/ja/r/topic/4334/latexのr-with-knitr>

15: R Markdown ノートブック RStudio

き

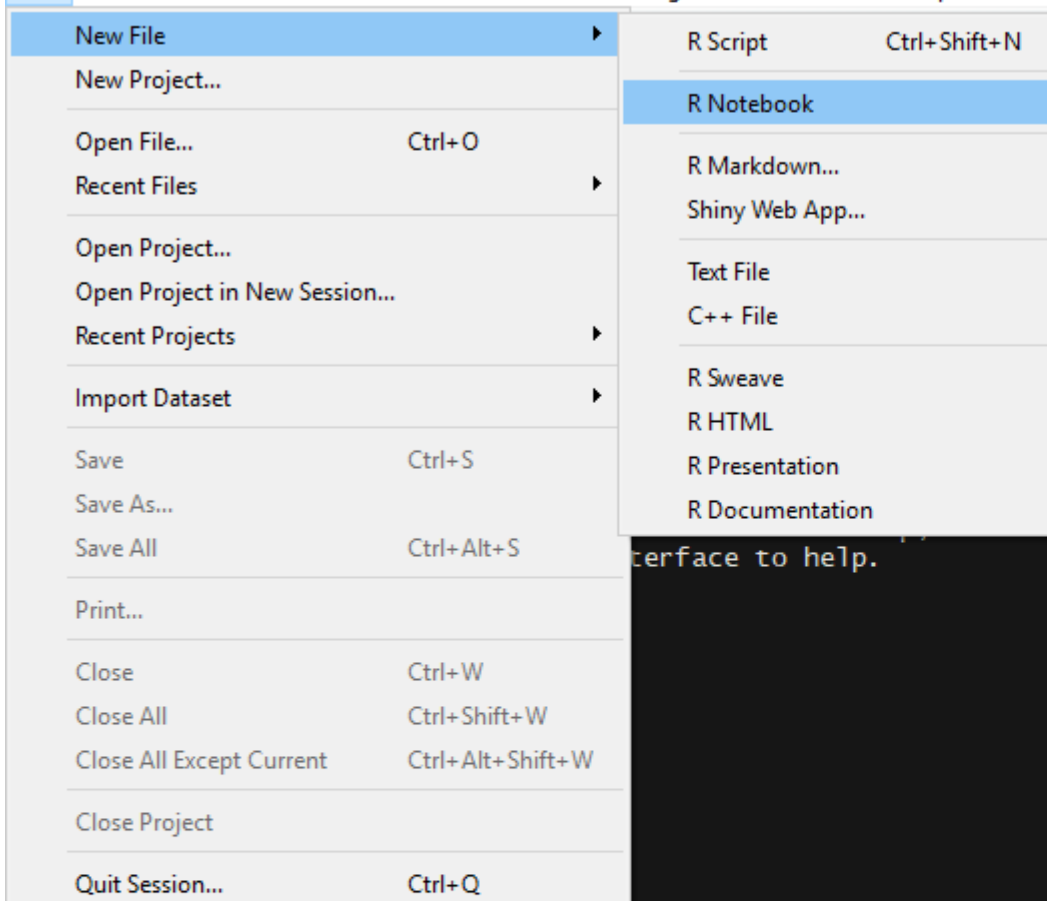
R Notebookは、してインタラクティブにできるチャンクをつR Markdownドキュメントであり、のすぐにながされます。 R Markdownドキュメントとていますが、レンダリングされたではなくR ノートブックの/モードでがされます。 RノートブックはRStudioのであり、RStudioのバージョン1.0でのみできます。

Examples

ノートブックの

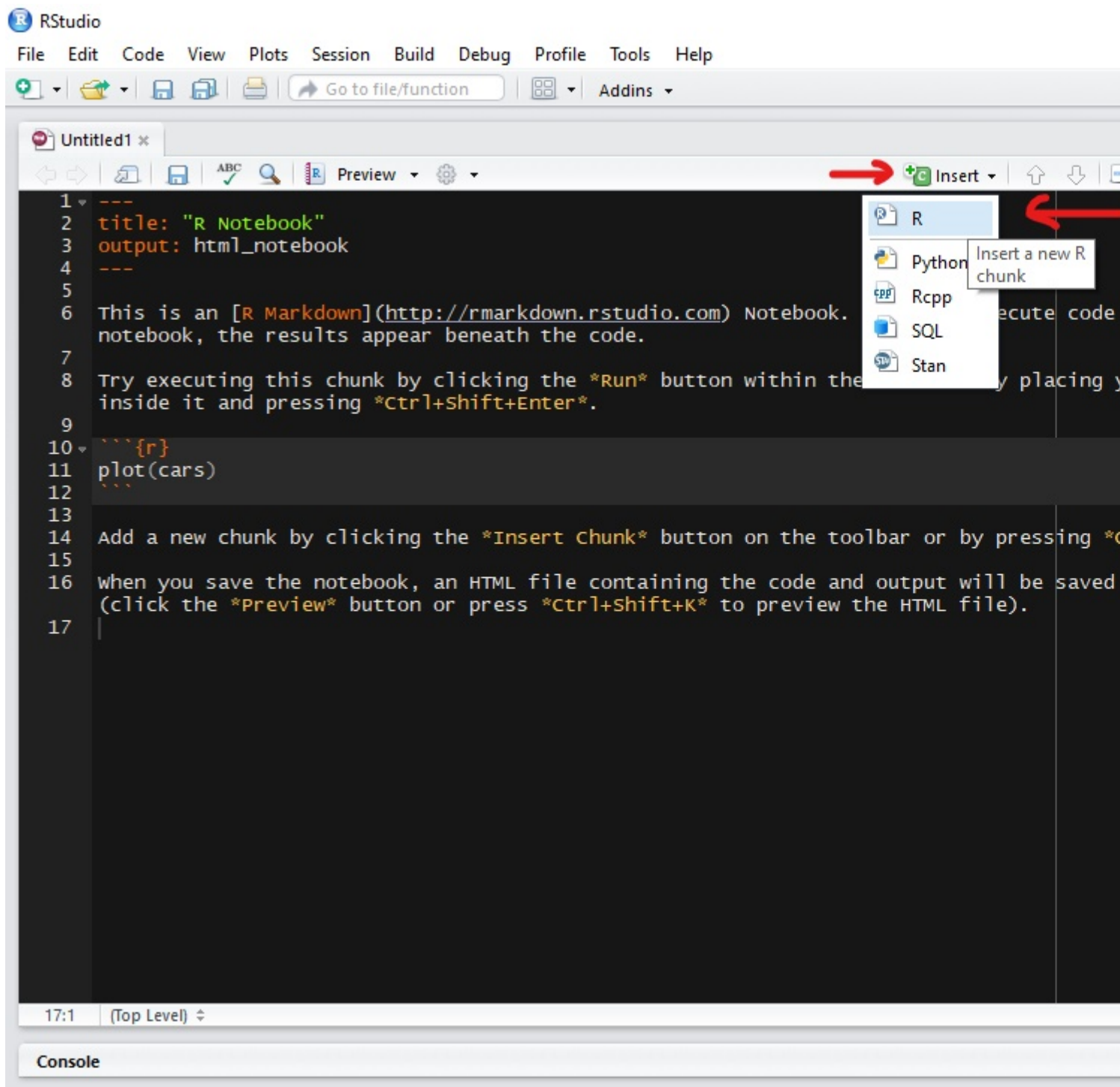
メニューコマンドのFile -> New File -> R Notebookをして、RStudioでしいノートブックをすることが出来ます

R Notebookのオプションがされないは、RStudioのバージョンをするがあります。 RStudioのインストールについては、 [このガイドにしてください](#)



チャンクの

チャンクは、インタラクティブにできるコードのです。ノートブックツールバーにあるボタンをクリックしてしいチャンクをし、するコードプラットフォームこのはRコードをきたいのでRをします。あるいは、キーボードショートカットをしてしいチャンクをすることもできます。Ctrl + Alt + IOS XCmd + Option + I



チャンクコードの

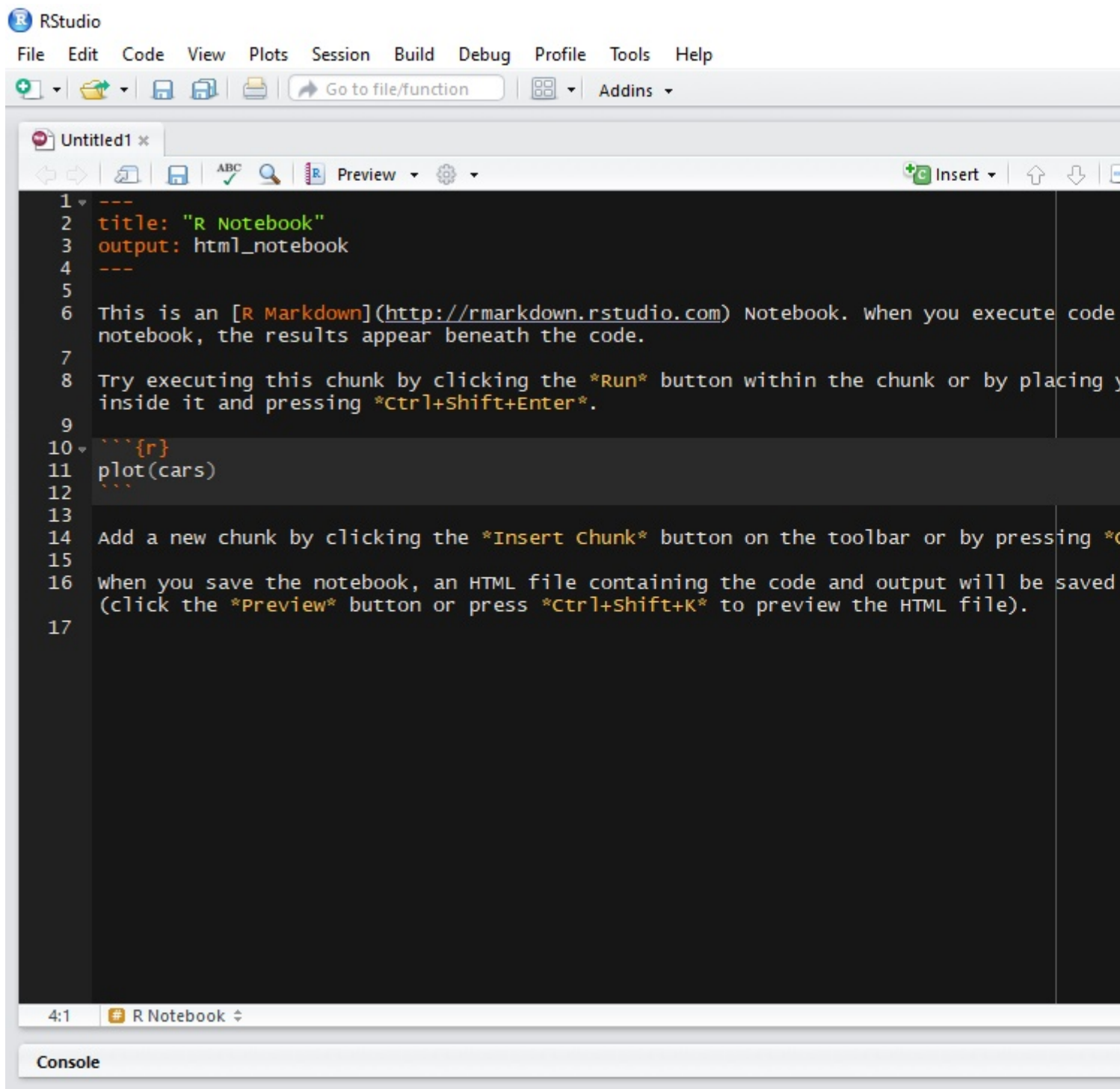
チャンクのにあるのチャンクののボタンをクリックすると、のチャンクをできます。あるいは、キーボードショートカット **Ctrl + Shift + Enter** **OS X Cmd + Shift + Enter**をして、チャンクのすべてのからのがチャンクのにされます。

コードをチャンクにする

チャンクはチャンクのにをするので、のをするのチャンクにのコードがある、チャンクが1つのを

するようにのチャンクにするとです。

これをうには、しいチャンクにして **Ctrl + Alt + IOS XCmd + Option + I** をすコードをし、



の

ノートブックでコードをすると、ガターにインジケータがされ、のがされます。Rにられたコードのはいでマークされています。まだRにられていないはいでマークされています。

のチャンクの

にあるすべてのチャンクにしてキーをして々のチャンクをまたはすることはしいことがあります。ツールバーの[]メニューから[すべて]をして、ノートブックにあるすべてのチャンクをすることができます。キーボードショートカットは、**Ctrl + Alt + ROS XCmd + Option + R**です。

また、**R**をし、すべてのチャンクをするコマンドエディタツールバーの[]メニューにありますがあり、すべてのチャンクをするにしいRセッションをします。

また、したチャンクのまたはにチャンクをするために、のすべてのチャンクをする、のすべてのチャンクをするなどのオプションがあります。

The screenshot shows the RStudio interface with a script editor and a preview window. The script editor contains the following code:

```
14 data("iris")
15 head(iris,5)
16
17 # Divide Iris data to x (containt the all features) and y (only the classes)
18 {r}
19 x <- subset(iris, select=-species)
20 y <- iris$Species
21
22
23
24 Create SVM Model and show summary
25 {r}
26 svm_model <- svm(x,y)
27
28 summary(svm_model)
29
30
31 Run Prediction
32 {r}
33 pred <- predict(svm_model,x)
34
35
36 you can time taken by using system.time
```

The preview window shows the first 5 rows of the iris dataset:

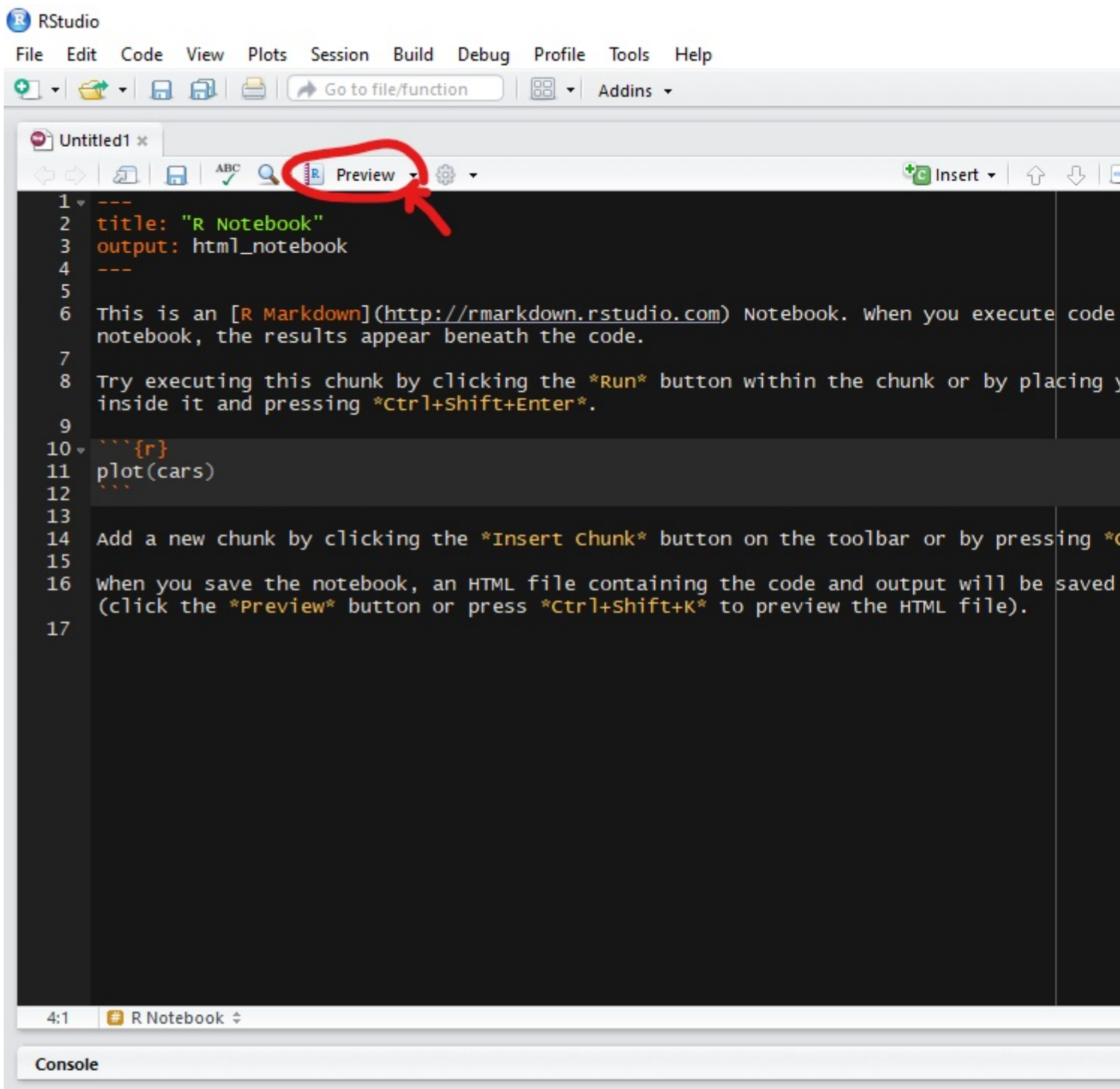
	Sepal.Length <dbl>	Sepal.Width <dbl>	Petal.Length <dbl>	Petal.Width <dbl>
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2

At the bottom of the script editor, there is a "Run All" button with a green progress bar, which is highlighted with a red arrow.

プレビュー

ノートブックのをレンダリングするに、をプレビューすることができます。ツールバーの[プレビュー]ボタンをクリックして、のをします。

「pdf_document」または「html_notebook」というオプションをして、のタイプをすることができます。



と

ノート.Rmdがされると、.nb.htmlファイルがそのにされます。このファイルは、のすべてのチャックウェブサイトにするのにしているとノートブック.RmdのコピーのをむノートブックのレンダリングされたコピーをむのHTMLファイルです。

はRStudioのドキュメントをご覧ください

オンラインでR MarkdownノートブックRStudioをむ <https://riptutorial.com/ja/r/topic/10728/r-markdownノートブック-rstudio->

16: Rcpp

Examples

インラインコードコンパイル

Rcppには、インラインでコードをコンパイルし、R `cppFunction()` と `evalCpp()` `cppFunction()` できる2つのがあります。ばれる3の `sourceCpp()` ただしにしていることができるのファイルのC++コードでみるために `cppFunction()`

は、RのC++をコンパイルするです。ソースをむために `""` をすることにしてください。

```
# Note - This is R code.
# cppFunction in Rcpp allows for rapid testing.
require(Rcpp)

# Creates a function that multiplies each element in a vector
# Returns the modified vector.
cppFunction("
NumericVector exfun(NumericVector x, int i){
  x = x*i;
  return x;
}")

# Calling function in R
exfun(1:5, 3)
```

C++のいをすばやくするには

```
# Use evalCpp to evaluate C++ expressions
evalCpp("std::numeric_limits<double>::max()")
## [1] 1.797693e+308
```

Rcppの

Rcpp Attributesは、RおよびC++でのプロセスをにします。のはのとおります。

```
// [[Rcpp::attribute]]
```

のは、ののものとしています。

```
// [[Rcpp::export]]
```

`sourceCpp()` してC++ファイルを `sourceCpp()` ときにされたヘッダーのにされます。

は、をするC++ファイルののです。

```
// Add code below into C++ file Rcpp_example.cpp

#include <Rcpp.h>
using namespace Rcpp;

// Place the export tag right above function declaration.
// [[Rcpp::export]]
double muRcpp(NumericVector x){

    int n = x.size(); // Size of vector
    double sum = 0; // Sum value

    // For loop, note cpp index shift to 0
    for(int i = 0; i < n; i++){
        // Shorthand for sum = sum + x[i]
        sum += x[i];
    }

    return sum/n; // Obtain and return the Mean
}

// Place dependent functions above call or
// declare the function definition with:
double muRcpp(NumericVector x);

// [[Rcpp::export]]
double varRcpp(NumericVector x, bool bias = true){

    // Calculate the mean using C++ function
    double mean = muRcpp(x);
    double sum = 0;

    int n = x.size();

    for(int i = 0; i < n; i++){
        sum += pow(x[i] - mean, 2.0); // Square
    }

    return sum/(n-bias); // Return variance
}
```

RでこのC++ファイルをするには、のをいます。

```
require(Rcpp)

# Compile File
sourceCpp("path/to/file/Rcpp_example.cpp")

# Make some sample data
x = 1:5

all.equal(muRcpp(x), mean(x))
## TRUE

all.equal(varRcpp(x), var(x))
## TRUE
```

プラグインによる**Rcpp**の

C++では、のようになるコンパイルフラグをできます。

```
// [[Rcpp::plugins(name)]]
```

みみのプラグインのリスト

```
// built-in C++11 plugin
// [[Rcpp::plugins(cpp11)]]

// built-in C++11 plugin for older g++ compiler
// [[Rcpp::plugins(cpp0x)]]

// built-in C++14 plugin for C++14 standard
// [[Rcpp::plugins(cpp14)]]

// built-in C++1y plugin for C++14 and C++17 standard under development
// [[Rcpp::plugins(cpp1y)]]

// built-in OpenMP++11 plugin
// [[Rcpp::plugins(openmp)]]
```

のの

Rcppエコシステムののにのパッケージをするには、しいヘッダファイルがないかもしれません
Rcpp.hが、 Rcpp<PACKAGE>.hなどとしてRcppArmadillo。はインポートするがあります。

```
// [[Rcpp::depends(Rcpp<PACKAGE>)]]
```

```
// Use the RcppArmadillo package
// Requires different header file from Rcpp.h
#include <RcppArmadillo.h>
// [[Rcpp::depends(RcppArmadillo)]]

// Use the RcppEigen package
// Requires different header file from Rcpp.h
#include <RcppEigen.h>
// [[Rcpp::depends(RcppEigen)]]
```

オンラインでRcppをむ <https://riptutorial.com/ja/r/topic/1404/rcpp>

17: RESTfulなRサービス

き

OpenCPUは、Rパッケージをして、Webアプリケーションを、およびします。

Examples

opencpu アプリ

のウェブサイトには、アプリのいがまれています <https://www.opencpu.org/apps.html>

Rセッションをするには、のコードをします。

```
library(opencpu)
opencpu$start(port = 5936)
```

このコードをすると、URLをしてRセッションのにアクセスできます。は、XML、html、JSONまたはそののされたになります。

えば、のRセッションは、cURLびしによってアクセスできます。

```
#curl uses http post method for -X POST or -d "arg=value"
curl http://localhost:5936/ocpu/library/MASS/scripts/ch01.R -X POST
curl http://localhost:5936/ocpu/library/stats/R/rnorm -d "n=10&mean=5"
```

びしはです。つまり、びしがするのをっているにRセッションがブロックされていないことをしますにして。

びしのは、 /ocpu/tmp/ されているセッションにされます。

セッションをするの

```
curl https://public.opencpu.org/ocpu/library/stats/R/rnorm -d n=5
/ocpu/tmp/x009f9e7630/R/.val
/ocpu/tmp/x009f9e7630/stdout
/ocpu/tmp/x009f9e7630/source
/ocpu/tmp/x009f9e7630/console
/ocpu/tmp/x009f9e7630/info
```

x009f9e7630はセッションのです。

/ocpu/tmp/x009f9e7630/R/.val を /ocpu/tmp/x009f9e7630/R/.val と、 rnorm(5) としてされます
/ocpu/tmp/x009f9e7630/R/console は、 rnorm(5) などのコンソールのをします。

オンラインでRESTfulなRサービスをむ <https://riptutorial.com/ja/r/topic/8323/restfulなrサービス>

18: RMarkdownとknitrプレゼンテーション

- ヘッダ
 - なパラメータとメタデータをするためにスクリプトがコンパイルされるときにされるYAML

パラメーター

パラメータ	
タイトル	のタイトル
	の
	の " r format (Sys.time(), '%d %B, %Y') "
	の
	のなくとも10がです。 htmlドキュメントの、 html_output。 PDFドキュメント、 pdf_document、 ..

サブオプションのパラメータ

サブオプション		html	pdf	ワード	ODT	rtf	md	ギブス	イオスライド	りやすい	ビーマー
citation_package	、natbib、biblatex、またはnoneをするLaTeXパッケージ		バツ				バツ				バツ
コードフォールディング	にRコードのを「なし」、「」、または「」にりえさせる	バツ									
コロールテーマ	するビーマーの										バ

サブオプション		html	pdf	ワード	ODT	rtf	md	ギブス	イオスライド	りやすい	ビーマー
テーマ									ツ		
CSS	ドキュメントをスタイルするためにするCSSファイル	バツ							バツ	バツ	
dev	Figureにするグラフィックスデバイス "png"	バツ	バツ				バツ	バツ	バツ	バツ	バツ
	スライドのフッターにカウントダウンタイマーをでする									バツ	
fig_caption	はキャプションですか	バツ	バツ	バツ	バツ				バツ	バツ	バツ
fig_height、fig_width	ドキュメントのデフォルトのFigureのさとインチ	バツ	バツ	バツ	バツ	バツ	バツ	バツ	バツ	バツ	バツ
ハイライト	ハイライト "タンゴ"、 "ピグ"、 "ケイト"、 "ゼンバーン"、 "テキストメイト"	バツ	バツ	バツ						バツ	バツ
む	ドキュメントにするコンテンツのファイル in_header、 before_body、 after_body	バツ	バツ		バツ		バツ	バツ	バツ	バツ	バツ
きはに1つずつ									バ	バ	バ

サブオプション		html	pdf	ワード	ODT	rtf	md	ギブス	イオスライド	りやすい	ビーマー
	レゼンターのマウスクリックで するがありますか								ツ	ツ	ツ
keep_md	ニットをむ.mdファイルのコピー をする	バツ		バツ	バツ	バツ			バツ	バツ	
keep_tex	ニットをむ.texファイルのコピー をする		バツ								バツ
latex_engine	ラテックスをレンダリングする エンジン、または "pdflatex"、 "xelatex"、 lualatex "		バツ								バツ
lib_dir	するファイルのディレクトリブ ートストラップ、MathJaxなど	バツ							バツ	バツ	
mathjax	ローカルまたはURLにして、ローカル/ URLバージョンのMathJaxをしてレンダ リングする	バツ							バツ	バツ	
md_extensions	デフォルトまたはR Markdownに するMarkdown	バツ	バツ	バツ	バツ	バツ	バツ	バツ	バツ	バツ	バツ
number_sections	ヘッダーにセクションをする	バツ	バツ								

サブオプション		html	pdf	ワード	ODT	rtf	md	ギブス	イオスライド	りやすい	ビーマー
pandoc_args	Pandocにすの	バツ	バツ	バツ	バツ	バツ	バツ	バツ	バツ	バツ	バツ
preserve_yaml	でYAMLのフロントマテリアルをしますか						バツ				
reference_docx	docxをするときにスタイルをコピーするがあるdocxファイル			バツ							
	をにめむ	バツ							バツ	バツ	
slide_level	々のスライドをするもいしレベル										バツ
さい	プレゼンテーションではさいフォントサイズをしますか								バツ		
スマート	っぐなをにし、ダッシュをemダッシュにし、...をなどにします。	バツ							バツ	バツ	
テンプレート	ファイルをレンダリングするときにするPandocテンプレート	バツ	バツ		バツ					バツ	バツ
テーマ	ページにするBootswatchまたはBeamerテーマ	バツ									バツ

サブオプション		html	pdf	ワード	ODT	rtf	md	ギブス	イオスライド	りやすい	ビーマー
toc	のにをする	バツ	バツ	バツ		バツ	バツ	バツ			バツ
toc_depth	にするしのレベル	バツ	バツ	バツ		バツ	バツ	バツ			
toc_float	メインコンテンツのにをかべる	バツ									

Examples

Rstudioの

これは.Rmdとしてされたスクリプトで、.Rとしてされたrスクリプトとはなりません。

スクリプトをするには、`render`をするか、Rstudioのショートカットボタンをします。

```

---
title: "Rstudio exemple of a rmd file"
author: 'stack user'
date: "22 July 2016"
output: html_document
---

The header is used to define the general parameters and the metadata.

## R Markdown

This is an R Markdown document.
It is a script written in markdown with the possibility to insert chunk of R code in it.
To insert R code, it needs to be encapsulated into inverted quote.

Like that for a long piece of code:

```{r cars}
summary(cars)
```

And like ``r cat("that")`` for small piece of code.

## Including Plots

You can also embed plots, for example:

```{r echo=FALSE}
```

```
plot(pressure)
````
```

ioslides プレゼンテーションへのフッターの

フッターをすることは、にではありません。いなことに、jQueryとCSSをして、knitrでレンダリングされたioslidesプレゼンテーションのスライドにフッターをすることができます。まず、jQueryプラグインをインクルードするがあります。これはによってわれます

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.min.js"></script>
```

これで、jQueryをして、プレゼンテーションのDOM ドキュメントオブジェクトモデル をできます。つまり、ドキュメントのHTMLをします。プレゼンテーションがロードされるとすぐに

`$(document).ready(function() { ... })`、クラスが`.title-slide`、`.backdrop`、または`.segue`ではないすべてのスライドをし、それぞれのスライドがじられるに`<footer></footer>`タグをしてください `</slide>`。 `label`は、でされるコンテンツがまれています。

はCSSでフッターをレイアウトするだけです。

```
<footer> footer::after footer::after
```

- `label` をする
- フォントサイズ12を
- フッターをしますスライドのから20ピクセル、から60ピクセル

のプロパティはできますが、プレゼンテーションがのスタイルテンプレートをするはするがあります。

```
---
title: "Adding a footer to presentaion slides"
author: "Martin Schmelzer"
date: "26 Juli 2016"
output: ioslides_presentation
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE)
```

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.min.js"></script>

<script>
  $(document).ready(function() {
    $('slide:not(.title-slide, .backdrop, .segue)').append('<footer label=\\"My amazing
footer!\\ "></footer>');
  })
</script>

<style>
  footer:after {
    content: attr(label);
    font-size: 12pt;
  }
</style>
```

```
    position: absolute;  
    bottom: 20px;  
    left: 60px;  
    line-height: 1.9;  
  }  
</style>
```

```
## Slide 1
```

```
This is slide 1.
```

```
## Slide 2
```

```
This is slide 2
```

```
# Test
```

```
## Slide 3
```

```
And slide 3.
```

はのようになります。

Slide 1

This is slide 1.

My amazing footer

オンラインでRMarkdownとknitrプレゼンテーションをむ

<https://riptutorial.com/ja/r/topic/2999/rmarkdownとknitrプレゼンテーション>

19: RMarkdownをしたレポートの

Examples

テーブルの

HTMLまたはLaTeXテーブルのでデータのをにするパッケージがいくつかあります。らはにがな
ります。

ここではパッケージをします

- ニット
- xtable
- パンダ

HTMLドキュメントの

```
---
title: "Printing Tables"
author: "Martin Schmelzer"
date: "29 Juli 2016"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(knitr)
library(xtable)
library(pander)
df <- mtcars[1:4,1:4]
```

# Print tables using `kable`
```{r, 'kable'}
kable(df)
```

# Print tables using `xtable`
```{r, 'xtable', results='asis'}
print(xtable(df), type="html")
```

# Print tables using `pander`
```{r, 'pander'}
pander(df)
```
```

Printing Tables

Martin Schmelzer

29 Juli 2016

Print tables using kable

```
kable(df)
```

	mpg	cyl	displ	hp
Mazda RX4	21.0	6	160	110
Mazda RX4 Wag	21.0	6	160	110
Datsun 710	22.8	4	108	93
Hornet 4 Drive	21.4	6	258	110

Print tables using xtable

```
print(xtable(df), type="html")
```

	mpg	cyl	displ	hp
Mazda RX4	21.000000	6	160.000000	110.00
Mazda RX4 Wag	21.000000	6	160.000000	110.00
Datsun 710	22.800000	4	108.000000	93.00
Hornet 4 Drive	21.400000	6	258.000000	110.00

Print tables using pander

```
pander(df)
```

	mpg	cyl	displ	hp
Mazda RX4	21	6	160	110
Mazda RX4 Wag	21	6	160	110
Datsun 710	22.8	4	108	93
Hornet 4 Drive	21.4	6	258	110

PDFドキュメントの

```
---
title: "Printing Tables"
author: "Martin Schmelzer"
date: "29 Juli 2016"
output: pdf_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(knitr)
library(xtable)
library(pander)
df <- mtcars[1:4,1:4]
```

# Print tables using `kable`
```{r, 'kable'}
kable(df)
```

# Print tables using `xtable`
```{r, 'xtable', results='asis'}
print(xtable(df, caption="My Table"))
```

# Print tables using `pander`
```{r, 'pander'}
pander(df)
```
```

Printing Tables

Martin Schmalzer
29 Juli 2016

Print tables using `kable`

```
kable(mf)
```

	mpg	cyl	disp	hp
Mazda RX4	21.0	6	160	110
Mazda RX4 Wag	21.0	6	160	110
Datsun 710	22.8	4	108	93
Hornet 4 Drive	21.4	6	258	110

Print tables using `xtable`

```
print(xtable(mf, options = "My Table"))
```

% latex table generated in R 3.3.1 by xtable 1.8-2 package % Fri Jul 29 10:18:04 2016

	mpg	cyl	disp	hp
Mazda RX4	21.00	6.00	160.00	110.00
Mazda RX4 Wag	21.00	6.00	160.00	110.00
Datsun 710	22.80	4.00	108.00	93.00
Hornet 4 Drive	21.40	6.00	258.00	110.00

Table 2: My Table

Print tables using `pander`

```
pander(mf)
```

	mpg	cyl	disp	hp
Mazda RX4	21	6	160	110
Mazda RX4 Wag	21	6	160	110
Datsun 710	22.8	4	108	93
Hornet 4 Drive	21.4	6	258	110

xtableがテーブルのにコメントをするのをめるにはどうすればいいですか

```
options(xtable.comment = FALSE)
```

LaTeXのプリサンプルコマンドをむ

LaTeXプリアンブルコマンド `\usepackage` をRMarkdownドキュメントにめるには、2りのがあります。

1. YAML オプション `header-includesheader-includes`

```
---
title: "Including LaTeX Preamble Commands in RMarkdown"
header-includes:
  - \renewcommand{\familydefault}{\cmss}
  - \usepackage[cm, slantedGreek]{sfmath}
  - \usepackage[T1]{fontenc}
output: pdf_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE, external=T)
```

# Section 1

As you can see, this text uses the Computer Modern Font!
```

Section 1

As you can see, this text uses the Computer Modern Font!

2. むコマンド `includes`、`in_header`

```
---
title: "Including LaTeX Preamble Commands in RMarkdown"
output:
  pdf_document:
    includes:
      in_header: includes.tex
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE, external=T)
```

# Section 1

As you can see, this text uses the Computer Modern Font!
```

ここで`include.tex`のは、`header-includes`まれる3つのコマンドとじです。

くしいテンプレートをく

なのは、のLaTeXのテンプレートをし、でそれをめることです`template`。しかしこれは、プリアンブルだけでなく、のくをカバーしています。

```
---
title: "My Template"
author: "Martin Schmelzer"
output:
  pdf_document:
    template: myTemplate.tex
---
```

をむ

`bibtex`カタログは、YAMLオプションの`bibliography:`にまれています`bibliography:`。ののスタイルは、`biblio-style:`することができます。はのにされます。

```
---
title: "Including Bibliography"
author: "John Doe"
output: pdf_document
bibliography: references.bib
---
```



```
# Abstract

@R_Core_Team_2016

# References
```

Including Bibliography
John Doe

Abstract

R Core Team (2016)

References

R Core Team. 2016. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.

なRマークダウン

Rマークダウンコードのチャンク

R-markdownは、チャンクと呼ばれるRコードがめまれたマークダウンファイルです。Rコードチャンクには、インラインとブロックの2があります。

インラインチャンクは、のをしてされます。

```
`r 2*2`
```

らはされ、をそのにされます。

ブロックチャンクのはのとおりです。

```
```${r name, echo=TRUE, include=TRUE, ...}

2*2

```
```

そして、いくつかのがあります。ここになものがありますしかしにもたくさんあります

- **echo** booleanは、チャンクのコードがドキュメントにまれるかどうかをします
- **is** は、にまれるべきをブーリアンコントロールをみます
- **fig.width** はのをします
- **fig.height** のさをします
- **fig.cap** characterはFigureのキャプションをします。

ののように、な`tag=value`でされています。

R-markdownドキュメントの

は、Rコードダウンのなであり、Rコードチャンクがrマークダウンにめまれるをしています。

```
# Title #

This is plain markdown text.

```{r code, include=FALSE, echo=FALSE}

Just declare variables

income <- 1000
taxes <- 125

...

My income is: `r income` dollars and I payed `r taxes` dollars in taxes.

Below is the sum of money I will have left:

```{r gain, include=TRUE, echo=FALSE}

gain <- income-taxes

gain

...

```{r plotOutput, include=TRUE, echo=FALSE, fig.width=6, fig.height=6}

pie(c(income,taxes), label=c("income", "taxes"))

...

```

## Rマークダウンをのフォーマットにする

R `knitr` パッケージは、R-markdown ファイルのRをし、の `knitr` するためにできます。

R-markdown ファイルを pdf / html にするには、のがです。

1. `knitr` をして R- `knitr` ファイルを `knitr` ファイルにします。
2. られたマークダウンファイルを、 *pandoc* のようなツールを使って pdf / html にします。

の `knitr` パッケージには、ラッパー `knit2html()` と `knit2pdf()` があり、これを `knit2pdf()` にするステップをわずにをすることができます。

のサンプルファイルが `income.Rmd` としてされているは、のRコマンドをして `pdf` ファイルにできます。

```
library(knitr)
knit2pdf("income.Rmd", "income.pdf")

```

なはのようになります。

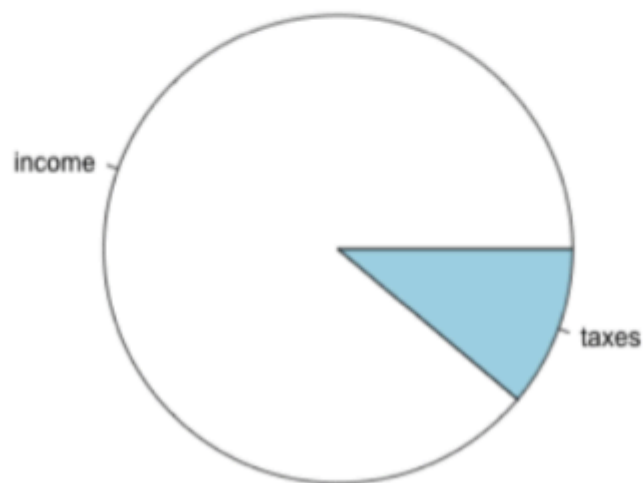
## Title

This is **plain markdown** text.

My income is: 1000 dollars and I payed 125 dollars in taxes.

Below is the sum of money I will have left:

```
[1] 875
```



オンラインでRMarkdownをしたレポートのをむ <https://riptutorial.com/ja/r/topic/4572/rmarkdown>  
をしたレポートの

## 20: RMDの

### パラメーター

| YAMLヘッダーのパラメーター |              |
|-----------------|--------------|
| toc             |              |
| number_sections | セクションににをける   |
| bibliography    | ファイルへのパス     |
| csl             | スタイルファイルへのパス |

- このドキュメントのは、をRMDファイルにすることです。
- のドキュメントをするには、Rに`install.packages("rmarkdown")`をして`rmarkdown`をインストールするがあります。
- にはRmarkdownはのハイパーリンクをします。これにするは、YAMLヘッダーにのコードをすることです。`link link-citations: true`
- にはのがあります。

| フォーマット        | ファイル     |
|---------------|----------|
| MODS          | .mods    |
| BibLaTeX      | .bib     |
| BibTeX        | .bibtex  |
| RIS           | .ris     |
| EndNote       | .enl     |
| EndNote XML   | .xml     |
| ISI           | .wos     |
| MEDLINE       | .medline |
| コパック          | コパック     |
| JSON citeproc | .json    |

# Examples

## をしてをする

RMDファイルのもなはYAMLヘッダーです。をくには、PDF、きセクション、およびtocをすることをめします。

```

title: "Writing an academic paper in R"
author: "Author"
date: "Date"
output:
 pdf_document:
 number_sections: yes
toc: yes
bibliography: bibliography.bib

```

このでは、ファイル**bibliography.bib**はのようになります。

```
@ARTICLE{Meyer2000,
 AUTHOR="Bernd Meyer",
 TITLE="A constraint-based framework for diagrammatic reasoning",
 JOURNAL="Applied Artificial Intelligence",
 VOLUME= "14",
 ISSUE = "4",
 PAGES= "327--344",
 YEAR=2000
}
```

.bibファイルにされているをするには、@とBibkey **Meyer2000**ます。

```
Introduction

`@Meyer2000` results in @Meyer2000.

`@Meyer2000 [p. 328]` results in @Meyer2000 [p. 328]

`[@Meyer2000]` results in [@Meyer2000]

`[-@Meyer2000]` results in [-@Meyer2000]

Summary

References
```

RStudioCtrl + Shift + Kまたはコンソール`rmarkdown::render("<path-to-your-RMD-file">)`をしてRMDファイルをレンダリングすると、

# Writing an academic paper in

*Author*

*Date*

## Contents

**1 Introduction**

**2 Summary**

**References**

## 1 Introduction

@Meyer2000 results in Meyer (2000).

@Meyer2000 [p. 328] results in Meyer (2000, 328)

[@Meyer2000] results in (Meyer 2000)

[-@Meyer2000] results in (2000)

## 2 Summary

## References

Meyer, Bernd. 2000. “A Constraint-Based Framework for Diagrammatic Reasoning.” *Artificial Intelligence* 14 (4): 327–44.

YAMLでエンコードされたのがまれていなければなりません。

```

title: "Writing an academic paper in R"
author: "Author"
date: "Date"
output:
 pdf_document:
 number_sections: yes
toc: yes
references:
 - id: Meyer2000
 title: A Constraint-Based Framework for Diagrammatic Reasoning
 author:
 - family: Meyer
 given: Bernd
 volume: 14
 issue: 4
 publisher: Applied Artificial Intelligence
 page: 327-344
 type: article-journal
 issued:
 year: 2000

Introduction

`@Meyer2000` results in @Meyer2000.

`@Meyer2000 [p. 328]` results in @Meyer2000 [p. 328]

`[@Meyer2000]` results in [@Meyer2000]

`[-@Meyer2000]` results in [-@Meyer2000]

Summary

References
```

このファイルをレンダリングすると、「の」とじになります。

## スタイル

デフォルトでは、`pandoc`はシカゴのをやにします。のスタイルをするには、`csl`メタデータフィールドに`CSL 1.0`スタイルファイルをするがあります。では、にされるスタイル、`elsevier`スタイルがされま<sup>す</sup> <https://github.com/citation-style-language/styles>でダウンロード。スタイルファイルはRMDファイルとじディレクトリにするか、ファイルへのパスをするがあります。

デフォルトのものからのスタイルをするには、のコードをします。

```

title: "Writing an academic paper in R"
author: "Author"
date: "Date"
output:
 pdf_document:
```

```
 number_sections: yes
toc: yes
bibliography: bibliography.bib
csl: elsevier-harvard.csl

Introduction

`@Meyer2000` results in @Meyer2000.

`@Meyer2000 [p. 328]` results in @Meyer2000 [p. 328]

`[@Meyer2000]` results in [@Meyer2000]

`[-@Meyer2000]` results in [-@Meyer2000]

Summary

Reference
```



# Writing an academic paper in R

*Author*

*Date*

## Contents

|          |                     |          |
|----------|---------------------|----------|
| <b>1</b> | <b>Introduction</b> | <b>1</b> |
| <b>2</b> | <b>Summary</b>      | <b>1</b> |
|          | <b>Reference</b>    | <b>1</b> |

## 1 Introduction

@Meyer2000 results in Meyer (2000).

@Meyer2000 [p. 328] results in Meyer (2000, p. 328)

[@Meyer2000] results in (Meyer, 2000)

[-@Meyer2000] results in (2000)

## 2 Summary

## Reference

Meyer, B., 2000. A constraint-based framework for diagrammatic reasoning. Applied Artificial Intelligence 14, 327–344.

「をしてをする」のとのいにしてください。

オンラインでRMDのをむ <https://riptutorial.com/ja/r/topic/7606/rmd>の

# 21: RODBC

## Examples

### RODBCでExcelファイルにする

RODBCは、RとのターゲットRDBMSとのものであるアーキテクチャをえたWindowsコンピュータにされていますが、その名の1つは、ExcelファイルをSQLデータベースとじょうにうことです。

```
require(RODBC)
con = odbcConnectExcel("myfile.xlsx") # open a connection to the Excel file
sqlTables(con)$TABLE_NAME # show all sheets
df = sqlFetch(con, "Sheet1") # read a sheet
df = sqlQuery(con, "select * from [Sheet1 $]") # read a sheet (alternative SQL syntax)
close(con) # close the connection to the file
```

### SQL Serverデータベースをしてのテーブルをする

RODBCのは、SQL Server Management Databaseにすることです。ここではSQL Serverの「ドライバ」、つまりデータベース「Atilla」をsqlQueryから、sqlQueryをしてテーブルまたはそのをするがあります。

```
library(RODBC)
cn <- odbcDriverConnect(connection="Driver={SQL
Server};server=localhost;database=Atilla;trusted_connection=yes;")
tbl <- sqlQuery(cn, 'select top 10 * from table_1')
```

### リレーショナルデータベースへの

```
library(RODBC)
con <- odbcDriverConnect("driver={Sql Server};server=servername;trusted connection=true")
dat <- sqlQuery(con, "select * from table");
close(con)
```

これはSQL Serverインスタンスにします。がどのようにえるかについては、[connectionstrings.com](https://connectionstrings.com)

また、データベースがされていないため、このdatabasename.schema.objectnameのようにするオブジェクトをするがあります

オンラインでRODBCをむ <https://riptutorial.com/ja/r/topic/2471/rodbc>

## 22: Rコードベクトルのベストプラクティス

### Examples

による

Rコードをベクトルするのは、「による」またはRのメソッドディスパッチをまたはすることです。

つまり、してのをするなどの「によって」となるにずっと、にねるがあります。

- がっているデータセットのクラスはですか
- Rのなをとせずにこれをできるのコンパイルみコードはありますか
- そうでないは、ではなくでこれらのをできますか
- に、な `apply` ループをするのではなく、なベクターされたコードをすることにくのをやす  
 `apply` ますか  
 いえれば、Rがなループをしてにできないほどデータをきく/されたものにして  
 いますか

メモリのりりとするオブジェクトをループにれて、ループでのRのディスパッチやを `apply` について、このでりげます。

ごとにをする/なはのとおります。

|                        |                     |                   |                  |                |        |       |
|------------------------|---------------------|-------------------|------------------|----------------|--------|-------|
| apply(mtcars, 1, mean) |                     |                   |                  |                |        |       |
|                        | Mazda RX4           | Mazda RX4 Wag     | Datsun 710       | Hornet 4 Drive | Hornet |       |
| Sportabout             | Valiant             | Duster 360        |                  |                |        |       |
|                        | 29.90727            | 29.98136          | 23.59818         | 38.73955       |        |       |
| 53.66455               | 35.04909            | 59.72000          |                  |                |        |       |
|                        | Merc 240D           | Merc 230          | Merc 280         | Merc 280C      |        | Merc  |
| 450SE                  | Merc 450SL          | Merc 450SLC       |                  |                |        |       |
|                        | 24.63455            | 27.23364          | 31.86000         | 31.78727       |        |       |
| 46.43091               | 46.50000            | 46.35000          |                  |                |        |       |
| Cadillac Fleetwood     | Lincoln Continental | Chrysler Imperial |                  | Fiat 128       |        | Honda |
| Civic                  | Toyota Corolla      | Toyota Corona     |                  |                |        |       |
|                        | 66.23273            | 66.05855          | 65.97227         | 19.44091       |        |       |
| 17.74227               | 18.81409            | 24.88864          |                  |                |        |       |
| Dodge Challenger       | AMC Javelin         | Camaro Z28        | Pontiac Firebird |                |        | Fiat  |
| X1-9                   | Porsche 914-2       | Lotus Europa      |                  |                |        |       |
|                        | 47.24091            | 46.00773          | 58.75273         | 57.37955       |        |       |
| 18.92864               | 24.77909            | 24.88027          |                  |                |        |       |
|                        | Ford Pantera L      | Ferrari Dino      | Maserati Bora    | Volvo 142E     |        |       |
|                        | 60.97182            | 34.50818          | 63.15545         | 26.26273       |        |       |

しかし、もっとうまくいくことができますかここがこったのかてみましょう

1. まず、 `data.frame` を `matrix` しました。は `apply` でこることにしてください。これは、です。  
 `matrix` はにのをすることはできません。したがって、そのようなは、おそらくのやをくにつ  
 ながら `apply(iris, 2, class)` と `str(iris)` または `sapply(iris, class)` と `sapply(iris, class)` して

```
sapply(iris, class)。
```

- もう1つは、にして1のをりしたことです。、私たちはいくつかのRの`nrow(mtcars)`をしなればなり`nrow(mtcars)`。この、`mean`はなではないため、きなデータセットであってもRがにできるがありますが、ごとにをするがあるなをうそれはのポイントにたちをもたらしす
- もRをしましたが、すでにこののコンパイルがありますか

には、

```
rowMeans(mtcars)
 Mazda RX4 Mazda RX4 Wag Datsun 710 Hornet 4 Drive Hornet
Sportabout 29.90727 29.98136 23.59818 38.73955
53.66455 35.04909 59.72000
 Merc 240D Merc 230 Merc 280 Merc 280C Merc
450SE 24.63455 27.23364 31.86000 31.78727
46.43091 46.50000 46.35000
Cadillac Fleetwood Lincoln Continental Chrysler Imperial Fiat 128 Honda
Civic Toyota Corolla Toyota Corona 66.05855 65.97227 19.44091
17.74227 18.81409 24.88864
Dodge Challenger AMC Javelin Camaro Z28 Pontiac Firebird Fiat
X1-9 Porsche 914-2 Lotus Europa 58.75273 57.37955
18.92864 24.77909 24.88027
Ford Pantera L Ferrari Dino Maserati Bora Volvo 142E
60.97182 34.50818 63.15545 26.26273
```

これは、によるものではないため、Rのなはありません。しかし、私たちはまだ`data.frame`を`matrix`しました。`rowMeans`にはエラーメカニズムがあり、できないデータセットではされませんが、それでもはします。

```
rowMeans(iris)
Error in rowMeans(iris) : 'x' must be numeric
```

しかし、それでもなお、私たちはよりいことができますかエラーをうではなく、`mtcars`をベクトルとしてするの`data.frame`はに`list`あり、`list`は`vector`なのでを`data.frame`ことが`data.frame`ます。

```
Reduce(`+`, mtcars)/ncol(mtcars)
[1] 29.90727 29.98136 23.59818 38.73955 53.66455 35.04909 59.72000 24.63455 27.23364 31.86000
31.78727 46.43091 46.50000 46.35000 66.23273 66.05855
[17] 65.97227 19.44091 17.74227 18.81409 24.88864 47.24091 46.00773 58.75273 57.37955 18.92864
24.77909 24.88027 60.97182 34.50818 63.15545 26.26273
```

のがになると、やエラー`NA`をむがわれました。

もうつのは、Rをとてグループごとにをすることです

```
aggregate(. ~ cyl, mtcars, mean)
cyl mpg disp hp drat wt qsec vs am gear
```

```
carb
1 4 26.66364 105.1364 82.63636 4.070909 2.285727 19.13727 0.9090909 0.7272727 4.090909
1.545455
2 6 19.74286 183.3143 122.28571 3.585714 3.117143 17.97714 0.5714286 0.4285714 3.857143
3.428571
3 8 15.10000 353.1000 209.21429 3.229286 3.999214 16.77214 0.0000000 0.1428571 3.285714
3.500000
```

それでも、私たちはにループのRをしています、ループはのCにされていますCループかRループかはほとんどありません。

私たちはそれをけることができますかまあ、Rにはrowsumというコンパイルされたがあるので、のよようにすることができます

```
rowsum(mtcars[-2], mtcars$cyl)/table(mtcars$cyl)
mpg disp hp drat wt qsec vs am gear carb
4 26.66364 105.1364 82.63636 4.070909 2.285727 19.13727 0.9090909 0.7272727 4.090909 1.545455
6 19.74286 183.3143 122.28571 3.585714 3.117143 17.97714 0.5714286 0.4285714 3.857143 3.428571
8 15.10000 353.1000 209.21429 3.229286 3.999214 16.77214 0.0000000 0.1428571 3.285714 3.500000
```

にしなければならなかったのですが。

これで、のデータがもかどうかというがあるかもしれません。 data.frameはベストプラクティスですかまたはをけるためにmatrixデータにりえるだけでよいのでしょうか

なをするようになるにつれて、ではがさらにになります。のによるをえてみましょう。

私たちはmをとっていることができます

```
set.seed(100)
m <- matrix(sample(1e2), 10)
m
 [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,] 8 33 39 86 71 100 81 68 89 84
[2,] 12 16 57 80 32 82 69 11 41 92
[3,] 62 91 53 13 42 31 60 70 98 79
[4,] 66 94 29 67 45 59 20 96 64 1
[5,] 36 63 76 6 10 48 85 75 99 2
[6,] 18 4 27 19 44 56 37 95 26 40
[7,] 3 24 21 25 52 51 83 28 49 17
[8,] 46 5 22 43 47 74 35 97 77 65
[9,] 55 54 78 34 50 90 30 61 14 58
[10,] 88 73 38 15 9 72 7 93 23 87
```

1つはにうことができます

```
apply(m, 1, var)
[1] 871.6556 957.5111 699.2111 941.4333 1237.3333 641.8222 539.7889 759.4333 500.4889
1255.6111
```

では、のにつて、このをにベクトルすることもできる

```
RowVar <- function(x) {
 rowSums((x - rowMeans(x))^2)/(dim(x)[2] - 1)
}
RowVar(m)
[1] 871.6556 957.5111 699.2111 941.4333 1237.3333 641.8222 539.7889 759.4333 500.4889
1255.6111
```

オンラインでRコードベクトルのベストプラクティスをむ <https://riptutorial.com/ja/r/topic/3327/rコードベクトルのベストプラクティス>

## 23: RでODEをく

- odev、times、func、parms、method、...

### パラメーター

| パラメータ |                                              |
|-------|----------------------------------------------|
| y     | きベクトルODEシステムの<br>がな。ののはでなければなりません            |
|       | ODEシステムののをするの                                |
| parms | きベクトルfuncにされるパラメータ<br>するインテグレータ、デフォルトではlsoda |

のとじでをすがあることにしてください。"Lorenzモデル"のでは、これは、"Lorenz"コマンド

```
return(list(c(dX, dY, dZ)))
```

のとじをつ

```
yini <- c(X = 1, Y = 1, Z = 1)
```

## Examples

### ローレンツモデル

ローレンツモデルは、3つのX、Y、Zのダイナミクスをする。モデルは、

$$\frac{dX}{dt} = a \cdot X + Y \cdot Z$$

$$\frac{dY}{dt} = b \cdot (Y - Z)$$

$$\frac{dZ}{dt} = -X \cdot Y + c \cdot Y - Z$$

はのとおりです。

$$X(0) = Y(0) = Z(0) = 1$$

a、b、cは3つのパラメータです。

$$a = -8/3$$

$$b = -10$$

$$c = 28$$

```
library(deSolve)

Define R-function

Lorenz <- function (t, y, parms) {
 with(as.list(c(y, parms)), {
 dX <- a * X + Y * Z
 dY <- b * (Y - Z)
 dZ <- -X * Y + c * Y - Z

 return(list(c(dX, dY, dZ)))
 })
}

Define parameters and variables

parms <- c(a = -8/3, b = -10, c = 28)
yini <- c(X = 1, Y = 1, Z = 1)
times <- seq(from = 0, to = 100, by = 0.01)

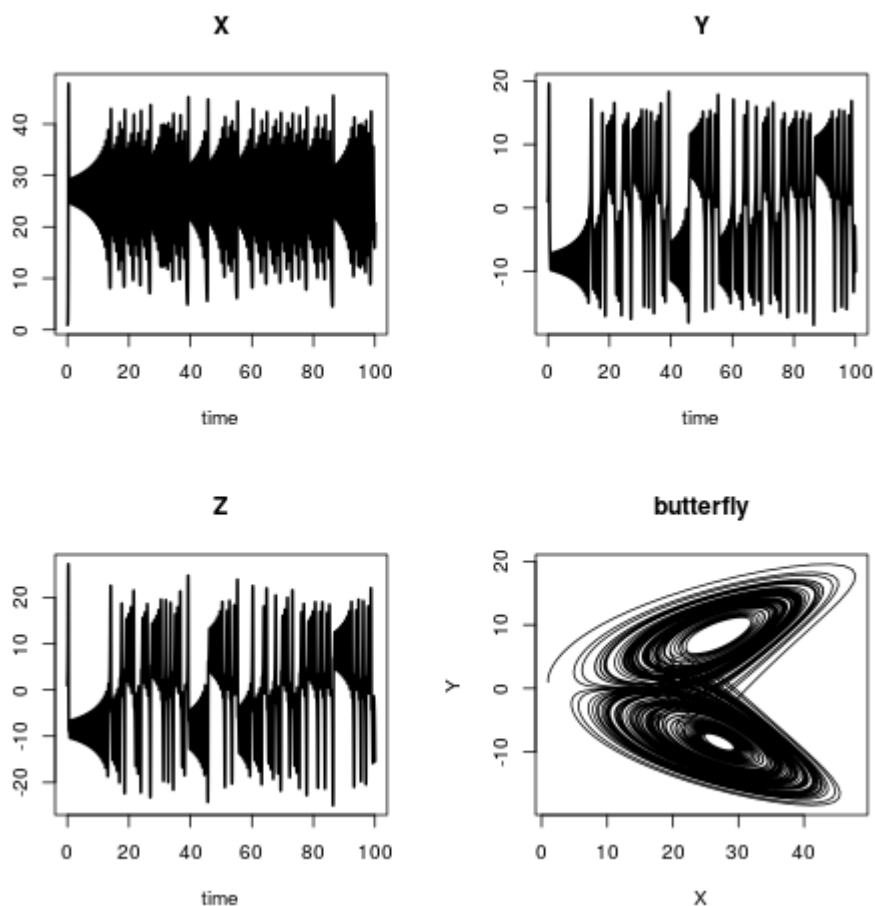
Solve the ODEs

out <- ode(y = yini, times = times, func = Lorenz, parms = parms)

Plot the results

plot(out, lwd = 2)
plot(out[, "X"], out[, "Y"],
 type = "l", xlab = "X",
 ylab = "Y", main = "butterfly")
```





ロトカ・ヴォルテッラまたは

```
library(deSolve)

Define R-function

LV <- function(t, y, parms) {
 with(as.list(c(y, parms)), {

 dP <- rG * P * (1 - P/K) - rI * P * C
 dC <- rI * P * C * AE - rM * C

 return(list(c(dP, dC), sum = C+P))
 })
}

Define parameters and variables

parms <- c(rI = 0.2, rG = 1.0, rM = 0.2, AE = 0.5, K = 10)
yini <- c(P = 1, C = 2)
times <- seq(from = 0, to = 200, by = 1)

Solve the ODEs

```

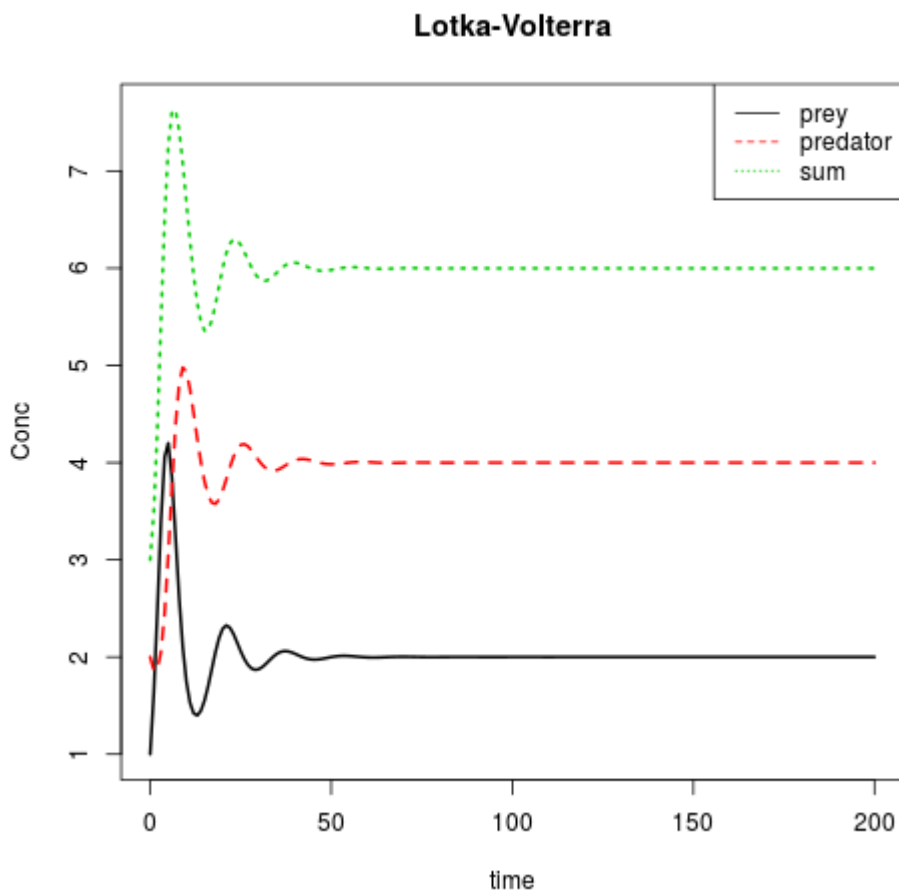
```

out <- ode(y = yini, times = times, func = LV, parms = parms)

Plot the results

matplot(out[,1], out[,2:4], type = "l", xlab = "time", ylab = "Conc",
 main = "Lotka-Volterra", lwd = 2)
legend("topright", c("prey", "predator", "sum"), col = 1:3, lty = 1:3)

```



コンパイルされたの**ODE - R**の

```

library(deSolve)

Define parameters and variables

eps <- 0.01;
M <- 10
k <- M * eps^2/2
L <- 1
L0 <- 0.5
r <- 0.1
w <- 10
g <- 1

```

```

parameter <- c(eps = eps, M = M, k = k, L = L, L0 = L0, r = r, w = w, g = g)

yini <- c(xl = 0, yl = L0, xr = L, yr = L0,
 ul = -L0/L, vl = 0,
 ur = -L0/L, vr = 0,
 lam1 = 0, lam2 = 0)

times <- seq(from = 0, to = 3, by = 0.01)

Define R-function

caraxis_R <- function(t, y, parms) {
 with(as.list(c(y, parms)), {

 yb <- r * sin(w * t)
 xb <- sqrt(L * L - yb * yb)
 Ll <- sqrt(xl^2 + yl^2)
 Lr <- sqrt((xr - xb)^2 + (yr - yb)^2)

 dxl <- ul; dyl <- vl; dxr <- ur; dyr <- vr

 dul <- (L0-Ll) * xl/Ll + 2 * lam2 * (xl-xr) + lam1*xb
 dvl <- (L0-Ll) * yl/Ll + 2 * lam2 * (yl-yr) + lam1*yb - k * g

 dur <- (L0-Lr) * (xr-xb)/Lr - 2 * lam2 * (xl-xr)
 dvr <- (L0-Lr) * (yr-yb)/Lr - 2 * lam2 * (yl-yr) - k * g

 c1 <- xb * xl + yb * yl
 c2 <- (xl - xr)^2 + (yl - yr)^2 - L * L

 return(list(c(dxl, dyl, dxr, dyr, dul, dvl, dur, dvr, c1, c2)))
 })
}

```

## コンパイルされたODE - Cの

```

sink("caraxis_C.c")
cat("
/* suitable names for parameters and state variables */

#include <R.h>
#include <math.h>
static double parms[8];

#define eps parms[0]
#define m parms[1]
#define k parms[2]
#define L parms[3]
#define L0 parms[4]
#define r parms[5]
#define w parms[6]
#define g parms[7]

/*-----
 initialising the parameter common block

```

```

*/
void init_C(void (* daeparms)(int *, double *)) {
 int N = 8;
 daeparms(&N, parms);
}
/* Compartments */

#define xl y[0]
#define yl y[1]
#define xr y[2]
#define yr y[3]
#define lam1 y[8]
#define lam2 y[9]

/*-----
the residual function
-----*/
*/
void caraxis_C (int *neq, double *t, double *y, double *ydot,
 double *yout, int* ip)
{
 double yb, xb, Lr, Ll;

 yb = r * sin(w * *t) ;
 xb = sqrt(L * L - yb * yb);
 Ll = sqrt(xl * xl + yl * yl) ;
 Lr = sqrt((xr-xb)*(xr-xb) + (yr-yb)*(yr-yb));

 ydot[0] = y[4];
 ydot[1] = y[5];
 ydot[2] = y[6];
 ydot[3] = y[7];

 ydot[4] = (L0-Ll) * xl/Ll + lam1*xb + 2*lam2*(xl-xr) ;
 ydot[5] = (L0-Ll) * yl/Ll + lam1*yb + 2*lam2*(yl-yr) - k*g;
 ydot[6] = (L0-Lr) * (xr-xb)/Lr - 2*lam2*(xl-xr) ;
 ydot[7] = (L0-Lr) * (yr-yb)/Lr - 2*lam2*(yl-yr) - k*g ;

 ydot[8] = xb * xl + yb * yl;
 ydot[9] = (xl-xr) * (xl-xr) + (yl-yr) * (yl-yr) - L*L;
}
", fill = TRUE)
sink()
system("R CMD SHLIB caraxis_C.c")
dyn.load(paste("caraxis_C", .Platform$dynlib.ext, sep = ""))
dllname_C <- dyn.load(paste("caraxis_C", .Platform$dynlib.ext, sep = ""))[[1]]

```

## コンパイルされたODE - Fortranでの

```

sink("caraxis_fortran.f")
cat("
c-----
c Initialiser for parameter common block
c-----
 subroutine init_fortran(daeparms)

 external daeparms
 integer, parameter :: N = 8

```

```

double precision parms(N)
common /myparms/parms

call daeparms(N, parms)
return
end

c-----
c rate of change
c-----

subroutine caraxis_fortran(neq, t, y, ydot, out, ip)
implicit none
integer neq, IP(*)
double precision t, y(neq), ydot(neq), out(*)
double precision eps, M, k, L, L0, r, w, g
common /myparms/ eps, M, k, L, L0, r, w, g

double precision xl, yl, xr, yr, ul, vl, ur, vr, lam1, lam2
double precision yb, xb, Ll, Lr, dxl, dyl, dxr, dyr
double precision dul, dvl, dur, dvr, c1, c2

c expand state variables
xl = y(1)
yl = y(2)
xr = y(3)
yr = y(4)
ul = y(5)
vl = y(6)
ur = y(7)
vr = y(8)
lam1 = y(9)
lam2 = y(10)

yb = r * sin(w * t)
xb = sqrt(L * L - yb * yb)
Ll = sqrt(xl**2 + yl**2)
Lr = sqrt((xr - xb)**2 + (yr - yb)**2)

dxl = ul
dyl = vl
dxr = ur
dyr = vr

dul = (L0-Ll) * xl/Ll + 2 * lam2 * (xl-xr) + lam1*xb
dvl = (L0-Ll) * yl/Ll + 2 * lam2 * (yl-yr) + lam1*yb - k*g
dur = (L0-Lr) * (xr-xb)/Lr - 2 * lam2 * (xl-xr)
dvr = (L0-Lr) * (yr-yb)/Lr - 2 * lam2 * (yl-yr) - k*g

c1 = xb * xl + yb * yl
c2 = (xl - xr)**2 + (yl - yr)**2 - L * L

c function values in ydot
ydot(1) = dxl
ydot(2) = dyl
ydot(3) = dxr
ydot(4) = dyr
ydot(5) = dul
ydot(6) = dvl
ydot(7) = dur
ydot(8) = dvr
ydot(9) = c1

```

```

 ydot(10) = c2
 return
 end
", fill = TRUE)

sink()
system("R CMD SHLIB caraxis_fortran.f")
dyn.load(paste("caraxis_fortran", .Platform$dynlib.ext, sep = ""))
dllname_fortran <- dyn.load(paste("caraxis_fortran", .Platform$dynlib.ext, sep = ""))[[1]]

```

## コンパイルされたのODE - ベンチマークテスト

の3つのコンパイルされたのODE - コンパイルされたのODE - コンパイルされたのCとODEの - fortranののコードをコンパイルしてロードすると、ベンチマークテストをできます。

```

library(microbenchmark)

R <- function(){
 out <- ode(y = yini, times = times, func = caraxis_R,
 parms = parameter)
}

C <- function(){
 out <- ode(y = yini, times = times, func = "caraxis_C",
 initfunc = "init_C", parms = parameter,
 dllname = dllname_C)
}

fortran <- function(){
 out <- ode(y = yini, times = times, func = "caraxis_fortran",
 initfunc = "init_fortran", parms = parameter,
 dllname = dllname_fortran)
}

```

がしいかどうかをする

```

all.equal(tail(R()), tail(fortran()))
all.equal(R()[,2], fortran()[,2])
all.equal(R()[,2], C()[,2])

```

ベンチマークをしてくださいあなたのマシンでは、はもちろんなります

```

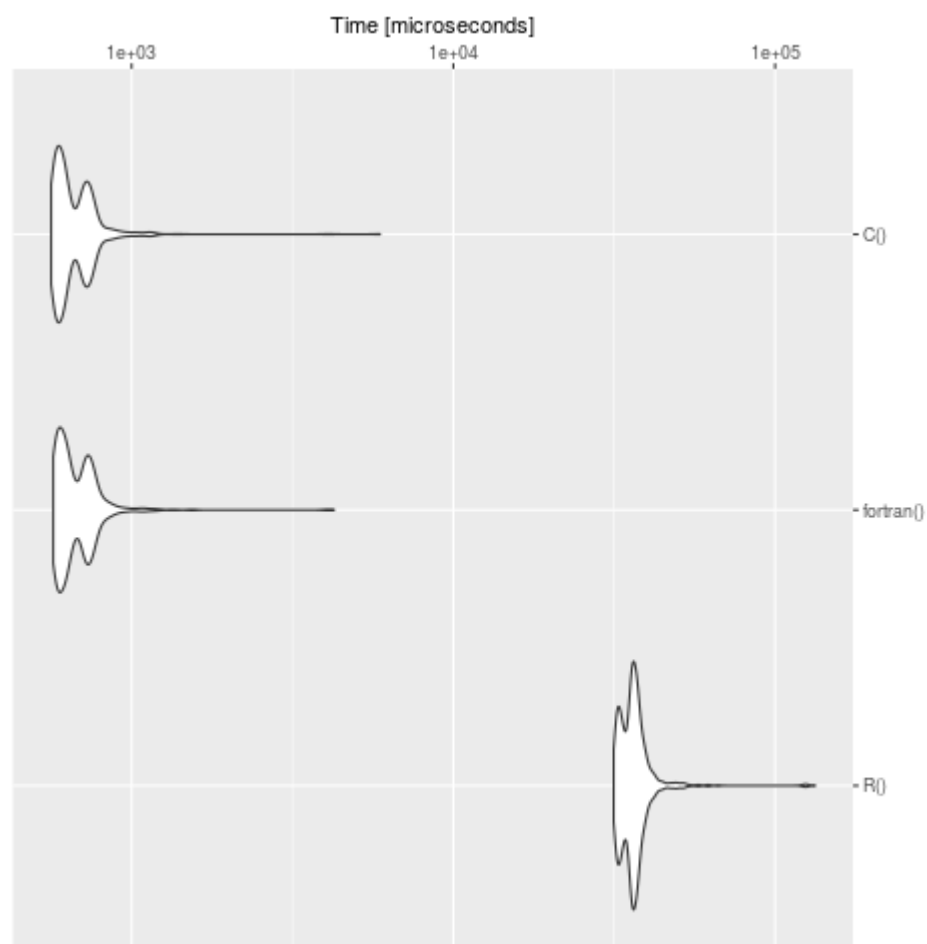
bench <- microbenchmark::microbenchmark(
 R(),
 fortran(),
 C(),
 times = 1000
)

summary(bench)

```

| expr      | min       | lq        | mean       | median     | uq         | max        | neval | cld |
|-----------|-----------|-----------|------------|------------|------------|------------|-------|-----|
| R()       | 31508.928 | 33651.541 | 36747.8733 | 36062.2475 | 37546.8025 | 132996.564 | 1000  | b   |
| fortran() | 570.674   | 596.700   | 686.1084   | 637.4605   | 730.1775   | 4256.555   | 1000  | a   |

C() 562.163 590.377 673.6124 625.0700 723.8460 5914.347 1000 a



RはCとfortranのとはにいいことがはっきりとかります。きなモデルの、コンパイルされたでをする  
があります。パッケージの`c0de`は、ODEをRからCにする1つのです。

オンラインでRでODEをくをむ <https://riptutorial.com/ja/r/topic/7448/rでodeをく>

## 24: Rでツイートをする

ち

オプションすべてのトピックにフォーカスがあります。にここをつけるかをえ、のにがするの  
かをらせる。

## Examples

## ツイートをダウンロードする

にあなたがするがあるとうツイートをダウンロードすることです。あなたのトゥイーターアカウントをセットアップするがあります。くのは、どのようにそれをうにはインターネットでつけることができます。の2つのリンクは、セットアップにちました20175ににチェック

にはな20175ににチェックされたの2つのリンクをつけました

リンク1

リンク2

# Rライブラリ

のRパッケージがです

```
library("devtools")
library("twitterR")
library("ROAuth")
```

あなたはあなたのをっているとのコードをするがあります

```
api_key <- XXXXXXXXXXXXXXXXXXXXXXXX
api_secret <- XXXXXXXXXXXXXXXXXXXXXXXX
access_token <- XXXXXXXXXXXXXXXXXXXXXXXX
access_token_secret <- XXXXXXXXXXXXXXXXXXXXXXXX

setup_twitter_oauth(api_key, api_secret)
```

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxをあなたのキーにしてくださいあなたがしたツイッターアカウントをとっているのは、がするキーをしています。

つぶやきをコピーにダウンロードしたいとしましょう。のコードはそれをいます



```
search.string <- "#coffee"
no.of.tweets <- 1000

c_tweets <- searchTwitter(search.string, n=no.of.tweets, lang="en")
```

あなたは "コーヒー" で1000のつぶやきをるでしょう。

つぶやくのテキストを

は、つぶやきのテキストにアクセスするがあります。それで、私たちはこのようにしますサプリーをえたのようなから、つぶやきをえるがあります。

```
coffee_tweets = sapply(c_tweets, function(t) t$text())

coffee_tweets <- sapply(coffee_tweets,function(row) iconv(row, "latin1", "ASCII", sub=""))
```

あなたは`head`であなたのつぶやきをすることができます。

```
head(coffee_tweets)
```

オンラインでRでツイートをするをむ <https://riptutorial.com/ja/r/topic/10086/rでツイートをする>

## 25: Rとパッケージライブラリの

### Examples

#### Windowsの

プログラムファイルのRバージョンごとのフォルダに、WindowsファイルしたがってライブラリにRをデフォルトでインストールします。

つまり、デフォルトでは、Rのいくつかのバージョンをしてし、ライブラリをすることになります。

これがあなたがむものではなく、あなたが々にしたくないのRインスタンスでにしたいは、Rインストールフォルダをすることをおめします。ウィザードでは、このフォルダをしますには `c:\stats\R` します。そして、アップグレードのために、このRをきするがあります。すべてのパッケージをアップグレードしたいかどうかは、コードのをするがあるため、なですこれは `tm` パッケージでされます。してもいいです

- パッケージをアップグレードするにまずすべてのライブラリのコピーをしてください
- のソースパッケージリポジトリをするえば、`package miniCRAN` を

チェックなしですべてのパッケージをアップグレードするは、`packageStatus` をのようにしてびすことができます。

```
pkgs <- packageStatus() # choose mirror
upgrade(pkgs)
```

に、すべての、つまり `installr` をするためのになパッケージがあります。のGUIをすることもできます。 `gui` をするは、`Rgui` をし、`RStudio` でパッケージをロードしないでください。コードきのパッケージをするのはです

```
install.packages("installr") # install
setInternet2(TRUE) # only for R versions older than 3.3.0
installr::updateR() # updating R.
```

はなドキュメント <https://www.r-statistics.com/tag/installr/> をしてください。にはWindowsのスクリーンショットによるステップバイステップのをしてください <https://www.r-statistics.com/2015/06/a-step-by-step-screenshots-tutorial-upgrading-for-windows/>

それでも、は1つのディレクトリ、すなわち、インストールフォルダのRバージョンへのをします。

オンラインでRとパッケージライブラリのをむ <https://riptutorial.com/ja/r/topic/4088/rとパッケージ>

ライブラリの

## 26: Rにおけるオブジェクトプログラミング

き

このドキュメントのページでは、Rの4つのオブジェクトシステムとそのレベルのについてします。々のシステムのについては、それぞれのトピックページをしてください。

4つのシステムは、S3、S4、Reference Classes、およびS6です。

### Examples

#### S3

S3オブジェクトシステムは、RのになOOシステムです。

すべてのオブジェクトにはS3クラスがあります。それは`class`でることができます。

```
> class(3)
[1] "numeric"
```

また、`class`することもできます

```
> bicycle <- 2
> class(bicycle) <- 'vehicle'
> class(bicycle)
[1] "vehicle"
```

`attr`することもできます

```
> velocipede <- 2
> attr(velocipede, 'class') <- 'vehicle'
> class(velocipede)
[1] "vehicle"
```

オブジェクトにはくのクラスがあります。

```
> class(x = bicycle) <- c('human-powered vehicle', class(x = bicycle))
> class(x = bicycle)
[1] "human-powered vehicle" "vehicle"
```

ジェネリックをする、Rは、なジェネリックをつクラスののをします。

えば

```
> summary.vehicle <- function(object, ...) {
+ message('this is a vehicle')
```

```
+ }
> summary(object = my_bike)
this is a vehicle
```

しかし、ここで `summary.bicycle` すると

```
> summary.bicycle <- function(object, ...) {
+ message('this is a bicycle')
+ }
> summary(object = my_bike)
this is a bicycle
```

オンラインでRにおけるオブジェクトプログラミングをむ <https://riptutorial.com/ja/r/topic/9723/r>に  
おけるオブジェクトプログラミング

## 27: Rにく

### Examples

き

Rはでいっぱいですが、それはプログラミングのですが、なながBaseリソースにはまれていないことがあります。あなたはおそらくをむパッケージをインストールすることができますが、おそらくあなたのはあまりにもものにられたは、にしないですかそれから、でるというがっています。

は、にであるというで、になものにすることができます。をするはありません。

```
one <- function() { 1 }
one()
[1] 1

two <- function() { 1 + 1 }
two()
[1] 2
```

{ }にあるものがなです。すべてをのにめることができます、にはではありませんが、をするのにです。

はにでありながら、になものにすることができます。このは、ベクトルこのではvec をとしてけり、ベクトルのさこのは6をつじベクトルをベクトルのからしいてします。

```
vec <- 4:9
subtract.length <- function(x) { x - length(x) }
subtract.length(vec)
[1] -2 -1 0 1 2 3
```

length() はあらかじめされているすなわちBaseであることにしてください。もちろん、にしたをのでしたり、をりてたり、いくつかのにまたがってのをしたりすることができます。

```
vec2 <- (4:7)/2

msdf <- function(x, multiplier=4) {
 mult <- x * multiplier
 subl <- subtract.length(x)
 data.frame(mult, subl)
}

msdf(vec2, 5)
 mult subl
1 10.0 -2.0
2 12.5 -1.5
3 15.0 -1.0
4 17.5 -0.5
```

multiplier=4

は、`multiplier`デフォルトが4であることをします。

のように、`one`、`two`、`subtract.length`など

は、そのがすように、がりてられていません。これは、がよりきいのであるにですが、それではあまりしません。のなユースケースの1つは、ベースの`*apply`ファミリーです。

`data.frame`のをします。

```
df <- data.frame(first=5:9, second=(0:4)^2, third=-1:3)

apply(df, 2, function(x) { sqrt(sum(x^2)) })
 first second third
15.968719 18.814888 3.872983
```

ののからまでのステップのシーケンスをします。

```
x <- sample(1:6, 12, replace=TRUE)
mat <- matrix(x, nrow=3)

apply(mat, 1, function(x) { seq(min(x), max(x)) })
```

は、でもします

```
(function() { 1 })()
[1] 1
```

は

```
f <- function() { 1 }
f()
[1] 1
```

## RStudioのコードスニペット

これは、をにするのためのちょっとしたハックです。

「しい」RStudio IDEとし、Tabキーをします。

1  
2  
3  
4  
5  
6  
7  
8

fun

|                                                                                   |                |           |
|-----------------------------------------------------------------------------------|----------------|-----------|
|  | fun            | {snippet} |
|  | function       | {base}    |
|  | functionBody   | {methods} |
|  | functionBody<- | {methods} |

```
${1:name} <- fun
 ${3:code}
}
```

はしいのスケルトンになります。

```
name <- function(variables) {

}
```

のスニペットテンプレートをにすることができます。のようなテンプレート

```
name <- function(df, x, y) {
 require(tidyverse)
 out <-
 return(out)
}
```

オプションは、[Global Options -> Code] Global Options -> Code ]メニューのGlobal Options -> Code Edit Snippets ]です。

のとしてをす

によっては、データフレームからにのをしたいがあります。それらはとしてされ、 [[。のをてみましょう.Rのコンソールには、したのながされます。

```
basic.stats <- function(dset, vars){
 for(i in 1:length(vars)){
 print(vars[i])
 print(summary(dset[[vars[i]]]))
 }
}

basic.stats(iris, c("Sepal.Length", "Petal.Width"))
```

のコードをした、したのとその、1、、、3およびがRコンソールにされます。コード

dset[[vars[i]]]はvarsからiのをし、されたデータセットdsetするをします。例えば、  
iris[["Sepal.Length"]]だけをすると、 irisデータセットのSepal.Lengthカラムがベクトルとしてさ



れます。

オンラインでRにをくをむ <https://riptutorial.com/ja/r/topic/7937/rにをく>

---

## 28: RのWebクロール

### Examples

RCurlパッケージをしたなスクレーピングアプローチ

たちはimdbのトップチャートのやをしようします

```
R> library(RCurl)
R> library(XML)
R> url <- "http://www.imdb.com/chart/top"
R> top <- getURL(url)
R> parsed_top <- htmlParse(top, encoding = "UTF-8")
R> top_table <- readHTMLTable(parsed_top)[[1]]
R> head(top_table[1:10, 1:3])
```

```
Rank & Title IMDb Rating
1 1. The Shawshank Redemption (1994) 9.2
2 2. The Godfather (1972) 9.2
3 3. The Godfather: Part II (1974) 9.0
4 4. The Dark Knight (2008) 8.9
5 5. Pulp Fiction (1994) 8.9
6 6. The Good, the Bad and the Ugly (1966) 8.9
7 7. Schindler's List (1993) 8.9
8 8. 12 Angry Men (1957) 8.9
9 9. The Lord of the Rings: The Return of the King (2003) 8.9
10 10. Fight Club (1999) 8.8
```

オンラインでRのWebクロールをむ <https://riptutorial.com/ja/r/topic/4336/rのwebクロール>

## 29: RのバイナリのI/O

### Examples

#### RdsとRDataRdaファイル

`.rds`と`.Rdata`としてもられている`.rda`ファイルは、例えば、ネイティブのストレージ・アプローチとすると、このをするくのがありますRへのネイティブでRのオブジェクトをするためにすることができます`write.table`

- データをRにするがです
- それは、データにされたRの例えば、タイプなどをする。

`saveRDS` / `readRDS`はのRオブジェクトのみをします。しかし、されたオブジェクトのオブジェクトが、オブジェクトがされたときのオブジェクトとしてであるはないというので、マルチオブジェクトストレージアプローチよりもがあります。

たとえば、`.rds`ファイルをして、する`iris`データセットをします。

```
saveRDS(object = iris, file = "my_data_frame.rds")
```

データを読み込むには

```
iris2 <- readRDS(file = "my_data_frame.rds")
```

のオブジェクトをするには、`save()`をして`.Rdata`としてすることができます。

2つのデータフレームをするアイリスと

```
save(iris, cars, file = "myIrisAndCarsData.Rdata")
```

ロードするには

```
load("myIrisAndCarsData.Rdata")
```

`save`と`load`は、オブジェクトがホストされるをすることをします

```
save(iris, cars, file = "myIrisAndCarsData.Rdata", envir = foo <- new.env())
load("myIrisAndCarsData.Rdata", envir = foo)
foo$cars

save(iris, cars, file = "myIrisAndCarsData.Rdata", envir = foo <- new.env())
load("myIrisAndCarsData.Rdata", envir = foo)
foo$cars
```

オンラインでRのバイナリのI/Oをむ <https://riptutorial.com/ja/r/topic/5540/rのバイナリのi---o>

## 30: Rの - の

### Examples

ゼロまたはゼロに近いフィーチャをする

がゼロに近いフィーチャがのになります。

のしきいのをですることができます。

```
data("GermanCredit")
variances<-apply(GermanCredit, 2, var)
variances[which(variances<=0.0025)]
```

または、キャレット・パッケージをして、ゼロに近いをつけることができます。ここでののは、のではなく、むしろのとしてゼロに近いをすることです。

"nearZeroVarは、1つのすなわちのをつまはのをつをします。これらは、サンプルおよびもの2のものにはきいです..."

```
library(caret)
names(GermanCredit)[nearZeroVar(GermanCredit)]
```

いNAのフィーチャをする

にのデータがしているは、のになります。

```
library(VIM)
data(sleep)
colMeans(is.na(sleep))
```

| BodyWgt    | BrainWgt   | NonD       | Dream      | Sleep      | Span       | Gest       |
|------------|------------|------------|------------|------------|------------|------------|
| 0.00000000 | 0.00000000 | 0.22580645 | 0.19354839 | 0.06451613 | 0.06451613 | 0.06451613 |
| Pred       | Exp        | Danger     |            |            |            |            |
| 0.00000000 | 0.00000000 | 0.00000000 |            |            |            |            |

この、それぞれが20のをつNonDとDreamをすることができますカットオフはなるがあります

にするの

につけられたフィーチャは、モデルにし、ペアの1つをすると、そのフィーチャをらすのにちます。をするはたくさんあります。ここに1つあります

```
library(purrr) # in order to use keep()

select correlatable vars
toCorrelate<-mtcars %>% keep(is.numeric)
```

```
calculate correlation matrix
correlationMatrix <- cor(toCorrelate)

pick only one out of each highly correlated pair's mirror image
correlationMatrix[upper.tri(correlationMatrix)]<-0

and I don't remove the highly-correlated-with-itself group
diag(correlationMatrix)<-0

find features that are highly correlated with another feature at the +- 0.85 level
apply(correlationMatrix,2, function(x) any(abs(x)>=0.85))

 mpg cyl disp hp drat wt qsec vs am gear carb
TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

は、MPGがどんなにくしているのかをて、をし、をげるべきかをめたいとうでしょう。cylとdispとじです。あるいは、くするをみわせるがあるかもしれません。

オンラインでRの - のをむ <https://riptutorial.com/ja/r/topic/7561/rの---の>

## 31: Rの

き

このドキュメントでは、Rでされるのをします。Rののについては、[?regex](#)してください。のなりストについては、[このICUのにするガイド](#)をしてください。

## Examples

ベクトルのをつけるには、`grep`をいます

```
General syntax:
grep(<pattern>, <character vector>)

mystring <- c('The number 5',
 'The number 8',
 '1 is the loneliest number',
 'Company, 3 is',
 'Git SSH tag is git@github.com',
 'My personal site is www.personal.org',
 'path/to/my/file')

grep('5', mystring)
[1] 1
grep('@', mystring)
[1] 5
grep('number', mystring)
[1] 1 2 3
```

`x|y`は "x"または "y"

```
grep('5|8', mystring)
[1] 1 2
grep('com|org', mystring)
[1] 5 6
```

`.`のです。これは、"のに"

```
grep('The number .', mystring)
[1] 1 2
```

ドットをマッチさせるときはしてください

```
tricky <- c('www.personal.org', 'My friend is a cyborg')
grep('.org', tricky)
[1] 1 2
```

リテラルをするには、をバックスラッシュ `\` でエスケープするがあります。しかし、Rはをする

にエスケープをしているので、にはバックスラッシュをエスケープする必要がありますをにエスケープする必要があります。

```
grep('\.org', tricky)
Error: '\.' is an unrecognized escape in character string starting "'\.'"
grep('\\.org', tricky)
[1] 1
```

ののいずれかにさせたいは、それらのをカッコ [] でむことができます

```
grep('[13]', mystring)
[1] 3 4
grep('[@/]', mystring)
[1] 5 7
```

シーケンスをすることはかもしれません。たとえば、[0-4]は0,1,2,3,4とし、[AZ]はにし、[Az]はまたはにし、[A-z0-9]はのにしますまたはすなわち、すべての

```
grep('[0-4]', mystring)
[1] 3 4
grep('[A-Z]', mystring)
[1] 1 2 4 5 6
```

Rには、カッコでできるいくつかのショートカットクラスもあります。たとえば、[:lower:]はaz、[:upper:]はAZ、[:alpha:]はAz、[:digit:]は0-9、[:alnum:]はA-z0-9。これらのなはカッコですがあります。たとえば、1のとさせるには、[:digit:]しますにしてください。のとして、[@[:digit:]/]は@、/または0-9とします。

```
grep('[:digit:]', mystring)
[1] 1 2 3 4
grep('@[:digit:]/', mystring)
[1] 1 2 3 4 5 7
```

は、カラット ^ とのをするためにもできます。たとえば、[^5]は"5"のとします。

```
grep('The number [^5]', mystring)
[1] 2
```

オンラインでRのをむ <https://riptutorial.com/ja/r/topic/9743/r>



## 32: Rバージョンの

き

ソフトウェアをインストールまたはアップデートすると、やバグフィックスにアクセスできます。Rのインストールをするには、いくつかのがあります。1つのなは、[RのWebサイトにアクセス](#)し、システムのバージョンをダウンロードすることです。

### Examples

#### Rウェブサイトからインストールする

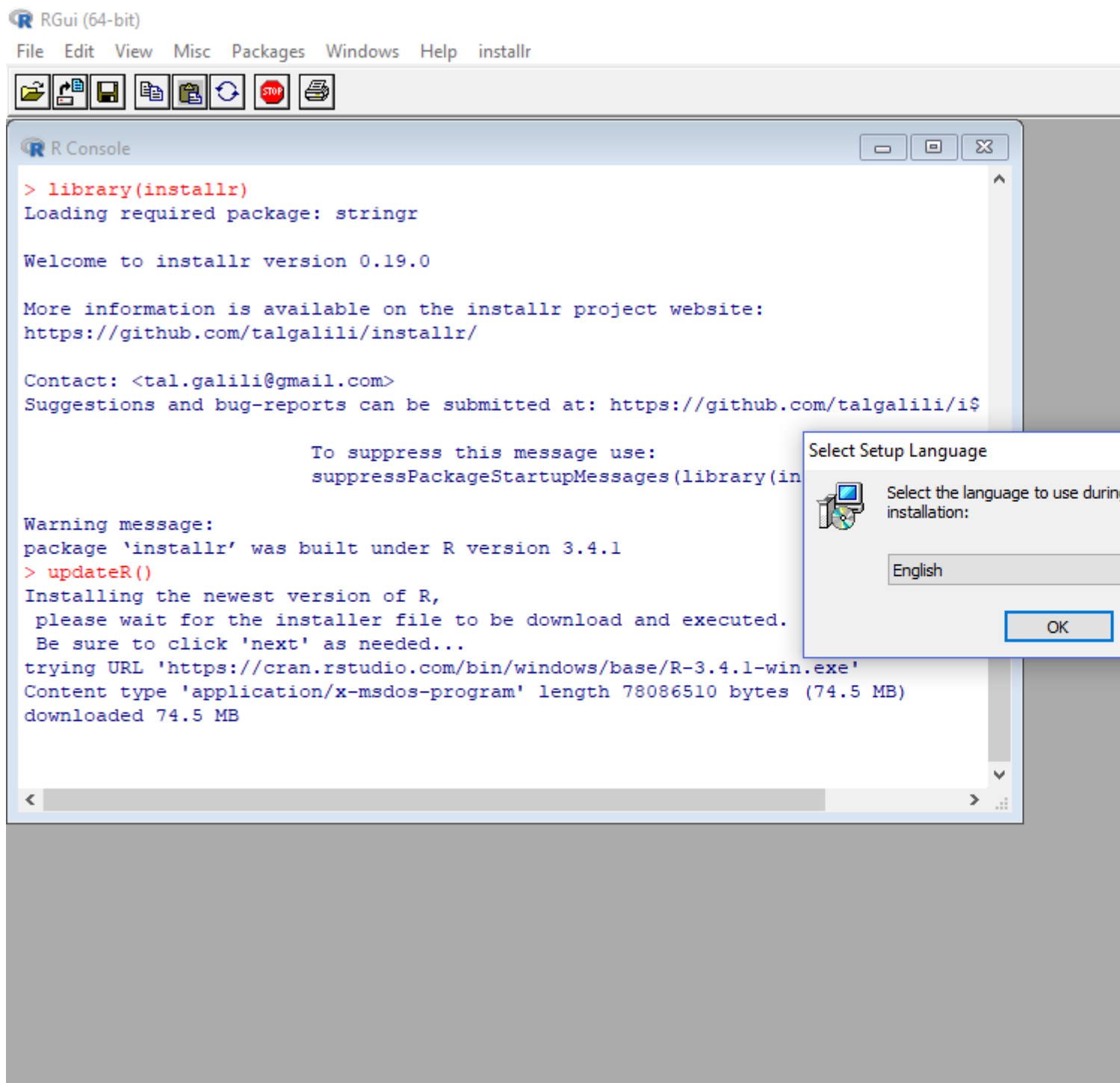
のリリースをするには、<https://cran.r-project.org/>にアクセスし、このオペレーティングシステムのファイルをダウンロードしてください。ダウンロードしたファイルをき、のインストールにいます。のをしないり、すべてのをデフォルトのままにすることができます。

#### installrパッケージをしてRからする

installrというなパッケージをとって、RからRをすることもできます。

RコンソールをくRStudioではなくRStudioではしません、のコードをしてパッケージをインストールしてアップデートをします。

```
install.packages("installr")
library("installr")
updateR()
```



いパッケージをする

インストールがしたら、[]ボタンをクリックします。

はあなたのパッケージをいバージョンのRからしいバージョンのRにコピーするかどうかをねます。  
はいをすると、すべてのパッケージがしいバージョンのRにコピーされます。



R Console

```
> library(installr)
Loading required package: stringr

Welcome to installr version 0.19.0

More information is available on the installr project website:
https://github.com/talgalili/installr/

Contact: <tal.galili@gmail.com>
Suggestions and bug-reports can be submitted at: https://github.com/talgalili/i$

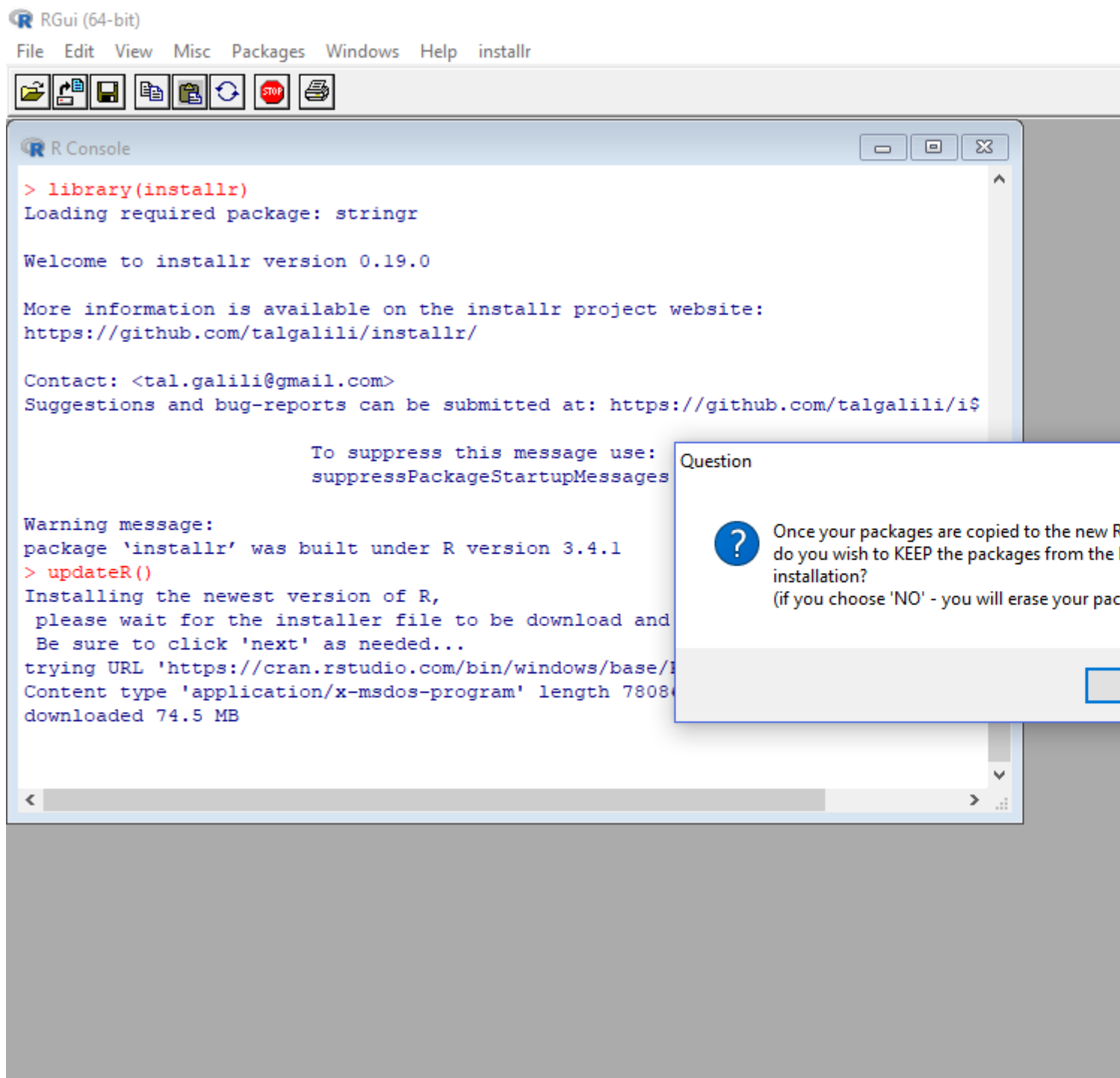
To suppress this message use:
suppressPackageStartupMessages()

Warning message:
package 'installr' was built under R version 3.4.1
> updateR()
Installing the newest version of R,
please wait for the installer file to be download and
Be sure to click 'next' as needed...
trying URL 'https://cran.rstudio.com/bin/windows/base/R341.exe'
Content type 'application/x-msdos-program' length 7808640 bytes
downloaded 74.5 MB
```

Question

Do you wish to copy your packages from the  
newer version of R?

その、いパッケージをしたい、またはするかどうかをできます。



いバージョンからRprofile.siteをして、カスタマイズしたをすべてすることもできます。



```
> library(installr)
Loading required package: stringr

Welcome to installr version 0.19.0

More information is available on the installr project website:
https://github.com/talgalili/installr/

Contact: <tal.galili@gmail.com>
Suggestions and bug-reports can be submitted at: https://github.com/talgalili/i$

To suppress this message use:
suppressPackageStartupMessages()

Warning message:
package 'installr' was built under R version 3.4.1
> updateR()
Installing the newest version of R,
please wait for the installer file to be download and
Be sure to click 'next' as needed...
trying URL 'https://cran.rstudio.com/bin/windows/base/R
Content type 'application/x-msdos-program' length 78086
downloaded 74.5 MB
```

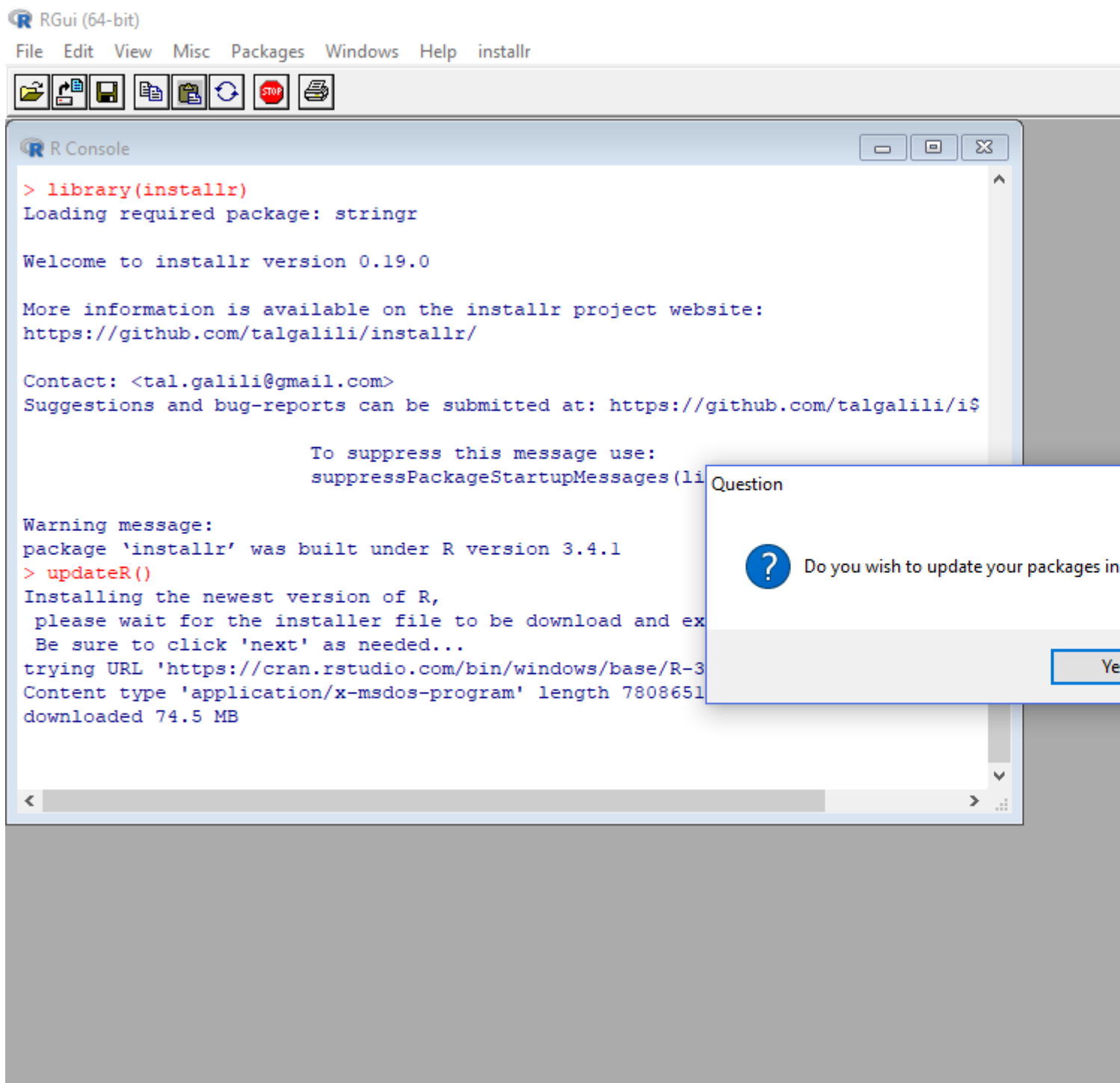
Question



Do you wish to copy your 'Rprofile.site' from the newer version of R?

パッケージの

Rのがしたら、インストールされたパッケージをすることができます。



したら、Rをしてをしってください。

## Rバージョンをする

コンソールをってRのバージョンをすることができます

```
version
```

オンラインでRバージョンのをむ <https://riptutorial.com/ja/r/topic/10729/rバージョンの>

## 33: Rをいた

### Examples

PDFとPMFをRのなるディストリビューションにする

のPMF

なダイが10がったとします。に2つの6をげるはどれくらいですか

dbinomをしてにえることができます。

```
> dbinom(2, 10, 1/6)
[1] 0.29071
```

POISSONのPMF

あるのレストランでされるのは、20のポアソンにうことがられています。はに18のがされるはありますか

あなたはdpoisでにえることができます

```
> dpois(18, 20)
[1] 0.08439355
```

ノーマルディストリビューションのためのPDF

が5でが2であるにして、 $x = 2.5$ でのpdfのをめるには、のコマンドをします。

```
> dnorm(2.5, mean=5, sd=2)
[1] 0.09132454
```

オンラインでRをいたをむ <https://riptutorial.com/ja/r/topic/4333/rをいた>

# 34: S4クラスをしたステートマシンパターンの

## き

マシンは、オブジェクトプログラミングOOP、例えばGOFでされたパターンについてJavaをしてされている「デザインパターン」を。

Rはオブジェクトのパラダイムをシミュレートするいくつかのメカニズムをします。このパターンをするためにS4 Object Systemをしましょう。

## Examples

ステートマシンをしたラインの

RのS4 Classのをして、のパターンののをするためのState Machineパターンをしましょう。

のりみ

り", "をして、がにするをするファイルをするがありますが、されるのにはオプションで、のフィールドをするわりにしています。には、Name; [Address;]Phoneがあります。アドレスがオプションの、にはアドレスをとっていることもあれば、しないこともあります。

```
GREGORY BROWN; 25 NE 25TH; +1-786-987-6543
DAVID SMITH; 786-123-4567
ALAN PEREZ; 25 SE 50TH; +1-786-987-5553
```

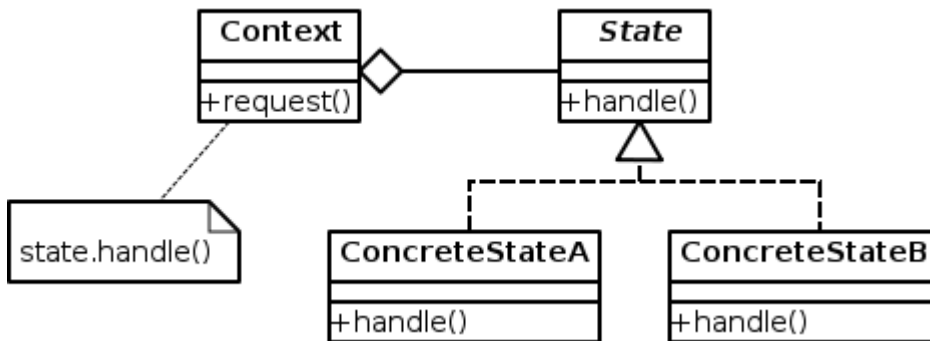
2はアドレスをしません。したがって、りののは、こののように、1つのりとの2つのりとはながあります。りのがわるがあるため、このをするの1つは、パターンについてフィールドのをすることです。そのような、そのようなパターンをするためにをすることができます。例えば

- `"^([AZ]'?\s+)* *[AZ](\s+[AZ]{1,2}\.?.?+)*[AZ](\s+[AZ]+)*$"` ます。たとえば RAFAEL REAL, DAVID R. SMITH, ERNESTO PEREZ GONZALEZ, O' CONNOR BROWN, LUIS PEREZ-MENA など
- アドレス `"^\s[0-9]{1,4}(\s+[AZ]{1,2}[0-9]{1,2}[AZ]{1,2}|[AZ]\s[0-9]+)$"` ます。 11020 LE JEUNE ROAD、 87 SW 27TH。 わかりやすくするために、ここには、はまれていませんが、はこのフィールドにめるか、のフィールドをすることができます。
- `"^\s*(\s+(-|\s+))*[0-9]{3}(-|\s+)[0-9]{3}(-|\s+)[0-9]{4}$"`。 えば 305-123-4567, 305 123 4567, +1-786-123-4567。
- はのアドレスとのもなパターンをしています、よりなをするのはにできます。
- Rでは、"\s"はにとってなをちますので、それをエスケープするがあります。
- をするプロセスをするため、のWebページ[regex101.com](https://regex101.com)をすることをおめします。なすべてのみわせについてされるがられるまで、えられたでそれをできます。

アイデアは、にされたパターンについてラインフィールドをすることです。 Stateパターンは、



のをするためにするのエンティティクラスをします Stateパターンはパターンです。



のコンテキストをしてをしましょう。

- **Context** プロセスのコンテキスト、すなわちのをし、プロセスをする。について、アクション `handle()` がされますが、コンテキストはの `State` クラスの `handle()` にしてされたアクションメソッドにして、についてそれをします。これは、クライアントにとってのあるインターフェースをします。たちの `Context` クラスはのようになすことができます
  - `state`
  - メソッド `handle()` 、 ...
- **State** **State Machine**のをすクラス。これは、コンテキストののにけられたをカプセルするためのインタフェースをします。それはのようになすことができます
  - `name` , `pattern`
  - メソッド `doAction()` 、 `isState` `pattern`をして、がこのパターンにしているかどうかをする、 ...
- **Concrete States** サブクラスクラスのサブクラス `State` のにけられたを `Context` 。たちのサブクラスは `InitState` 、 `NameState` 、 `AddressState` 、 `PhoneState` です。そのようなクラスは、そのようにのにロジックをしてメソッドをするだけです。のはありません。

**action**、 `handle()` 、 `doAction()` または `goNext()` をするメソッドにをけるは、です。メソッド `doAction()` は、はじでクラスはなる2つのジェネリックメソッドをするのをけるため、 `Context` クラスの `handle()` としてしたのクラス `State` または `Context` でじになすことができます。

パーソナルクラス

S4をして、 `Person` クラスをのようになす。

```
setClass(Class = "Person",
 slots = c(name = "character", address = "character", phone = "character")
)
```

クラスをすることをおめします。 `setClass` ドキュメントでは、 `prototype` , `representation` などののをするわりに、 `"initialize"` というラベルのいたメソッドをすることをしてしています。

```
setMethod("initialize", "Person",
 definition = function(.Object, name = NA_character_,
 address = NA_character_, phone = NA_character_) {
 .Object@name <- name
 .Object@address <- address
 })
```

```

 .Object@phone <- phone
 .Object
 }
)

```

initializeメソッドはすでにパッケージ `methods` メソッドであるため、のをするがあります。 Rプロンプトでをできます

```
> initialize
```

これはをします。ののにのようなものがあります。

```
function (.Object, ...) {...}
```

したがって、 `setMethod` をするときは、 `.Object` で *exactly* をするがあります。

ののながある `show` それはである、 `toString()` Javaからメソッドと、クラスドメインののをつことはいいアイデアです。

```

setMethod("show", signature = "Person",
 definition = function(object) {
 info <- sprintf("%s@[name='%s', address='%s', phone='%s']",
 class(object), object@name, object@address, object@phone)
 cat(info)
 invisible(NULL)
 }
)

```

デフォルトの `toString()` Javaと同じをします。

された `Person` オブジェクトのリストをデータセットにしたい、まずオブジェクトのリストをRができるものにしますがありますたとえば、オブジェクトをリストとしてにする。ののメソッドをすることができますは [post](#) をしてください

```

setGeneric(name = "as.list", signature = c('x'),
 def = function(x) standardGeneric("as.list"))

Suggestion taken from here:
http://stackoverflow.com/questions/30386009/how-to-extend-as-list-in-a-canonical-way-to-s4-objects
setMethod("as.list", signature = "Person",
 definition = function(x) {
 mapply(function(y) {
 #apply as.list if the slot is again an user-defined object
 #therefore, as.list gets applied recursively
 if (inherits(slot(x,y), "Person")) {
 as.list(slot(x,y))
 } else {
 #otherwise just return the slot
 slot(x,y)
 }
 },
 slotNames(class(x)),

```

```

 SIMPLIFY=FALSE)
 }
)

```

Rはにとってなをするためににされたものであるため、Oはをしません。したがって、ユーザメソッドは、1 `setGeneric`と2 `setMethod`の2つのをとします。ののように。

クラス

S4のにつて、`State`クラスをしましょう。

```

setClass(Class = "State", slots = c(name = "character", pattern = "character"))

setMethod("initialize", "State",
 definition = function(.Object, name = NA_character_, pattern = NA_character_) {
 .Object@name <- name
 .Object@pattern <- pattern
 .Object
 }
)

setMethod("show", signature = "State",
 definition = function(object) {
 info <- sprintf("%s@[name='%s', pattern='%s']", class(object),
 object@name, object@pattern)
 cat(info)
 invisible(NULL)
 }
)

setGeneric(name = "isState", signature = c('obj', 'input'),
 def = function(obj, input) standardGeneric("isState"))

setGeneric(name = "doAction", signature = c('obj', 'input', 'context'),
 def = function(obj, input, context) standardGeneric("doAction"))

```

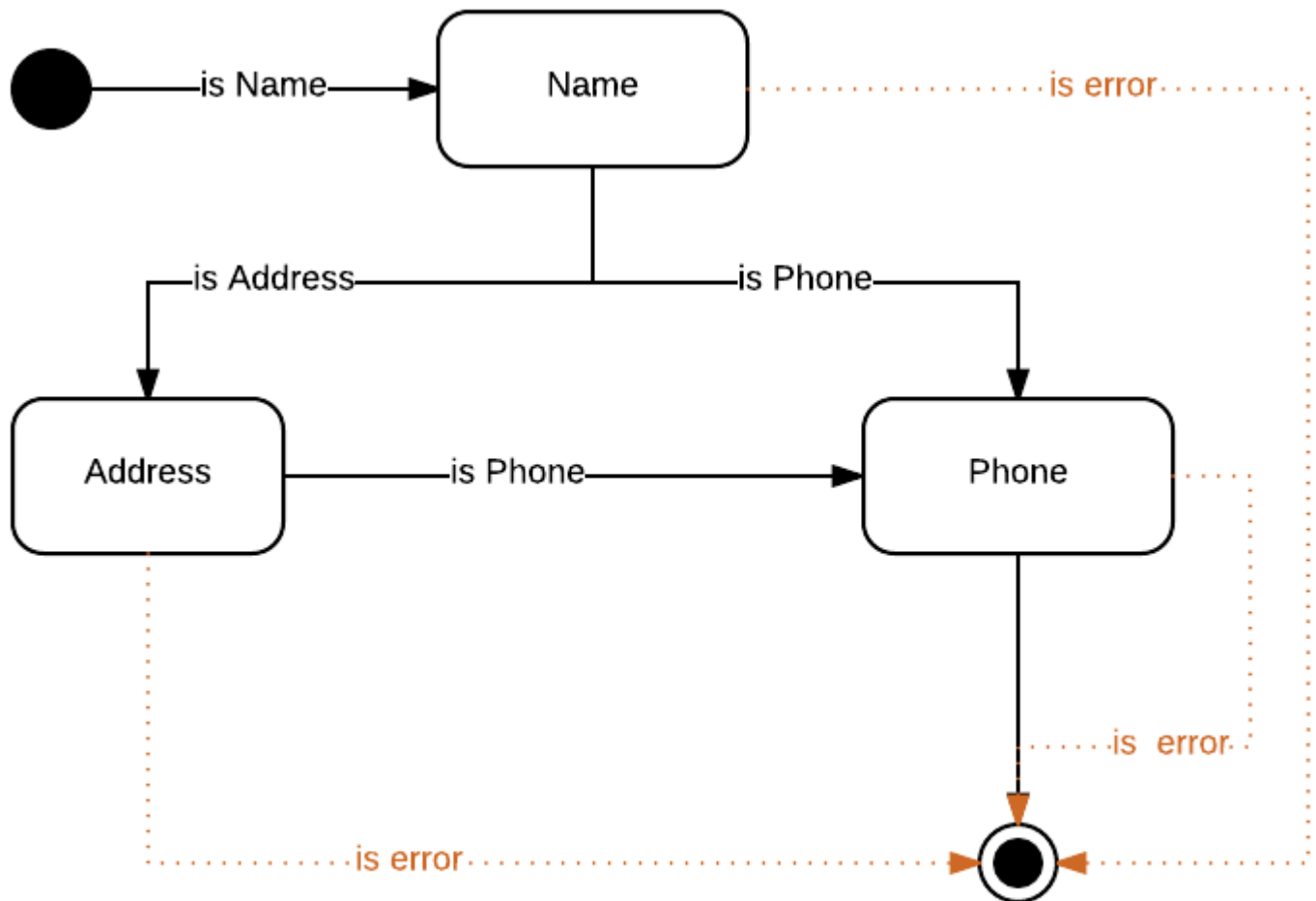
`State`からのすべてのサブクラスには`name`と`pattern`がけられているだけでなく、されたがこのにしているかどうかをする `isState()` メソッド、およびこの `doAction()`。

このプロセスをするために、けったについてのをしましょう

|    |     |  |  |    |
|----|-----|--|--|----|
| 1の | それに |  |  |    |
|    |     |  |  |    |
|    |     |  |  |    |
| わり |     |  |  | わり |

`cell [row, col]=[i, j]` は、`i`をけったときのの`j`をします。

ののでは、またはの2つのをけることができます。トランザクションテーブルをすもう1つののは、のUMLステートマシンのをするです。



**is error:** when the input argument has an invalid pattern

クラスのサブとして、それぞれのをしてみましょう<sub>State</sub>

のサブクラス

は、のクラスをしてされます

```

setClass("InitState", contains = "State")

setMethod("initialize", "InitState",
 definition = function(.Object, name = "init", pattern = NA_character_) {
 .Object@name <- name
 .Object@pattern <- pattern
 .Object
 }
)

setMethod("show", signature = "InitState",
 definition = function(object) {
 callNextMethod()
 }
)

```

Rで、クラスがのクラスのサブクラスであることをするには、 `contains` をし、クラスのクラスをします。

サブクラスは、のをせずにメソッドをするだけなので、 `show` メソッドは、のクラスからのメソッドをびすだけ `callNextMethod()` メソッド `callNextMethod()`

にはパターンがけられておらず、プロセスのをしているだけで、クラスを `NA` でします。

に、 `State` クラスからジェネリックメソッドをすることができます。

```
setMethod(f = "isState", signature = "InitState",
 definition = function(obj, input) {
 nameState <- new("NameState")
 result <- isState(nameState, input)
 return(result)
 }
)
```

このの `pattern` なしでは、のフィールドをしているプロセスをするというアイデアは `name` になります。それのはエラーになります。

```
setMethod(f = "doAction", signature = "InitState",
 definition = function(obj, input, context) {
 nameState <- new("NameState")
 if (isState(nameState, input)) {
 person <- context@person
 person@name <- trimws(input)
 context@person <- person
 context@state <- nameState
 } else {
 msg <- sprintf("The input argument: '%s' cannot be identified", input)
 stop(msg)
 }
 return(context)
 }
)
```

`doAction` メソッドは、をし、されたでコンテキストをします。ここでは、 `@-operator` してコンテキストにアクセスしています。わりに、このプロセスをカプセルする `get/set` メソッド `OO` ベストプラクティスではカプセルができます `get/set`、こののではをせずに `get-set` ごとに4つのメソッドをします。

すべての `doAction` では、がしくされないにをすることをおめします。

の

このクラスのはのとおりです。

```
setClass ("NameState", contains = "State")

setMethod("initialize", "NameState",
 definition=function(.Object, name="name",
```

```

 pattern = "^[A-Z]'?\\s+)* *[A-Z]+(\\s+[A-Z]{1,2}\\\\.?.? +)*[A-Z]+((-|\\s+) [A-Z]+)*$"
 {
 .Object@pattern <- pattern
 .Object@name <- name
 .Object
 }
)

setMethod("show", signature = "NameState",
 definition = function(object) {
 callNextMethod()
 }
)

```

がえられたパターンにすることをするために `grepl` をいます。

```

setMethod(f="isState", signature="NameState",
 definition=function(obj, input) {
 result <- grepl(obj@pattern, input, perl=TRUE)
 return(result)
 }
)

```

に、のにしてするアクションをします。

```

setMethod(f = "doAction", signature = "NameState",
 definition=function(obj, input, context) {
 addressState <- new("AddressState")
 phoneState <- new("PhoneState")
 person <- context@person
 if (isState(addressState, input)) {
 person@address <- trimws(input)
 context@person <- person
 context@state <- addressState
 } else if (isState(phoneState, input)) {
 person@phone <- trimws(input)
 context@person <- person
 context@state <- phoneState
 } else {
 msg <- sprintf("The input argument: '%s' cannot be identified", input)
 stop(msg)
 }
 return(context)
 }
)

```

ここでは、なをする。1つはアドレスであり、もう1つはである。どのようなでも、コンテキストをします。

- `person` をつ `address` または `phone` 。
- プロセスの `state`

をするは、のの `isState()` メソッドをびすことです。デフォルトのの `addressState`, `phoneState` をし、のをします。

のサブクラス1つのごとに1つののロジックはにています。

の

```
setClass("AddressState", contains = "State")

setMethod("initialize", "AddressState",
 definition = function(.Object, name="address",
 pattern = "^\\s[0-9]{1,4}(\\s+[A-Z]{1,2}[0-9]{1,2}[A-Z]{1,2}|[A-Z]\\s0-9]+)$") {
 .Object@pattern <- pattern
 .Object@name <- name
 .Object
 }
)

setMethod("show", signature = "AddressState",
 definition = function(object) {
 callNextMethod()
 }
)

setMethod(f="isState", signature="AddressState",
 definition=function(obj, input) {
 result <- grepl(obj@pattern, input, perl=TRUE)
 return(result)
 }
)

setMethod(f = "doAction", "AddressState",
 definition=function(obj, input, context) {
 phoneState <- new("PhoneState")
 if (isState(phoneState, input)) {
 person <- context@person
 person@phone <- trimws(input)
 context@person <- person
 context@state <- phoneState
 } else {
 msg <- sprintf("The input argument: '%s' cannot be identified", input)
 stop(msg)
 }
 return(context)
 }
)
```

の

```
setClass("PhoneState", contains = "State")

setMethod("initialize", "PhoneState",
 definition = function(.Object, name = "phone",
 pattern = "^\\s*(\\s+1(-|\\s+))*[0-9]{3}(-|\\s+)[0-9]{3}(-|\\s+)[0-9]{4}$") {
 .Object@pattern <- pattern
 .Object@name <- name
 .Object
 }
)

setMethod("show", signature = "PhoneState",
 definition = function(object) {
```

```

 callNextMethod()
 }
)

setMethod(f = "isState", signature = "PhoneState",
 definition = function(obj, input) {
 result <- grepl(obj@pattern, input, perl = TRUE)
 return(result)
 }
)

```

ここでは、`context`の`persons`のリストにし`context`。

```

setMethod(f = "doAction", "PhoneState",
 definition = function(obj, input, context) {
 context <- addPerson(context, context@person)
 context@state <- new("InitState")
 return(context)
 }
)

```

## コンテキストクラス

ここで、`Context`クラスについてします。のをしてできます。

```

setClass(Class = "Context",
 slots = c(state = "State", persons = "list", person = "Person")
)

```

## どこで

- `state` プロセスの
- `person` の。のからすでにしたをします。
- `persons` されたみのリスト。

オプションで、のパーサータイプをうにえて、をけてコンテキストをするための`name`をします。

```

setMethod(f="initialize", signature="Context",
 definition = function(.Object) {
 .Object@state <- new("InitState")
 .Object@persons <- list()
 .Object@person <- new("Person")
 return(.Object)
 }
)

setMethod("show", signature = "Context",
 definition = function(object) {
 cat("An object of class ", class(object), "\n", sep = "")
 info <- sprintf("[state='%s', persons='%s', person='%s']", object@state,
 toString(object@persons), object@person)
 cat(info)
 invisible(NULL)
 }
)

```



```

 }
)

 setGeneric(name = "handle", signature = c('obj', 'input', 'context'),
 def = function(obj, input, context) standardGeneric("handle"))

 setGeneric(name = "addPerson", signature = c('obj', 'person'),
 def = function(obj, person) standardGeneric("addPerson"))

 setGeneric(name = "parseLine", signature = c('obj', 's'),
 def = function(obj, s) standardGeneric("parseLine"))

 setGeneric(name = "parseLines", signature = c('obj', 's'),
 def = function(obj, s) standardGeneric("parseLines"))

 setGeneric(name = "as.df", signature = c('obj'),
 def = function(obj) standardGeneric("as.df"))

```

このようなメソッドでは、プロセスのなをします。

- `handle()` の `state` の `doAction()` メソッドをびします。
- `addPerson` 々はにすると、々は、するがあり `person` リストに `persons` 々がされてきました。
- `parseLine()` 1をする
- `parseLines()` のをするの
- `as.df()` `persons` リストからをデータフレームオブジェクトにします。

するについて、します。

`handle()` メソッドは、 `context` の `state` から `doAction()` `handle()` メソッドにし `context` 。

```

setMethod(f = "handle", signature = "Context",
 definition = function(obj, input) {
 obj <- doAction(obj@state, input, obj)
 return(obj)
 }
)

setMethod(f = "addPerson", signature = "Context",
 definition = function(obj, person) {
 obj@persons <- c(obj@persons, person)
 return(obj)
 }
)

```

まず、のをりをってにし、`Rstrsplit()` をしてをした、について、えられたのとしてをいます。

`handle()` メソッドは、された `state`、`person`、`persons` をつ `context` をびします。

```

setMethod(f = "parseLine", signature = "Context",
 definition = function(obj, s) {
 elements <- strsplit(s, ";")[[1]]
 # Adding an empty field for considering the end state.
 elements <- c(elements, "")
 n <- length(elements)
 input <- NULL
 for (i in (1:n)) {

```

```

 input <- elements[i]
 obj <- handle(obj, input)
 }
 return(obj@person)
}
)

```

Rがのコピーをすると、コンテキスト `obj` をすがあります。

```

setMethod(f = "parseLines", signature = "Context",
 definition = function(obj, s) {
 n <- length(s)
 listOfPersons <- list()
 for (i in (1:n)) {
 ipersons <- parseLine(obj, s[i])
 listOfPersons[[i]] <- ipersons
 }
 obj@persons <- listOfPersons
 return(obj)
 }
)

```

`persons`は、`S4 Person`クラスのインスタンスのリストです。Rはユーザーのクラスのインスタンスをうことをらないので、このかをするにはできません。は、にした`as.list`メソッドをして`Person`をリストにすることです。その、々は、リストのに、このをすることができ`persons`をして、`lapply()`。に`lapply()`ののびしで、`persons.list`をデータフレームにする`data.frame`をする`data.frame`に`persons.list`ました。に、`rbind()`されたデータフレームのしいとしてされたをするために`rbind()`がびされますはこのをしてください

```

Sugestion taken from this post:
http://stackoverflow.com/questions/4227223/r-list-to-data-frame
setMethod(f = "as.df", signature = "Context",
 definition = function(obj) {
 persons <- obj@persons
 persons.list <- lapply(persons, as.list)
 persons.ds <- do.call(rbind, lapply(persons.list, data.frame, stringsAsFactors = FALSE))
 return(persons.ds)
 }
)

```

にすべてをす

に、ソリューションをテストできます。2のアドレスがないをするをします。

```

s <- c(
 "GREGORY BROWN; 25 NE 25TH; +1-786-987-6543",
 "DAVID SMITH; 786-123-4567",
 "ALAN PEREZ; 25 SE 50TH; +1-786-987-5553"
)

```

に、`context`をし、をします。

```

context <- new("Context")

```

```
context <- parseLines(context, s)
```

に、するデータセットをしてします。

```
df <- as.df(context)
> df
```

|   | name          | address    | phone           |
|---|---------------|------------|-----------------|
| 1 | GREGORY BROWN | 25 NE 25TH | +1-786-987-6543 |
| 2 | DAVID SMITH   | <NA>       | 786-123-4567    |
| 3 | ALAN PEREZ    | 25 SE 50TH | +1-786-987-5553 |

showメソッドをテストしましょう

```
> show(context@persons[[1]])
Person@[name='GREGORY BROWN', address='25 NE 25TH', phone='+1-786-987-6543']
```

そして、いくつかのサブについて

```
> show(new("PhoneState"))
PhoneState@[name='phone', pattern='^\s*(\+1(-|\s+))*[0-9]{3}(-|\s+)[0-9]{3}(-|\s+)[0-9]{4}$']
```

に、 as.list()メソッドをテストし as.list()。

```
> as.list(context@persons[[1]])
$name
[1] "GREGORY BROWN"

$address
[1] "25 NE 25TH"

$phone
[1] "+1-786-987-6543"

>
```

これは、OOパラダイムをするためにRからなメカニズムの1つをして、Stateパターンをするをしています。それにもかかわらず、R OOソリューションはユーザーフレンドリーではなく、の OOPとはきくなります。がまったくうので、あなたのえをえるがあります。それはプログラミングのパラダイムをいこさせます。たとえば、Java / Cのように `object.setID("A1")` わりに、Rのは `setID(object, "A1")` ようにメソッドをびすがあります。したがって、のコンテキストをするためには、にオブジェクトをとめてめるがあります。に、 `this` クラスもなものもなく、 `..` されたクラスのメソッドまたはにアクセスするための。クラスまたはメソッドをするためには、 `"Person"`、 `"isState"` などをするため、これはエラープロンプトのくです。

のS4クラスのソリューションには、なタスクをするためののJava / Cよりもくのコードがです。とにかく、ステートパターンは、このようなのにするれたソリューションです。これは、ロジックをのにするプロセスをします。すべてのをするためのきな `if-else` ブロックをつわりに、でするアクションをするために、 `State` サブクラスののにさな `if-else` ブロックをしています。

ファイル [ここ](#)では、スクリプトをダウンロードできます。

どんなものです。

オンラインでS4クラスをしたステートマシンパターンのをむ

<https://riptutorial.com/ja/r/topic/9126/s4クラスをしたステートマシンパターンの>

# 35: sqldf

## Examples

な

`sqldf()` パッケージから `sqldf` RのSQLクエリでデータをしてするためのSQLiteクエリのはとして  
されていることができます。

`ggplot2` パッケージから "diamonds" データセットのの10をするには、の `ggplot2` にします。

```
data("diamonds")
head(diamonds)
```

```
A tibble: 6 x 10
 carat cut color clarity depth table price x y z
 <dbl> <ord> <ord> <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl>
1 0.23 Ideal E SI2 61.5 55 326 3.95 3.98 2.43
2 0.21 Premium E SI1 59.8 61 326 3.89 3.84 2.31
3 0.23 Good E VS1 56.9 65 327 4.05 4.07 2.31
4 0.29 Premium I VS2 62.4 58 334 4.20 4.23 2.63
5 0.31 Good J SI2 63.3 58 335 4.34 4.35 2.75
6 0.24 Very Good J VVS2 62.8 57 336 3.94 3.96 2.48
```

```
require(sqldf)
sqldf("select * from diamonds limit 10")
```

```
 carat cut color clarity depth table price x y z
1 0.23 Ideal E SI2 61.5 55 326 3.95 3.98 2.43
2 0.21 Premium E SI1 59.8 61 326 3.89 3.84 2.31
3 0.23 Good E VS1 56.9 65 327 4.05 4.07 2.31
4 0.29 Premium I VS2 62.4 58 334 4.20 4.23 2.63
5 0.31 Good J SI2 63.3 58 335 4.34 4.35 2.75
6 0.24 Very Good J VVS2 62.8 57 336 3.94 3.96 2.48
7 0.24 Very Good I VVS1 62.3 57 336 3.95 3.98 2.47
8 0.26 Very Good H SI1 61.9 55 337 4.07 4.11 2.53
9 0.22 Fair E VS2 65.1 61 337 3.87 3.78 2.49
10 0.23 Very Good H VS1 59.4 61 338 4.00 4.05 2.39
```

"E"のの10をするには

```
sqldf("select * from diamonds where color = 'E' limit 10")
```

```
 carat cut color clarity depth table price x y z
1 0.23 Ideal E SI2 61.5 55 326 3.95 3.98 2.43
2 0.21 Premium E SI1 59.8 61 326 3.89 3.84 2.31
3 0.23 Good E VS1 56.9 65 327 4.05 4.07 2.31
4 0.22 Fair E VS2 65.1 61 337 3.87 3.78 2.49
5 0.20 Premium E SI2 60.2 62 345 3.79 3.75 2.27
```

|    |      |           |   |     |      |    |     |      |      |      |
|----|------|-----------|---|-----|------|----|-----|------|------|------|
| 6  | 0.32 | Premium   | E | I1  | 60.9 | 58 | 345 | 4.38 | 4.42 | 2.68 |
| 7  | 0.23 | Very Good | E | VS2 | 63.8 | 55 | 352 | 3.85 | 3.92 | 2.48 |
| 8  | 0.23 | Very Good | E | VS1 | 60.7 | 59 | 402 | 3.97 | 4.01 | 2.42 |
| 9  | 0.23 | Very Good | E | VS1 | 59.5 | 58 | 402 | 4.01 | 4.06 | 2.40 |
| 10 | 0.23 | Good      | E | VS1 | 64.1 | 59 | 402 | 3.83 | 3.85 | 2.46 |

のでは、SQLクエリーのきのは、のクエリーがでまれているにはでまれていますこれはもです。

1カラットをえるプレミアムカットダイヤモンドのをカウントするしいをしたいとします。

```
sqldf("select count(*) from diamonds where carat > 1 and color = 'E'")
```

|   |          |
|---|----------|
|   | count(*) |
| 1 | 1892     |

されたのは、しいとしてすこともできます。

```
sqldf("select *, count(*) as cnt_big_E_colored_stones from diamonds where carat > 1 and color = 'E' group by clarity")
```

|                          | carat | cut       | color | clarity | depth | table | price | x    | y    | z    |
|--------------------------|-------|-----------|-------|---------|-------|-------|-------|------|------|------|
| cnt_big_E_colored_stones |       |           |       |         |       |       |       |      |      |      |
| 1                        | 1.30  | Fair      | E     | I1      | 66.5  | 58    | 2571  | 6.79 | 6.75 | 4.50 |
| 65                       |       |           |       |         |       |       |       |      |      |      |
| 2                        | 1.28  | Ideal     | E     | IF      | 60.7  | 57    | 18700 | 7.09 | 6.99 | 4.27 |
| 28                       |       |           |       |         |       |       |       |      |      |      |
| 3                        | 2.02  | Very Good | E     | SI1     | 59.8  | 59    | 18731 | 8.11 | 8.20 | 4.88 |
| 499                      |       |           |       |         |       |       |       |      |      |      |
| 4                        | 2.03  | Premium   | E     | SI2     | 61.5  | 59    | 18477 | 8.24 | 8.16 | 5.04 |
| 666                      |       |           |       |         |       |       |       |      |      |      |
| 5                        | 1.51  | Ideal     | E     | VS1     | 61.5  | 57    | 18729 | 7.34 | 7.40 | 4.53 |
| 158                      |       |           |       |         |       |       |       |      |      |      |
| 6                        | 1.72  | Very Good | E     | VS2     | 63.4  | 56    | 18557 | 7.65 | 7.55 | 4.82 |
| 318                      |       |           |       |         |       |       |       |      |      |      |
| 7                        | 1.20  | Ideal     | E     | VVS1    | 61.8  | 56    | 16256 | 6.78 | 6.87 | 4.22 |
| 52                       |       |           |       |         |       |       |       |      |      |      |
| 8                        | 1.55  | Ideal     | E     | VVS2    | 62.5  | 55    | 18188 | 7.38 | 7.40 | 4.62 |
| 106                      |       |           |       |         |       |       |       |      |      |      |

もしcut じてダイヤモンドの priceはかがあれば

```
sqldf("select cut, max(price) from diamonds group by cut")
```

|   | cut       | max(price) |
|---|-----------|------------|
| 1 | Fair      | 18574      |
| 2 | Good      | 18788      |
| 3 | Ideal     | 18806      |
| 4 | Premium   | 18823      |
| 5 | Very Good | 18818      |

オンラインでsqldfをむ <https://riptutorial.com/ja/r/topic/2100/sqldf>

## 36: stringiパッケージによる

パッケージをインストールするには、をします。

```
install.packages("stringi")
```

それをロードする

```
require("stringi")
```

## Examples

のパターンカウント

パターンで

```
stri_count_fixed("babab", "b")
[1] 3
stri_count_fixed("babab", "ba")
[1] 2
stri_count_fixed("babab", "bab")
[1] 1
```

ネイティブ

```
length(gregexpr("b", "babab")[[1]])
[1] 3
length(gregexpr("ba", "babab")[[1]])
[1] 2
length(gregexpr("bab", "babab")[[1]])
[1] 1
```

はとパターンにしてベクトルされます

```
stri_count_fixed("babab", c("b", "ba"))
[1] 3 2
stri_count_fixed(c("babab", "bbb", "bca", "abc"), c("b", "ba"))
[1] 3 0 1 0
```

ベースR

```
sapply(c("b", "ba"), function(x) length(gregexpr(x, "babab")[[1]]))
b ba
3 2
```

で

の - に<sub>a</sub>とのをつける

の-つけたやのを<sub>a</sub>

```
stri_count_regex("a1 b2 a3 b4 aa", "a.")
[1] 3
stri_count_regex("a1 b2 a3 b4 aa", "a\\d")
[1] 2
```

の

```
stri_dup("abc", 3)
[1] "abccabccabc"
```

じことをするRのは、のようになります。

```
paste0(rep("abc", 3), collapse = "")
[1] "abccabccabc"
```

ベクターをりける

```
stri_paste(LETTERS, "-", 1:13)
[1] "A-1" "B-2" "C-3" "D-4" "E-5" "F-6" "G-7" "H-8" "I-9" "J-10" "K-11" "L-12" "M-13"
[14] "N-1" "O-2" "P-3" "Q-4" "R-5" "S-6" "T-7" "U-8" "V-9" "W-10" "X-11" "Y-12" "Z-13"
```

ネイティブに、たちはRでこれをうことができます

```
> paste(LETTERS, 1:13, sep="-")
[1] "A-1" "B-2" "C-3" "D-4" "E-5" "F-6" "G-7" "H-8" "I-9" "J-10" "K-11" "L-12" "M-13"
[14] "N-1" "O-2" "P-3" "Q-4" "R-5" "S-6" "T-7" "U-8" "V-9" "W-10" "X-11" "Y-12" "Z-13"
```

いくつかのパターンでテキストをする

1つのパターンをしてテキストのベクトルをする

```
stri_split_fixed(c("To be or not to be.", "This is very short sentence."), " ")
[[1]]
[1] "To" "be" "or" "not" "to" "be."
#
[[2]]
[1] "This" "is" "very" "short" "sentence."
```

くのパターンをして1つのテキストをします。

```
stri_split_fixed("Apples, oranges and pineaplles.", c(" ", ",", "s"))
```



```
[[1]]
[1] "Apples," "oranges" "and" "pineapllles."
#
[[2]]
[1] "Apples" " oranges and pineapllles."
#
[[3]]
[1] "Apple" ", orange" " and pineaplle" "."
```

オンラインでstringiパッケージによるをむ <https://riptutorial.com/ja/r/topic/1670/stringi>パッケージによる

## 37: strsplit

- strsplit
- バツ
- スプリット
- fixed = FALSE
- perl = FALSE
- useBytes = FALSE

### Examples

き

strsplitは、ベクトルをあるパターンのリストにするのになです。なRツールでは、リストをdata.frameにみみすることができます。リストのは、グラフのでできます。

strsplitないはstrsplitです。カンマリによってベクトルをstrsplitします。

```
temp <- c("this,that,other", "hat,scarf,food", "woman,man,child")
get a list split by commas
myList <- strsplit(temp, split=",")
print myList
myList
[[1]]
[1] "this" "that" "other"

[[2]]
[1] "hat" "scarf" "food"

[[3]]
[1] "woman" "man" "child"
```

でしたように、splitはにされず、でされたパターンにうかもしれません。たとえば、temp2は、ごとにセパレータがされているをいて、のtempとじです。されたがベクトルのをするためにをけるというをすることができます。

```
temp2 <- c("this, that, other", "hat,scarf ,food", "woman; man ; child")
myList2 <- strsplit(temp2, split=" ?[,;] ?")
myList2
[[1]]
[1] "this" "that" "other"

[[2]]
[1] "hat" "scarf" "food"

[[3]]
[1] "woman" "man" "child"
```

1. このでは、のをしています。
- 2.

によっては、にするとプロセスがなくなることがあります。のをするくのRとに、は、Rにを  
りさせるようにするためにできます。

オンラインでstrsplitをむ <https://riptutorial.com/ja/r/topic/2762/strsplit>

---

## 38: texreg をしてにしたでモデルをエクスポートする

き

texreg パッケージは、にすぐれたでモデルまたはのモデルをエクスポートするのにちます。は HTML または .doc MS Office Word としてエクスポートできます。

---

### リンク

- [CRAN ページ](#)

## Examples

の

```
models
fit1 <- lm(mpg ~ wt, data = mtcars)
fit2 <- lm(mpg ~ wt+hp, data = mtcars)
fit3 <- lm(mpg ~ wt+hp+cyl, data = mtcars)

export to html
texreg::htmlreg(list(fit1, fit2, fit3), file='models.html')

export to doc
texreg::htmlreg(list(fit1, fit2, fit3), file='models.doc')
```

はこのテーブルのようにえます。

|             | Model 1            | Model 2            | Model 3            |
|-------------|--------------------|--------------------|--------------------|
| (Intercept) | 37.29***<br>(1.88) | 37.23***<br>(1.60) | 38.75***<br>(1.79) |
| wt          | -5.34***<br>(0.56) | -3.88***<br>(0.63) | -3.17***<br>(0.74) |
| hp          |                    | -0.03**<br>(0.01)  | -0.02<br>(0.01)    |
| cyl         |                    |                    | -0.94<br>(0.55)    |
| R2          | 0.75               | 0.83               | 0.84               |
| Adj. R2     | 0.74               | 0.81               | 0.83               |
| Num. obs.   | 32                 | 32                 | 32                 |
| RMSE        | 3.05               | 2.59               | 2.51               |

\*\*\*p < 0.001, \*\*p < 0.01, \*p < 0.05

Statistical models

texreg::htmlreg() には、さらになパラメータがいくつかあります。もなパラメータのです。

```
export to html
texreg::htmlreg(list(fit1, fit2, fit3), file='models.html',
 single.row = T,
 custom.model.names = LETTERS[1:3],
 leading.zero = F,
 digits = 3)
```

このようなテーブルの

|             | A                 | B                 | C                 |
|-------------|-------------------|-------------------|-------------------|
| (Intercept) | 37.285 (1.878)*** | 37.227 (1.599)*** | 38.752 (1.787)*** |
| wt          | -5.344 (0.559)*** | -3.878 (0.633)*** | -3.167 (0.741)*** |
| hp          |                   | -0.032 (0.009)**  | -0.018 (0.012)    |
| cyl         |                   |                   | -0.942 (0.551)    |
| R2          | 0.753             | 0.827             | 0.843             |
| Adj. R2     | 0.745             | 0.815             | 0.826             |
| Num. obs.   | 32                | 32                | 32                |
| RMSE        | 3.046             | 2.593             | 2.512             |

\*\*\*p < 0.001, \*\*p < 0.01, \*p < 0.05

Statistical models

オンラインでtexregをしてにしたでモデルをエクスポートするをむ

<https://riptutorial.com/ja/r/topic/9037/texreg>をしてにしたでモデルをエクスポートする

## 39: Webスクレイプと

スクレイピングとは、コンピュータをしてWebページのコードをすることをします。コードをしたら、それをRでさらにするためになにするがあります。

Base Rにはこれらのプロセスになツールがないため、はスクレイピングとはパッケージでわれます。いくつかのパッケージは、スクレイピング `RSelenium`、`httr`、`curl`、`RCurl`、`XML`、`xml2`、`rvest` のためにもです。

するプロセスは、Web APIとはなり、であることをしたデータをすWeb APIをスクラップしています。じパッケージのくがにされています。

サーバのがえたか、データのがされているかにかかわらず、のウェブサイトはきられることにしていますウェブサイトがそのをすることをじている、それをきることはです。

## Examples

**rvest**によるなりり

`rvest`は、Pythonの[Beautiful Soup](#)にされたHadley WickhamによるWeb `rvest`とのパッケージです。Hadleyの`xml2`パッケージの`libxml2`バインディングをHTMLにしています。

tidyverseのとして、`rvest`が**パイプ**されます。それはしています

- `xml2::read_html` をしてWebページのHTMLをスクラップし、
- これは、そのサブセットとすることができる`html_node`と`html_nodes` CSSをして、またはXPathのセクタ、および
- `html_text`や`html_table`ようなでRオブジェクトにパースされます。

[RのWikipediaページ](#)からマイルストーンのテーブルをるために、コードはのようになります

```
library(rvest)

url <- 'https://en.wikipedia.org/wiki/R_(programming_language) '

scrape HTML from website
url %>% read_html() %>%
 # select HTML tag with class="wikitable"
 html_node(css = '.wikitable') %>%
 # parse table into data.frame
 html_table() %>%
 # trim for printing
 dplyr::mutate(Description = substr(Description, 1, 70))

Release Date Description
1 0.16 This is the last alpha version developed primarily by Ihaka
2 0.49 1997-04-23 This is the oldest source release which is currently availab
```

```
3 0.60 1997-12-05 R becomes an official part of the GNU Project. The code is h
4 0.65.1 1999-10-07 First versions of update.packages and install.packages funct
5 1.0 2000-02-29 Considered by its developers stable enough for production us
6 1.4 2001-12-19 S4 methods are introduced and the first version for Mac OS X
7 2.0 2004-10-04 Introduced lazy loading, which enables fast loading of data
8 2.1 2005-04-18 Support for UTF-8 encoding, and the beginnings of internatio
9 2.11 2010-04-22 Support for Windows 64 bit systems.
10 2.13 2011-04-14 Adding a new compiler function that allows speeding up funct
11 2.14 2011-10-31 Added mandatory namespaces for packages. Added a new paralle
12 2.15 2012-03-30 New load balancing functions. Improved serialization speed f
13 3.0 2013-04-03 Support for numeric index values 231 and larger on 64 bit sy
```

これによりdata.frameがされますが、スクレイプされたデータにはなように、ここでデータをす  
るがあります。ここでは、の、NAのなどがあります。

のいのデータは、をさせるためにループやそののをとすることにしてください。ウェブサイトが  
jQueryやそののをってコンテンツをすると、read\_htmlするにはであり、RSeleniumのようなよりな  
スクレーパーがながあります。

## ログインがなにrvestをする

はなは、Webをするとき、WebサイトにログインするためのユーザーIDとパスワードをするで  
す。

のをするためにしたこのでは、ここにスタックオーバーフローがしました。なれはログインする  
こと、ウェブページのをすること、データフレームをしてのページにすることです。

```
library(rvest)

#Address of the login webpage
login<-
"https://stackoverflow.com/users/login?ssrc=head&returnurl=http%3a%2f%2fstackoverflow.com%2f"

#create a web session with the desired login address
pgsession<-html_session(login)
pgform<-html_form(pgsession)[[2]] #in this case the submit is the 2nd form
filled_form<-set_values(pgform, email="*****", password="*****")
submit_form(pgsession, filled_form)

#pre allocate the final results dataframe.
results<-data.frame()

#loop through all of the pages with the desired info
for (i in 1:5)
{
 #base address of the pages to extract information from
 url<-"http://stackoverflow.com/users/*****?tab=answers&sort=activity&page="
 url<-paste0(url, i)
 page<-jump_to(pgsession, url)

 #collect info on the question votes and question title
 summary<-html_nodes(page, "div .answer-summary")
 question<-matrix(html_text(html_nodes(summary, "div"), trim=TRUE), ncol=2, byrow = TRUE)

 #find date answered, hyperlink and whether it was accepted
```

```
dateans<-html_node(summary, "span") %>% html_attr("title")
hyperlink<-html_node(summary, "div a") %>% html_attr("href")
accepted<-html_node(summary, "div") %>% html_attr("class")

#create temp results then bind to final results
rtemp<-cbind(question, dateans, accepted, hyperlink)
results<-rbind(results, rtemp)
}

#Dataframe Clean-up
names(results)<-c("Votes", "Answer", "Date", "Accepted", "HyperLink")
results$Votes<-as.integer(as.character(results$Votes))
results$Accepted<-ifelse(results$Accepted=="answer-votes default", 0, 1)
```

このループは5ページにされていますが、これはアプリケーションにわせてするがあります。はユーザーのを\*\*\*\*\*にきえました。うまくいけば、これはあなたのためのいくつかのガイダンスをします。

オンラインでWebスクレープとをむ <https://riptutorial.com/ja/r/topic/2890/webスクレープと>



# 40: xgboost

## Examples

### クロスバリデーションとxgboostによるチューニング

```
library(caret) # for dummyVars
library(RCurl) # download https data
library(Metrics) # calculate errors
library(xgboost) # model

#####
Load data from UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets.html)
urlfile <- 'https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data'
x <- getURL(urlfile, ssl.verifypeer = FALSE)
adults <- read.csv(textConnection(x), header=F)

adults <- read.csv('https://archive.ics.uci.edu/ml/machine-learning-
databases/adult/adult.data', header=F)
names(adults)=c('age', 'workclass', 'fnlwt', 'education', 'educationNum',
 'maritalStatus', 'occupation', 'relationship', 'race',
 'sex', 'capitalGain', 'capitalLoss', 'hoursWeek',
 'nativeCountry', 'income')

clean up data
adults$income <- ifelse(adults$income==' <=50K', 0, 1)
binarize all factors
library(caret)
dmy <- dummyVars(" ~ .", data = adults)
adultsTrsf <- data.frame(predict(dmy, newdata = adults))
#####

what we're trying to predict adults that make more than 50k
outcomeName <- c('income')
list of features
predictors <- names(adultsTrsf)[!names(adultsTrsf) %in% outcomeName]

play around with settings of xgboost - eXtreme Gradient Boosting (Tree) library
https://github.com/tqchen/xgboost/wiki/Parameters
max.depth - maximum depth of the tree
nrounds - the max number of iterations

take first 10% of the data only!
trainPortion <- floor(nrow(adultsTrsf)*0.1)

trainSet <- adultsTrsf[1:floor(trainPortion/2),]
testSet <- adultsTrsf[(floor(trainPortion/2)+1):trainPortion,]

smallestError <- 100
for (depth in seq(1,10,1)) {
 for (rounds in seq(1,20,1)) {

 # train
 bst <- xgboost(data = as.matrix(trainSet[,predictors]),
 label = trainSet[,outcomeName],
 max.depth=depth, nround=rounds,
 objective = "reg:linear", verbose=0)
```

```

 gc()

 # predict
 predictions <- predict(bst, as.matrix(testSet[,predictors]),
outputmargin=TRUE)
 err <- rmse(as.numeric(testSet[,outcomeName]), as.numeric(predictions))

 if (err < smallestError) {
 smallestError = err
 print(paste(depth, rounds, err))
 }
 }
}

cv <- 30
trainSet <- adultsTrsf[1:trainPortion,]
cvDivider <- floor(nrow(trainSet) / (cv+1))

smallestError <- 100
for (depth in seq(1,10,1)) {
 for (rounds in seq(1,20,1)) {
 totalError <- c()
 indexCount <- 1
 for (cv in seq(1:cv)) {
 # assign chunk to data test
 dataTestIndex <- c((cv * cvDivider):(cv * cvDivider + cvDivider))
 dataTest <- trainSet[dataTestIndex,]
 # everything else to train
 dataTrain <- trainSet[-dataTestIndex,]

 bst <- xgboost(data = as.matrix(dataTrain[,predictors]),
 label = dataTrain[,outcomeName],
 max.depth=depth, nround=rounds,
 objective = "reg:linear", verbose=0)

 gc()
 predictions <- predict(bst, as.matrix(dataTest[,predictors]),
outputmargin=TRUE)

 err <- rmse(as.numeric(dataTest[,outcomeName]),
as.numeric(predictions))
 totalError <- c(totalError, err)
 }
 if (mean(totalError) < smallestError) {
 smallestError = mean(totalError)
 print(paste(depth, rounds, smallestError))
 }
 }
}

#####
Test both models out on full data set

trainSet <- adultsTrsf[1:trainPortion,]

assign everything else to test
testSet <- adultsTrsf[(trainPortion+1):nrow(adultsTrsf),]

bst <- xgboost(data = as.matrix(trainSet[,predictors]),
 label = trainSet[,outcomeName],
 max.depth=4, nround=19, objective = "reg:linear", verbose=0)
pred <- predict(bst, as.matrix(testSet[,predictors]), outputmargin=TRUE)

```

```
rmse(as.numeric(testSet[,outcomeName]), as.numeric(pred))

bst <- xgboost(data = as.matrix(trainSet[,predictors]),
 label = trainSet[,outcomeName],
 max.depth=3, nround=20, objective = "reg:linear", verbose=0)
pred <- predict(bst, as.matrix(testSet[,predictors]), outputmargin=TRUE)
rmse(as.numeric(testSet[,outcomeName]), as.numeric(pred))
```

オンラインでxgboostをむ <https://riptutorial.com/ja/r/topic/3239/xgboost>

## 41: インtrosペクション

### Examples

をするための

くの、`R`では、しているオブジェクトやにすることをりたいことがあります。これは、のコードをんだり、のコードをんだりするとき、にあなたにとってめてのパッケージをうときにです。

`a`をするとします。

```
a <- matrix(1:9, 3, 3)
```

これはどのデータですかあなたとることができます

```
> class(a)
[1] "matrix"
```

これはなので、がします

```
> a %*% t(a)
 [,1] [,2] [,3]
[1,] 66 78 90
[2,] 78 93 108
[3,] 90 108 126
```

`a`のはですか

```
> dim(a)
[1] 3 3
> nrow(a)
[1] 3
> ncol(a)
[2] 3
```

さまざまなデータにしてするそののなは、`head`、`tail`、および`str`です。

```
> head(a, 1)
 [,1] [,2] [,3]
[1,] 1 4 7
> tail(a, 1)
 [,1] [,2] [,3]
[3,] 3 6 9
> str(a)
int [1:3, 1:3] 1 2 3 4 5 6 7 8 9
```

これらは、きなオブジェクトきなデータセットなどにはるかにちます。`str`はリストのネストについてするのにもです。`a`のようなをりし`a`ください

```
a <- c(a)
```

クラスはじまますか

```
> class(a)
[1] "integer"
```

いいえ、`a`はもはやではありません。がをめるなら、はいえをられません

```
> dim(a)
NULL
```

わりに、はさをめることができます

```
> length(a)
[1] 9
```

はどう

```
> class(a * 1.0)
[1] "numeric"
```

くの、`data.frames`することができます

```
a <- as.data.frame(a)
names(a) <- c("var1", "var2", "var3")
```

をしてください

```
> names(a)
[1] "var1" "var2" "var3"
```

これらは、`R`をするときにさまざまなでちます。

オンラインでイントロスペクションをむ <https://riptutorial.com/ja/r/topic/3565/イントロスペクション>

## 42: キャレット

き

`caret`はのになデータをするRパッケージです。とののです。のデータセットのモデルをするには、データの、なの、テストセットのモデルの、なるモデルのなど、するがあるのアルゴリズムのタスクがいくつかあります。

`caret`は、されるのアルゴリズムとはして、これらのシナリオでちます。

### Examples

キャレットでは、`preProcess()`によってわれます。えられたまたはデータフレームタイプのオブジェクト`x`、`preProcess()`がトレーニングデータにをし、テストデータにすることができます。

`preProcess()`のは、`method`です。メソッドはのでされます。

1. ゼロフィルタ
2. ゼロにいフィルタ
3. Box-Cox / Yeo-Johnson /
4. センタリング
5. スケーリング
- 6.
- 7.
8. PCA
9. ICA
- 10.

では、`mtcars`データセットをし、センタリング、スケーリング、およびをします。

```
auto_index <- createDataPartition(mtcars$mpg, p = .8,
 list = FALSE,
 times = 1)

mt_train <- mtcars[auto_index,]
mt_test <- mtcars[-auto_index,]

process_mtcars <- preProcess(mt_train, method = c("center", "scale", "spatialSign"))

mtcars_train_transf <- predict(process_mtcars, mt_train)
mtcars_test_tranf <- predict(process_mtcars, mt_test)
```

オンラインでキャレットをむ <https://riptutorial.com/ja/r/topic/4271/キャレット>

## 43: クラス

き

データオブジェクトのクラスは、どのがそのをするかをしします。 `class` はベクトルであり、オブジェクトはゼロ、1つのクラスをつことができます。クラスがないでも、オブジェクトの `mode` によってまるのクラスがし `mode`。クラスは `class` ででき、 `class<- function` でまたはできます。S3クラスのシステムは、Sののにされました。よりなS4クラスのシステムがにされた

オブジェクトの "タイプ"をするためのいくつかのがあります。そのようなもなは `class` ですが、にはオブジェクトの `mode` をべるがあります。たちは、「タイプ」をしているので、つはどうかもしれない `typeof` であろうが、からの `mode` な「クラス」をつオブジェクトは `-attribute` は、ディスパッチがされた「のクラス」によってされますので、よりになりますらのモードによって。

### Examples

ベクトル

Rでできるもなデータはベクトルです。 `c()` をすると、、、およびのベクトルをできます。えは

```
c(1, 2, 3)
[1] 1 2 3
c(TRUE, TRUE, FALSE)
[1] TRUE TRUE FALSE
c("a", "b", "c")
[1] "a" "b" "c"
```

また、 `c()` をってベクトルにすることもできます。

```
x <- c(1, 2, 5)
y <- c(3, 4, 6)
z <- c(x, y)
z
[1] 1 2 5 3 4 6
```

ベクトルをするのよりなは、[「ベクトルの」トピック](#)でつけることができます

クラスをする

Rのすべてのオブジェクトにはクラスがりてられます。 `class()` をしてオブジェクトのクラスをし、 `str()` をして、にまれるクラスをむをすることができます。えは

```
class(iris)
[1] "data.frame"

str(iris)
```

```
'data.frame': 150 obs. of 5 variables:
 $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...

class(iris$Species)
[1] "factor"
```

には `data.frame` というクラスがあり、 `str()` をうとのデータをべることが出来ます。データフレームのは、クラスファクタであり、クラスであるのとはである。 `str()` はのさもし、の2つのをしますが、 `class()` はオブジェクトのクラスのみをします。

## ベクトルとリスト

Rのデータはベクトルにされます。なベクトルは、すべてじモードえば、ベクトル、ベクトルをするのである。してください?atomic のクラスとそれにするストレージモードのについては、  
"logical", "integer", "numeric" (synonym "double"), "complex", "character"と"raw"。くのクラスは、に `class` をつベクトルです。

```
x <- 1826
class(x) <- "Date"
x
[1] "1975-01-01"
x <- as.Date("1970-01-01")
class(x)
[1] "Date"
is(x, "Date")
[1] TRUE
is(x, "integer")
[1] FALSE
is(x, "numeric")
[1] FALSE
mode(x)
[1] "numeric"
```

リストはなタイプのベクトルであり、はか、のリストであってもよく、リストのRは「ベクトル」です。

```
mylist <- list(A = c(5,6,7,8), B = letters[1:10], CC = list(5, "Z"))
```

リストには2つのになががあります。

- はのしかすことができないので、なをリストにすのがです。

```
f <- function(x) list(xplus = x + 10, xsq = x^2)

f(7)
$xplus
[1] 17
#
```



```
$xsq
[1] 49
```

- リストは、[データフレーム](#)のとなるクラスでもあります。フードのでは、データフレームはすべてじさをつベクトルのリストです。

```
L <- list(x = 1:2, y = c("A", "B"))
DF <- data.frame(L)
DF
x y
1 1 A
2 2 B
is.list(DF)
[1] TRUE
```

ベクトルののクラスはRであり、これは「」 - オブジェクト

[オンラインでクラスをむ](#) <https://riptutorial.com/ja/r/topic/3563/クラス>

---

## 44: グラフィックスのカラースキーム

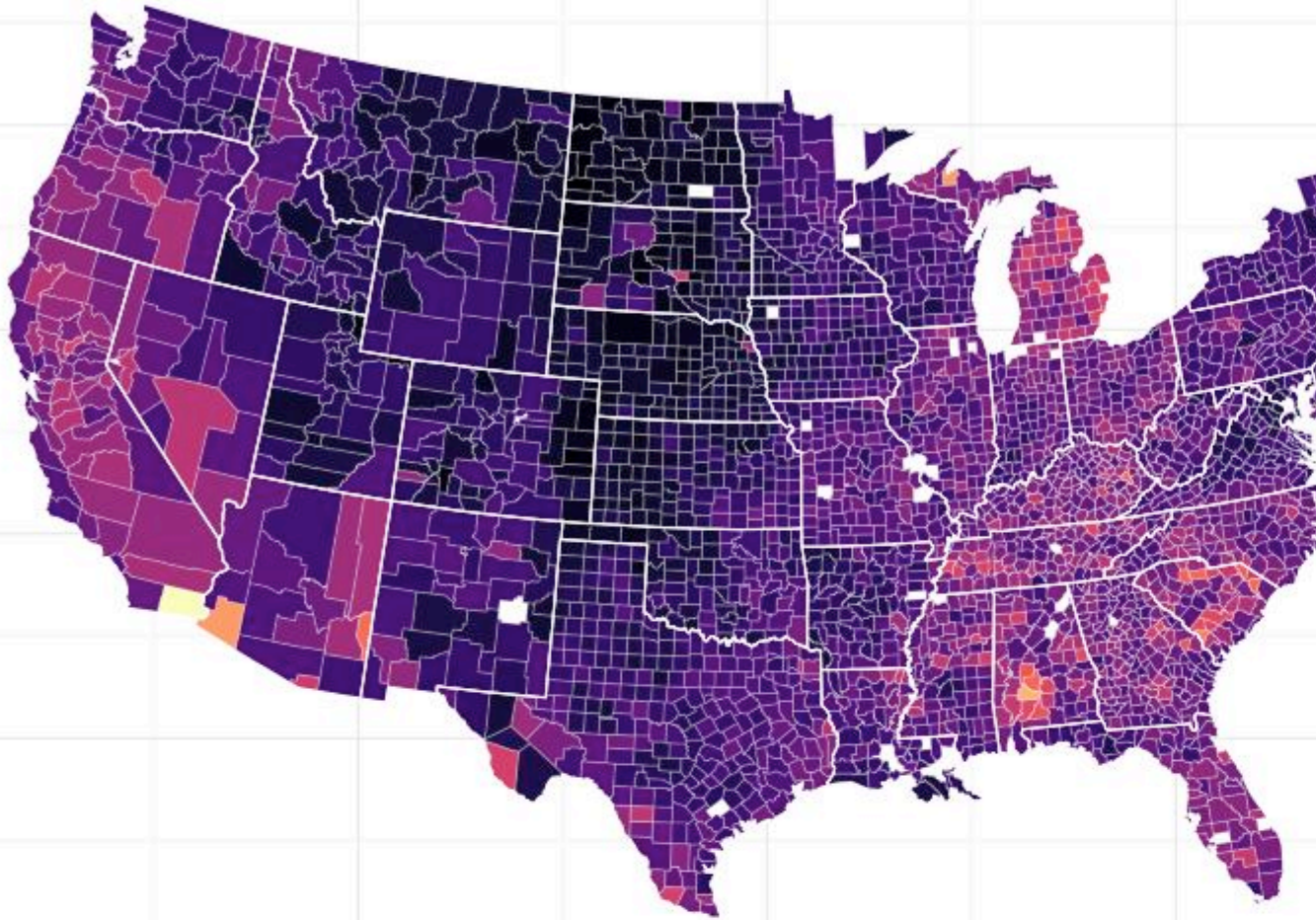
### Examples

**viridis** - フレンドリーなパレットをしてカラーブラインドする

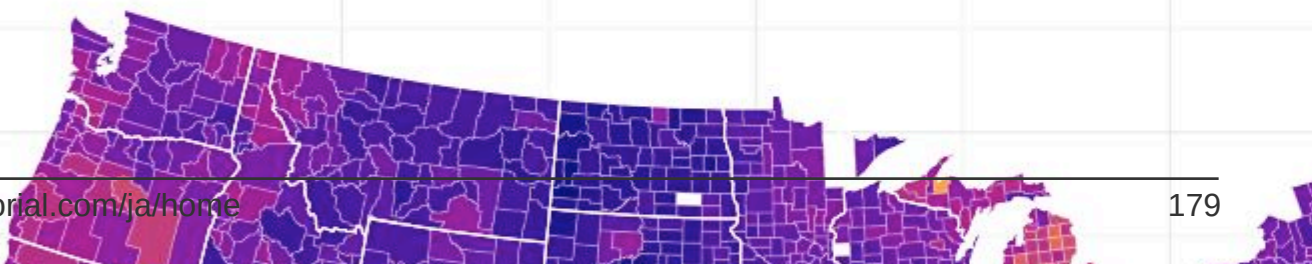
Viridis [chromis viridis fish](#)にちなんでは、[されたPythonライブラリmatplotlib](#)です このリンクのビデオプレゼンテーションでは、がどのようにされ、そのながあるかしています。これはRシームレスにされます。

には、magma、plasma、inferno、viridis 4がありますデフォルト。これらはoptionパラメータでoptionされ、してA、B、C、Dとしてコードされます。4つののをたせるには、をてください

option A aka 'magma'



option C aka 'plasma'



## ソース

---

パッケージは[CRAN](#)または[github](#)からインストールできます。

---

`viridis`パッケージのビネットはです。

---

`viridis`カラスキームの`ggplot2`は、`ggplot2`との`ggplot2`。パッケージには、`scale_color_viridis()`と`scale_fill_viridis()` 2つの`ggplot2`がされています。のをしてください。

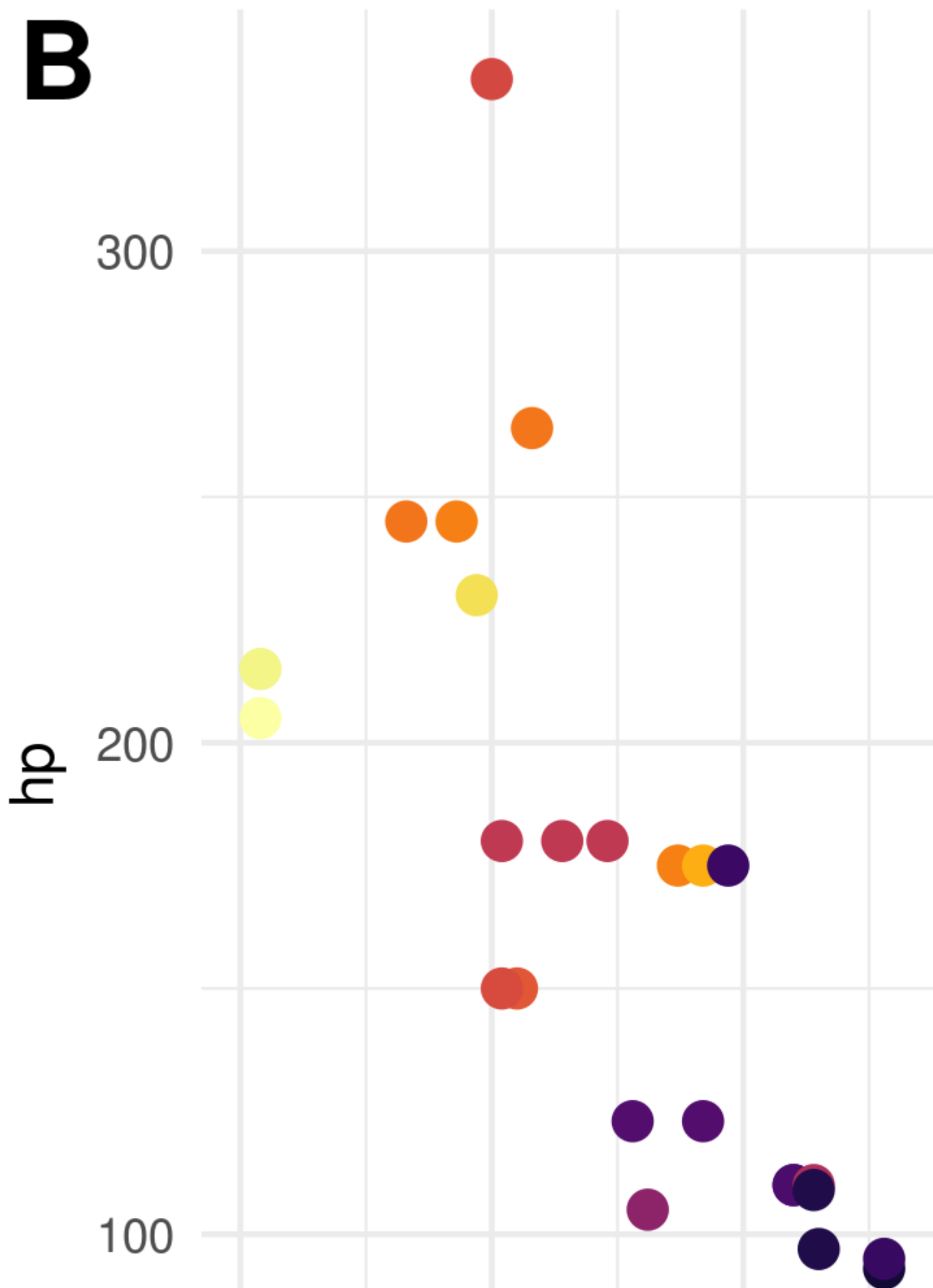
```
library(viridis)
library(ggplot2)

gg1 <- ggplot(mtcars)+
 geom_point(aes(x = mpg, y = hp, color = disp), size = 3)+
 scale_color_viridis(option = "B")+
 theme_minimal()+
 theme(legend.position = c(.8,.8))

gg2 <- ggplot(mtcars)+
 geom_violin(aes(x = factor(cyl), y = hp, fill = factor(cyl)))+
 scale_fill_viridis(discrete = T)+
 theme_minimal()+
 theme(legend.position = 'none')

library(cowplot)
output <- plot_grid(gg1,gg2, labels = c('B','D'),label_size = 20)
print(output)
```

**B**



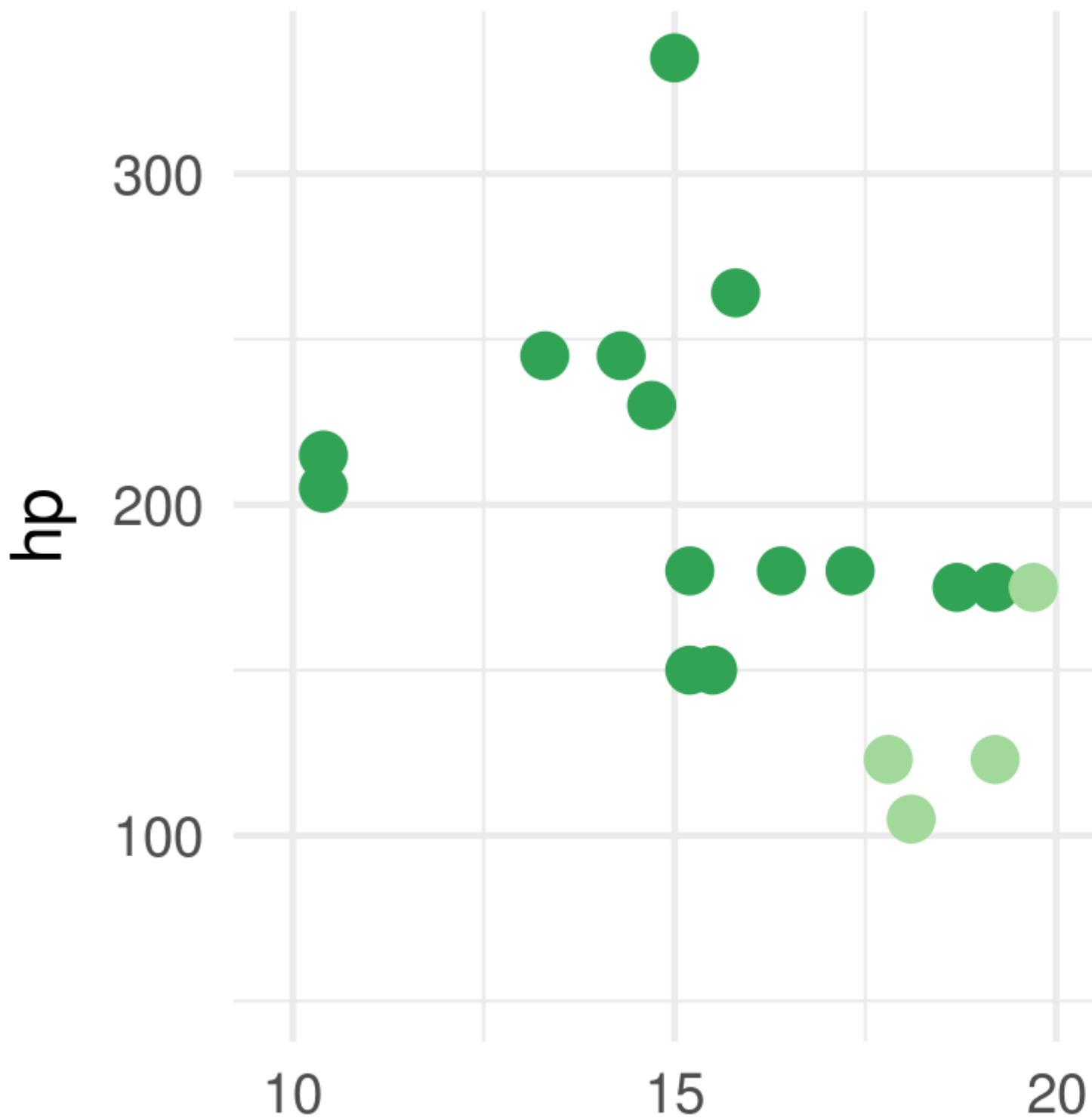
プロジェクトのポートで、カラーブラインドにしたパレットもします。

---

```
colors_vec <- brewer.pal(5, name = 'BrBG')
print(colors_vec)
[1] "#A6611A" "#DFC27D" "#F5F5F5" "#80CDC1" "#018571"
```

RColorBrewerのためののオプションをggplot2 scale\_color\_brewerとscale\_fill\_brewer。

```
library(ggplot2)
ggplot(mtcars) +
 geom_point(aes(x = mpg, y = hp, color = factor(cyl)), size = 3) +
 scale_color_brewer(palette = 'Greens') +
 theme_minimal() +
 theme(legend.position = c(.8, .8))
```



のベクトルをえるな

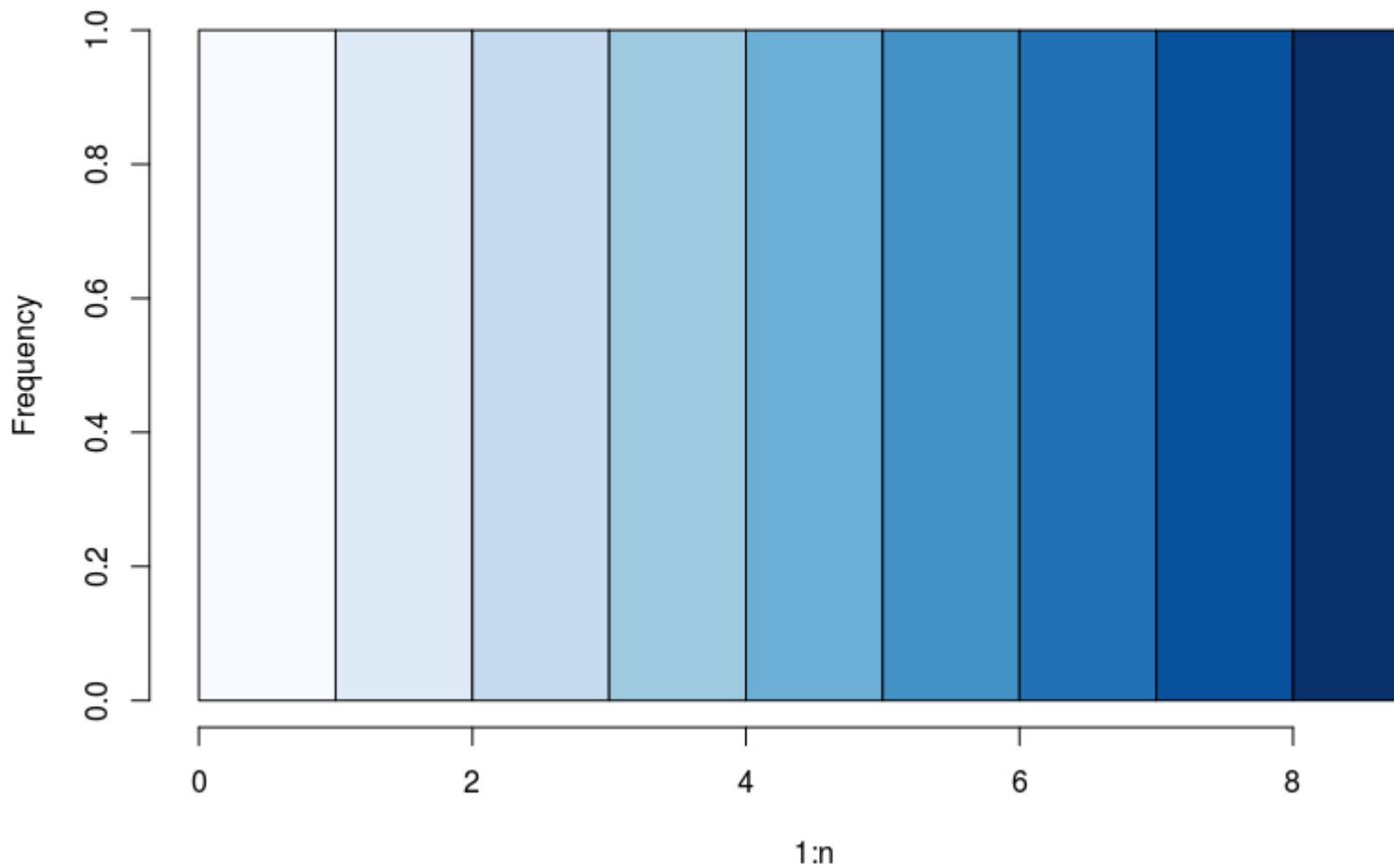
かなりのので、したカラーパレットをるがあります。なソリューションの1つは、のです。

```
color_glimpse <- function(colors_string){
 n <- length(colors_string)
```

```
hist(1:n,breaks=0:n,col=colors_string)
}
```

```
color_glimpse(blues9)
```

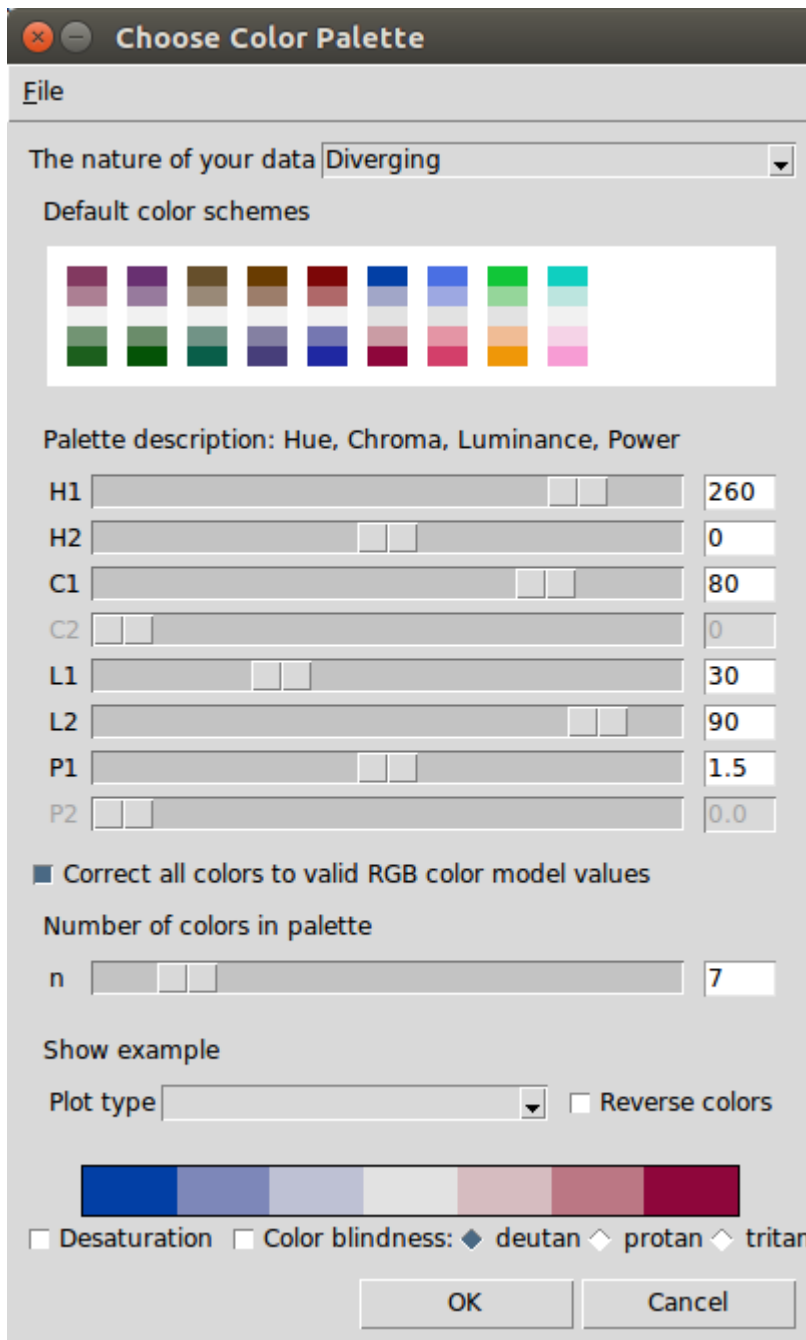
Histogram of 1:n



**colorspace** - のインターフェースをクリックドラッグする

パッケージの `colorspace` は、パレットをするためのGUIをします。 `choose_palette()` のびしでは、のウィンドウがポップアップします。





パレットがされたら、OKだけで、を、例えば`pal`にすることをしないでください。

```
pal <- choose_palette()
```

は`n` をとし、したパレットにじてさ`n` カラーベクトルをします。

```
pal(10)
[1] "#023FA5" "#6371AF" "#959CC3" "#BEC1D4" "#DBDCE0" "#E0DBDC" "#D6BCC0" "#C6909A" "#AE5A6D"
"#8E063B"
```

なRカラー

の`colors()` は、Rによってされるすべてのをリストします。にそれらのをることができるなPDFがあります。

`colorRampPalette`は、与えられた色のセットをしてほしいカラーパレットを作るをします。このは  $n$  をとし、`colorRampPalette` をする  $n$  カラーベクトルをします。

```
pal <- colorRampPalette(c('white','red'))
pal(5)
[1] "#FFFFFF" "#FFBFBF" "#FF7F7F" "#FF3F3F" "#FF0000"
```

これは、`rgb()` でできます。

```
rgb(0,1,0)
```

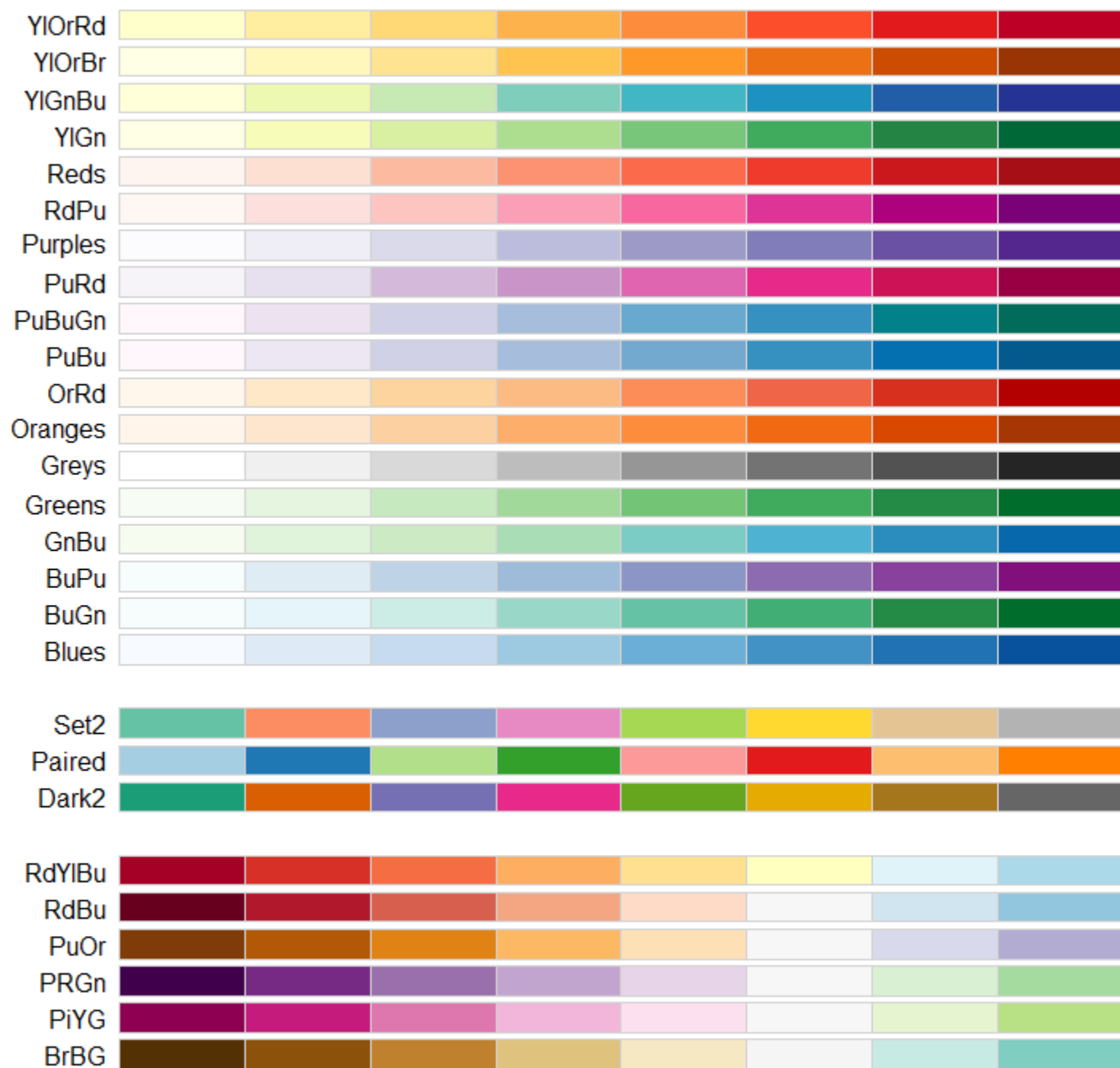
green する。

カラーブラインドにしいパレット

はのをすることができますが、のをすることはしいかもしれません。

`RColorBrewer` はカラーブラインドにしいパレットをします

```
library(RColorBrewer)
display.brewer.all(colorblindFriendly = T)
```



のカラーユニバーサルデザインは、のパレットをしています。

```
#palette using grey
```

```
cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00",
"#CC79A7")

#palette using black
cbbPalette <- c("#000000", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00",
"#CC79A7")
```

オンラインでグラフィックスのカラースキームをむ <https://riptutorial.com/ja/r/topic/8005/グラフィックスのカラースキーム>

## 45: コードプロファイリング

### Examples

#### システム

システムは、RをするためになCPUをします。たとえば、のようになります。

```
system.time(print("hello world"))

[1] "hello world"
user system elapsed
0 0 0
```

をして、よりきなコードをすることができます。

```
system.time({
 library(numbers)
 Primes(1,10^5)
})
```

または、をテストするためにします。

```
fibb <- function (n) {
 if (n < 3) {
 return(c(0,1)[n])
 } else {
 return(fibb(n - 2) + fibb(n -1))
 }
}

system.time(fibb(30))
```

#### proc.time

もな、`proc.time()`は、のプロセスのCPUをで`proc.time()`ます。コンソールですと、のようながられます。

```
proc.time()

user system elapsed
284.507 120.397 515029.305
```

これは、のコードをベンチマークするににです。えは

```
t1 <- proc.time()
fibb <- function (n) {
 if (n < 3) {
 return(c(0,1)[n])
 }
}
```

```

 } else {
 return(fibb(n - 2) + fibb(n -1))
 }
 }
print("Time one")
print(proc.time() - t1)

t2 <- proc.time()
fibb(30)

print("Time two")
print(proc.time() - t2)

```

これにより、のがられます。

```

source('~/.active-rstudio-document')

[1] "Time one"
user system elapsed
0 0 0

[1] "Time two"
user system elapsed
1.534 0.012 1.572

```

`system.time()` は、のコマンド/のをす `proc.time()` ラッパーです。

```

print(t1 <- system.time(replicate(1000, 12^2)))
user system elapsed
0.000 0.000 0.002

```

クラス `proc.time` されるオブジェクトは、サーフェスにされるよりもややです。

```

str(t1)
Class 'proc_time' Named num [1:5] 0 0 0.002 0 0
..- attr(*, "names")= chr [1:5] "user.self" "sys.self" "elapsed" "user.child" ...

```

## ラインプロファイリング

ラインプロファイリングのための1つのパッケージは、Hadley Wickhamによってかれされている [lineprof](#) です。パッケージの `auto.arima` でどのようにするかをにします。

```

library(lineprof)
library(forecast)

l <- lineprof(auto.arima(AirPassengers))
shine(l)

```

これはるアプリをします。これにより、すべてのびしをくりげることができます。これにより、Rコードがくなっているをにすることができます。のようなのあるアプリのスクリーンショットがあります

# Line profiling

[Back](#)

| #  | Source code              | t | r | a | d |
|----|--------------------------|---|---|---|---|
| 1  | nsdiffs/OCSBtest         |   |   |   |   |
| 2  | nsdiffs/diff             |   |   |   |   |
| 3  | nsdiffs/OCSBtest         |   |   |   |   |
| 4  | diff/diff.ts             |   |   |   |   |
| 5  | ndiffs/suppressWarnings  |   |   |   |   |
| 6  | ndiffs/diff              |   |   |   |   |
| 7  | diff/diff.ts             |   |   |   |   |
| 8  | try/tryCatch             |   |   |   |   |
| 9  | myarima/suppressWarnings |   |   |   |   |
| 10 |                          |   |   |   |   |
| 11 | myarima/suppressWarnings |   |   |   |   |
| 12 | myarima                  |   |   |   |   |
| 13 | myarima/suppressWarnings |   |   |   |   |
| 14 |                          |   |   |   |   |
| 15 | myarima/suppressWarnings |   |   |   |   |
| 16 | data.frame               |   |   |   |   |

## マイクロベンチマーク

Microbenchmarkは、そののではのみにちます。たとえば、こんにちををするのにかかるをもることをしてください。

```
system.time(print("hello world"))
```

```
[1] "hello world"
user system elapsed
0 0 0
```

これは、`system.time`がに`proc.time`ラッパーで`proc.time`、でされるため`proc.time`。 "hello world" のには1もかかりませんので、は1ですが、これはてはまりません。これをるために、パッケージのマイクロベンチマークをすることができます

```
library(microbenchmark)
microbenchmark(print("hello world"))
```

```
Unit: microseconds
expr min lq mean median uq max neval
print("hello world") 26.336 29.984 44.11637 44.6835 45.415 158.824 100
```

ここで`print("hello world")` 100した、して44マイクロであった。このコードをすると "hello

world"がコンソールに100されることにしてください。

これをのき `cat("hello world\n")` として、 `print("hello world")` よりかどうかをべることができ `print("hello world")`

```
microbenchmark(cat("hello world\n"))

Unit: microseconds
expr min lq mean median uq max neval
cat("hello world\n") 14.093 17.6975 23.73829 19.319 20.996 119.382 100
```

この、 `cat()` は `print()` ほぼ2のさです。

あるいは、 `microbenchmark` コールの2つのをすることもできます。

```
microbenchmark(print("hello world"), cat("hello world\n"))
Unit: microseconds
expr min lq mean median uq max neval
print("hello world") 29.122 31.654 39.64255 34.5275 38.852 192.779 100
cat("hello world\n") 9.381 12.356 13.83820 12.9930 13.715 52.564 100
```

## マイクロベンチマークをしたベンチマーク

`microbenchmark` パッケージをして、「ミリのなのタイミング」をできます。

このでは、のについて、グループのをするための6つの `data.table` のをしています。

すなわち

`id`、 `time`、 `status` 3の `data.table`。 `id` にして、のレコードをしたい - ステータスが `true` ならそのレコードのために `7` よりきいは `false` にしたい

```
library(microbenchmark)
library(data.table)

set.seed(20160723)
dt <- data.table(id = c(rep(seq(1:10000), each = 10)),
 time = c(rep(seq(1:10000), 10)),
 status = c(sample(c(TRUE, FALSE), 10000*10, replace = TRUE)))
setkey(dt, id, time) ## create copies of the data so the 'updates-by-reference' don't affect
other expressions
dt1 <- copy(dt)
dt2 <- copy(dt)
dt3 <- copy(dt)
dt4 <- copy(dt)
dt5 <- copy(dt)
dt6 <- copy(dt)

microbenchmark(

 expression_1 = {
 dt1[dt1[order(time), .I[.N], by = id]$V1, status := status * time < 7]
 },
```



```

expression_2 = {
 dt2[,status := c(.SD[.N, status], .SD[.N, status * time > 7]), by = id]
},

expression_3 = {
 dt3[dt3[, .N, by = id][,cumsum(N)], status := status * time > 7]
},

expression_4 = {
 y <- dt4[,.SD[.N],by=id]
 dt4[y, status := status & time > 7]
},

expression_5 = {
 y <- dt5[, .SD[.N, .(time, status)], by = id][time > 7 & status]
 dt5[y, status := FALSE]
},

expression_6 = {
 dt6[dt6[, .I == .I[which.max(time)], by = id]$V1 & time > 7, status := FALSE]
},

times = 10L ## specify the number of times each expression is evaluated
)

Unit: milliseconds
expr min lq mean median uq max neval
expression_1 11.646149 13.201670 16.808399 15.643384 18.78640 26.321346 10
expression_2 8051.898126 8777.016935 9238.323459 8979.553856 9281.93377 12610.869058 10
expression_3 3.208773 3.385841 4.207903 4.089515 4.70146 5.654702 10
expression_4 15.758441 16.247833 20.677038 19.028982 21.04170 36.373153 10
expression_5 7552.970295 8051.080753 8702.064620 8861.608629 9308.62842 9722.234921 10
expression_6 18.403105 18.812785 22.427984 21.966764 24.66930 28.607064 10

```

は、このテストでは`expression_3`がもいことをしています。

[data.table - のと](#)

[data.table - data.tableのなグループシンボル](#)

オンラインでコードプロファイリングをむ <https://riptutorial.com/ja/r/topic/2149/コードプロファイリング>

## 46: コンビナトリアル

### Examples

されたさのみわせをする

なし

`combn`、ベクトルがにされます。

```
combn(LETTERS, 3)

Showing only first 10.
 [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,] "A" "A" "A" "A" "A" "A" "A" "A" "A" "A"
[2,] "B" "B" "B" "B" "B" "B" "B" "B" "B" "B"
[3,] "C" "D" "E" "F" "G" "H" "I" "J" "K" "L"
```

する

`expand.grid`、ベクトルが1にされます。

```
expand.grid(LETTERS, LETTERS, LETTERS)
or
do.call(expand.grid, rep(list(LETTERS), 3))

Showing only first 10.
 Var1 Var2 Var3
1 A A A
2 B A A
3 C A A
4 D A A
5 E A A
6 F A A
7 G A A
8 H A A
9 I A A
10 J A A
```

なペアのは、`outer`をしてベクトルをセルにれます

```
FUN here is used as a function executed on each resulting pair.
in this case it's string concatenation.
outer(LETTERS, LETTERS, FUN=paste0)

Showing only first 10 rows and columns
 [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,] "AA" "AB" "AC" "AD" "AE" "AF" "AG" "AH" "AI" "AJ"
```

```
[2,] "BA" "BB" "BC" "BD" "BE" "BF" "BG" "BH" "BI" "BJ"
[3,] "CA" "CB" "CC" "CD" "CE" "CF" "CG" "CH" "CI" "CJ"
[4,] "DA" "DB" "DC" "DD" "DE" "DF" "DG" "DH" "DI" "DJ"
[5,] "EA" "EB" "EC" "ED" "EE" "EF" "EG" "EH" "EI" "EJ"
[6,] "FA" "FB" "FC" "FD" "FE" "FF" "FG" "FH" "FI" "FJ"
[7,] "GA" "GB" "GC" "GD" "GE" "GF" "GG" "GH" "GI" "GJ"
[8,] "HA" "HB" "HC" "HD" "HE" "HF" "HG" "HH" "HI" "HJ"
[9,] "IA" "IB" "IC" "ID" "IE" "IF" "IG" "IH" "II" "IJ"
[10,] "JA" "JB" "JC" "JD" "JE" "JF" "JG" "JH" "JI" "JJ"
```

されたさのみわせをえる

---

なし

```
choose(length(LETTERS), 5)
[1] 65780
```

---

する

```
length(letters)^5
[1] 11881376
```

オンラインでコンビナトリアルをむ <https://riptutorial.com/ja/r/topic/5836/コンビナトリアル>

## 47: サブセッティング

き

Rオブジェクトがえられている、そのオブジェクトに含まれるデータの1つのについてののがあります。のオブジェクトからデータのこれらのをするプロセスは、 `subsetting` とされます。

がありません

`NA` とサブセットに `S [リターン NA ので NA`

のをするので、するに `NA` がされます。

`NA` の「デフォルト」タイプは、サブセットにされる「」ベクトルがサブセットされたオブジェクトのさにするようにリサイクルされることをする「」タイプ `typeof(NA)` である。そう `x[NA]` にする `x[as.logical(NA)]` にする `x[rep_len(as.logical(NA), length(x))]` と、これにより、をす `NA x` について。として

```
x <- 1:3
x[NA]
[1] NA NA NA
```

「」/「」でけしながら、 `NA` 、のピック `NA NA` インデックスを

```
x[as.integer(NA)]
[1] NA

x[c(NA, 1, NA, NA)]
[1] NA 1 NA NA
```

のサブセット

[ 1つのがされた ]は、 `> length(x)` インデックスをし、アトミックベクトルのは `NA` 、ジェネリックベクトルのは `NULL` を `NULL` 。に、 [[と [のがされすなわち、のサブセットをいてオブジェクトの `length(dim(x)) > 2` エラーがさです。

```
(1:3)[10]
[1] NA
(1:3)[[10]]
Error in (1:3)[[10]] : subscript out of bounds
as.matrix(1:3)[10]
[1] NA
as.matrix(1:3)[, 10]
Error in as.matrix(1:3)[, 10] : subscript out of bounds
list(1, 2, 3)[10]
[[1]]
NULL
list(1, 2, 3)[[10]]
```

```
Error in list(1, 2, 3)[[10]] : subscript out of bounds
```

これは、オブジェクトの「」でない「」ベクトルでサブセットするもじです。

```
c(a = 1, b = 2)["c"]
<NA>
NA
list(a = 1, b = 2)["c"]
<NA>
NULL
```

ヘルプトピックス

については、「`?Extract`」をしてください。

## Examples

ベクトル

ベクトルリストともしますが、これもベクトルでは、`[`

```
create an example vector
v1 <- c("a", "b", "c", "d")

select the third element
v1[3]
[1] "c"
```

`[`はとしてベクトルをとることもできます。たとえば、と3のをするには

```
v1 <- c("a", "b", "c", "d")

v1[c(1, 3)]
[1] "a" "c"
```

にはベクトルからののをするがあるもあります。これは、そののインデックスのにマイナス `-` をしてできます。たとえば、`v1`からののをするは、`v1[-1]`します。これは、2つのにすることができま  
す。たとえば、`v1[-c(1,3)]`ます。

```
> v1[-1]
[1] "b" "c" "d"
> v1[-c(1,3)]
[1] "b" "d"
```

によっては、に、ベクトルのさがいときにののインデックスがするはそれをりたいといいます。

```
> v1=="c"
[1] FALSE FALSE TRUE FALSE
> which(v1=="c")
[1] 3
```

アトムベクトルに `names` があるは、のベクトルをしてサブセットにすることができます。

```
v <- 1:3
names(v) <- c("one", "two", "three")

v
one two three
1 2 3

v["two"]
two
2
```

`[`は、さが1のインデックスベクトルをけり、するのをりくというで、ベクトルをインデックスけするためにもできます。

```
v[[c(1, 2)]]
Error in v[[c(1, 2)]] :
attempt to select more than one element in vectorIndex

v[["two"]]
[1] 2
```

また、ベクトルはベクトルをしてサブセットにすることもできます。ベクトルとベクトルによるサブセットとはに、サブセットにされるベクトルは、がされるベクトルのさとしくなければならぬ。したがって、ベクトル $y$ が $x$ をサブセットするためにされる、すなわち $x[y]$   $\text{length}(y) < \text{length}(x)$   $y$ は $\text{length}(x)$ とするようにリサイクルされます

```
v[c(TRUE, FALSE, TRUE)]
one three
1 3

v[c(FALSE, TRUE)] # recycled to 'c(FALSE, TRUE, FALSE)'
two
2

v[TRUE] # recycled to 'c(TRUE, TRUE, TRUE)'
one two three
1 2 3

v[FALSE] # handy to discard elements but save the vector's type and basic structure
named integer(0)
```

## リスト

リストは`[`

```
l1 <- list(c(1, 2, 3), 'two' = c("a", "b", "c"), list(10, 20))
l1
[[1]]
[1] 1 2 3
##
$two
```

```
[1] "a" "b" "c"
##
[[3]]
[[3]][[1]]
[1] 10
##
[[3]][[2]]
[1] 20

l1[1]
[[1]]
[1] 1 2 3

l1['two']
$two
[1] "a" "b" "c"

l1[[2]]
[1] "a" "b" "c"

l1[['two']]
[1] "a" "b" "c"
```

`l1`はリストのをしてよりさなリストをすので、`l1[2]`のはとしてリストであることにしてください。`l1`はリストをし、リストののオブジェクトをします。

は、またはのでできますする。のは、`l1`のやのをすことで`l1`ですることができます。ベクトルとし`length > 1`で`l1`および`l1`、それぞれ、されたとサブセットなで「リスト」をします。

```
l1[c(3, 1)]
[[1]]
[[1]][[1]]
[1] 10
##
[[1]][[2]]
[1] 20
##
##
[[2]]
[1] 1 2 3
```

にべ

```
l1[[c(3, 1)]]
[1] 10
```

これはとです

```
l1[[3]][[1]]
[1] 10
```

`$`をすると、だけでリストをできますが、`l1`と`l1`、はありません。として、`$`は1つのしかることをできません

```
l1$two
[1] "a" "b" "c"
```

また、`$`はデフォルトでをにします

```
l1$t
[1] "a" "b" "c"
```

がされるかどうかをするがある[[

```
l1[["t"]]
NULL
l1[["t", exact = FALSE]]
[1] "a" "b" "c"
```

`options(warnPartialMatchDollar = TRUE)``options(warnPartialMatchDollar = TRUE)`、`$`こったときに "`$`"がえられます

```
l1$t
[1] "a" "b" "c"
Warning message:
In l1$t : partial match of 't' to 'two'
```

オブジェクトのについて、`[`は1つのをとります。ベクトルはが1つあり、は1つです。とデータフレームは2つのをち、`[i, j]`としてえられる2つのをる`[i, j]`ここで、`i`はであり、`j`はである。インデックスは1からまります。

```
a sample matrix
mat <- matrix(1:6, nrow = 2, dimnames = list(c("row1", "row2"), c("col1", "col2", "col3")))

mat
col1 col2 col3
row1 1 3 5
row2 2 4 6
```

`mat[i, j]`は、`mat`の`i`の`j`ののです。たとえば、`i`が`2`で`j`が`1`は、の2のと1ののがえられます。`i`または`j`をすると、そののすべてのがされます。

```
mat[, 3]
row1 row2
5 6

mat[1,]
col1 col2 col3
1 3 5
```

にまたはののではないがあるは、これらをサブセットにできます。

```
mat[, 'col1']
row1 row2
1 2
```



デフォルトでは、サブセットのはなりされます。ののようにサブセットにが1つしかない、は2のではなく1のベクトルになります。このデフォルトは、 `drop = FALSE`で[

```
This selects the first row as a vector
class(mat[1,])
[1] "integer"

Whereas this selects the first row as a 1x3 matrix:
class(mat[1, , drop = F])
[1] "matrix"
```

もちろん、に2つのディメンションがある、ディメンションをすることはできません。

```
mat[1:2, 2:3] ## A 2x2 matrix
col2 col3
row1 3 5
row2 4 6
```

それらのによって々のマトリクスエントリをする

Nx2マトリックスをして、マトリックスからNの々のをすることもできますのみなど。ベクトル (1st row, 1st column), (1st row, 3rd column), (2nd row, 3rd column), (2nd row, 1st column)のエントリをベクトルでするは、これらのでインデックスをし、それをつてをサブセットすることによってえます

```
mat
col1 col2 col3
row1 1 3 5
row2 2 4 6

ind = rbind(c(1, 1), c(1, 3), c(2, 3), c(2, 1))
ind
[,1] [,2]
[1,] 1 1
[2,] 1 3
[3,] 2 3
[4,] 2 1

mat[ind]
[1] 1 5 6 2
```

のでは、 `ind`の1は`mat`をし、2の`ind`は`mat`をします。

## データフレーム

よりさなデータフレームにデータフレームをサブセットすることは、リストをサブセットすることと同じようにすることができる。

```
> df3 <- data.frame(x = 1:3, y = c("a", "b", "c"), stringsAsFactors = FALSE)

> df3
```

```
x y
1 1 a
2 2 b
3 3 c

> df3[1] # Subset a variable by number
x
1 1
2 2
3 3

> df3["x"] # Subset a variable by name
x
1 1
2 2
3 3

> is.data.frame(df3[1])
TRUE

> is.list(df3[1])
TRUE
```

データフレームをベクトルにサブセットするには、`[]`またはドル`$`をします。

```
> df3[[2]] # Subset a variable by number using [[]]
[1] "a" "b" "c"

> df3[["y"]] # Subset a variable by name using [[]]
[1] "a" "b" "c"

> df3$x # Subset a variable by name using $
[1] 1 2 3

> typeof(df3$x)
"integer"

> is.vector(df3$x)
TRUE
```

データを $2$ としてサブセットすることは、 $i$ および $j$ をしてすることができる。

```
> df3[1, 2] # Subset row and column by number
[1] "a"

> df3[1, "y"] # Subset row by number and column by name
[1] "a"

> df3[2,] # Subset entire row by number
x y
2 2 b

> df3[, 1] # Subset all first variables
[1] 1 2 3

> df3[, 1, drop = FALSE]
x
1 1
```

```
2 2
3 3
```

j によるサブセットだけで、のiにされますが、iだけでサブセットするとdata.frameがされます。なるはなるとクラスをつががあります。dropパラメータをFALSEすると、データフレームがされFALSE。

```
> is.vector(df3[, 2])
TRUE

> is.data.frame(df3[2,])
TRUE

> is.data.frame(df3[, 2, drop = FALSE])
TRUE
```

## そのオブジェクト

[と[はなプリミティブです。これは、RのオブジェクトにisTRUE(is.object(x)) ieがな"class"をつが、サブセットされたときにそののされたいをつことができることをします。すなわちのためのメソッドをとっている[およびまたは[[

えは、これは[.data.frame is.object(iris) オブジェクトので、[.data.frameと[[.data.frameメソッドがされていて、それらがとも"matrix"のようにされる「リスト」のようなサブセッティングがまれます。「data.frame」をサブセット、々がいることがわかり、エラーをすると、には、[.data.frame、々が-just-ときにびされた[。

```
iris[invalidArgument,]
Error in `[.data.frame'(iris, invalidArgument,) :
object 'invalidArgument' not found
```

## のトピックにするながないり、[メソッド

```
x = structure(1:5, class = "myClass")
x[c(3, 2, 4)]
[1] 3 2 4
' [.myClass' = function(x, i) cat(sprintf("We'd expect '%s[%s]'" to be returned but this a
custom `[` method and should have a `?[.myClass` help page for its behaviour\n",
deparse(substitute(x))), deparse(substitute(i))))

x[c(3, 2, 4)]
We'd expect 'x[c(3, 2, 4)]' to be returned but this a custom `[` method and should have a
`?[.myClass` help page for its behaviour
NULL
```

たちは、[ genericのな.subset と[[ .subset2 をすることで、[ディスパッチをすることができます。これは、たちの"クラス"をプログラミングするとき、そして"クラス"をにメソッドのディスパッチとオブジェクトのコピーをけてするときの unclass(x) をけたいときににです。

```
.subset(x, c(3, 2, 4))
```

```
[1] 3 2 4
```

## ベクトルけ

ここでは、のベクトルをします。

```
> x <- 11:20
> x
[1] 11 12 13 14 15 16 17 18 19 20
```

Rベクトルは1つのインデックスがけられているので、えは`x[1]`は11をします。また、えベクトルをブラケットにすことで、`x`サブベクトルをすることもできます。

```
> x[c(2,4,6)]
[1] 12 14 16
```

のインデックスのベクトルをすと、されたインデックスがされたサブベクトルがされます。

```
> x[c(-1,-3)]
[1] 12 14 15 16 17 18 19 20
```

ブールにブールベクトルをすこともできます。この、インデックスベクトルが`TRUE`にするサブベクトルがされます。

```
> x[c(rep(TRUE,5),rep(FALSE,5))]
[1] 11 12 13 14 15 16
```

インデックスベクトルがのさよりもいは、のようになりされます。

```
> x[c(TRUE,FALSE)]
[1] 11 13 15 17 19
> x[c(TRUE,FALSE,FALSE)]
[1] 11 14 17 20
```

の

AとBをじの2つのとする。+、-、/、\*、^は、じのとにすると、のするにしてなをし、じのしをします。これらは、ごとのとばれます。

| オペレーター | オペラB  |              |
|--------|-------|--------------|
| +      | A + B | AとBのするの      |
| -      | A - B | BのをAのするからきます |
| /      | A / B | AのをBのするでします。 |

| オペレーター | オペラ B    |              |
|--------|----------|--------------|
| *      | $A * B$  | AのにBのするをします。 |
| ^      | $A^{-1}$ | えは、がAのであるをえる |

にられるように、"の"の、`%%`します。たとえば、AとBのは、`A %% B`です。があること`ncol()`のとじである`A nrow()`のB

でされる

| nrow      | ネロ—A       | Aのをする                       |
|-----------|------------|-----------------------------|
| ncol      | ncolA      | Aのをする                       |
| rownames  | rownamesA  | Aのをします。                     |
| colnames  | colnamesA  | Aのをします。                     |
| rowMeans  | rowMeansA  | Aののをする                      |
| colMeans  | colMeansA  | Aののをする                      |
| upper.tri | upper.triA | がであるベクトルをします                |
|           |            | Aの                          |
| lower.tri | lower.triA | がのベクトルをします。                 |
|           |            | Aの                          |
| det       | detA       | としてAのがられる                   |
| する        | するA        | Aのがられる                      |
| diag      | ダイアグA      | をします。そのは、がゼロであり、            |
|           |            | はAのとじである                    |
| t         | tA         | Aのをします                      |
|           | A          | Aのとベクトルをする                  |
| is.matrix | is.matrixA | AがかどうかによってTRUEまたはFALSEをします。 |
| as.matrix | as.matrixx | ベクトル $x$ からをする              |

オンラインでサブセッティングをむ <https://riptutorial.com/ja/r/topic/1686/サブセッティング>

---

## 48: シャイニー

### Examples

アプリをする

Shinyは、[RStudio](#)によってされたRパッケージで、RのをインタラクティブにするWebページのをにします。

Shinyアプリケーションをするには2つのながあります

- 1つの.Rファイル、または
- 2つのファイル `ui.R`と`server.R`。

Shinyアプリは2つのにかれています

- **ui** アプリケーションのレイアウトとをするユーザーインターフェイススクリプト。
- サーバー アプリケーションがするためのコードをむサーバースクリプト。

---

## 1つのファイル

```
library(shiny)

Create the UI
ui <- shinyUI(fluidPage(
 # Application title
 titlePanel("Hello World!")
))

Create the server function
server <- shinyServer(function(input, output){})

Run the app
shinyApp(ui = ui, server = server)
```

---

## 2つのファイル

`ui.R` ファイルをする

```
library(shiny)

Define UI for application
shinyUI(fluidPage(
 # Application title
 titlePanel("Hello World!")
))
```

```
))
```

## server.R ファイルをする

```
library(shiny)

Define server logic
shinyServer(function(input, output){})
```

### ラジオボタン

リストからをするためのラジオボタンのセットをできます。

をすることはです

- にしたしないは0
- インラインまたは
- 

HTMLをすることもです。

```
library(shiny)

ui <- fluidPage(
 radioButtons("radio",
 label = HTML('Welcome

Your favorite color is red ?'),
 choices = list("TRUE" = 1, "FALSE" = 2),
 selected = 1,
 inline = T,
 width = "100%"),
 fluidRow(column(3, textOutput("value"))))

server <- function(input, output){
 output$value <- renderPrint({
 if(input$radio == 1){return('Great !')}
 else{return("Sorry !")}}})

shinyApp(ui = ui, server = server)
```

**Welcome**

**Your favorite color is red ?**

☒ TRUE ☐ FALSE

[1] "Great !"

### チェックボックスグループ



のをにりえるためにできるチェックボックスのグループをします。サーバーは、されたのベクトルとしてをけります。

```
library(shiny)

ui <- fluidPage(
 checkboxGroupInput("checkGroup1", label = h3("This is a Checkbox group"),
 choices = list("1" = 1, "2" = 2, "3" = 3),
 selected = 1),
 fluidRow(column(3, verbatimTextOutput("text_choice")))
)

server <- function(input, output){
 output$text_choice <- renderPrint({
 return(paste0("You have chosen the choice ",input$checkGroup1))
 })
}

shinyApp(ui = ui, server = server)
```

## This is a Checkbox group

- ☒ 1
- ☐ 2
- ☐ 3

```
[1] "You have chosen the choice 1"
```

をすることはです

- ラベルtitle
- された
- にしたしないはNULL
- インラインまたは
- 

HTMLをすることもです。

ボックス

リストから1つまたはのをするためにできるリストをします。

```
library(shiny)

ui <- fluidPage(
 selectInput("id_selectInput",
 label = HTML('What is your favorite color ?'),
 multiple = TRUE,
 choices = list("red" = "red", "green" = "green", "blue" = "blue", "yellow" =
"yellow"),
```

```

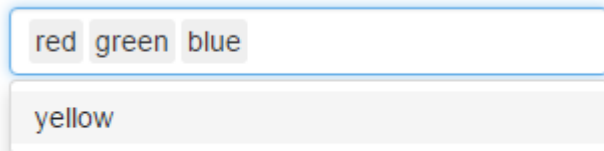
 selected = NULL),
br(), br(),
fluidRow(column(3, textOutput("text_choice"))))

server <- function(input, output){
 output$text_choice <- renderPrint({
 return(input$id_selectInput)})
}

shinyApp(ui = ui, server = server)

```

## What is your favorite color ?



```
[1] "red" "green" "blue"
```

をすることはです

- ラベルtitle
- された
- にしたしないはNULL
- multipleTRUEまたはFALSE
- 
- サイズ
- selectizeTRUEまたはFALSEselectize.jsをするかどうか、をする

HTMLをすることもです。

シャイニーアプリをする

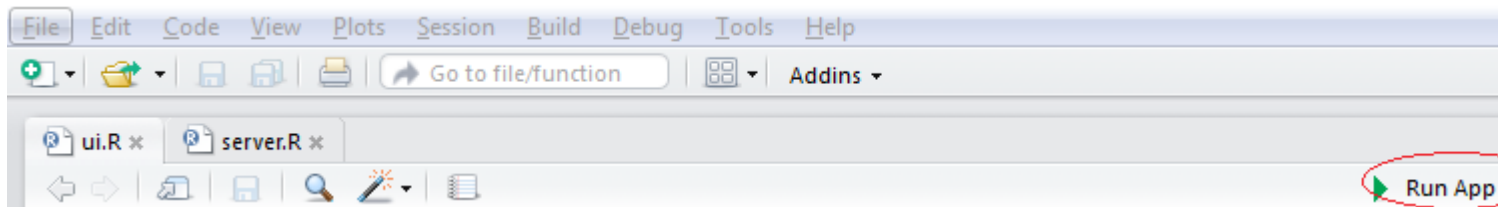
アプリケーションののじて、いくつかのでアプリケーションをできます。アプリが`ui.R`と`server.R` 2つのファイルにされている、またはすべてのアプリが1つのファイルにまれている。

## 1. 2つのファイル app

あなたの2つのファイル`ui.R`と`server.R`はじフォルダになければなりません。コンソールで`shinyApp()`をし、Shinyアプリケーションをむディレクトリのパスをすることでアプリケーションをすることが出来ます。

```
shinyApp("path_to_the_folder_containing_the_files")
```

`ui.R`または`server.R`ファイルをいたときにRstudioにされる**Run App**ボタンをして、Rstudioからアプリをすることも出来ます。



または、ディレクトリがShiny Appディレクトリのは、コンソールに`runApp()`をするだけです。

## 2. 1つのファイルアプリ

1つのRファイルをするは、`shinyApp()`をしてすることもできます。

- あなたのコードの

```
library(shiny)

ui <- fluidPage() #Create the ui
server <- function(input, output){} #create the server

shinyApp(ui = ui, server = server) #run the App
```

- コンソールで、シャイニーアプリケーションをむ.Rファイルへのパスを`appFile`ます。

```
shinyApp(appFile="path_to_my_R_file_containig_the_app")
```

### コントロールウィジェット

|                                 | ウィジェット                |
|---------------------------------|-----------------------|
| <code>actionButton</code>       | アクションボタン              |
| <code>checkboxGroupInput</code> | チェックボックスのグループ         |
| <code>checkboxInput</code>      | 1つのチェックボックス           |
| <code>dateInput</code>          | をするカレンダー              |
| <code>dateRangeInput</code>     | をするカレンダーのペア           |
| <code>fileInput</code>          | ファイルアップロードコントロールウィザード |
| ヘルプテキスト                         | フォームにできるヘルプテキスト       |
| <code>numericInput</code>       | をするフィールド              |
| ラジオボタン                          | ラジオボタンのセット            |
| <code>selectInput</code>        | のあるボックス               |

|              | ウィジェット       |
|--------------|--------------|
| sliderInput  | スライダーバー      |
| submitButton | ボタン          |
| textInput    | テキストをするフィールド |

```
library(shiny)

Create the UI
ui <- shinyUI(fluidPage(
 titlePanel("Basic widgets"),

 fluidRow(

 column(3,
 h3("Buttons"),
 actionButton("action", label = "Action"),
 br(),
 br(),
 submitButton("Submit")),

 column(3,
 h3("Single checkbox"),
 checkboxInput("checkbox", label = "Choice A", value = TRUE)),

 column(3,
 checkboxGroupInput("checkGroup",
 label = h3("Checkbox group"),
 choices = list("Choice 1" = 1,
 "Choice 2" = 2, "Choice 3" = 3),
 selected = 1)),

 column(3,
 dateInput("date",
 label = h3("Date input"),
 value = "2014-01-01"))
),

 fluidRow(

 column(3,
 dateRangeInput("dates", label = h3("Date range"))),

 column(3,
 fileInput("file", label = h3("File input"))),

 column(3,
 h3("Help text"),
 helpText("Note: help text isn't a true widget,",
 "but it provides an easy way to add text to",
 "accompany other widgets.")),

 column(3,
 numericInput("num",
 label = h3("Numeric input"),
 value = 1))
),
)
```

```

fluidRow(

 column(3,
 radioButtons("radio", label = h3("Radio buttons"),
 choices = list("Choice 1" = 1, "Choice 2" = 2,
 "Choice 3" = 3), selected = 1)),

 column(3,
 selectInput("select", label = h3("Select box"),
 choices = list("Choice 1" = 1, "Choice 2" = 2,
 "Choice 3" = 3), selected = 1)),

 column(3,
 sliderInput("slider1", label = h3("Sliders"),
 min = 0, max = 100, value = 50),
 sliderInput("slider2", "",
 min = 0, max = 100, value = c(25, 75))
),

 column(3,
 textInput("text", label = h3("Text input"),
 value = "Enter text...")
)
)

))

Create the server function
server <- shinyServer(function(input, output){})

Run the app
shinyApp(ui = ui, server = server)

```

## デバッグ

`debug()` と `debugonce()` はほとんどのShinyデバッグのコンテキストでうまくしません。しかし、なにされた `browser()` は、Shinyコードがどのようにしているかについてくのをえることができます。

`browser()` [デバッグ](#) `browser()`

## ショーケースモード

**ショーケースモード**では、アプリケーションをするコードとともにアプリケーションをし、している `server.R` のコードをハイライトします。

ショーケースモードをにするには、の2つのがあります。

- `runApp("MyApp", display.mode = "showcase")`、`display.mode = "showcase"`でShinyアプリケーションをし `runApp("MyApp", display.mode = "showcase")`。
- Shiny appフォルダに `DESCRIPTION` というファイルをし、このを `DisplayMode: Showcase` します。

## ログビジュアライザ

**Reactive Log Visualizer**は、アプリケーションのなやをするための、インタラクティブなブラウザベースのツールをします。 Reactive Log Visualizerをにするには、Rコンソールで `options(shiny.reactlog=TRUE)` をするか、`server.R`ファイルにそのコードをします。 Reactive Log Visualizerをするには、WindowsのはCtrl + F3を、アプリのはMacのCommand + F3をします。 Reactive Log Visualizerをナビゲートするには、のキーをします。

オンラインでシャイニーをむ <https://riptutorial.com/ja/r/topic/2044/シャイニー>

## 49: スタンドアロンのRスクリプトをしてをする

き

Rをくの々のデータファイルににしたいや、りしメソッドをのにしたいは、なRスクリプトはいや  
すいす。あなたやあなたのユーザがRをびして `source(.)` やびしをしてRのスクリプトをするので  
はなく、スクリプトをにプログラムであるかのようにびすことができます。

/チャンネルをすには、`file("stdin")` パイプでまたはのプログラムから、`stdout()` および `stderr()`  
エラーをします。 `stdin()` はしますが、 `stdin()` スクリプトをRにするときはできません。ユーザー  
ではなくスクリプトののをみむためです。

### Examples

スタンドアローンRプログラムのとそのび

## のスタンドアロンのRスクリプト

スタンドアロンのRスクリプトは、プログラムR Windowsでは `R.exe` ではなく、デフォルトでRイン  
ストールにまれる `Rscript Rscript.exe` というプログラムによってされます。

このをするには、スタンドアロンのRスクリプトは、のをしているシェバングラインとばれるな  
ライン、でまる `#!/usr/bin/env Rscript`。 Windowsでは、のがであり、でしくします。

のなスタンドアロンのRスクリプトは、ヒストグラムをファイル "hist.png"のに、としてけるから  
します。

```
#!/usr/bin/env Rscript

User message (\n = end the line)
cat("Input numbers, separated by space:\n")
Read user input as one string (n=1 -> Read only one line)
input <- readLines(file('stdin'), n=1)
Split the string at each space (\\s == any space)
input <- strsplit(input, "\\s")[[1]]
convert the obtained vector of strings to numbers
input <- as.numeric(input)

Open the output picture file
png("hist.png",width=400, height=300)
Draw the histogram
hist(input)
Close the output file
dev.off()
```

スタンドアローンRスクリプトのいくつかのなをすることができます。のでは、Shebangがされます

。それに `cat("....\n")` は、ユーザーにメッセージをするためにされます。 `file("stdin")` コンソールのユーザー」をデータのとしてするはいつでも `file("stdin")` してください。これは、いくつかのデータ `scan`、`read.table`、`read.csv`、... でファイルのわりにできます。ユーザーがからにされると、プロットがされます。そこには、ファイルにきまれることをしたプロットコマンドは、2つのコマンドでまなければならないことがわかります。これらはこの `png(.)` と `dev.off()` です。のは、のファイル `jpeg(.)` と `pdf(.)` でまります。2の `dev.off()` はにです。プロットをファイルにきみ、プロットプロセスをします。

## スタンドアロンRスクリプトの

### Linux / Mac

スタンドアロンスクリプトのファイルは、まずにするがあります。これは、ファイルをクリックしてきメニューの[プロパティ]をき、[アクセス]タブの[ファイル]チェックボックスをオンにすることでできます。あるいは、コマンド

```
chmod +x PATH/TO/SCRIPT/SCRIPTNAME.R
```

ターミナルでびすことができます。

### Windows

スタンドアロンのスクリプトでは、バッチファイルはのがあります。

```
"C:\Program Files\R-XXXXXXX\bin\Rscript.exe" "%~dp0\XXXXXXX.R" %*
```

バッチファイルはのテキストファイルですが、は `*.txt` のが `*.bat` です。 `notepad Word` ではなくなどのテキストエディタをしてし、ダイアログでファイルを `"FILENAME.bat"` れます。のバッチファイルをするには、そのファイルをクリックして[]をします。

のコードをどこにでもさせるがあります `xxx...` かれています

- あなたのRインストールがするしいフォルダをしてください
- スクリプトのしいをし、このバッチファイルとじディレクトリにきます。

コードのの `"C:\...\Rscript.exe"` は、Windowsに `Rscript.exe` プログラムのをします。2の `"%~dp0\XXX.R"` は、 `Rscript` したRスクリプトをバッチファイルとじフォルダに `%~dp0` ます `%~dp0` はバッチファイルフォルダをします。に、 `.*` はバッチファイルにしたコマンドラインをRスクリプトにします。

バッチファイルをダブルクリックすると、Rスクリプトがされます。バッチファイルにファイルをドラッグすると、するファイルがコマンドラインとしてRスクリプトにえられます。



## littlerをしてRスクリプトをする

**littler** されたさな **r** **cran** は、のにて、littlerの **r** コマンドでコマンドラインからRスクリプトをする2つのをしますLinuxまたはMacOSです。

## littlerのインストール

### Rから

```
install.packages("littler")
```

ターミナルには、**r**のパスがされています。

```
You could link to the 'r' binary installed in
'/home/*USER*/R/x86_64-pc-linux-gnu-library/3.4/littler/bin/r'
from '/usr/local/bin' in order to use 'r' for scripting.
```

システムのコマンドラインから**r**をびすには、シンボリックリンクがです。

```
ln -s /home/*USER*/R/x86_64-pc-linux-gnu-library/3.4/littler/bin/r /usr/local/bin/r
```

### apt-getをうDebian、Ubuntu

```
sudo apt-get install littler
```

## litrとの.rスクリプトをする

**r**をlittlerからlittlerすると、スクリプトをせずにスタンドアロンのRスクリプトをすることができます。スクリプト

```
User message (\n = end the line)
cat("Input numbers, separated by space:\n")
Read user input as one string (n=1 -> Read only one line)
input <- readLines(file('stdin'), n=1)
Split the string at each space (\s == any space)
input <- strsplit(input, "\\s")[[1]]
convert the obtained vector of strings to numbers
input <- as.numeric(input)

Open the output picture file
png("hist.png",width=400, height=300)
Draw the histogram
hist(input)
Close the output file
dev.off()
```

スクリプトのにはシバンがないことにしてください。えはhist.rようにすると、システムコマン

ドから`hist.r`ことができます

```
r hist.r
```

## なスクリプトで***littler***をう

また、`littler`をしてなRスクリプトをすることもです `shebang`をします。

```
#!/usr/bin/env r
```

スクリプトのにされます。するRスクリプトは`chmod +X /path/to/script.r`にするがあり、システムから`chmod +X /path/to/script.r`ことができます。

オンラインでスタンドアロンのRスクリプトをしてをするをむ

<https://riptutorial.com/ja/r/topic/9937/スタンドアロンのrスクリプトをしてをする>

## 50: スパーク API SparkR

SparkR パッケージをすると、[Spark クラスタ](#)にデータフレームを SparkR ことができます。これにより、大きなデータセットでの、フィルタリング、などのことができます。 [SparkR の SparkR のパッケージドキュメント](#)

### Examples

スパーク コンテキストの

### R で Spark コンテキストをする

Sparks データフレームのをするには、R プログラムをの Spark Cluster にするがあります。

```
library(SparkR)
sc <- sparkR.init() # connection to Spark context
sqlContext <- sparkRSQL.init(sc) # connection to SQL context
```

[ここに](#)です スパーク クラスタにあなたの IDE をする。

### スパーク クラスタをする

インストール [について](#)は、[Apache Spark のトピック](#) があります。には、Spark Cluster を Java [を](#) または [ではないクラウドアプリケーション](#) [Microsoft Azure \[トピックサイト\]](#)、[IBM など](#) をして ローカルにすることができます。

### データを キャッシュする

キャッシングは Spark でをできます。キャッシングはデータをメモリにし、のなケースです。 [こ](#) [こ](#)では、Spark で RDD をキャッシュするとどうなるかします。

なぜ

には、キャッシュは、のデータののはをします。したがって、キャッシュされた RDD をすると、の をすることなく、メモリからすでにされたデータにアクセスできます。

どうやって

ここでは、データここでは 3GB の csv を、メモリのストレージからアクセスするとすぐにそのデータにすばやくアクセスするのをします。

```
library(SparkR)
next line is needed for direct csv import:
Sys.setenv('SPARKR_SUBMIT_ARGS'='"--packages" "com.databricks:spark-csv_2.10:1.4.0" "sparkr-
```

```
shell"')
sc <- sparkR.init()
sqlContext <- sparkRSQL.init(sc)

loading 3 GB big csv file:
train <- read.df(sqlContext, "/train.csv", source = "com.databricks.spark.csv", inferSchema =
"true")
cache(train)
system.time(head(train))
output: time elapsed: 125 s. This action invokes the caching at this point.
system.time(head(train))
output: time elapsed: 0.2 s (!!)
```

## RDDのあるデータセット

### データフレームから

```
mtrdd <- createDataFrame(sqlContext, mtcars)
```

### csvから

csvの、Sparkコンテキストをするに、に[csvパッケージ](#)をするがあります。

```
Sys.setenv('SPARKR_SUBMIT_ARGS'='"--packages" "com.databricks:spark-csv_2.10:1.4.0" "sparkr-
shell"') # context for csv import read csv ->
sc <- sparkR.init()
sqlContext <- sparkRSQL.init(sc)
```

に、のデータのデータスキーマをすることによってcsvをロードすることができます。

```
train <- read.df(sqlContext, "/train.csv", header= "true", source =
"com.databricks.spark.csv", inferSchema = "true")
```

または、にデータスキーマをすることによって

```
customSchema <- structType(
 structField("margin", "integer"),
 structField("gross", "integer"),
 structField("name", "string"))

train <- read.df(sqlContext, "/train.csv", header= "true", source =
"com.databricks.spark.csv", schema = customSchema)
```

オンラインでスパークAPI SparkRをむ <https://riptutorial.com/ja/r/topic/5349/スパーク-api-sparkr->

## 51: データのクリーニング

き

をうには、Rでのデータのクリーニングがもです。あなたがっているデータは、でのやウェブからられたものであろうと、それをしたり、したり、フィルタリングしてにしたものにするがあります。このドキュメントでは、のトピックについてします。 - したデータによるの - データの - なの

### Examples

ベクターからけているデータをりく

まず、Vector1というベクトルをします。

```
set.seed(123)
Vector1 <- rnorm(20)
```

しているデータをする

```
set.seed(123)
Vector1[sample(1:length(Vector1), 5)] <- NA
```

に、is.naをしてVectorのサブセットをします

```
Vector1 <- Vector1[!is.na(Vector1)]
```

のベクトルはのVector1のNAsをします

なの

データフレームがあり、NAをむがあるすべてのをしたいがあります。これは、 *complete.cases* がなオプションであるためです。

々は、にNAsをしているので、としてデータセットのの6をする

```
x <- head(airquality)
```

これには、Solar.RにNAsをむ2があります。これらのをするには、のをいます

```
x_no_NA <- x[complete.cases(x),]
```

のデータフレームx\_no\_NAは、NAsのないなしかたない

オンラインでデータのクリーニングをむ <https://riptutorial.com/ja/r/topic/8165/データのクリーニング>

## 52: データフレーム

- `data.frame...`、`row.names = NULL`、`check.rows = FALSE`、`check.names = TRUE`、`stringsAsFactors = default.stringsAsFactors`
- `as.data.frame(x)`、`row.names = NULL`、オプション = `FALSE`、...ジェネリック
- `as.data.frame(x)`、...、`stringsAsFactors = default.stringsAsFactors` クラス "のS3メソッド
- クラス "のS3メソッドは、のようになります。 `asStatusAsFactors = default.stringsAsFactors`
- `is.data.frame(x)`

## Examples

## のdata.frameをする

`data.frame`は名のリストで、です。リストのはじさをち、には「」があります。にはのクラスがありますが、あるのクラスはののクラスとはなるがありますとはなり、すべてのがじクラスでなければなりません。

として、data.frameにははなく、はありません。

```
> structure(list(character()), class = "data.frame")
NULL
<0 rows> (or 0-length row.names)
```

しかしこれはいいことです。 `data.frame`がく のとをつのがよりです。 は32の`data.frame`です <sup>a</sup>  
はクラス、 <sup>b</sup>はクラス。

```
> structure(list(a = 1:3, b = letters[1:3]), class = "data.frame")
[1] a b
<0 rows> (or 0-length row.names)
```

`data.frame`がするためには、いくつかのをするがあります。ここでは、13のだけをします。

```
> structure(list(a = 1:3, b = letters[1:3]), class = "data.frame", row.names = 1:3)
 a b
1 1 a
2 2 b
3 3 c
```

32のdata.frameがあることは確かです。 `nrow()`、`ncol()`、および`dim()`を使ってこれをチェックすることができます

```
> x <- structure(list(a = numeric(3), b = character(3)), class = "data.frame", row.names = 1:3)
> nrow(x)
[1] 3
> ncol(x)
[1] 2
> dim(x)
[1] 3 2
```

Rは、`data.frame()`にできる2つの`structure()`にをします。はに`data.frame()`とばれます。したがであり、リストがすべてじさであり、にされたをいくつかするかどうかチェックされます。これは、`data.frame()`がいつもあなたがしているものとじになることをします

```
> str(data.frame("a a a" = numeric(3), "b-b-b" = character(3)))
'data.frame': 3 obs. of 2 variables:
 $ a.a.a: num 0 0 0
 $ b.b.b: Factor w/ 1 level "": 1 1 1
```

のは`as.data.frame()`とばれます。これは、`data.frame()`して`data.frame()`することによって、`data.frame`ではないオブジェクトをに`data.frame`にするためにできます。として、をえてみましょう。

```
> m <- matrix(letters[1:9], nrow = 3)
> m
 [,1] [,2] [,3]
[1,] "a" "d" "g"
[2,] "b" "e" "h"
[3,] "c" "f" "i"
```

そして

```
> as.data.frame(m)
 V1 V2 V3
1 a d g
2 b e h
3 c f i
> str(as.data.frame(m))
'data.frame': 3 obs. of 3 variables:
 $ V1: Factor w/ 3 levels "a","b","c": 1 2 3
 $ V2: Factor w/ 3 levels "d","e","f": 1 2 3
 $ V3: Factor w/ 3 levels "g","h","i": 1 2 3
```

データフレームからとをサブセットする

## とにアクセスするための `[]`、`[[`、`$`

このトピックでは、データフレームののとにアクセスするためのものについてします。これらは

- のかっこをつ`matrix`ように、`data[rows, columns]`
  - とをする
  -



## およびの

- `list`
  - データフレームをするためのの `data[columns]`
  - `data[[one_column]]` をってベクトルをる
- `1` の `data$column_name` をすると

たちはみみの `mtcars` データフレームを `mtcars` ています。

## のように `data[rows, columns]`

### インデックスをする

みみのデータフレーム `mtcars` をして、カンマをむ `[]` をしてとをできます。コンマののインデックスはです

```
get the first row
mtcars[1,]
get the first five rows
mtcars[1:5,]
```

### に、カンマがのに

```
get the first column
mtcars[, 1]
get the first, third and fifth columns:
mtcars[, c(1, 3, 5)]
```

のように、またはのいずれかをそのままにすると、すべてがされます。 `mtcars[1, ]` は、すべてのをむのをします。

## および

ここまでは、のとにどのようにアクセスするかとじです。 `data.frame` では、ほとんどの、インデックスにをすることをお `data.frame` ます。これは、 `numeric` わりにをつ `character` をとともにすることによってされます。

```
get the mpg column
mtcars[, "mpg"]
get the mpg, cyl, and disp columns
mtcars[, c("mpg", "cyl", "disp")]
```

あまりではありませんが、もできます。

```
mtcars["Mazda Rx4",]
```

## とをまとめて

とのほにできます

```
first four rows of the mpg column
mtcars[1:4, "mpg"]

2nd and 5th row of the mpg, cyl, and disp columns
mtcars[c(2, 5), c("mpg", "cyl", "disp")]
```

## ディメンションにする

これらのメソッドをする、のをすると、データフレームがされます。ただし、のをすると、デフォルトオプションのにデータフレームではなくベクトルがられます。

```
multiple columns returns a data frame
class(mtcars[, c("mpg", "cyl")])
[1] "data.frame"
single column returns a vector
class(mtcars[, "mpg"])
[1] "numeric"
```

これには2つのがあります。1つはデータフレームをリストとしてい、もう1つは`drop = FALSE`をすることです。これは、Rに「のディメンションをしない」ようにします。

```
class(mtcars[, "mpg", drop = FALSE])
[1] "data.frame"
```

はじめですることにしてください。デフォルトでは、のまたはがベクトルになりますが、`drop = FALSE`をすると、1または1のとしてでき`drop = FALSE`。

## リストのように

データフレームはに`lists`です。つまり、ベクトルのリストすべてがじさでなければなりません。リストは、の`[`サブリストのは`[`のの`[`をしてサブセットにすることができます。

のかっこで`data[columns]`

のかっこをし、カンマをしない、データフレームはのリストなので、がされます。

```
mtcars["mpg"]
mtcars[c("mpg", "cyl", "disp")]
my_columns <- c("mpg", "cyl", "hp")
mtcars[my_columns]
```

リストのようなのどのようなの

`data[columns]`と`data[, columns]`のいは、`data.frame`を`list`のカンマなしとしてうとき、されるオブジェクトは`data.frame`ます。コンマをして`data.frame matrix`ようにう、のをするとベクトルがされますが、のをすると`data.frame`がされます。

```
When selecting a single column
like a list will return a data frame
class(mtcars["mpg"])
[1] "data.frame"
like a matrix will return a vector
class(mtcars[, "mpg"])
[1] "numeric"
```

でまれた `data[[one_column]]`

`data.frame` を `list` としてうときに、のをベクトルとしてするには、`[[`。これは、に1つのにしての  
みします。

```
extract a single column by name as a vector
mtcars[["mpg"]]

extract a single column by name as a data frame (as above)
mtcars["mpg"]
```

## \$をしたへのアクセス

でまれたをせずに、のショートカット `$` をしてのをすることができます

```
get the column "mpg"
mtcars$mpg
```

`$` アクセスされるはにデータフレームではなくベクトルです。

にアクセスするための `$`

`$` はなショートカットです。にこのはをするRStudioなどでしているにです。しかし、`$` はもあ  
りますのをけるためにのをします。つまり、がにされているとしません。

```
my_column <- "mpg"
the below will not work
mtcars$my_column
but these will work
mtcars[, my_column] # vector
mtcars[my_column] # one-column data frame
mtcars[[my_column]] # vector
```

これらののため、`$` は、がの、Rセッションでもよくされます。たとえばのなるさまざまなデー  
タセットでされるなをくなど、プログラムでののは、`$` をけるがあります。

また、デフォルトでは、オブジェクトをくから `$`

```
give you the values of "mpg" column
as "mtcars" has only one column having name starting with "m"
mtcars$m
will give you "NULL"
```

```
as "mtcars" has more than one columns having name starting with "d"
mtcars$d
```

## なインデックスのインデックスとインデックス

インデックスにをするオプションがあるは、のをしてのインデックスをするか、ブールベクトルをして、するアイテムをにすこともできます。

### のインデックスはをします

```
mtcars[1,] # first row
mtcars[-1,] # everything but the first row
mtcars[-(1:10),] # everything except the first 10 rows
```

### ベクトルは、するのをします

<などのをしてベクトルをし、をたすのみをすることが出来ます。

```
logical vector indicating TRUE when a row has mpg less than 15
FALSE when a row has mpg >= 15
test <- mtcars$mpg < 15

extract these rows from the data frame
mtcars[test,]
```

### をするステップをすることもできます

```
extract all columns for rows where the value of cyl is 4.
mtcars[mtcars$cyl == 4,]
extract the cyl, mpg, and hp columns where the value of cyl is 4
mtcars[mtcars$cyl == 4, c("cyl", "mpg", "hp")]
```

## data.framesをするな

data.framesをするいくつかのなは、subset()、transform()、with()およびwithin()です。

### サブセット

subset()をすると、data.frameをよりにサブセットすることができますサブセットはのクラスでもします。

```
subset(mtcars, subset = cyl == 6, select = c("mpg", "hp"))
 mpg hp
Mazda RX4 21.0 110
Mazda RX4 Wag 21.0 110
Hornet 4 Drive 21.4 110
```

|              |      |     |
|--------------|------|-----|
| Valiant      | 18.1 | 105 |
| Merc 280     | 19.2 | 123 |
| Merc 280C    | 17.8 | 123 |
| Ferrari Dino | 19.7 | 175 |

のコードでは、 `cyl == 6` のと `mpg` と `hp` のみをめています。 `[]` でのコードでじをすることができます

```
mtcars[mtcars$cyl == 6, c("mpg", "hp")]
```

## する

`transform()` は、 `data.frame` のをするな `data.frame` 。 たとえば、のコードは、の `mpg2` のと `mpg^2` の `mtcars data.frame`

```
mtcars <- transform(mtcars, mpg2 = mpg^2)
```

## とに

`with()` と `within()` をする `with()` 、 `data.frame` のをすることができ、より `data.frame` されたがになり、  
\$ または `[]` をできます。

たとえば、 `airquality data.frame` のを、 またはするは、 のようにします。

```
aq <- within(airquality, {
 logOzone <- log(Ozone) # creates new column
 Month <- factor(month.abb[Month]) # changes Month Column
 cTemp <- round((Temp - 32) * 5/9, 1) # creates new column
 S.cT <- Solar.R / cTemp # creates new column
 rm(Day, Temp) # removes columns
})
```

## き

データフレームは、でもくするデータです。データフレームは、なるクラスのじさのベクトルをするなのリストです。 `data.frame` をしてデータフレームをします。のは、とのベクトルをデータフレームにみわせることによってこれをしています。これは `:` をして、1から3までのすべてののをむベクトルをします。

```
df1 <- data.frame(x = 1:3, y = c("a", "b", "c"))
df1
x y
1 1 a
2 2 b
3 3 c
class(df1)
[1] "data.frame"
```

データフレームオブジェクトはではされないため、のクラスは `data.frame` ではありません。

```
df2 <- data.frame(x = c("1", "2", "3"), y = c("a", "b", "c"))
df2
x y
1 1 a
2 2 b
3 3 c
```

さらなるがなければ、`df1`および`df2`の「x」は`df2`できない。`str`は、クラスよりもなオブジェクトをするためにできます。

```
str(df1)
'data.frame': 3 obs. of 2 variables:
$ x: int 1 2 3
$ y: Factor w/ 3 levels "a","b","c": 1 2 3
str(df2)
'data.frame': 3 obs. of 2 variables:
$ x: Factor w/ 3 levels "1","2","3": 1 2 3
$ y: Factor w/ 3 levels "a","b","c": 1 2 3
```

ここでは、`data.frame`あり、2つの「x」と「y」の3つの`df1`があることが`df1`ます。に、「x」はデータこのクラスではではありませんが、のためにはのようにしますと「y」は3つのレベルのデータクラスについてしていません。デフォルトでは、データフレームはをにすることにするのがです。デフォルトのは`stringsAsFactors`パラメータでできます。

```
df3 <- data.frame(x = 1:3, y = c("a", "b", "c"), stringsAsFactors = FALSE)
str(df3)
'data.frame': 3 obs. of 2 variables:
$ x: int 1 2 3
$ y: chr "a" "b" "c"
```

、「y」はです。したように、データフレームの「」はじさでなければならない。さのなるベクトルから`data.frame`をしようとする、エラーがします。のエラーをするには、`data.frame(x = 1:3, y = 1:4)`をしてください。

データフレームのテストケースとして、ではRによっていくつかのデータがされます。それらの1つはで、のようにロードされます

```
mydataframe <- iris
str(mydataframe)
```

## do.callをしてリストにされたデータをのデータフレームにする

データをリストにしてい、このリストをデータフレームにしたい、`do.call`はこれをするなです。ただし、しないのリサイクルをぐためには、すべてのリストのさがじであることがです。

```
dataList <- list(1:3, 4:6, 7:9)
dataList
[[1]]
[1] 1 2 3
#
```

```
[[2]]
[1] 4 5 6
#
[[3]]
[1] 7 8 9

dataframe <- data.frame(do.call(rbind, dataList))
dataframe
X1 X2 X3
1 1 2 3
2 4 5 6
3 7 8 9
```

リストがデータフレームでされているにもします。

```
dataframeList <- list(data.frame(a = 1:2, b = 1:2, c = 1:2),
 data.frame(a = 3:4, b = 3:4, c = 3:4))

dataframeList
[[1]]
a b c
1 1 1 1
2 2 2 2

[[2]]
a b c
1 3 3 3
2 4 4 4

dataframe <- do.call(rbind, dataframeList)
dataframe
a b c
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
```

## data.frameのすべてのをクラスにする

なは、データフレームをRDBMSにするや、データでレベルがなるのあるをむデータをするなど、をにするためにdata.frameのすべてのをクラスにすることです。。

データフレームをするほとんどすべてのメソッドには、`FALSE`にできるoptions `stringsAsFactors`オプションがあり`FALSE`。

データがすでにされている、はのようになにできます。

```
bob <- data.frame(jobs = c("scientist", "analyst"),
 pay = c(160000, 100000), age = c(30, 25))
str(bob)
```

```
'data.frame': 2 obs. of 3 variables:
 $ jobs: Factor w/ 2 levels "analyst","scientist": 2 1
 $ pay : num 160000 100000
 $ age : num 30 25
```

```
Convert *all columns* to character
bob[] <- lapply(bob, as.character)
str(bob)
```

```
'data.frame': 2 obs. of 3 variables:
 $ jobs: chr "scientist" "analyst"
 $ pay : chr "160000" "1e+05"
 $ age : chr "30" "25"
```

```
Convert only factor columns to character
bob[] <- lapply(bob, function(x) {
 if is.factor(x) x <- as.character(x)
 return(x)
})
```

## によるのサブセット

ビルトインは、をたす `columns` をむ `rows` をサブセットできます。

```
df <- data.frame(item = c(1:10),
 price_Elasticity = c(-0.57667, 0.03205, -0.04904, 0.10342, 0.04029,
 0.0742, 0.1669, 0.0313, 0.22204, 0.06158),
 total_Margin = c(-145062, 98671, 20576, -56382, 207623, 43463, 1235,
 34521, 146553, -74516))
```

`price_Elasticity > 0` rows をするには

```
df[df$price_Elasticity > 0,]
```

|    | item | price_Elasticity | total_Margin |
|----|------|------------------|--------------|
| 2  | 2    | 0.03205          | 98671        |
| 4  | 4    | 0.10342          | -56382       |
| 5  | 5    | 0.04029          | 207623       |
| 6  | 6    | 0.07420          | 43463        |
| 7  | 7    | 0.16690          | 1235         |
| 8  | 8    | 0.03130          | 34521        |
| 9  | 9    | 0.22204          | 146553       |
| 10 | 10   | 0.06158          | -74516       |

サブセットは `price_Elasticity > 0`、`total_Margin > 0`

```
df[df$price_Elasticity > 0 & df$total_Margin > 0,]
```

|   | item | price_Elasticity | total_Margin |
|---|------|------------------|--------------|
| 2 | 2    | 0.03205          | 98671        |
| 5 | 5    | 0.04029          | 207623       |
| 6 | 6    | 0.07420          | 43463        |
| 7 | 7    | 0.16690          | 1235         |
| 8 | 8    | 0.03130          | 34521        |
| 9 | 9    | 0.22204          | 146553       |

オンラインでデータフレームをむ <https://riptutorial.com/ja/r/topic/438/データフレーム>



## 53: データフレームの

き

はRのもの1つです。Rでこれをうにはいくつかのがありますが、ここです。

### Examples

Rとの

このために、`aggregate`をします。これはのようになります。

```
aggregate(formula,function,data)
```

のコードは、をするさまざまなをしています。

コード

```
df = data.frame(group=c("Group 1","Group 1","Group 2","Group 2","Group 2"), subgroup =
c("A","A","A","A","B"),value = c(2,2.5,1,2,1.5))

sum, grouping by one column
aggregate(value~group, FUN=sum, data=df)

mean, grouping by one column
aggregate(value~group, FUN=mean, data=df)

sum, grouping by multiple columns
aggregate(value~group+subgroup,FUN=sum,data=df)

custom function, grouping by one column
in this example we want the sum of all values larger than 2 per group.
aggregate(value~group, FUN=function(x) sum(x[x>2]), data=df)
```

```
> df = data.frame(group=c("Group 1","Group 1","Group 2","Group 2","Group 2"), subgroup =
c("A","A","A","A","B"),value = c(2,2.5,1,2,1.5))
> print(df)
 group subgroup value
1 Group 1 A 2.0
2 Group 1 A 2.5
3 Group 2 A 1.0
4 Group 2 A 2.0
5 Group 2 B 1.5
>
> # sum, grouping by one column
> aggregate(value~group, FUN=sum, data=df)
 group value
1 Group 1 4.5
2 Group 2 4.5
>
> # mean, grouping by one column
```

```

> aggregate(value~group, FUN=mean, data=df)
 group value
1 Group 1 2.25
2 Group 2 1.50
>
> # sum, grouping by multiple columns
> aggregate(value~group+subgroup,FUN=sum,data=df)
 group subgroup value
1 Group 1 A 4.5
2 Group 2 A 3.0
3 Group 2 B 1.5
>
> # custom function, grouping by one column
> # in this example we want the sum of all values larger than 2 per group.
> aggregate(value~group, FUN=function(x) sum(x[x>2]), data=df)
 group value
1 Group 1 2.5
2 Group 2 0.0

```

## dplyrとの

dplyrとののはですこのためにgroup\_byとsummarizeをすることができます。いくつかのをにします。

### コード

```

Aggregating with dplyr
library(dplyr)

df = data.frame(group=c("Group 1","Group 1","Group 2","Group 2","Group 2"), subgroup =
c("A","A","A","A","B"),value = c(2,2.5,1,2,1.5))
print(df)

sum, grouping by one column
df %>% group_by(group) %>% summarize(value = sum(value)) %>% as.data.frame()

mean, grouping by one column
df %>% group_by(group) %>% summarize(value = mean(value)) %>% as.data.frame()

sum, grouping by multiple columns
df %>% group_by(group,subgroup) %>% summarize(value = sum(value)) %>% as.data.frame()

custom function, grouping by one column
in this example we want the sum of all values larger than 2 per group.
df %>% group_by(group) %>% summarize(value = sum(value[value>2])) %>% as.data.frame()

```

```

> library(dplyr)
>
> df = data.frame(group=c("Group 1","Group 1","Group 2","Group 2","Group 2"), subgroup =
c("A","A","A","A","B"),value = c(2,2.5,1,2,1.5))
> print(df)
 group subgroup value
1 Group 1 A 2.0
2 Group 1 A 2.5
3 Group 2 A 1.0
4 Group 2 A 2.0
5 Group 2 B 1.5

```

```

>
> # sum, grouping by one column
> df %>% group_by(group) %>% summarize(value = sum(value)) %>% as.data.frame()
 group value
1 Group 1 4.5
2 Group 2 4.5
>
> # mean, grouping by one column
> df %>% group_by(group) %>% summarize(value = mean(value)) %>% as.data.frame()
 group value
1 Group 1 2.25
2 Group 2 1.50
>
> # sum, grouping by multiple columns
> df %>% group_by(group, subgroup) %>% summarize(value = sum(value)) %>% as.data.frame()
 group subgroup value
1 Group 1 A 4.5
2 Group 2 A 3.0
3 Group 2 B 1.5
>
> # custom function, grouping by one column
> # in this example we want the sum of all values larger than 2 per group.
> df %>% group_by(group) %>% summarize(value = sum(value[value>2])) %>% as.data.frame()
 group value
1 Group 1 2.5
2 Group 2 0.0

```

## data.tableでする

data.tableパッケージをしたグループは、`dt[i, j, by]`というをしてわれます。「`dt`をり、`i`をしてをサブセットし、に`j`でグループします。のやグループをリストにれるがあります。`list()`は`.`、どちらもじでできます。のでは、`.`をし`.`。

### コード

```

Aggregating with data.table
library(data.table)

dt = data.table(group=c("Group 1","Group 1","Group 2","Group 2","Group 2"), subgroup =
c("A","A","A","A","B"),value = c(2,2.5,1,2,1.5))
print(dt)

sum, grouping by one column
dt[,.(value=sum(value)),group]

mean, grouping by one column
dt[,.(value=mean(value)),group]

sum, grouping by multiple columns
dt[,.(value=sum(value)),.(group,subgroup)]

custom function, grouping by one column
in this example we want the sum of all values larger than 2 per group.
dt[,.(value=sum(value[value>2])),group]

```

```
> # Aggregating with data.table
```

```

> library(data.table)
>
> dt = data.table(group=c("Group 1","Group 1","Group 2","Group 2","Group 2"), subgroup =
c("A","A","A","A","B"),value = c(2,2.5,1,2,1.5))
> print(dt)
 group subgroup value
1: Group 1 A 2.0
2: Group 1 A 2.5
3: Group 2 A 1.0
4: Group 2 A 2.0
5: Group 2 B 1.5
>
> # sum, grouping by one column
> dt[,.(value=sum(value)),group]
 group value
1: Group 1 4.5
2: Group 2 4.5
>
> # mean, grouping by one column
> dt[,.(value=mean(value)),group]
 group value
1: Group 1 2.25
2: Group 2 1.50
>
> # sum, grouping by multiple columns
> dt[,.(value=sum(value)),.(group,subgroup)]
 group subgroup value
1: Group 1 A 4.5
2: Group 2 A 3.0
3: Group 2 B 1.5
>
> # custom function, grouping by one column
> # in this example we want the sum of all values larger than 2 per group.
> dt[,.(value=sum(value[value>2])),group]
 group value
1: Group 1 2.5
2: Group 2 0.0

```

オンラインでデータフレームのをむ <https://riptutorial.com/ja/r/topic/10792/データフレームの>

## 54: データベーステーブルのI/O

### パッケージ

- RMySQL
- RODBC

## Examples

### MySQL データベースからのデータのみのみ

パッケージRMyをすると、MariaDBデータベースだけでなくMySQLにもクエリをし、そのデータをRデータフレームにできます。

```
library(RMySQL)

mydb <- dbConnect(MySQL(), user='user', password='password', dbname='dbname', host='127.0.0.1')

queryString <- "SELECT * FROM table1 t1 JOIN table2 t2 on t1.id=t2.id"
query <- dbSendQuery(mydb, queryString)
data <- fetch(query, n=-1) # n=-1 to return all results
```

### をう

たとえば、の100,000だけをするなど、をすることもできます。これをうには、のにするSQLクエリをするだけです。のパッケージは、これらのオプションをします。

```
queryString <- "SELECT * FROM table1 limit 100000"
```

### MongoDB データベースからのデータのみのみ

MongoDBデータベースからRデータフレームにデータをロードするには、ライブラリMongoLiteをします。

```
Use MongoLite library:
#install.packages("mongolite")
library(jsonlite)
library(mongolite)

Connect to the database and the desired collection as root:
db <- mongo(collection = "Tweets", db = "TweetCollector", url =
"mongodb://USERNAME:PASSWORD@HOSTNAME")
```

```
Read the desired documents i.e. Tweets inside one dataframe:
documents <- db$find(limit = 100000, skip = 0, fields = '{ "_id" : false, "Text" : true }')
```

コードはPASSWORD USERNAMEというサーバーのHOSTNAMEし、データベースTweetCollectorをき、コレクションのTweetsをみます。クエリは、フィールド、つまりTextをみもうとします。

は、されたデータセットとしてののをデータフレームです。このの、データフレームにはTextがまれています documents\$Text。

オンラインでデータベーステーブルのI/Oをむ <https://riptutorial.com/ja/r/topic/5537/データベーステーブルのi-o>

---

## 55: データ

き

Rセッションにデータをする。Rのれたの1つは、データのさです。Rパッケージをしたデータにはいくつかのがあります。

### Examples

みみデータセット

Rにはなビルトインデータセットがあります。、それらは、かつになをするためのでされます。みみのデータセットをリストしたらしいWebページがあります

<https://vincentarelbundock.github.io/Rdatasets/datasets.html>

---

---

スイスのと1888データ。カトリックののつづいてのいをべましょう。

```
library(tidyverse)

swiss %>%
 ggplot(aes(x = Agriculture, y = Fertility,
 color = Catholic > 50)) +
 geom_point() +
 stat_ellipse()
```

Fertility

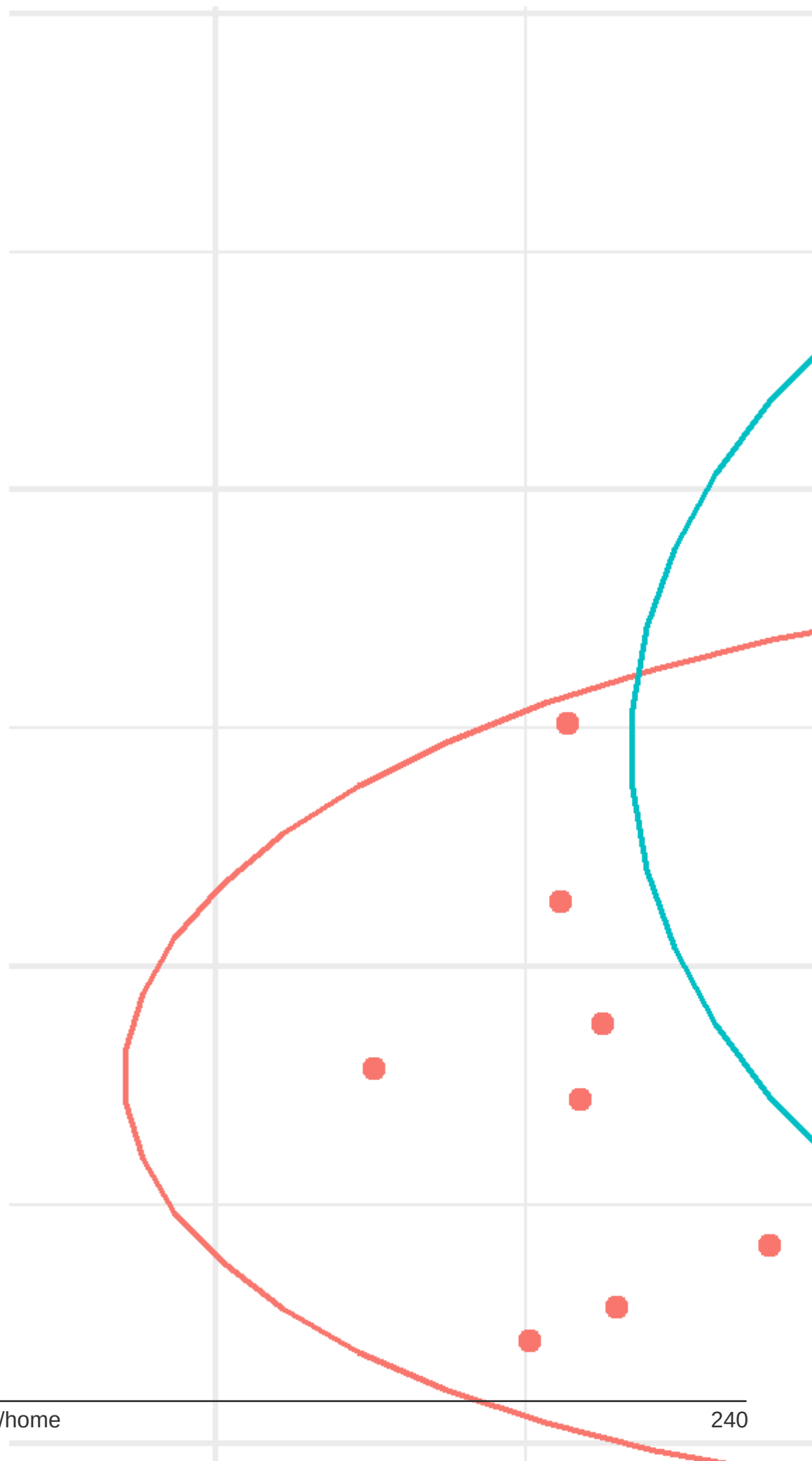
110

90

70

50

240





パッケージがロードされると `library(pkg)` 、されたデータセットはRオブジェクトとしてになります。またはそれらを `data()` でびすがあります。

---

## Gapminder

のにするなデータセット。

```
library(tidyverse)
library(gapminder)

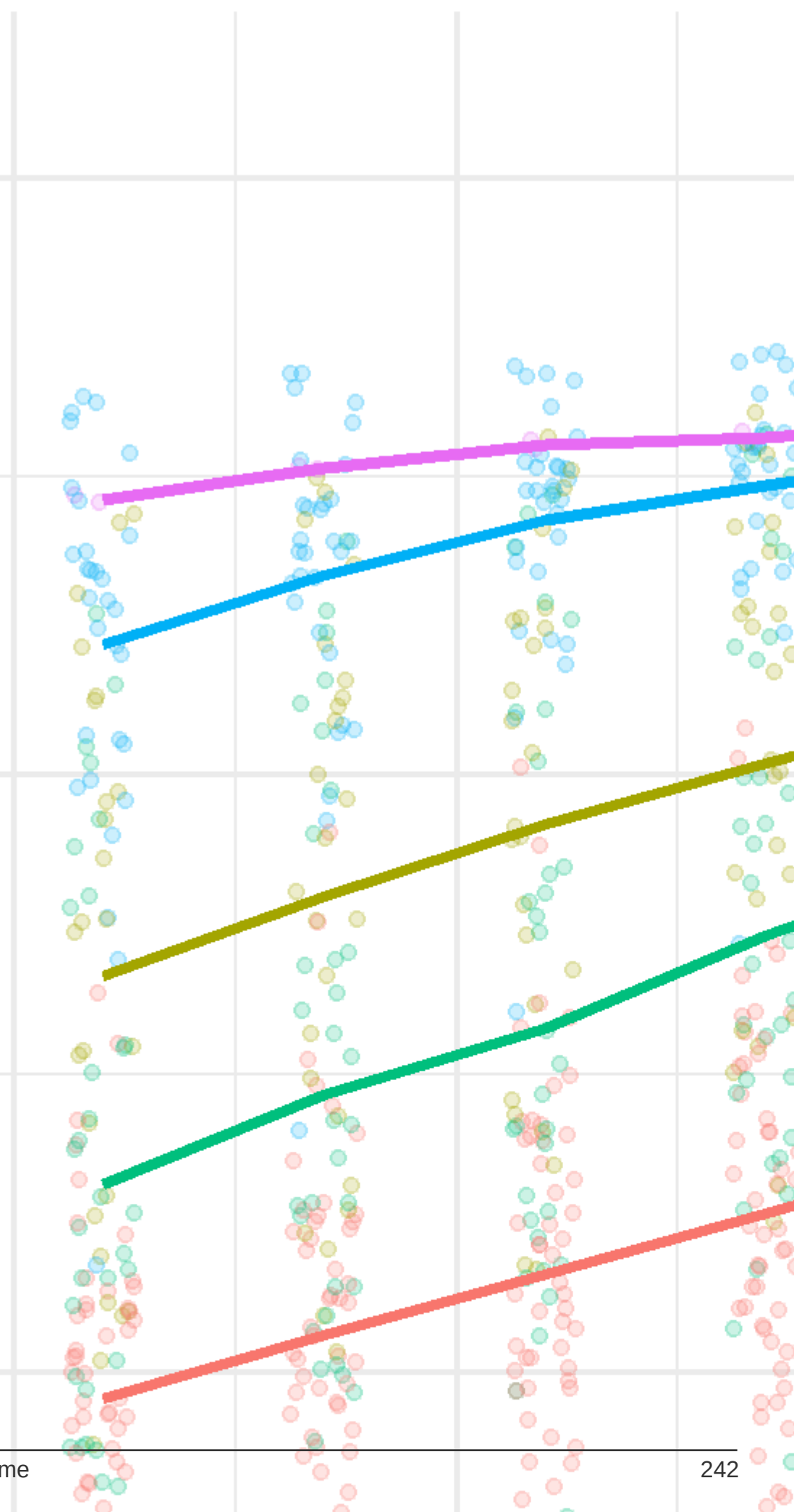
gapminder %>%
 ggplot(aes(x = year, y = lifeExp,
 color = continent)) +
 geom_jitter(size = 1, alpha = .2, width = .75) +
 stat_summary(geom = "path", fun.y = mean, size = 1) +
 theme_minimal()
```

lifeExp

80

60

40



のみみ/にをできます。

---

## ユーロスタット

にもかかわらず、`eurostat` パッケージには、がある `search_eurostat()` それはすべてのデータセットがつけられません。これはユーロスタットのウェブサイトで、データセットのコードをでするがです [データベース](#) または [データベース](#)。ダウンロードがしないは、[Bulk Download Facility](#) でデータをできます。

```
library(tidyverse)
library(lubridate)
library(forcats)
library(eurostat)
library(geofacet)
library(viridis)
library(ggthemes)
library(extrafont)

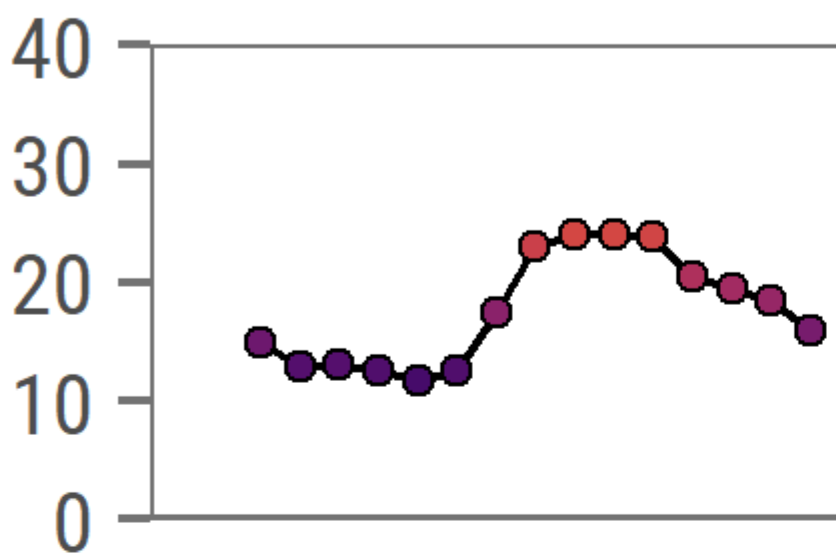
download NEET data for countries
neet <- get_eurostat("edat_lfse_22")

neet %>%
 filter(geo %>% paste %>% nchar == 2,
 sex == "T", age == "Y18-24") %>%
 group_by(geo) %>%
 mutate(avg = values %>% mean()) %>%
 ungroup() %>%
 ggplot(aes(x = time %>% year(),
 y = values)) +
 geom_path(aes(group = 1)) +
 geom_point(aes(fill = values), pch = 21) +
 scale_x_continuous(breaks = seq(2000, 2015, 5),
 labels = c("2000", "'05", "'10", "'15")) +
 scale_y_continuous(expand = c(0, 0), limits = c(0, 40)) +
 scale_fill_viridis("NEET, %", option = "B") +
 facet_geo(~ geo, grid = "eu_grid1") +
 labs(x = "Year",
 y = "NEET, %",
 title = "Young people neither in employment nor in education and training in
Europe",
 subtitle = "Data: Eurostat Regional Database, 2000-2016",
 caption = "ikashnitsky.github.io") +
 theme_few(base_family = "Roboto Condensed", base_size = 15) +
 theme(axis.text = element_text(size = 10),
 panel.spacing.x = unit(1, "lines"),
 legend.position = c(0, 0),
 legend.justification = c(0, 0))
```

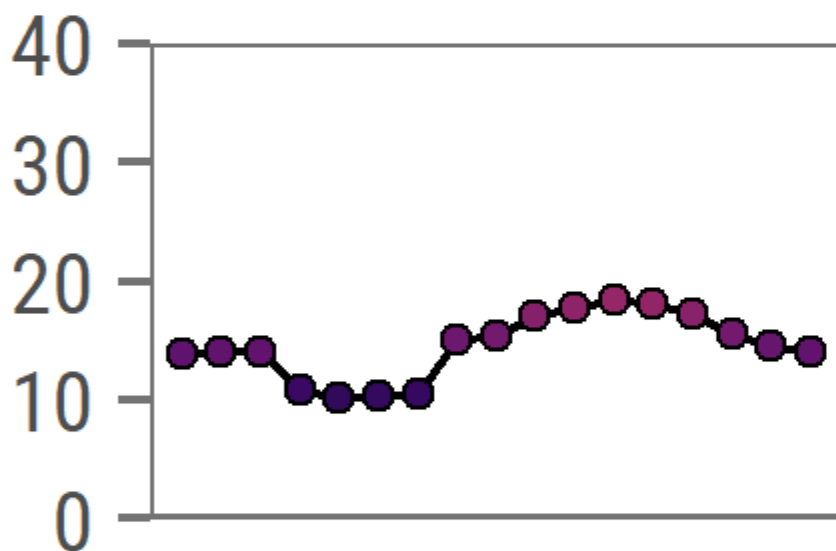
# Young people neither

Data: Eurostat Regional D

IE



UK



マックスブランクのプロジェクトです。

```
load required packages
library(tidyverse)
library(extrafont)
library(HMDHFDplus)

country <- getHMDcountries()

exposures <- list()
for (i in 1:length(country)) {
 cnt <- country[i]
 exposures[[cnt]] <- readHMDweb(cnt, "Exposures_1x1", user_hmd, pass_hmd)
 # let's print the progress
 paste(i,'out of',length(country))
} # this will take quite a lot of time
```

`user_hmd`と`pass_hmd`はHuman Mortality DatabaseのWebサイトのログインです。データにアクセスするには、<http://www.mortality.org/>でアカウントをし、のを `readHMDweb()` にするがあります。

```
sr_age <- list()

for (i in 1:length(exposures)) {
 di <- exposures[[i]]
 sr_agei <- di %>% select(Year, Age, Female, Male) %>%
 filter(Year %in% 2012) %>%
 select(-Year) %>%
 transmute(country = names(exposures)[i],
 age = Age, sr_age = Male / Female * 100)
 sr_age[[i]] <- sr_agei
}
sr_age <- bind_rows(sr_age)

remove optional populations
sr_age <- sr_age %>% filter(!country %in% c("FRACNP", "DEUTE", "DEUTW", "GBRCENW", "GBR_NP"))

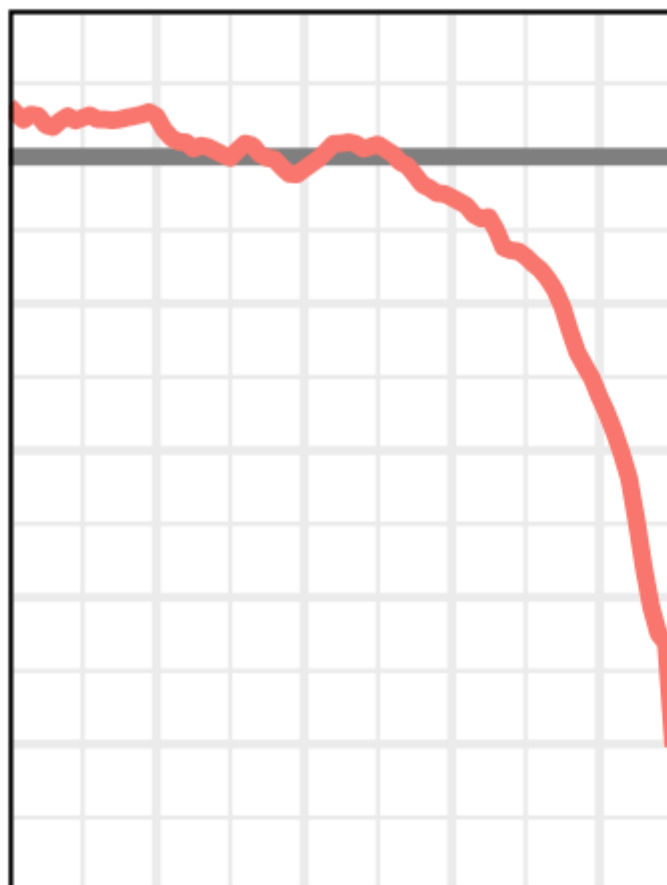
summarize all ages older than 90 (too jerky)
sr_age_90 <- sr_age %>% filter(age %in% 90:110) %>%
 group_by(country) %>% summarise(sr_age = mean(sr_age, na.rm = T)) %>%
 ungroup() %>% transmute(country, age=90, sr_age)

df_plot <- bind_rows(sr_age %>% filter(!age %in% 90:110), sr_age_90)

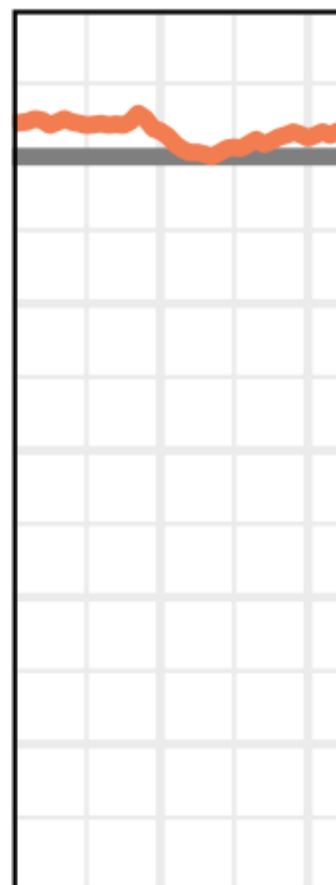
finally - plot
df_plot %>%
 ggplot(aes(age, sr_age, color = country, group = country))+
 geom_hline(yintercept = 100, color = 'grey50', size = 1)+
 geom_line(size = 1)+
 scale_y_continuous(limits = c(0, 120), expand = c(0, 0), breaks = seq(0, 120, 20))+
 scale_x_continuous(limits = c(0, 90), expand = c(0, 0), breaks = seq(0, 80, 20))+
 xlab('Age')+
 ylab('Sex ratio, males per 100 females')+
 facet_wrap(~country, ncol=6)+
 theme_minimal(base_family = "Roboto Condensed", base_size = 15)+
 theme(legend.position='none',
 panel.border = element_rect(size = .5, fill = NA))
```

AUT

120  
100  
80  
60  
40  
20  
0

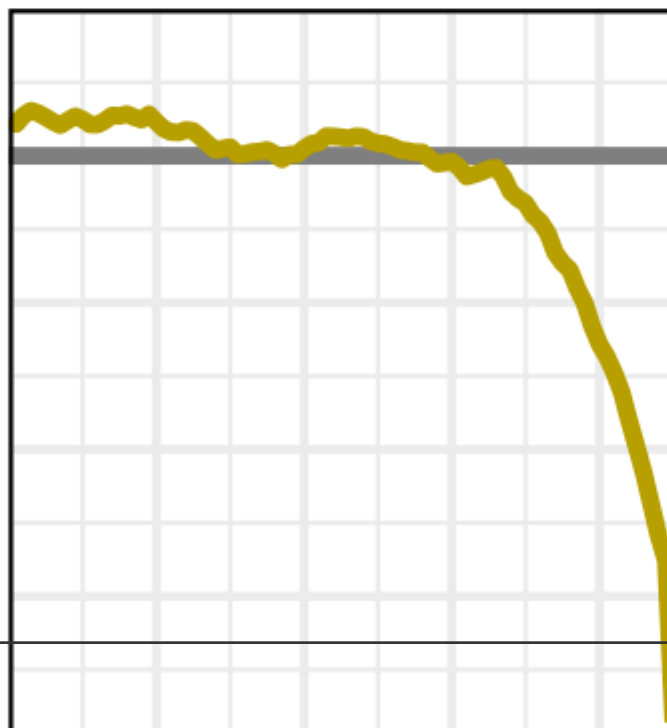


BI

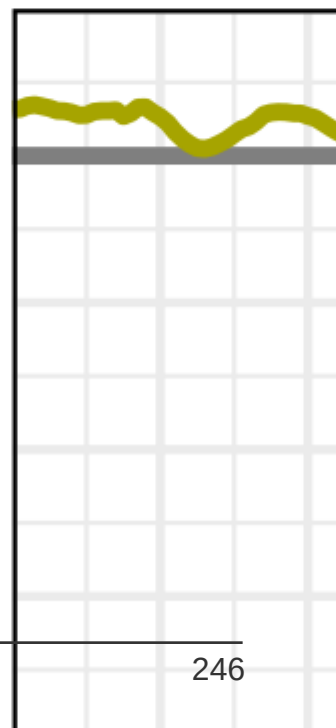


DNK

120  
100  
80  
60  
40  
20  
0



ES



## 56: データ

き

Data.tableは、Rからのデータフレームのをし、にパフォーマンスとをさせるパッケージです。については、パッケージのドキュメント [data.table](#)をしてください。

- DT[i, j, by]  
DT [where | select | update | do、 by]
- DT[...][...]
- ##### Shortcuts, special functions and special symbols inside DT[...]
- 。  
listのわりにいくつかのに
- J  
リストのがiのできえられました。
- =  
in j、のまたはにされる
- N  
iで、の  
in j、グループの
- 。  
in j、テーブルののベクトルiでフィルタリングされた
- 。 SD  
in j、データのサブセット  
.SDcolsでされた
- .GRP  
in j、データのサブセットののインデックス
- 。 によって  
in j、データのサブセットのによるリスト
- V1、 V2、 ...  
jでされたののデフォルト
- ##### Joins inside DT[...]
- DT1 [DT2、 on、 j]  
2つのテーブルをする
- 。 \*  
のDT2ののな
- by = EACHI  
オプションはでのみです
- DT1 [DT2、 on、 j]  
2つのテーブルのアンチ・ジョイン
- DT1 [DT2、 on、 roll、 j]  
2つのテーブルをし、のをon =
- ##### Reshaping, stacking and splitting
-

DT、id.vars、measure.vars

いにする

ののは、measure.vars = patterns...をします。

- dcastDT、  
ワイドフォーマットにする
- rbindDT1、DT2、 ...  
stackされたdata.tables
- rbindlistDT\_list、idcol  
データテーブルのリストをスタックする
- DT、by  
data.tableをリストにする
- ##### Some other functions specialized for data.tables
- ホバリング
- マージ  
2つのテーブルをするの
- セット  
をまたはするの
- fintersect、fsetdiff、funion、fsetequal、ユニーク、、anyDuplicated  
をとするset-theory
- ユニークN  
なるの
- rowidvDT、cols  
colsでされたグループのID1N
- rleidvDT、cols  
colsのによってされるグループのグループID1.GRP
- シフトDT、n、タイプ=c「ラグ」、「リード」  
すべてののにシフトをする
- setorder、setcolorder、setnames、setkey、setindex、setattr  
との

---

## インストールとサポート

data.tableパッケージをインストールするには

```
install from CRAN
install.packages("data.table")

or install development version
install.packages("data.table", type = "source", repos =
"http://Rdatatable.github.io/data.table")

and to revert from devel to CRAN, the current version must first be removed
remove.packages("data.table")
install.packages("data.table")
```



パッケージの[サイト](#)には、ヘルプのをするwikiページ、およびウェブのプレゼンテーションとのリストがあります。 StackOverflowやのをするに、[サポートページ](#)をおみください。

---

## パッケージのロード

のののくは、`data.table`にします。それらをするには、`library(data.table)`のようなをにするか、に`fread`わりに`data.table::fread`のようなフルパスをするがあります。々のヘルプについては、は`help("fread")`または`?fread`です。パッケージがロードされていないは、`?data.table::fread`のようなフルネームをしてください。

## Examples

### `data.table`の

`data.table`は、Rからの`data.frame`クラスのバージョンです。したがって、`class()`はvector `"data.table" "data.frame"`あり、`data.frame`でするも`data.table`でします。 `data.table`を、ロード、またはするはたくさんあります。

---

## ビルド

`data.table`パッケージをインストールしてアクティブすることをれないでください

```
library(data.table)
```

じのコンストラクタがあります

```
DT <- data.table(
 x = letters[1:5],
 y = 1:5,
 z = (1:5) > 3
)
x y z
1: a 1 FALSE
2: b 2 FALSE
3: c 3 FALSE
4: d 4 TRUE
5: e 5 TRUE
```

`data.frame`とはなり、`data.table`はをファクタにしません。

```
sapply(DT, class)
x y z
"character" "integer" "logical"
```



テキストファイルからみむことができます

```
dt <- fread("my_file.csv")
```

`read.csv`とはなり、`fread`はをではなくとしてみみます。

## data.frameをする

をけるため、`data.table`は`data.frame`または`list`をして、コピーをせずに、またはメモリのをせずに`data.table`をそのでするをします

```
example data.frame
DF <- data.frame(x = letters[1:5], y = 1:5, z = (1:5) > 3)
modification
setDT(DF)
```

オブジェクト`DF`はインプレースでされているため、を<-りてないことにしてください。  
`data.frame`のクラスはされます

```
sapply(DF, class)
x y z
"factor" "integer" "logical"
```

## オブジェクトをdata.tableにする

あなたがっているは`list`、`data.frame`、または`data.table`、あなたがするがあります`setDT`にするを`data.table`それはコピーをこれはるのではなく、することにより、をいますので、`as.data.table`ありません。なデータセットをうは、これがです。

あなたはこのようなマトリックスのようなRオブジェクトをっているは、するがあります  
`as.data.table`にそれをする`data.table`。

```
mat <- matrix(0, ncol = 10, nrow = 10)

DT <- as.data.table(mat)
or
DT <- data.table(mat)
```

のと

`DT[where, select|update|do, by]`は、`data.table`のをするためにされます。

- "where"が<sub>i</sub>です。
- "select | update | do"は<sub>j</sub>です

これらの2つの、<sub>i</sub>ではなくによってされます。

このデータは

```
mtcars = data.table(mtcars, keep.rownames = TRUE)
```

の

しいをりてには、<sub>j</sub>で:=をします。

```
mtcars[, mpg_sq := mpg^2]
```

NULLしてをする

```
mtcars[, mpg_sq := NULL]
```

:=のをしてのをする

```
mtcars[, `:=`(mpg_sq = mpg^2, wt_sqrt = sqrt(wt))]
or
mtcars[, c("mpg_sq", "wt_sqrt") := .(mpg^2, sqrt(wt))]
```

がしており、にするがあるは、1つのがあります。

```
mtcars[, c("mpg_sq", "mpg2_hp") := .(temp1 <- mpg^2, temp1/hp)]
```

.()は、LHS := RHSのがのリストであるにされます。

にされるのは、かっこをします。

```
vn = "mpg_sq"
mtcars[, (vn) := mpg^2]
```

ほとんどの、カラムは<sub>set</sub>ですることできます。

```
set(mtcars, j = "hp_over_wt", v = mtcars$hp/mtcars$wt)
```

のサブセットの

<sub>i</sub>をして、のを「」するをします。

```
mtcars[1:3, newvar := "Hello"]
or
set(mtcars, j = "newvar", i = 1:3, v = "Hello")
```

`data.frame`の他に、またはテストをしてサブセットすることができます。 `i`で「」をすることもできますが、のではよりながまれています。

## の

`levels<-`または`names<-`などのをするは、にオブジェクトをされたコピーでできます。 `data.table`の1つのでのみされても、オブジェクトがコピーされてきえられます。

コピーをたないオブジェクトをするには、 `setnames`をして`data.table`または`data.frame`および`setattr`のをし、オブジェクトのをします。

```
Print a message to the console whenever the data.table is copied
tracemem(mtcars)
mtcars[, cyl2 := factor(cyl)]

Neither of these statements copy the data.table
setnames(mtcars, old = "cyl2", new = "cyl_fac")
setattr(mtcars$cyl_fac, "levels", c("four", "six", "eight"))

Each of these statements copies the data.table
names(mtcars)[names(mtcars) == "cyl_fac"] <- "cf"
levels(mtcars$cf) <- c("IV", "VI", "VIII")
```

これらののはによってわれるため、グローバルにされています。あるでそれらをする、すべてのオブジェクトにします。

```
This function also changes the levels in the global environment
edit_levels <- function(x) setattr(x, "levels", c("low", "med", "high"))
edit_levels(mtcars$cyl_factor)
```

## `data.table`の

### 。SD

`.SD`は、 `data.table`で `by`すべてのをいて、グループの `data.table`のサブセットをします。

`.SD`と`lapply`は、 `data.table`グループ `data.table`にのラムにのをするためにできます

じみみデータセット `mtcars`をききし `mtcars`。

```
mtcars = data.table(mtcars) # Let's not include rownames to keep things simpler
```

データセットのすべてのの、 シリンダ、 `cyl`

```
mtcars[, lapply(.SD, mean), by = cyl]
```

```
cyl mpg disp hp drat wt qsec vs am gear
carb
#1: 6 19.74286 183.3143 122.28571 3.585714 3.117143 17.97714 0.5714286 0.4285714 3.857143
3.428571
#2: 4 26.66364 105.1364 82.63636 4.070909 2.285727 19.13727 0.9090909 0.7272727 4.090909
1.545455
#3: 8 15.10000 353.1000 209.21429 3.229286 3.999214 16.77214 0.0000000 0.1428571 3.285714
3.500000
```

cyl とはに、 vs 、 am 、 gear 、 carb などのデータセットにはのカテゴリのがあります。これらの mean をすることはにはがありません。だから、これらのをしましょう。これは、 .SDcols がになるです。

## .SDcols

.SDcols のを data.table にまれています .SD 。

ので、データセットのすべてのの gear 、 び 、 cyl によって、 gear と cyl

```
All the continuous variables in the dataset
cols_chosen <- c("mpg", "disp", "hp", "drat", "wt", "qsec")

mtcars[order(gear, cyl), lapply(.SD, mean), by = .(gear, cyl), .SDcols = cols_chosen]
```

```
gear cyl mpg disp hp drat wt qsec
#1: 3 4 21.500 120.1000 97.0000 3.700000 2.465000 20.0100
#2: 3 6 19.750 241.5000 107.5000 2.920000 3.337500 19.8300
#3: 3 8 15.050 357.6167 194.1667 3.120833 4.104083 17.1425
#4: 4 4 26.925 102.6250 76.0000 4.110000 2.378125 19.6125
#5: 4 6 19.750 163.8000 116.5000 3.910000 3.093750 17.6700
#6: 5 4 28.200 107.7000 102.0000 4.100000 1.826500 16.8000
#7: 5 6 19.700 145.0000 175.0000 3.620000 2.770000 15.5000
#8: 5 8 15.400 326.0000 299.5000 3.880000 3.370000 14.5500
```

たぶん、グループに mean をしたくないかもしれません。データセットのすべてののをするために、 by はしません。

```
mtcars[, lapply(.SD, mean), .SDcols = cols_chosen]
```

```
mpg disp hp drat wt qsec
#1: 20.09062 230.7219 146.6875 3.596563 3.21725 17.84875
```

- あらかじめ cols\_chosen をするはありません。 .SDcols はをすることができます
- .SDcols はのベクトルをすることもできます。のでは、これは mtcars[, lapply(.SD, mean), .SDcols = c(1,3:7)]

.Nはグループののです。

```
iris[, .(count=.N), by=Species]

Species count
#1: setosa 50
#2: versicolor 50
#3: virginica 50
```

**data.frame**と**data.table**のとのあるコードをく

## サブセットのい

**data.table**は、**data.frame**、**matrix**、**2D array**に、Rでないいくつかの2データの1つです。これらのクラスはすべてサブセットににようにしますが、**A[rows, cols]**スキーマです。

されているのデータを**matrix**、**data.frame**と**data.table**

```
ma <- matrix(rnorm(12), nrow=4, dimnames=list(letters[1:4], c('X', 'Y', 'Z')))
df <- as.data.frame(ma)
dt <- as.data.table(ma)

ma[2:3] #---> returns the 2nd and 3rd items, as if 'ma' were a vector (because it is!)
df[2:3] #---> returns the 2nd and 3rd columns
dt[2:3] #---> returns the 2nd and 3rd rows!
```

がされるかをしたいは、にするがいです。

のをするには、のにカンマをします。

```
ma[2:3,] # \
df[2:3,] # }---> returns the 2nd and 3rd rows
dt[2:3,] # /
```

しかし、をサブセットしたい、いくつかのケースはなってされます。にされていないまたはインデックスをして、3つのすべてをじてサブセットすることができます。

```
ma[, 2:3] # \
df[, 2:3] # \
dt[, 2:3] # }---> returns the 2nd and 3rd columns
ma[, c("Y", "Z")] # /
df[, c("Y", "Z")] # /
dt[, c("Y", "Z")] # /
```

ただし、でまれているではありません

```
mycols <- 2:3
ma[, mycols] # \
df[, mycols] # }---> returns the 2nd and 3rd columns
dt[, mycols, with = FALSE] # /
```

```
dt[, mycols] # ----> Raises an error
```

のケースでは、`mycols`はのとしてされます。 `dt`は`mycols`というのをつけることができないため、エラーがします。

`data.table`パッケージ`data.table`バージョンでは、このはなります。 `dt`をとしてしてのものはされていきました。だから、`dt[, 2:3]`と`dt[, mycols]`はベクトル`2:3`します。 `mycols`がにするため、2のケースではエラーはしません。

## data.frameとdata.tableとのをするための

`data.frame`と`data.table`することがされているコードをくはたくさんあります。おそらく、`data.frame`をなくされたり、がわからないコードをするがあるかもしれません。したがって、これをするためのいくつかのながのにあります

1. のクラスでじようにするをします。
2. のとじことをするのをしてください。
3. `data.table`を`data.table`としてさせ`data.frame`のメソッド`print.data.frame`びします。
4. それらをには`list`としています。
5. かをするにテーブルを`data.frame`にしますなテーブルのはいえです。
6. がでないは、を`data.table`します。

サブセットそのな、`[, ]`セクタをカンマでつけてください

```
A[1:10,]
A[A$var > 17,] # A[var > 17,] just works for data.table
```

サブセット。のがなは、`$`または`[[ ]]`セクタをします。

```
A$var
colname <- 'var'
A[[colname]]
A[[1]]
```

なでのをしたいは、`select`するがあります。

```
B <- `[.data.frame` (A, 2:4)

We can give it a better name
select <- `[.data.frame`
B <- select(A, 2:4)
C <- select(A, c('foo', 'bar'))
```

サブセット「け」`data.frame`は`data.frame`があり`row.names`、`data.table`はの`key`があります。のは、`row.names`にけ、`row.names`のはのを`data.table`することです。

```
B <- A[A$var != 0,]
or...
B <- with(A, A[var != 0,]) # data.table will silently index A by var before subsetting

stuff <- c('a', 'c', 'f')
C <- A[match(stuff, A$name),] # really worse than: setkey(A); A[stuff,]
```

**1**のテーブルをし、ベクトルとしてをします。これまでのたちがたことはです

```
B <- select(A, 2) #---> a table with just the second column
C <- unlist(A[1,]) #---> the first row as a vector (coerced if necessary)
```

## data.tableにキーをする

はい、あなたは1.9.6にSETKEYするがあります

には1.9.6より、テーブルのキーとしてカラムをすることで、にきなテーブルの、`data.table`がされました。[が544した20159の[イントロ・ビネット・ページ5](#)]テーブルをするときに、このキーを'setkey'ですか、'key ='をするいコードがあります。

```
library(data.table)
DT <- data.table(
 x = letters[1:5],
 y = 5:1,
 z = (1:5) > 3
)

#> DT
x y z
#1: a 5 FALSE
#2: b 4 FALSE
#3: c 3 FALSE
#4: d 2 TRUE
#5: e 1 TRUE
```

setkey コマンドでキーをします。のをつキーをつことができます。

```
setkey(DT, y)
```

テーブルのテーブルのキーをする

```
tables()

> tables()
 NAME NROW NCOL MB COLS KEY
[1,] DT 5 3 1 x,y,z y
Total: 1MB
```

これはあなたのデータをソートすることにしてください。

```
#> DT
x y z
```



```
#1: e 1 TRUE
#2: d 2 TRUE
#3: c 3 FALSE
#4: b 4 FALSE
#5: a 5 FALSE
```

はです

v1.9.6は、にテーブルをするののキーをするがありました。 data.tableのはスピードアップし、キーのをきえることができる"on="をしました。 [なについては、このSOのを](#)してください。

20171、は[インデックスのにバイナリ](#)をきんで、 "on"をし、インデックスのためにのをできるようにしました。

セカンダリインデックスをしますか

keyとたで、あなたはsetindex(DT, key.col)またはsetindex(DT, key.col) setindexv(DT, "key.col.string") できます。ここでDTはあなたのdata.tableです。 setindex(DT, NULL) すべてのインデックスをしsetindex(DT, NULL)。

indices(DT) をしてセカンダリインデックスをしてください。

なぜですか

これは、キーとはなりテーブルをソートしませんが、 "on"をしてなインデックスをにします。キーは1つだけですが、のセカンダリインデックスをすることができます。これにより、テーブルのキーをしてするがなくなります。サブセットするをすると、サブセットがされます。

のではyがDTのキーであったことをいしてください。

```
DT
x y z
1: e 1 TRUE
2: d 2 TRUE
3: c 3 FALSE
4: b 4 FALSE
5: a 5 FALSE

Let us set x as index
setindex(DT, x)

Use indices to see what has been set
indices(DT)
[1] "x"

fast subset using index and not keyed column
DT["c", on = "x"]
x y z
#1: c 3 FALSE

old way would have been rekeying DT from y to x, doing subset and
perhaps keying back to y (now we save two sorts)
This is a toy example above but would have been more valuable with big data sets
```

オンラインでデータをむ <https://riptutorial.com/ja/r/topic/849/データ>

## 57: テキストマイニング

### Examples

N グラムの**Word** クラウドをするためのデータのスクラップ

のでは、`tm` テキストマイニングパッケージをして、Webからのテキストデータをスクラップして  
りこし、シンボリックなシェーディングとけをつワードクラウドをしています。

```
require(RWeka)
require(tau)
require(tm)
require(tm.plugin.webmining)
require(wordcloud)

Scrape Google Finance -----
googlefinance <- WebCorpus(GoogleFinanceSource("NASDAQ:LFVN"))

Scrape Google News -----
lv.googlenews <- WebCorpus(GoogleNewsSource("LifeVantage"))
p.googlenews <- WebCorpus(GoogleNewsSource("Protandim"))
ts.googlenews <- WebCorpus(GoogleNewsSource("TrueScience"))

Scrape NYTimes -----
lv.nytimes <- WebCorpus(NYTimesSource(query = "LifeVantage", appid = nytimes_appid))
p.nytimes <- WebCorpus(NYTimesSource("Protandim", appid = nytimes_appid))
ts.nytimes <- WebCorpus(NYTimesSource("TrueScience", appid = nytimes_appid))

Scrape Reuters -----
lv.reutersnews <- WebCorpus(ReutersNewsSource("LifeVantage"))
p.reutersnews <- WebCorpus(ReutersNewsSource("Protandim"))
ts.reutersnews <- WebCorpus(ReutersNewsSource("TrueScience"))

Scrape Yahoo! Finance -----
lv.yahoofinance <- WebCorpus(YahooFinanceSource("LFVN"))

Scrape Yahoo! News -----
lv.yahoonews <- WebCorpus(YahooNewsSource("LifeVantage"))
p.yahoonews <- WebCorpus(YahooNewsSource("Protandim"))
ts.yahoonews <- WebCorpus(YahooNewsSource("TrueScience"))

Scrape Yahoo! Inplay -----
lv.yahooinplay <- WebCorpus(YahooInplaySource("LifeVantage"))

Text Mining the Results -----
corpus <- c(googlefinance, lv.googlenews, p.googlenews, ts.googlenews, lv.yahoofinance,
lv.yahoonews, p.yahoonews,
ts.yahoonews, lv.yahooinplay) #lv.nytimes, p.nytimes, ts.nytimes,lv.reutersnews,
p.reutersnews, ts.reutersnews,

inspect(corpus)
wordlist <- c("lfvn", "lifelivantage", "protandim", "truescience", "company", "fiscal",
"nasdaq")

ds0.lg <- tm_map(corpus, content_transformer(tolower))
```

```

ds1.1g <- tm_map(ds0.1g, content_transformer(removeWords), wordlist)
ds1.1g <- tm_map(ds1.1g, content_transformer(removeWords), stopwords("english"))
ds2.1g <- tm_map(ds1.1g, stripWhitespace)
ds3.1g <- tm_map(ds2.1g, removePunctuation)
ds4.1g <- tm_map(ds3.1g, stemDocument)

tdm.1g <- TermDocumentMatrix(ds4.1g)
dtm.1g <- DocumentTermMatrix(ds4.1g)

findFreqTerms(tdm.1g, 40)
findFreqTerms(tdm.1g, 60)
findFreqTerms(tdm.1g, 80)
findFreqTerms(tdm.1g, 100)

findAssocs(dtm.1g, "skin", .75)
findAssocs(dtm.1g, "scienc", .5)
findAssocs(dtm.1g, "product", .75)

tdm89.1g <- removeSparseTerms(tdm.1g, 0.89)
tdm9.1g <- removeSparseTerms(tdm.1g, 0.9)
tdm91.1g <- removeSparseTerms(tdm.1g, 0.91)
tdm92.1g <- removeSparseTerms(tdm.1g, 0.92)

tdm2.1g <- tdm92.1g

Creates a Boolean matrix (counts # docs w/terms, not raw # terms)
tdm3.1g <- inspect(tdm2.1g)
tdm3.1g[tdm3.1g>=1] <- 1

Transform into a term-term adjacency matrix
termMatrix.1gram <- tdm3.1g %*% t(tdm3.1g)

inspect terms numbered 5 to 10
termMatrix.1gram[5:10,5:10]
termMatrix.1gram[1:10,1:10]

Create a WordCloud to Visualize the Text Data -----
notsparse <- tdm2.1g
m = as.matrix(notsparse)
v = sort(rowSums(m),decreasing=TRUE)
d = data.frame(word = names(v),freq=v)

Create the word cloud
pal = brewer.pal(9,"BuPu")
wordcloud(words = d$word,
 freq = d$freq,
 scale = c(3,.8),
 random.order = F,
 colors = pal)

```



`random.order`とシーケンシャルパレットをすることにしてください。プログラマは、にをりて、をすることによって、クラウドでよりくのをできます。

は1グラムのです。

たちはn-gramにきなをげることができます。そうすることで、TDMをすることでn-gramをするのになをつテキストマイニングをどのようにするかをていきます。

Rのn-gramをつたののは、テキストマイニングのもなパッケージである`tm`、bi-gramやn-gramのトークンをにサポートしていないことです。トークンは、の、またはまたはのグループをトークンとばれるのデータとしてすプロセスです。

いにも、々はアップグレードされたトークナイザで`tm`をしけることをにするいくつかのハックをっています。これをするためのはあります。 `tau`の`textcnt()`をとって、のシンプルな`textcnt()`をくことができます

```
tokenize_ngrams <- function(x, n=3)
return(rownames(as.data.frame(unclass(textcnt(x, method="string", n=n)))))
```

あるいは`tm`で`RWeka`をびすことができます

```
BigramTokenize
BigramTokenizer <- function(x) NGramTokenizer(x, Weka_control(min = 2, max = 2))
```

このから、1グラムのとにめることができます。

```
Create an n-gram Word Cloud -----
tdm.ng <- TermDocumentMatrix(ds5.1g, control = list(tokenize = BigramTokenizer))
dtm.ng <- DocumentTermMatrix(ds5.1g, control = list(tokenize = BigramTokenizer))
```



## 58: デバッグ

### Examples

ブラウザをする

`browser` はブレークポイントのようにできます。コードのはびされたでします。に、ユーザーはのをし、のRコードをし、でコードをステップできます。

コードで `browser()` がヒットしたら、インタラクティブなインタプリタがします。どんなRコードもどおりにすることができ、さらにのコマンドがします。

| コマンド |                               |
|------|-------------------------------|
| c    | ブラウザをしてプログラムをする               |
| f    | のループまたはをする\                   |
| n    | ステップオーバーのステートメントの、びしのステップオーバー |
| s    | ステップインのステートメントの、びしのステップ       |
| どこで  | スタックトレースをする                   |
| r    | 「」をしてする                       |
| Q    | ブラウザをしてする                     |

たとえば、のようなスクリプトがあるとします。

```
toDebug <- function() {
 a = 1
 b = 2

 browser()

 for(i in 1:100) {
 a = a * b
 }
}

toDebug()
```

のスクリプトをすると、はのようなものがされます。

```
Called from: toDebug
Browser[1]>
```

プロンプトとにすることができます、

```
Called from: toDebug
Browser[1]> a
[1] 1
Browser[1]> b
[1] 2
Browse[1]> n
debug at #7: for (i in 1:100) {
 a = a * b
}
Browse[2]> n
debug at #8: a = a * b
Browse[2]> a
[1] 1
Browse[2]> n
debug at #8: a = a * b
Browse[2]> a
[1] 2
Browse[2]> Q
```

`browser()` は、のようなチェーンのとしてもできます。

```
mtcars %>% group_by(cyl) %>% {browser() }
```

## デバッグの

あなたはしてデバッグするためののことができます `debug`。

```
debug(mean)
mean(1:3)
```

のびしはデバッグモードになります。 `undebug` このをにすることができます。

```
undebug(mean)
mean(1:3)
```

のデバッグモードにだけるがあることがわかっているは、 `debugonce` のをして `debugonce`。

```
debugonce(mean)
mean(1:3)
mean(1:3)
```

オンラインでデバッグをむ <https://riptutorial.com/ja/r/topic/1695/デバッグ>



## 59: パイプ>など

き

`magrittr`、`dplyr`、およびのRパッケージでなパイプは、よりなれのRメソッドではなく、`dplyr`をしてのステップのとして1つのステップのをすることによってのをしてデータオブジェクトをしすびし。

パイプのされたは、かれたコードののをめることです。パフォーマンスのについては、「」をしてください。

- `lhs>rhs rhs(lhs)`パイプ
- `lhs>rhs a = 1 rhs(lhs, a = 1)`パイプ `rhs(lhs, a = 1)`
- `lhs>rhs a = 1、 b = 0 rhs(a = 1, b = lhs)`パイプ `rhs(a = 1, b = lhs)`
- `lhs<>rhs lhs <- rhs(lhs)`パイプ `lhs <- rhs(lhs)`
- `lhs$rhs with(lhs, rhs(lhs$a))`パイプ
- `lhsT>rhs { rhs(lhs); lhs }`パイプ `{ rhs(lhs); lhs }`

### パラメーター

| lhs                  | rhs                  |
|----------------------|----------------------|
| またはmagrittrプレースホルダー。 | magrittrセマンティクスをしすびし |

%>%をするパッケージ

`pipe`は`magrittr`パッケージでされて`magrittr`ますが、`dplyr`パッケージ `magrittr`からをインポートし  
ますできなとをしました。は`tidyverse`で`tidyverse`、のデータとAPIをしているため、してするパッ  
ケージのコレクションです。

`magrittr`パッケージには、りてパイプ`%<>%`、パイプ`%%`、ティーオペレータ`%T>%`など、のをとする  
にも、パイプのいくつかのバリエーションがされています。また、な`+`、`[`、`[[`など]をつをきえ  
ることで、のパイプでにできるエイリアスをしす。

ドキュメントの

のようなのとじように`+`、`*`、`^`、`&`、`%in%`あなたがでそれをく、あなたはドキュメントをつけ  
ることができます? '%>%' または `help('%>%') pkg:magrittr`をしたパッケージをロードしたとします。

## ホットキー

`Ctrl+Shift+M` *Windows*および*Linux*、`Cmd+Shift+M` *Mac* のパイプのなホットキーがRStudioにあります。

## パフォーマンスにする

パイプはですが、にのオーバーヘッドによりパフォーマンスにがぶことにしてください。パイプをするときは、の2つのをくしてください。

- マシンパフォーマンスループ
- `object %>% rm()` は `object` しません

## Examples

など

パイプ`%>%`は、にをするためにされます。のではなく、`magrittr`などのパッケージをしたにのみできます。パイプは、パイプのLHSをとり、パイプのRHSのののとしてします。えは

```
library(magrittr)

1:10 %>% mean
[1] 5.5

is equivalent to
mean(1:10)
[1] 5.5
```

パイプをして、のびしをきえることができます。のパイプをすると、からにあって、からへにみきを行うことができます。たとえば、としてされている`years`がありますが、にするがあるとします。がわれるのをぐため、まずにしてからにします。

```
years <- factor(2008:2012)

nesting
as.numeric(as.character(years))

piping
years %>% as.character %>% as.numeric
```

RHSののとしてLHSをしたくないは、のけやなどのがあります。パイプされたがどこにくのかをしします。

```
example with grepl
its syntax:
grepl(pattern, x, ignore.case = FALSE, perl = FALSE, fixed = FALSE, useBytes = FALSE)

note that the `substring` result is the *2nd* argument of grepl
```

```
grepl("Wo", substring("Hello World", 7, 11))

piping while naming other arguments
"Hello World" %>% substring(7, 11) %>% grepl(pattern = "Wo")

piping with .
"Hello World" %>% substring(7, 11) %>% grepl("Wo", .)

piping with . and curly braces
"Hello World" %>% substring(7, 11) %>% { c(paste('Hi', .)) }
#[1] "Hi World"

#using LHS multiple times in argument with curly braces and .
"Hello World" %>% substring(7, 11) %>% { c(paste(. , 'Hi', .)) }
#[1] "World Hi World"
```

りしするのをえると、それをにするとなくなることがよくあります。パイプをうと、のようなドットでシーケンスをすることで、このようなをみみなですることができます。

```
. %>% RHS
```

たとえば、があり、そのをしたいとします。

```
library(magrittr) # needed to include the pipe operators
library(lubridate)
read_year <- . %>% as.character %>% as.Date %>% year

Creating a dataset
df <- data.frame(now = "2015-11-11", before = "2012-01-01")
now before
1 2015-11-11 2012-01-01

Example 1: applying `read_year` to a single character-vector
df$now %>% read_year
[1] 2015

Example 2: applying `read_year` to all columns of `df`
df %>% lapply(read_year) %>% as.data.frame # implicit `lapply(df, read_year)`
now before
1 2015 2012

Example 3: same as above using `mutate_all`
library(dplyr)
df %>% mutate_all(funs(read_year))
if an older version of dplyr use `mutate_each`
now before
1 2015 2012
```

のをするか `functions` をして、のをできます。

```
read_year
Functional sequence with the following components:
#
1. as.character(.)
2. as.Date(.)
3. year(.)
```

```
#
Use 'functions' to extract the individual functions.
```

シーケンスのによってにアクセスすることもできます。

```
read_year[[2]]
function (.)
as.Date(.)
```

に、このアプローチは、さがよりもであるにであります。

## <>でのりて

magrittrパッケージには、に1つの`rhs`にパイプしてをし、をすることによってをする`%<>%`がまれています。これにより、オブジェクトを2<- のに1ずつするがなくなります。 `%<>%`はチェーンなのでなければなりません

```
library(magrittr)
library(dplyr)

df <- mtcars
```

## くのではなく

```
df <- df %>% select(1:3) %>% filter(mpg > 20, cyl == 6)
```

## または

```
df %>% select(1:3) %>% filter(mpg > 20, cyl == 6) -> df
```

は、`df`パイプしてりてします。

```
df %<>% select(1:3) %>% filter(mpg > 20, cyl == 6)
```

## \$でコンテンツをする

パイプ`%$%`は、のオブジェクトのRシンボルとしてをのにします。これは、`data`をたないえは`lm`とはなりとして`data.frame`と`column`のをらないほとんどのな`dplyr`にパイプするときです。

パイプ`%$%`すると、をすることがあるときにパイプラインをることをけることができます。たとえば、`data.frame`をフィルタし、`cor.test`をして2つのにしてテストをするとし`cor.test`。

```
library(magrittr)
library(dplyr)
mtcars %>%
 filter(wt > 2) %$%
 cor.test(hp, mpg)
```

```
#>
#> Pearson's product-moment correlation
#>
#> data: hp and mpg
#> t = -5.9546, df = 26, p-value = 2.768e-06
#> alternative hypothesis: true correlation is not equal to 0
#> 95 percent confidence interval:
#> -0.8825498 -0.5393217
#> sample estimates:
#> cor
#> -0.7595673
```

ここでの`%>%`パイプは`data.frame`を`filter()`にし、`%%`パイプはカラムを`cor.test()`します。

パイプは、ベースR `with()` のパイプバージョンのようにし、オブジェクトがとしてけられます。

## dplyrとggplot2でパイプをする

`%>%`をして、`dplyr`を`ggplot`にパイプすることもできます。これにより、にカスタマイズなされたデータEDAパイプラインがされます。このメソッドは、`ggplot`でにをするよりもで、なをけるというがあります。

```
library(dplyr)
library(ggplot2)

diamonds %>%
 filter(depth > 60) %>%
 group_by(cut) %>%
 summarize(mean_price = mean(price)) %>%
 ggplot(aes(x = cut, y = mean_price)) +
 geom_bar(stat = "identity")
```

## T>でをする

Rのには、、、、プロットなどのがあり、ずしものあるやましいをすとはらないもあります。

`%T>% tee`では、の`lhs`をそのまましたまま、にをすることができます。いえれば、`tee`は`%>%`のようにしますが、りは`rhs` / のではなく、`lhs`であるがなります。

オブジェクトを、パイプ、きみ、す。こので`%T>%`わりに`%>%`がされた、`all_letters`は、ソートされたオブジェクトのではなく`NULL`がまれます。

```
all_letters <- c(letters, LETTERS) %>%
 sort %T>%
 write.csv(file = "all_letters.csv")

read.csv("all_letters.csv") %>% head()
x
1 a
2 A
```

```
3 b
4 B
5 c
6 C
```

のないオブジェクトを`save()`パイプすると、きのオブジェクトがされ、`load()`をしてワークスペースにロードされたとき。ただし、ヘルパーをしたインラインでとしてすることもできますがです

。

```
all_letters <- c(letters, LETTERS) %>%
 sort %T>%
 save(file = "all_letters.RData")

load("all_letters.RData", e <- new.env())

get("all_letters", envir = e)
Error in get("all_letters", envir = e) : object 'all_letters' not found

get(".", envir = e)
[1] "a" "A" "b" "B" "c" "C" "d" "D" "e" "E" "f" "F" "g" "G" "h" "H" "i" "I" "j" "J"
[21] "k" "K" "l" "L" "m" "M" "n" "N" "o" "O" "p" "P" "q" "Q" "r" "R" "s" "S" "t" "T"
[41] "u" "U" "v" "V" "w" "W" "x" "X" "y" "Y" "z" "Z"

Work-around
save2 <- function(. = ., name, file = stop("'file' must be specified")) {
 assign(name, .)
 call_save <- call("save", ... = name, file = file)
 eval(call_save)
}

all_letters <- c(letters, LETTERS) %>%
 sort %T>%
 save2("all_letters", "all_letters.RData")
```

オンラインでパイプ>などをむ <https://riptutorial.com/ja/r/topic/652/パイプ---gt--など->

---

## 60: パターンマッチングと

き

このトピックでは、するパターンのときえについてします。なパターンののについては、[を](#)して  
ください。

- `grep "query"、 "subject"、 optional_args`
- `grepl "query"、 "subject"、 optional_args`
- `gsub "グループ1グループ2"、 "\\グループ"、 ""`

---

## のとのい

エスケープされたシンボル `\1` は、 `pattern` だけでなく、 `sub` と `gsub replacement` も `\\1` などの2にエス  
ケープするがあります。

デフォルトでは、すべてのコマンド `grep`、 `sub`、 `regexpr` のパターンは Perl Compatible Regular  
Expression PCRE ではないので、ルックアラウンドのようなものはサポートされていません。し  
かし、は `perl=TRUE` をけてそれらをにします。については、[Rののトピック](#) をしてください。

---

## パツケージ

- 
- `stringr`

## Examples

をう

```
example data
test_sentences <- c("The quick brown fox quickly", "jumps over the lazy dog")
```

のキツネをりましょう

```
sub("brown","red", test_sentences)
#[1] "The quick red fox quickly" "jumps over the lazy dog"
```

さあ、 `"fast"` キツネのを `"fastly"` `"fast"` ろう。これではできません

```
sub("quick", "fast", test_sentences)
#[1] "The fast red fox quickly" "jumps over the lazy dog"
```

subはになをうだけです。 **グローバル**のためにgsubがです。

```
gsub("quick", "fast", test_sentences)
#[1] "The fast red fox fastly" "jumps over the lazy dog"
```

のについては、「**によるの**」をしてください。

マッチの

```
example data
test_sentences <- c("The quick brown fox", "jumps over the lazy dog")
```

---

## マッチはありますか

grepl()は、またはがまたはベクトルにするかどうかをチェックするためにされます。はTRUE / FALSEまたは "Boolean"ベクトルをします。

に「fox」というがあるかどうかをし、そのわりにブールベクトルをけることができます。

```
grepl("fox", test_sentences)
#[1] TRUE FALSE
```

---

## の

grepはとをります。これは、インデックスのベクトルをします。これは、「fox」というをむをします。

```
grep("fox", test_sentences)
#[1] 1
```

---

## する

パターンにするをするには

```
each of the following lines does the job:
test_sentences[grep("fox", test_sentences)]
test_sentences[grepl("fox", test_sentences)]
grep("fox", test_sentences, value = TRUE)
[1] "The quick brown fox"
```



"fox"パターンはではなくなるなので、 `fixed = TRUE` することでパフォーマンスをさせることができます `grep` または `grepl` いずれか。

```
grep("fox", test_sentences, fixed = TRUE)
#[1] 1
```

パターンとしないをするには、 `invert = TRUE` `grep` をします。または `-grep(...)` または `!grepl(...)` を `-grep(...)` サブセットのルールにいます。

`grepl(pattern, x)` と `grep(pattern, x)` で、 `x` パラメータはベクトルされ、 `pattern` パラメータはベクトルされません。として、これらをして `pattern[1]` を `x[1]`、 `pattern[2]` を `x[2]` などとすることはできません。

## マッチの

`grepl` コマンドなどをした、 `TRUE` または `FALSE` をすることができ `FALSE`。これは、例えばきなデータセットのにである。これをうには、 `summary` コマンドをします。

```
example data
test_sentences <- c("The quick brown fox", "jumps over the lazy dog")

find matches
matches <- grepl("fox", test_sentences)

overview
summary(matches)
```

シングルとグローバルマッチ。

をとってする、PCREのはグローバルの `g` は `g` です。

Rマッチングとには、のマッチとグローバルマッチという2つのバージョンがあります。

- `sub(pattern, replacement, text)` は、 `pattern` ののオカレンスをテキストのできえます
- `gsub(pattern, replacement, text)` は `sub` とじことをい `gsub(pattern, replacement, text)` が、パターンのごとに
- `regexpr(pattern, text)` は `pattern` ののインスタンスののをします
- `gregexpr(pattern, text)` はすべてののをします。

いくつかのランダムなデータ

```
set.seed(123)
teststring <- paste0(sample(letters, 20), collapse="")

teststring
```

```
#[1] "htjuwakqxyzpgrsbncvyo"
```

をかのものにきえたいに、これがどのようにするかてみましょう。

```
sub("[aeiouy]", " ** HERE WAS A VOWEL ** ", teststring)
#[1] "htj ** HERE WAS A VOWEL ** wakqxyzpgrsbncvyo"

gsub("[aeiouy]", " ** HERE WAS A VOWEL ** ", teststring)
#[1] "htj ** HERE WAS A VOWEL ** w ** HERE WAS A VOWEL ** kqxyzpgrsbncv ** HERE WAS A VOWEL ** **
HERE WAS A VOWEL ** "
```

さて、1つまたはののにつけるをてみましょう。

```
regexpr("^[aeiou][aeiou]+", teststring)
#[1] 3
#attr(,"match.length")
#[1] 2
#attr(,"useBytes")
#[1] TRUE
```

たちは、さ2のの3にします。すなわち、 ju

すぐすべてのをたい

```
gregexpr("^[aeiou][aeiou]+", teststring)
#[[1]]
#[1] 3 5 19
#attr(,"match.length")
#[1] 2 2 2
#attr(,"useBytes")
#[1] TRUE
```

これはにらしいですが、これはマッチのをするだけであるものをするのはではありません。ここでは `regmatches` というのは `regmatches` からマッチしたをすることですが、がなります。

マッチをにしてのからしましょう

```
matches <- gregexpr("^[aeiou][aeiou]+", teststring)
regmatches(teststring, matches)
#[[1]]
#[1] "ju" "wa" "yo"
```

これはショートカットをたないのがにこえるかもしれませんが、これはのものとのマッチによるのからのをにします2つのベクトルをすることを教えてください。な

```
teststring2 <- "this is another string to match against"
regmatches(teststring2, matches)
#[[1]]
#[1] "is" " i" "ri"
```

デフォルトでパターンはPerlではありませんが、ルックアラウンドのようなものはサポートされ

ていませんが、ここでされているは`perl=TRUE`をしています。

きなデータセットでをつける

きなデータセットの、`grepl("fox", test_sentences)`のびしは`grepl("fox", test_sentences)`しません。きなデータセットは、えは、クロールされたウェブサイトまたはのツイートなどです。

のは、`perl = TRUE`オプションのです。よりくても、オプション`fixed = TRUE`です。なはのとおりです。

```
example data
test_sentences <- c("The quick brown fox", "jumps over the lazy dog")

grepl("fox", test_sentences, perl = TRUE)
#[1] TRUE FALSE
```

テキストマイニングの、しばしばコーパスがされます。コーパスは`grepl`することはできません。したがって、このをえてみましょう

```
searchCorpus <- function(corpus, pattern) {
 return(tm_index(corpus, FUN = function(x) {
 grepl(pattern, x, ignore.case = TRUE, perl = TRUE)
 })))
}
```

オンラインでパターンマッチングとをむ <https://riptutorial.com/ja/r/topic/1123/パターンマッチングと>

## 61: パッケージのインストール

- `install.packagespkgs`、`lib`、`repos`、メソッド、`destdir`、...

### パラメーター

| パラメータ                |                                                                                                                          |
|----------------------|--------------------------------------------------------------------------------------------------------------------------|
| <code>pkgs</code>    | パッケージのベクトル。 <code>repos = NULL</code> 、ファイルパスのベクトルです。                                                                    |
| <code>lib</code>     | パッケージをインストールするライブラリディレクトリをえるベクタ。                                                                                         |
| レポ                   | ベクトル、するリポジトリのベースURLは、ローカルファイルからインストールするは <code>NULL</code> にすることができます                                                    |
|                      | ダウンロード                                                                                                                   |
| <code>destdir</code> | ダウンロードしたパッケージがされているディレクトリ                                                                                                |
|                      | これらのパッケージがするアンインストールされたパッケージ// <code>import / suggest</code> へのリンクにをインストールするかどうかをにします。 <code>repos = NULL</code> はされません。 |
| ...                  | 'download.file'にす、またはOS XとWindowsのバイナリインストールのためのにす。                                                                      |

## するドキュメント

- [パッケージの](#)

## Examples

リポジトリからパッケージをダウンロードしてインストールする

パッケージは、R、データ、およびコンパイルされたコードのコレクションで、[にされたです](#)。パブリックおよびプライベートリポジトリは、Rパッケージのコレクションをホストするためにされます。RパッケージののコレクションはCRANからできます。

## CRANの

パッケージは、のコードをして [CRAN](#) からインストールできます。

```
install.packages("dplyr")
```

"dplyr" はベクトルとされます。

c() をし、パッケージのベクトルをすことによって、のパッケージをにインストールできます。

```
install.packages(c("dplyr", "tidyr", "ggplot2"))
```

によっては、getOption("repos") にじて、install.packages が [CRAN ミラー](#) のプロンプトをするか  
することがあります。これをぐには、repos として [CRAN ミラー](#) をします。

```
install.packages("dplyr", repos = "https://cloud.r-project.org/")
```

repos をすると、のリポジトリからインストールすることもできます。なすべてのオプションの  
については、 ?install.packages して ?install.packages。

ほとんどのパッケージは、のパッケージ例えば、パッケージ `data.table` にされたをとします。この  
パッケージでされているすべてのパッケージまたはのパッケージをインストールするには、の  
dependencies を TRUE にするがあります。

```
install.packages("data.table", dependencies = TRUE)
```

## Bioconductor

バイオ [コンダクタ](#) は、バイオインフォマティクスにするパッケージのなコレクションをホストし  
ています。それらは、biocLite をとしたのパッケージをします。

```
Try http:// if https:// URLs are not supported
source("https://bioconductor.org/biocLite.R")
biocLite()
```

デフォルトでは、もにされるをするパッケージのサブセットがインストールされます。のパッ  
ケージは、パッケージのベクトルをすことでインストールできます。たとえば、Bioconductor から  
RImmPort をインストールするには

```
source("https://bioconductor.org/biocLite.R")
biocLite("RImmPort")
```

ローカルソースからパッケージをインストールする

ローカルソースファイルからパッケージをインストールするには

```
install.packages(path_to_source, repos = NULL, type="source")

install.packages("~/Downloads/dplyr-master.zip", repos=NULL, type="source")
```

ここで、`path_to_source`はローカルソースファイルのパスです。

ダウンロードしたzipまたはtar.gzソースファイルをするためのウィンドウをくこのコマンドはのとおります。

```
install.packages(file.choose(), repos=NULL)
```

のとして、**GUI**ベースの**RStudio**をするがあります。

ステップ**1** ツールにします。

ステップ**2** パッケージのインストールにみます。

### 3 「パッケージアーカイブファイルとしてのインストール」.zip; .tar.gz

ステップ**4** **Browse**をクリックして、パッケージファイル `crayon_1.3.1.zip`をし、しばらくしてからパッケージアーカイブタブにパッケージパスとファイルをした

ローカルソースからRパッケージをインストールするのは、`devtools`パッケージの`install_local()`をすることです。

```
library(devtools)
install_local("~/Downloads/dplyr-master.zip")
```

## GitHubからパッケージをインストールする

パッケージをGitHubからインストールするには、`devtools`パッケージをします

```
library(devtools)
install_github("authorName/repositoryName")
```

githubから`ggplot2`をインストールするには

```
devtools::install_github("tidyverse/ggplot2")
```

のコマンドは、**master**ブランチにする`ggplot2`のバージョンをインストールします。リポジトリのブランチからインストールするには、`ref`をしてブランチのをします。たとえば、のコマンドは、`googleway`パッケージの`dev_general`ブランチをインストールします。

```
devtools::install_github("SymbolixAU/googleway", ref = "dev_general")
```

もうつのオプションは`ghit`パッケージをうこと`ghit`。 `github`からパッケージをインストールするためののをしします

```
install.packages("ghit")
ghit::install_github("google/CausalImpact")
```

Githubのプ ラ イベートリポジトリにあるパッケージをインストールするには、 <http://www.github.com/settings/tokens/>でアクセストークンをし **ます** じことについては、`install_github`をして **ください** 。 のをしします

1. 

```
install.packages(c("curl", "httr"))
```
2. 

```
config = httr::config(ssl_verifypeer = FALSE)
```
3. 

```
install.packages("RCurl")
options(RCurlOptions = c(getOption("RCurlOptions"), ssl.verifypeer = FALSE,
ssl.verifyhost = FALSE))
```
4. 

```
getOption("RCurlOptions")
```

がされます。

```
ssl.verifypeer ssl.verifyhost
FALSE FALSE
```

5. 

```
library(httr)
set_config(config(ssl_verifypeer = 0L))
```

これにより、「されたCAでピアをできません」というなエラーがされます。

6. に、 のコマンドをしてパッケージをシームレスにインストールします

```
install_github("username/package_name", auth_token="abc")
```

あるいは、 `GITHUB_PAT` しします。

```
Sys.setenv(GITHUB_PAT = "access_token")
devtools::install_github("organisation/package_name")
```

GithubでされたPATは、しかされません。つまり、にされたときにされるため、そのトークンを `.Rprofile` にするのが `.Rprofile` です。これは、にプライベートリポジトリがあるにもちます。

**CLI** パッケージマネージャをう **な** パックマンのい

`pacman` はRのなパッケージマネージャです。

`pacman`は、ユーザがのコードで `p_load` して、なすべてのパッケージをコンパクトにロードし、しているものおよびそのをコンパクトにロードすることをにします。 `pacman` では、パッケージのにをするはありません。なはのとおりです。

```
p_load(data.table, dplyr, ggplot2)
```

このアプローチで `library`、 `require`、または `install.packages` ステートメントが `require` なのパッケージは、 `pacman` です。

```
library(pacman)
p_load(data.table, dplyr, ggplot2)
```

または、にです。

```
pacman::p_load(data.table, dplyr, ggplot2)
```

`pacman`は、パッケージをするためのコードをなくしてをするだけでなく、なパッケージをインストールすることで、インストールなコードのをにします。

はわからないかもしれませんが、 `pacman` のコードののにでかあなたのコードをするユーザーのライブラリーにインストールされているベストプラクティスをインストールするには、めることです `pacman` それがまだないみまれた

```
if(!(require(pacman)) install.packages("pacman"))
pacman::p_load(data.table, dplyr, ggplot2)
```

## パッケージのローカルをインストールする

Rパッケージのにりんでいるに、パッケージのバージョンをインストールするがあることがよくあります。これは、まずパッケージのソースをしコマンドラインで

```
R CMD build my_package
```

それをRにインストールします。のバージョンのパッケージがロードされているのRセッションは、リロードするがあります。

```
unloadNamespace("my_package")
library(my_package)
```

よりなアプローチは、プロセスをするために `devtools` パッケージをします。ディレクトリーがパッケージディレクトリーにされたRセッション

```
devtools::install()
```

パッケージをビルド、インストール、ロードします。



オンラインでパッケージのインストールをむ <https://riptutorial.com/ja/r/topic/1719/パッケージのインストール>

## 62: パッケージの

き

パッケージはベースRにされます。このドキュメントでは、インストールされたパッケージとそれをするについてします。ドキュメント [パッケージのインストール](#)

Comprehensive R Archive NetworkCRANは、な[パッケージリポジトリ](#)です。

### Examples

パッケージをする

dplyrパッケージとそののにするをするには、のようになります。

```
help(package = "dplyr")
```

にパッケージをロードするはありません。

パッケージのビルト インデータセットをする

パッケージdplyrのみみデータセットをするには

```
data(package = "dplyr")
```

にパッケージをロードするはありません。

パッケージのエクスポートされたをする

パッケージdplyrののリストをするには、まずパッケージをロードするがあります。

```
library(dplyr)
ls("package:dplyr")
```

パッケージのバージョンをする

パッケージがなくともインストールされているがあります。のセッションにロードされていないはありません。

```
Checking package version which was installed at past or
installed currently but not loaded in the current session

packageVersion("seqinr")
[1] '3.3.3'
```

```
packageVersion("RWeka")
[1] '0.4.29'
```

のセッションでロードされたパッケージをする

ロードされたパッケージのリストをするには

```
search()
```

または

```
(.packages())
```

オンラインでパッケージのをむ <https://riptutorial.com/ja/r/topic/7408/パッケージの>

## 63: ハッシュマップ

### Examples

ハッシュマップとしての

のでは、ハッシュマップとハッシュテーブルというはじでされ、じ、つまりハッシュをしてなキールックアップをするデータをしています。

き

Rはネイティブのハッシュテーブルをしませんが、`new.env` デフォルトからされた`environment`オブジェクトがハッシュされたキーをするというをすることで、のをできます。の2つのはです。

`hash`パラメータのデフォルトは`TRUE`です。

```
H <- new.env(hash = TRUE)
H <- new.env()
```

さらに、ハッシュテーブルが、デフォルト29をする`size`パラメータをしてのサイズでりてされることをすることができる。のすべてのRオブジェクトとに、`environment`はのメモリをし、したがって、`size`にデフォルトのをするはありませんが、オブジェクトにくのがまれているパフォーマンスのわずかなメリットがあります。`size`をしてなスペースをりてることが、よりきなメモリフットプリントをつオブジェクトにならないことにしてください。

```
object.size(new.env())
56 bytes

object.size(new.env(size = 10e4))
56 bytes
```

のをいてうことができるのいずれか`[[<-`または`$<-`のためにされる`environment`ではなく、「シングるブラケット」りてをして、クラス`[[<-`

```
H <- new.env()

H[["key"]] <- rnorm(1)

key2 <- "xyz"
H[[key2]] <- data.frame(x = 1:3, y = letters[1:3])

H$another_key <- matrix(rbinom(9, 1, 0.5) > 0, nrow = 3)

H["error"] <- 42
#Error in H["error"] <- 42 :
object of type 'environment' is not subsettable
```

の、リテラルのわりにをできるため、Rののファセットとに、のメソッド `object[[key]] <- value` が2の `object$key <- value` `object$key <- value` えばのでは `key2`。

にハッシュマップの他に、 `environment` オブジェクトはキーをしません。のキーにキーとのペアをしようとすると、にされたがきえられます。

```
H[["key3"]] <- "original value"

H[["key3"]] <- "new value"

H[["key3"]]
[1] "new value"
```

## キールックアップ

に、は `[` または `$` でアクセスできますが、 `[`

```
H[["key"]]
[1] 1.630631

H[[key2]] ## assuming key2 <- "xyz"
x y
1 1 a
2 2 b
3 3 c

H$another_key
[,1] [,2] [,3]
[1,] TRUE TRUE TRUE
[2,] FALSE FALSE FALSE
[3,] TRUE TRUE TRUE

H[1]
Error in H[1] : object of type 'environment' is not subsettable
```

## ハッシュマップの

の `environment` であるため、ハッシュマップはなでできます。

```
names(H)
[1] "another_key" "xyz" "key" "key3"

ls(H)
[1] "another_key" "key" "key3" "xyz"

str(H)
#<environment: 0x7828228>

ls.str(H)
another_key : logi [1:3, 1:3] TRUE FALSE TRUE TRUE FALSE TRUE ...
key : num 1.63
```

```
key3 : chr "new value"
xyz : 'data.frame': 3 obs. of 2 variables:
$ x: int 1 2 3
$ y: chr "a" "b" "c"
```

は`rm`をしてできます。

```
rm(list = c("key", "key3"), envir = H)

ls.str(H)
another_key : logi [1:3, 1:3] TRUE FALSE TRUE TRUE FALSE TRUE ...
xyz : 'data.frame': 3 obs. of 2 variables:
$ x: int 1 2 3
$ y: chr "a" "b" "c"
```

`environment` オブジェクトをハッシュテーブルとしてするなの1つは、の`environment`であっても、あらゆるオブジェクトをととしてできることです。

```
H2 <- new.env()

H2[["a"]] <- LETTERS
H2[["b"]] <- as.list(x = 1:5, y = matrix(rnorm(10), 2))
H2[["c"]] <- head(mtcars, 3)
H2[["d"]] <- Sys.Date()
H2[["e"]] <- Sys.time()
H2[["f"]] <- (function() {
 H3 <- new.env()
 for (i in seq_along(names(H2))) {
 H3[[names(H2)[i]]] <- H2[[names(H2)[i]]]
 }
 H3
})()

ls.str(H2)
a : chr [1:26] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" ...
b : List of 5
$: int 1
$: int 2
$: int 3
$: int 4
$: int 5
c : 'data.frame': 3 obs. of 11 variables:
$ mpg : num 21 21 22.8
$ cyl : num 6 6 4
$ disp: num 160 160 108
$ hp : num 110 110 93
$ drat: num 3.9 3.9 3.85
$ wt : num 2.62 2.88 2.32
$ qsec: num 16.5 17 18.6
$ vs : num 0 0 1
$ am : num 1 1 1
$ gear: num 4 4 4
$ carb: num 4 4 1
d : Date[1:1], format: "2016-08-03"
e : POSIXct[1:1], format: "2016-08-03 19:25:14"
f : <environment: 0x91a7cb8>
```

```
ls.str(H2$f)
a : chr [1:26] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" ...
b : List of 5
$: int 1
$: int 2
$: int 3
$: int 4
$: int 5
c : 'data.frame': 3 obs. of 11 variables:
$ mpg : num 21 21 22.8
$ cyl : num 6 6 4
$ disp: num 160 160 108
$ hp : num 110 110 93
$ drat: num 3.9 3.9 3.85
$ wt : num 2.62 2.88 2.32
$ qsec: num 16.5 17 18.6
$ vs : num 0 0 1
$ am : num 1 1 1
$ gear: num 4 4 4
$ carb: num 4 4 1
d : Date[1:1], format: "2016-08-03"
e : POSIXct[1:1], format: "2016-08-03 19:25:14"
```

environment オブジェクトをハッシュマップとしてするのは、Rのくのと異なり、の/でベクトルがサポートされていないことです。

```
names(H2)
#[1] "a" "b" "c" "d" "e" "f"

H2[[c("a", "b")]]
#Error in H2[[c("a", "b")]] :
wrong arguments for subsetting an environment

Keys <- c("a", "b")
H2[[Keys]]
#Error in H2[[Keys]] : wrong arguments for subsetting an environment
```

オブジェクトにされているデータののにて、 `vapply` または `list2env` をしてのをにりてることができます。

```
E1 <- new.env()
invisible({
 vapply(letters, function(x) {
 E1[[x]] <- rnorm(1)
 logical(0)
 }, FUN.VALUE = logical(0))
})

all.equal(sort(names(E1)), letters)
#[1] TRUE

Keys <- letters
E2 <- list2env(
 setNames(
 as.list(rnorm(26)),
 nm = Keys),
```

```

 enviro = NULL,
 hash = TRUE
)

all.equal(sort(names(E2)), letters)
#[1] TRUE

```

のどちらにもではありませんが、キーとのペアのがいは `for` ループなどをするよりもましいがあります。

## パッケージハッシュ

**ハッシュパッケージ**はRでハッシュをしますが、とみりの**のタイミングというでは**、をハッシュとしてすることとはなりません。このドキュメントは、そのを、のでのサンプルタイミングコードをしています。のRコードでは、ハッシュがなであるというされたケースはありません。

してください

```

Generic unique string generator
unique_strings <- function(n){
 string_i <- 1
 string_len <- 1
 ans <- character(n)
 chars <- c(letters,LETTERS)
 new_strings <- function(len,pfx){
 for(i in 1:length(chars)){
 if (len == 1){
 ans[string_i] <- paste(pfx,chars[i],sep='')
 string_i <- string_i + 1
 } else {
 new_strings(len-1,pfx=paste(pfx,chars[i],sep=''))
 }
 if (string_i > n) return ()
 }
 }
 while(string_i <= n){
 new_strings(string_len,'')
 string_len <- string_len + 1
 }
 sample(ans)
}

Generate timings using an environment
timingsEnv <- plyr::adply(2^(10:15),.mar=1,.fun=function(i){
 strings <- unique_strings(i)
 ht1 <- new.env(hash=TRUE)
 lapply(strings, function(s){ ht1[[s]] <- 0L})
 data.frame(
 size=c(i,i),
 seconds=c(
 system.time(for (j in 1:i) ht1[[strings[j]]]==0L)[3]),
 type = c('1_hashedEnv')
)
})

timingsHash <- plyr::adply(2^(10:15),.mar=1,.fun=function(i){
 strings <- unique_strings(i)

```



```

ht <- hash::hash()
lapply(strings, function(s) ht[[s]] <- 0L)
data.frame(
 size=c(i,i),
 seconds=c(
 system.time(for (j in 1:i) ht[[strings[j]]]==0L)[3]),
 type = c('3_stringHash')
)
})

```

## パッケージ `listenv`

`package:listenv` にリストのようなインターフェースをしますが、ハッシュのようなためのにするパフォーマンスはハッシュではいずす。ただし、がのは、にかなりになります。しかし、の、例えば `package:future`。そののためにこのパッケージをカバーすることは、ののをえています。ただし、ここでされるタイミグコードは、write タイミグのための `package:hash` のとみわけてできます。

```

timingsListEnv <- plyr::adply(2^(10:15), .mar=1, .fun=function(i) {
 strings <- unique_strings(i)
 le <- listenv::listenv()
 lapply(strings, function(s) le[[s]] <- 0L)
 data.frame(
 size=c(i,i),
 seconds=c(
 system.time(for (k in 1:i) le[[k]]==0L)[3]),
 type = c('2_numericListEnv')
)
})

```

オンラインでハッシュマップをむ <https://riptutorial.com/ja/r/topic/5179/ハッシュマップ>

## 64: ヒートマップとヒートマップ.2

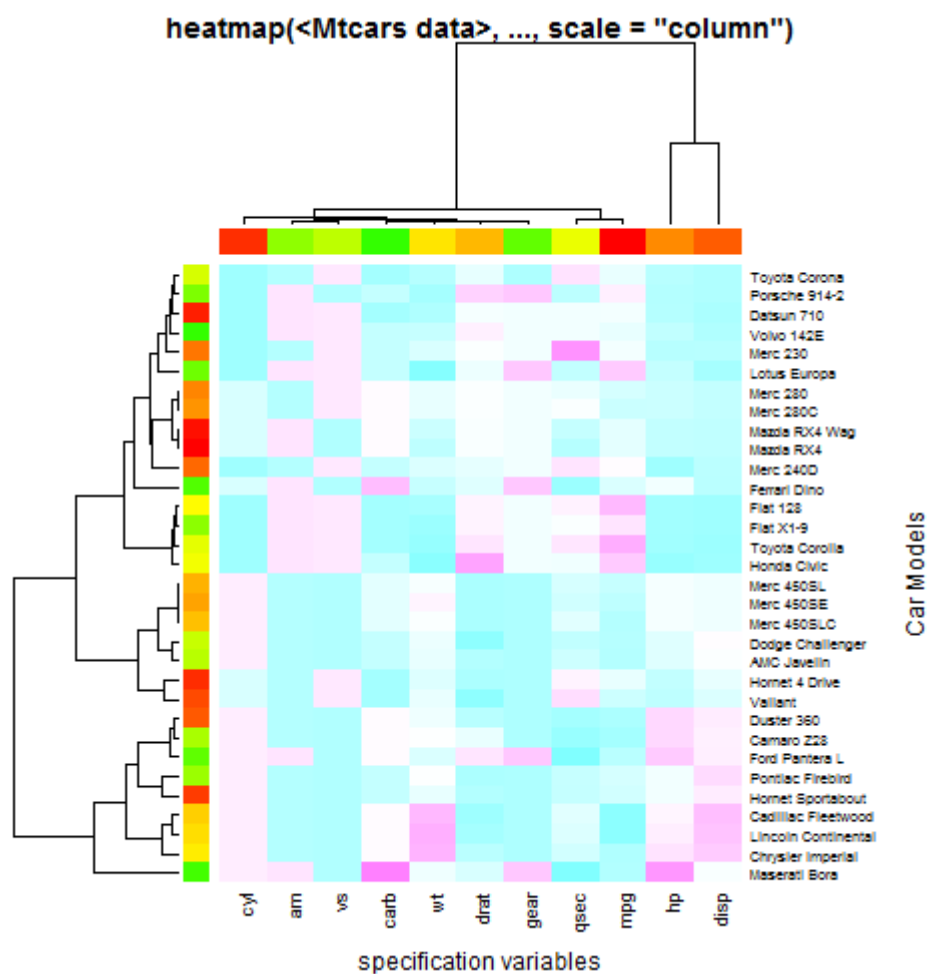
### Examples

からの

## ::ヒートマップ

1ない

```
require(graphics); require(grDevices)
x <- as.matrix(mtcars)
rc <- rainbow(nrow(x), start = 0, end = .3)
cc <- rainbow(ncol(x), start = 0, end = .3)
hv <- heatmap(x, col = cm.colors(256), scale = "column",
 RowSideColors = rc, ColSideColors = cc, margins = c(5,10),
 xlab = "specification variables", ylab = "Car Models",
 main = "heatmap(<Mtcars data>, ..., scale = \"column\")")
```

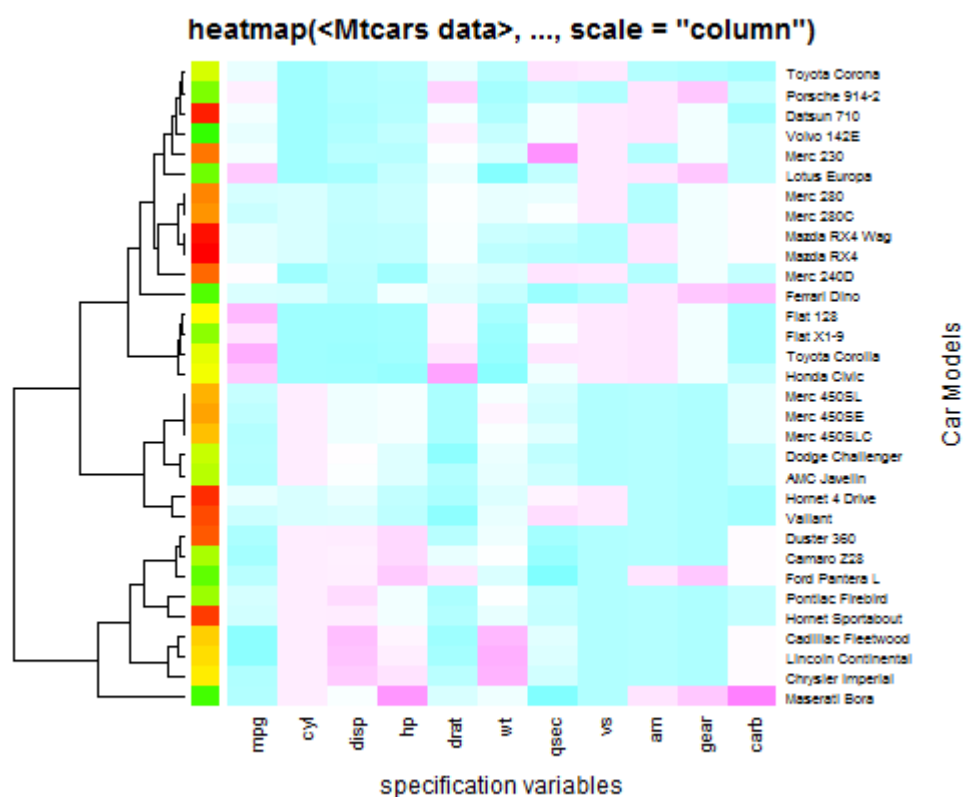


```
utils::str(hv) # the two re-ordering index vectors
```

```
List of 4
$ rowInd: int [1:32] 31 17 16 15 5 25 29 24 7 6 ...
$ colInd: int [1:11] 2 9 8 11 6 5 10 7 1 4 ...
$ Rowv : NULL
$ Colv : NULL
```

## 2 デンドログラムはまったくありません

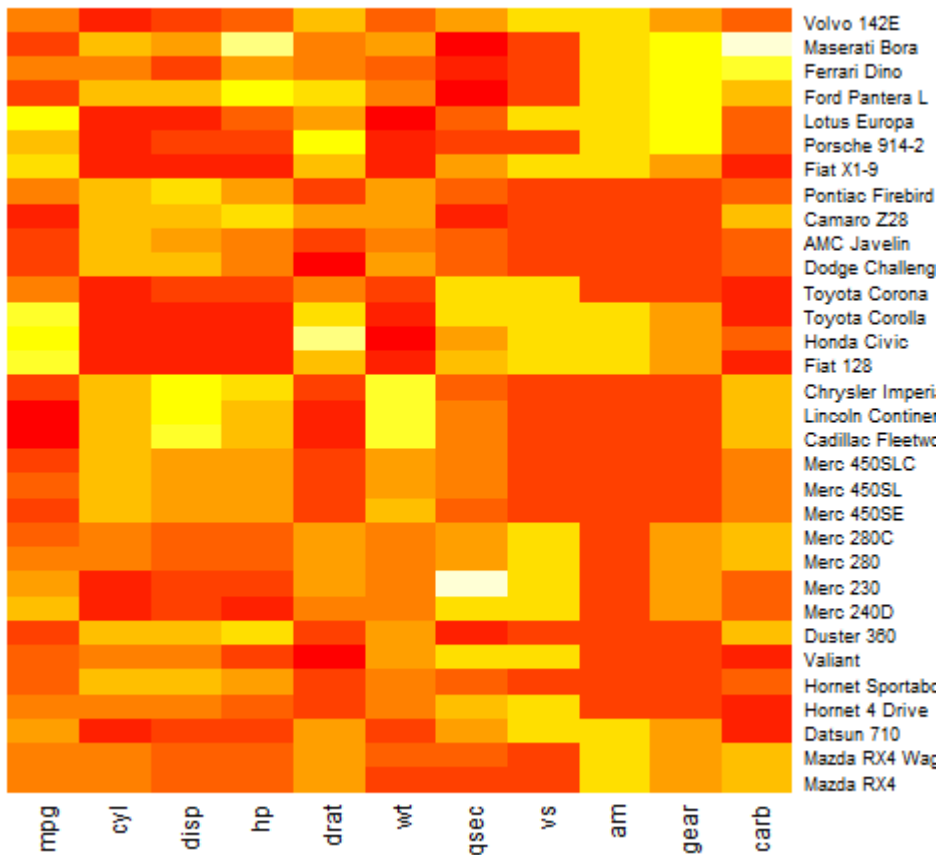
```
heatmap(x, Colv = NA, col = cm.colors(256), scale = "column",
 RowSideColors = rc, margins = c(5,10),
 xlab = "specification variables", ylab = "Car Models",
 main = "heatmap(<Mtcars data>, ..., scale = \"column\")")
```



## 3 「もなし」

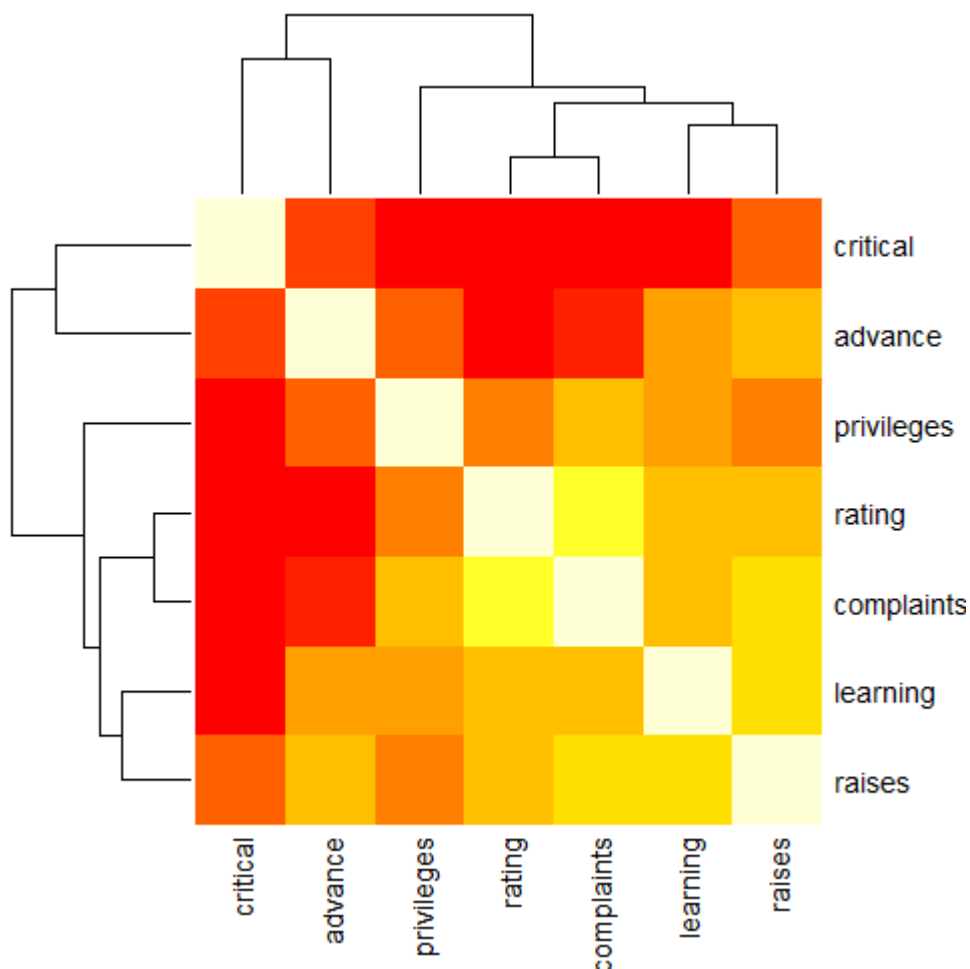
```
heatmap(x, Rowv = NA, Colv = NA, scale = "column",
 main = "heatmap(*, NA, NA) ~ image(t(x))")
```

heatmap(\*, NA, NA) ~= image(t(x))



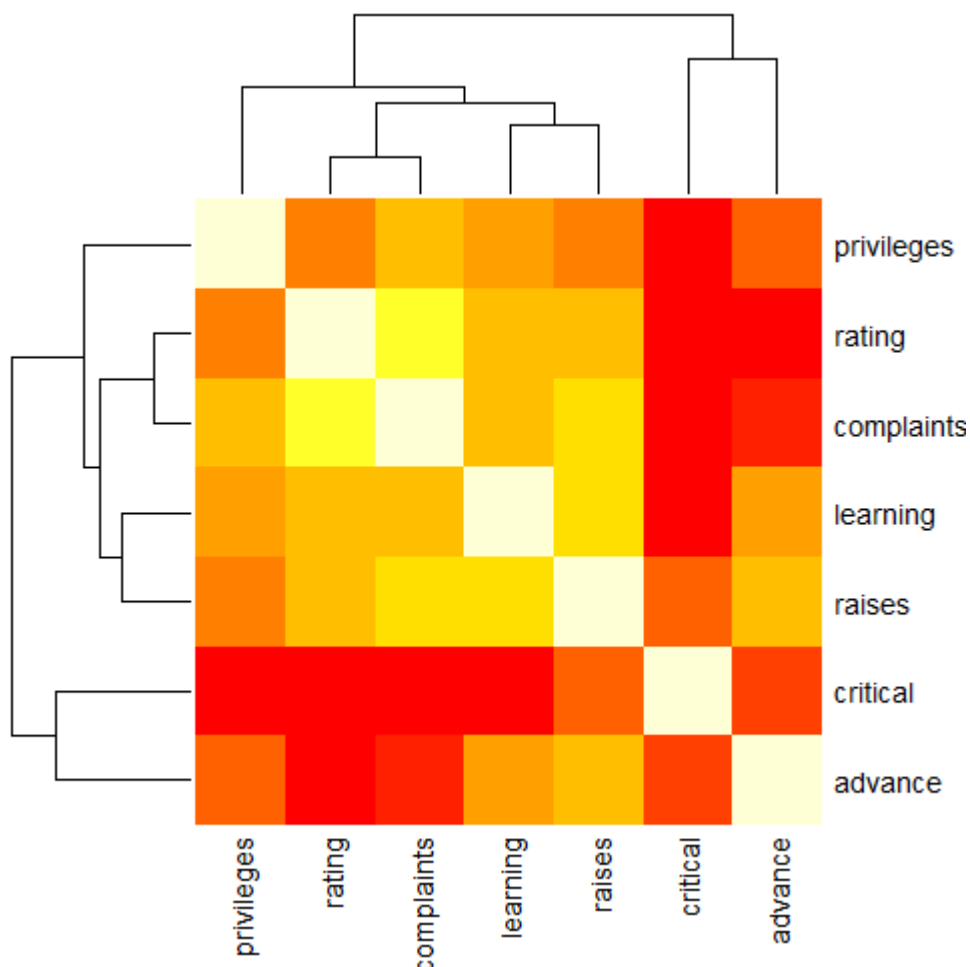
## 4reorderを

```
round(Ca <- cor(attendance), 2)
rating complaints privileges learning raises critical advance
rating 1.00 0.83 0.43 0.62 0.59 0.16 0.16
complaints 0.83 1.00 0.56 0.60 0.67 0.19 0.22
privileges 0.43 0.56 1.00 0.49 0.45 0.15 0.34
learning 0.62 0.60 0.49 1.00 0.64 0.12 0.53
raises 0.59 0.67 0.45 0.64 1.00 0.38 0.57
critical 0.16 0.19 0.15 0.12 0.38 1.00 0.28
advance 0.16 0.22 0.34 0.53 0.57 0.28 1.00
symnum(Ca) # simple graphic
rt cm p l rs cr a
rating 1
complaints + 1
privileges . . 1
learning , . . 1
raises . , . , 1
critical . . . 1
advance 1
attr("legend")
[1] 0 ' ' 0.3 ' ' 0.6 ' ' 0.8 '+' 0.9 '*' 0.95 'B' 1
heatmap(Ca, symm = TRUE, margins = c(6,6))
```



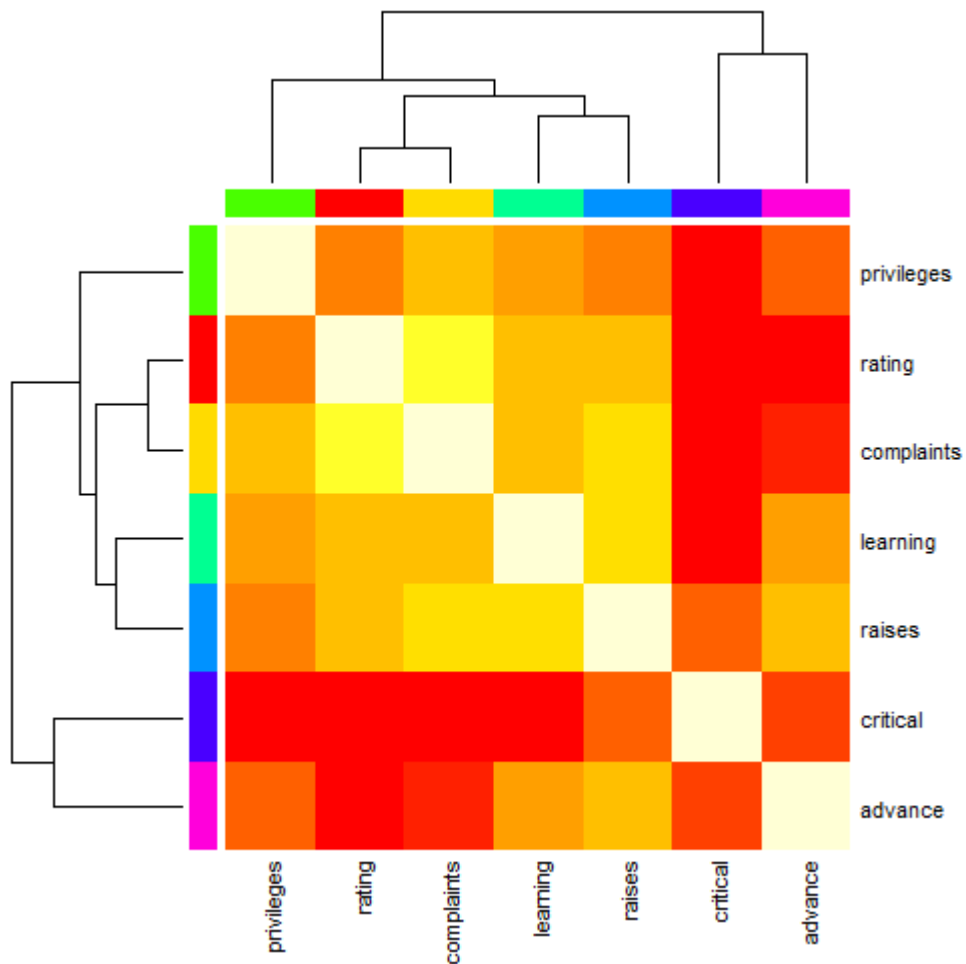
## 5 NO reorder

```
heatmap(Ca, Rowv = FALSE, symm = TRUE, margins = c(6,6))
```



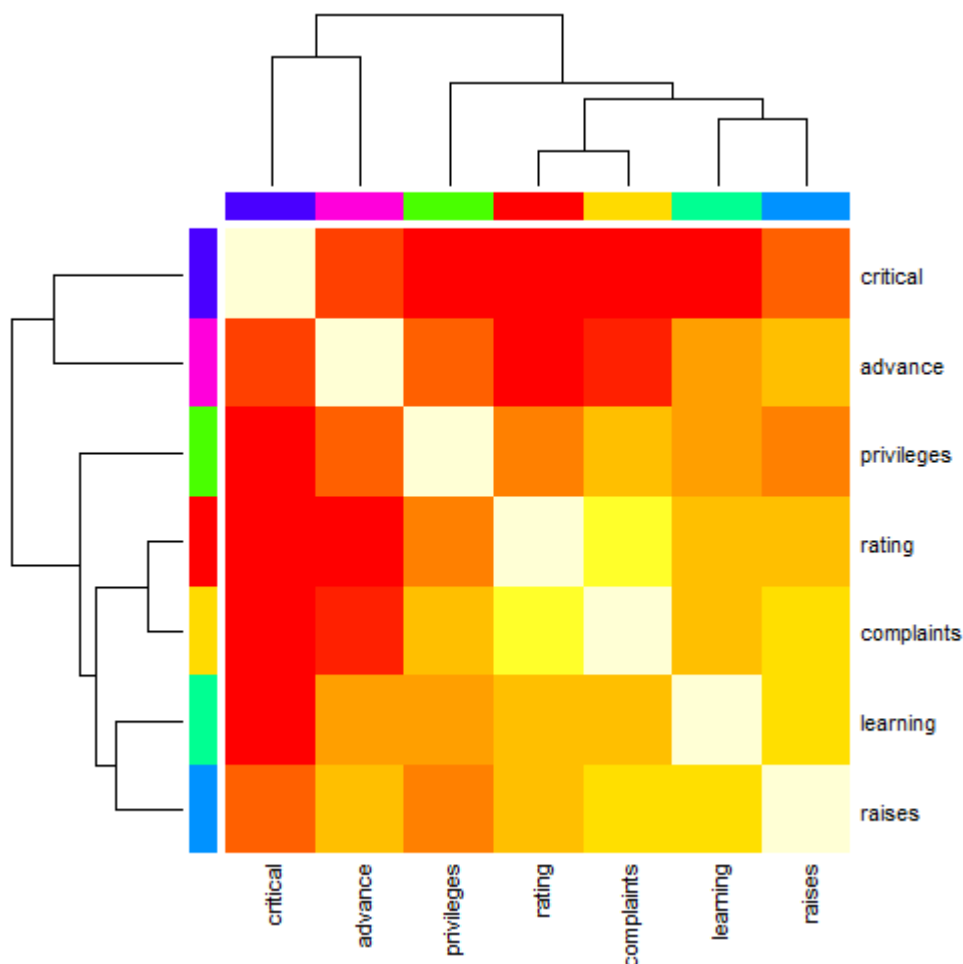
## 6 カラーバーでしに、なし

```
cc <- rainbow(nrow(Ca))
heatmap(Ca, Rowv = FALSE, symm = TRUE, RowSideColors = cc, ColSideColors = cc,
 margins = c(6,6))
```



## 7 わずかになカラーバーきの

```
heatmap(Ca, symm = TRUE, RowSideColors = cc, ColSideColors = cc,
 margins = c(6,6))
```



8のクラスタリングではなく、corにつくをする

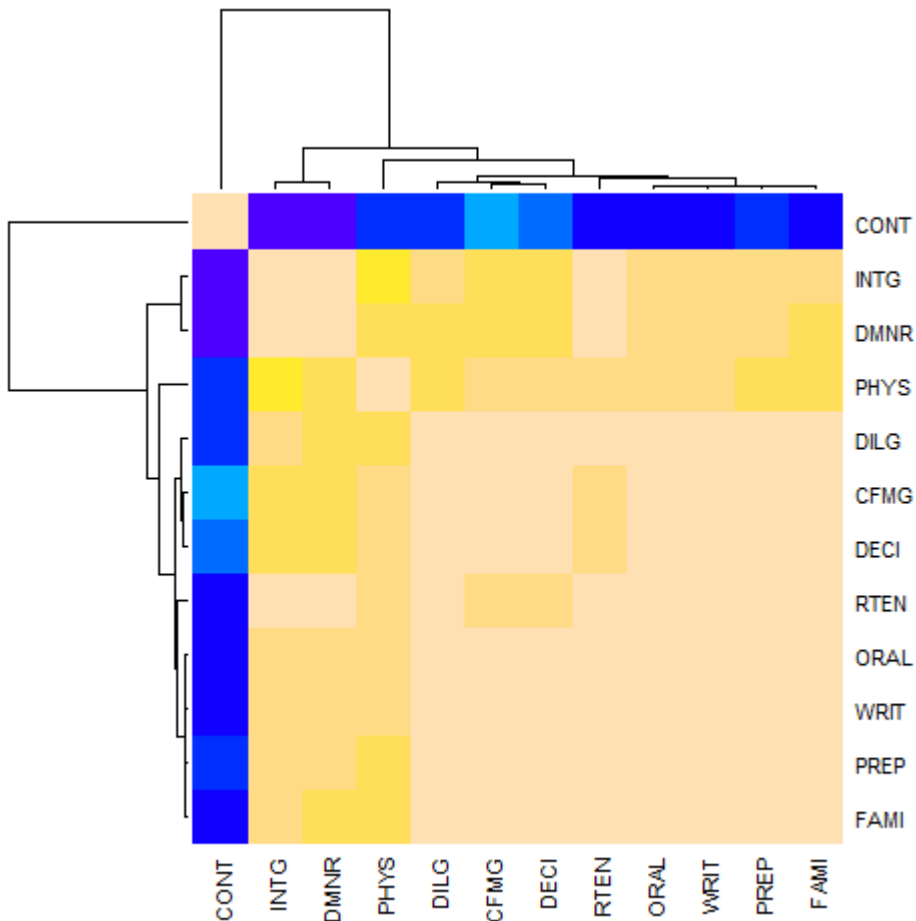
```

symnum(cU <- cor(USJudgeRatings))
CO I DM DI CF DE PR F O W PH R
CONT 1
INTG 1
DMNR B 1
DILG + + 1
CFMG + + B 1
DECI + + B B 1
PREP + + B B B 1
FAMI + + B * * B 1
ORAL * * B B * B B 1
WRIT * + B * * B B B 1
PHYS , , + + + + + + 1
RTEN * * * * * B * B B * 1
attr("legend")
[1] 0 \ ' 0.3 \ ' 0.6 \ , ' 0.8 \ + ' 0.9 \ * ' 0.95 \ B ' 1

hU <- heatmap(cU, Rowv = FALSE, symm = TRUE, col = topo.colors(16),
 distfun = function(c) as.dist(1 - c), keep.dendro = TRUE)

```





```
The Correlation matrix with same reordering:
round(100 * cU[hU[[1]], hU[[2]]])
CONT INTG DMNR PHYS DILG CFMG DECI RTEN ORAL WRIT PREP FAMI
CONT 100 -13 -15 5 1 14 9 -3 -1 -4 1 -3
INTG -13 100 96 74 87 81 80 94 91 91 88 87
DMNR -15 96 100 79 84 81 80 94 91 89 86 84
PHYS 5 74 79 100 81 88 87 91 89 86 85 84
DILG 1 87 84 81 100 96 96 93 95 96 98 96
CFMG 14 81 81 88 96 100 98 93 95 94 96 94
DECI 9 80 80 87 96 98 100 92 95 95 96 94
RTEN -3 94 94 91 93 93 92 100 98 97 95 94
ORAL -1 91 91 89 95 95 95 98 100 99 98 98
WRIT -4 91 89 86 96 94 95 97 99 100 99 99
PREP 1 88 86 85 98 96 96 95 98 99 100 99
FAMI -3 87 84 84 96 94 94 94 98 99 99 100
```

```
The column dendrogram:
utils::str(hU$Colv)
--[dendrogram w/ 2 branches and 12 members at h = 1.15]
|--leaf "CONT"
`--[dendrogram w/ 2 branches and 11 members at h = 0.258]
|--[dendrogram w/ 2 branches and 2 members at h = 0.0354]
| |--leaf "INTG"
| `--leaf "DMNR"
`--[dendrogram w/ 2 branches and 9 members at h = 0.187]
|--leaf "PHYS"
`--[dendrogram w/ 2 branches and 8 members at h = 0.075]
|--[dendrogram w/ 2 branches and 3 members at h = 0.0438]
| |--leaf "DILG"
```

```
| `--[dendrogram w/ 2 branches and 2 members at h = 0.0189]
| |--leaf "CFMG"
| `--leaf "DECI"
`--[dendrogram w/ 2 branches and 5 members at h = 0.0584]
|--leaf "RTEN"
`--[dendrogram w/ 2 branches and 4 members at h = 0.0187]
| |--[dendrogram w/ 2 branches and 2 members at h = 0.00657]
| | |--leaf "ORAL"
| | `--leaf "WRIT"
| `--[dendrogram w/ 2 branches and 2 members at h = 0.0101]
| |--leaf "PREP"
| `--leaf "FAMI"
```

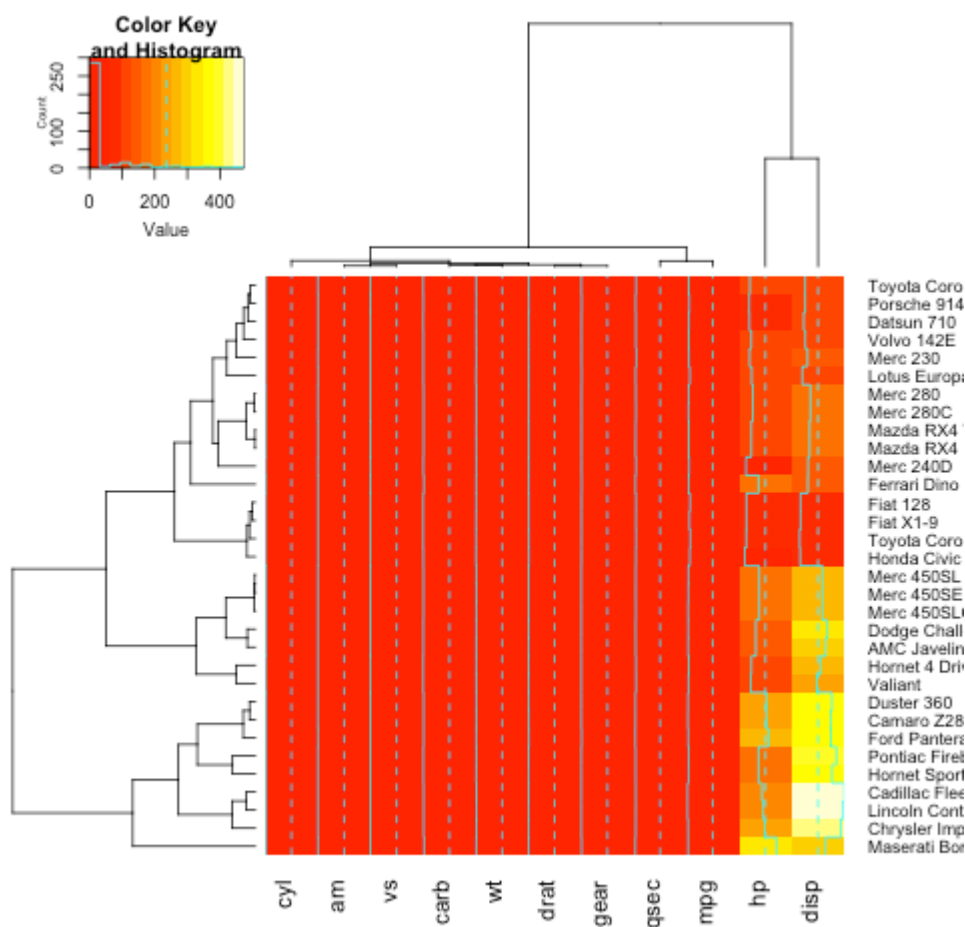
## ヒートマップのパラメータのチューニング2

えられた

```
x <- as.matrix(mtcars)
```

つはすることができます heatmap.2 のされたバージョン- heatmap 、 のライブラリをロードすることによって、

```
require(gplots)
heatmap.2(x)
```

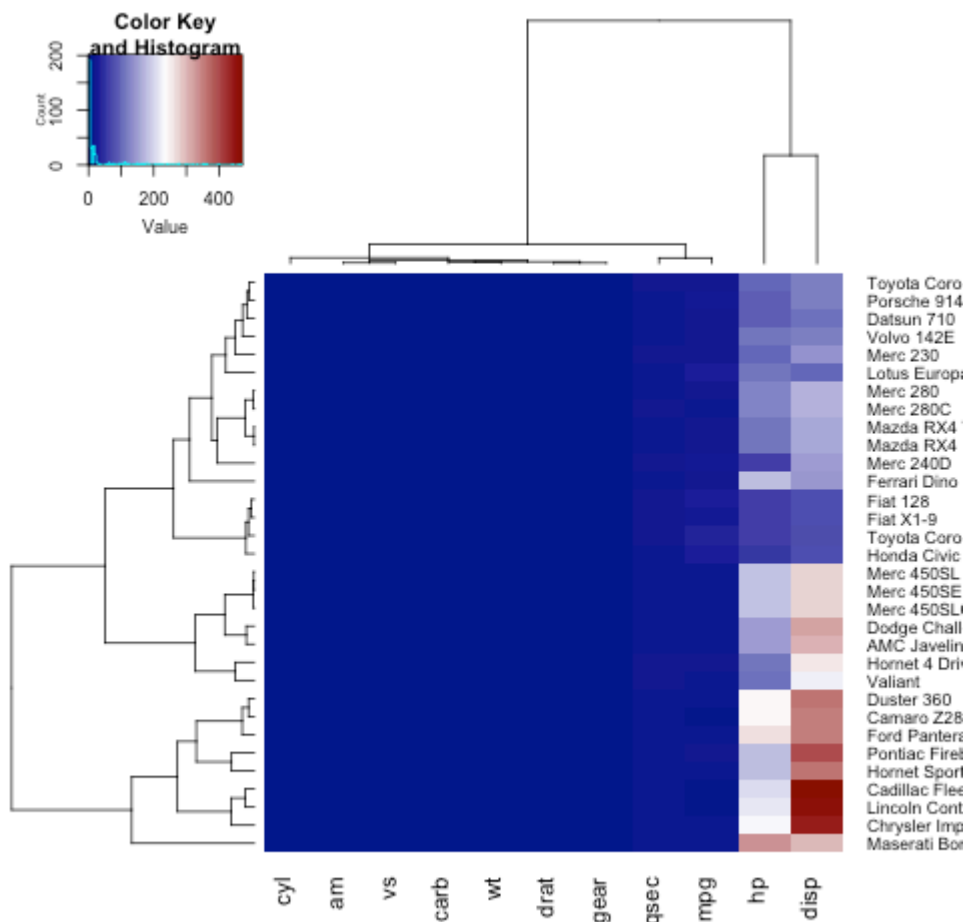


ヒートマップにタイトル、xまたはyラベルをするには、 `main`、 `xlab`、 `ylab` をするがあります。

```
heatmap.2(x, main = "My main title: Overview of car features", xlab="Car features", ylab = "Car brands")
```

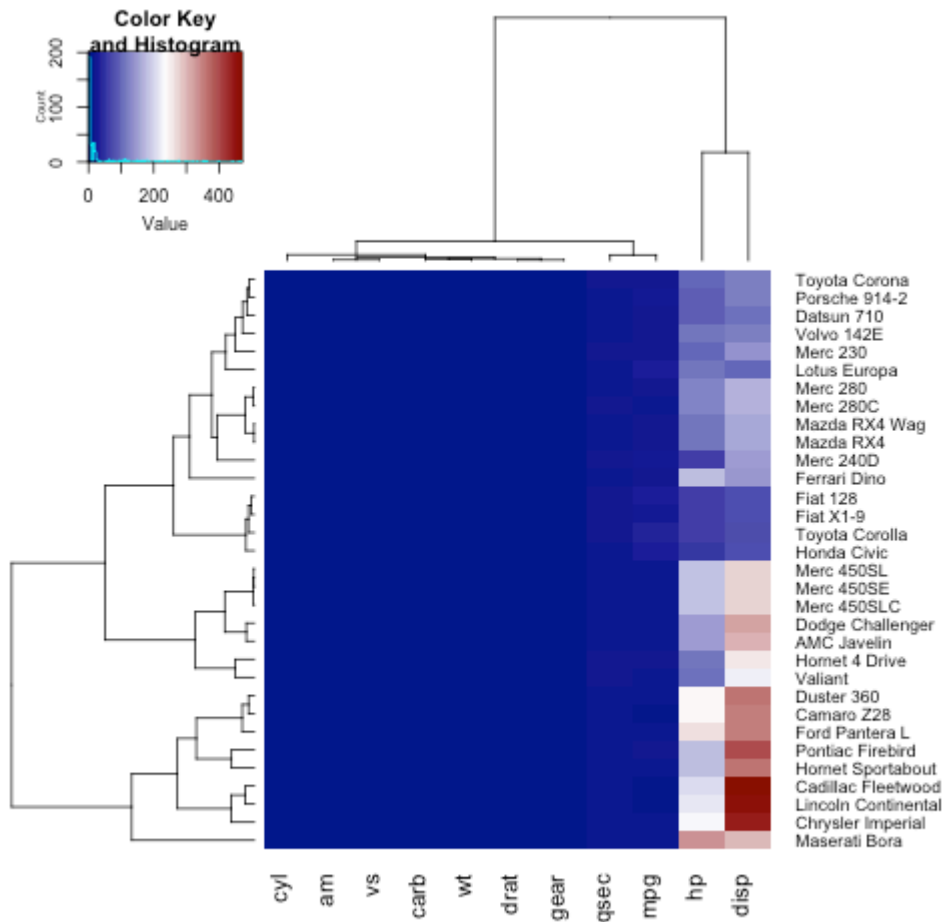
ヒートマップにのカラーパレットをするは、 `colorRampPalette` をして `col` パラメータをできます。

```
heatmap.2(x, trace="none", key=TRUE, Colv=FALSE, dendrogram = "row", col = colorRampPalette(c("darkblue", "white", "darkred"))(100))
```

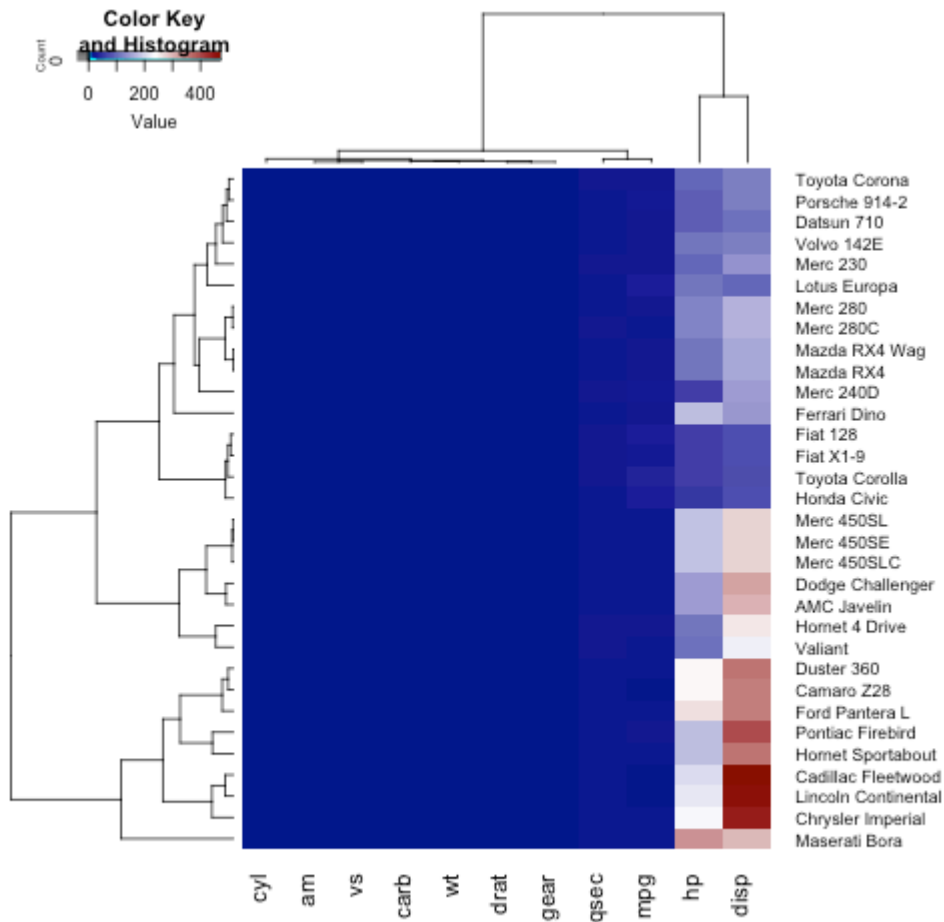


このとおり、yのラベルはにまらない。これをするために、ユーザーは `margins` パラメータをできます

```
heatmap.2(x, trace="none", key=TRUE, col = colorRampPalette(c("darkblue", "white", "darkred"))(100), margins=c(5,8))
```

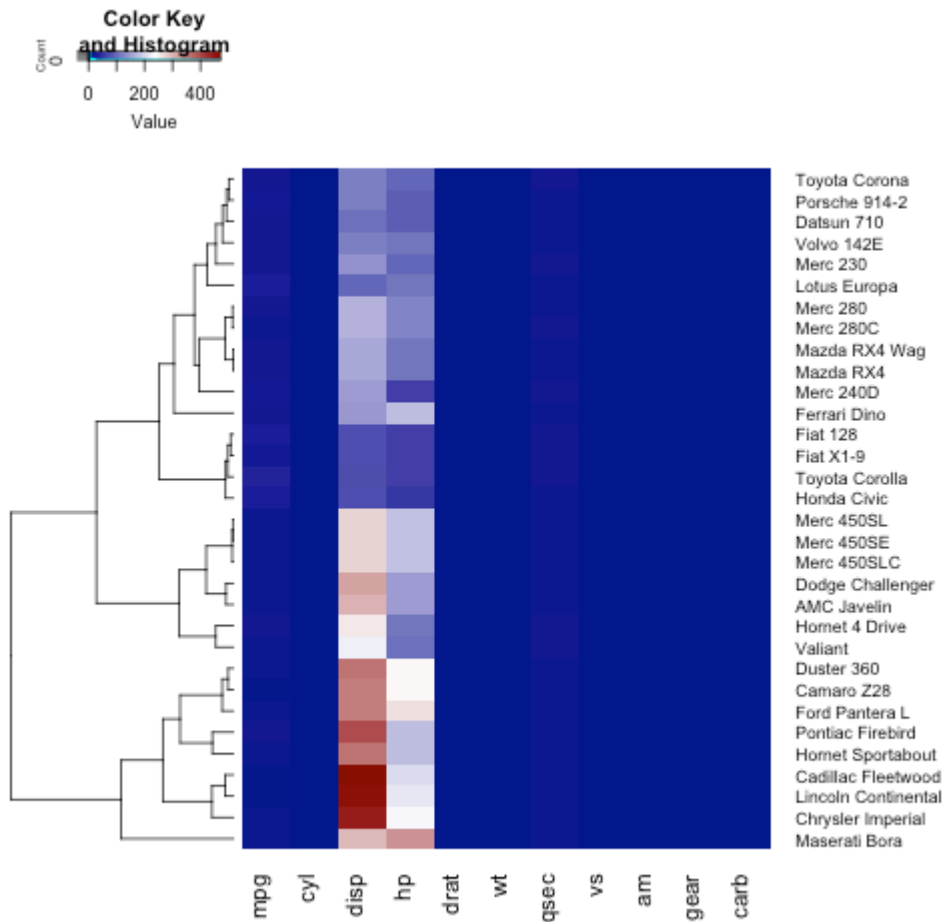


さらに、`lhei`と`lwid`する`lhei`、`lwid`のセクションのサイズキーヒストグラム、、ヒートマップを  
 することができます。



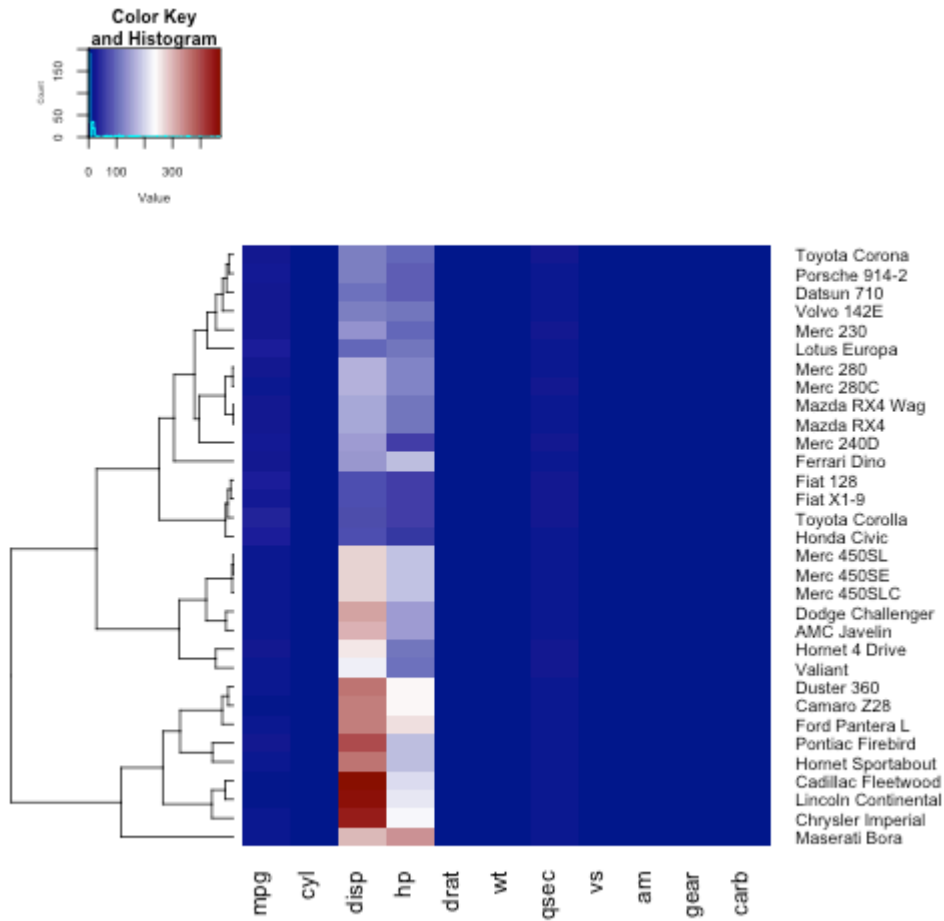
またはの `Colv=FALSE` だけをしたい、 `Colv=FALSE` または `Rowv=FALSE` をして、 `dendrogram` パラメータをするがあります。

```
heatmap.2(x, trace="none", key=TRUE, Colv=FALSE, dendrogram = "row", col =
colorRampPalette(c("darkblue","white","darkred"))(100), margins=c(5,8), lwid = c(5,15), lhei =
c(3,15))
```



のタイトル、ラベル、のフォントサイズをするには、ユーザーは`par`リストに`cex.main`, `cex.lab`, `cex.axis`をするがあります。

```
par(cex.main=1, cex.lab=0.7, cex.axis=0.7)
heatmap.2(x, trace="none", key=TRUE, Colv=FALSE, dendrogram = "row", col =
colorRampPalette(c("darkblue","white","darkred"))(100), margins=c(5,8), lwid = c(5,15), lhei =
c(5,15))
```



オンラインでヒートマップとヒートマップ.2をむ <https://riptutorial.com/ja/r/topic/4814/ヒートマップとヒートマップ-2>

## 65: フーリエと

フーリエは、のを、そののまたはラウドネスとしてどのようにできるかとに、それをするにする。のフーリエはのであり、そののはのにするのをし、のはそのにおけるのオフセットである。

フーリエはののとはれます。フーリエというは、とをにけるのをす。フーリエはのにされず、されたをするために、ののはにとばれる。ののあるくのにして、これをするをすることができる。のフーリエともばれるフーリエは、ののをするためになるのすべてのをみわせる。

あるドメインまたはでされるは、のドメインでするをちますが、これはにはするがです。におけるのはによるにするので、いくつかのはでのがです。また、におけるみみは、におけるのにする。には、これは、にされるフィルタのようなのシステムが、でのとしてにされることをする。したがって、なは、をにし、のをし、そのをにすることによってされることがい。

は、とのをにすることであり、なのほぼすべてのにくわっています。

においてされたは、にわたってがっているフーリエをし、もまたである。は、とにおける、また例えば、をすのにおいてになガウスであり、なによってフーリエのもとにむ。Joseph Fourierはのでをしました。ここでガウスはのとしてれます。

フーリエは、なりーマンとしてにすることができ、これをするが、このは、よりされたをとするくのアプリケーションにはしていない。

例えば、くのなアプリケーションでは、であるかのようににうことができるDiracデルタをしますが、にはにされたがです。フーリエはまた、ユークリッドのいくつかののにすることができ、3のを3のまたは4のおよびのにる。

このえは、のにおいてフーリエをになものにし、においても、またはのとしてののを、またにはそのとしてののをすことができる。に、フーリエがなはであり、によってはベクトルである。にすることができるのは、グループのであり、 $\mathbb{R}$ は $\mathbb{R}_n$ えてグループとしてられるののフーリエにえて、フーリエDTFT、グループ $=\mathbb{Z}$ 、フーリエDFT、グループ $=\mathbb{Z}_{\text{mod } N}$ とフーリエまたはフーリエグループ $=S_1$ 、 $\approx$ じた、がされている。は、なをするためににされています。フーリエFFTは、DFTをするためのアルゴリズムです。

## Examples

### フーリエ

ジョセフフーリエは、のがなのでされることをした。このはフーリエとはれます。フーリエは、システムがであるのみされます。いくらかのがえられてもがじままであるような、あるオーバーフローがある、がにり、およびをる。

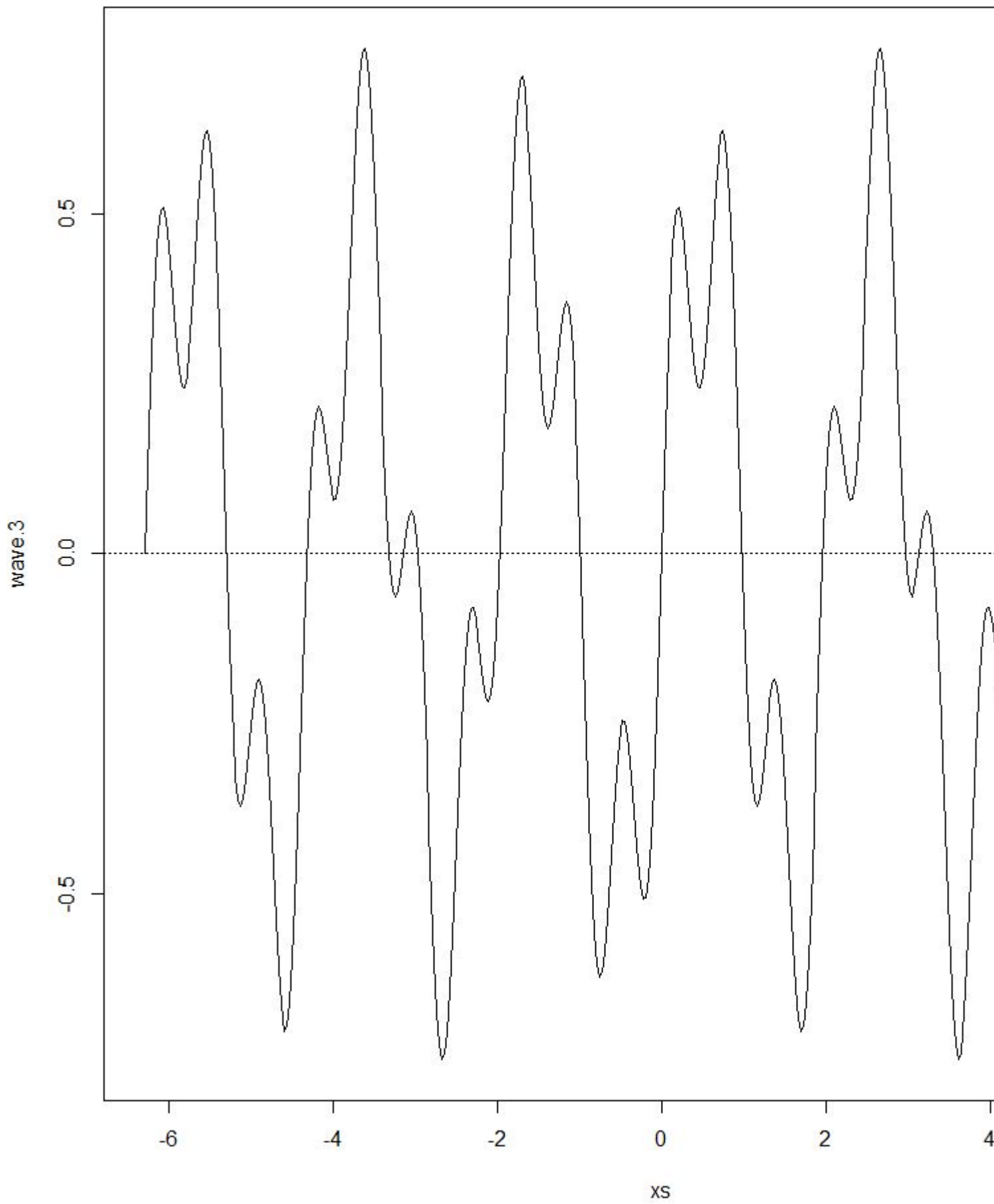
```
Sine waves
xs <- seq(-2*pi, 2*pi, pi/100)
```



```
wave.1 <- sin(3*xs)
wave.2 <- sin(10*xs)
par(mfrow = c(1, 2))
plot(xs, wave.1, type="l", ylim=c(-1, 1)); abline(h=0, lty=3)
plot(xs, wave.2, type="l", ylim=c(-1, 1)); abline(h=0, lty=3)

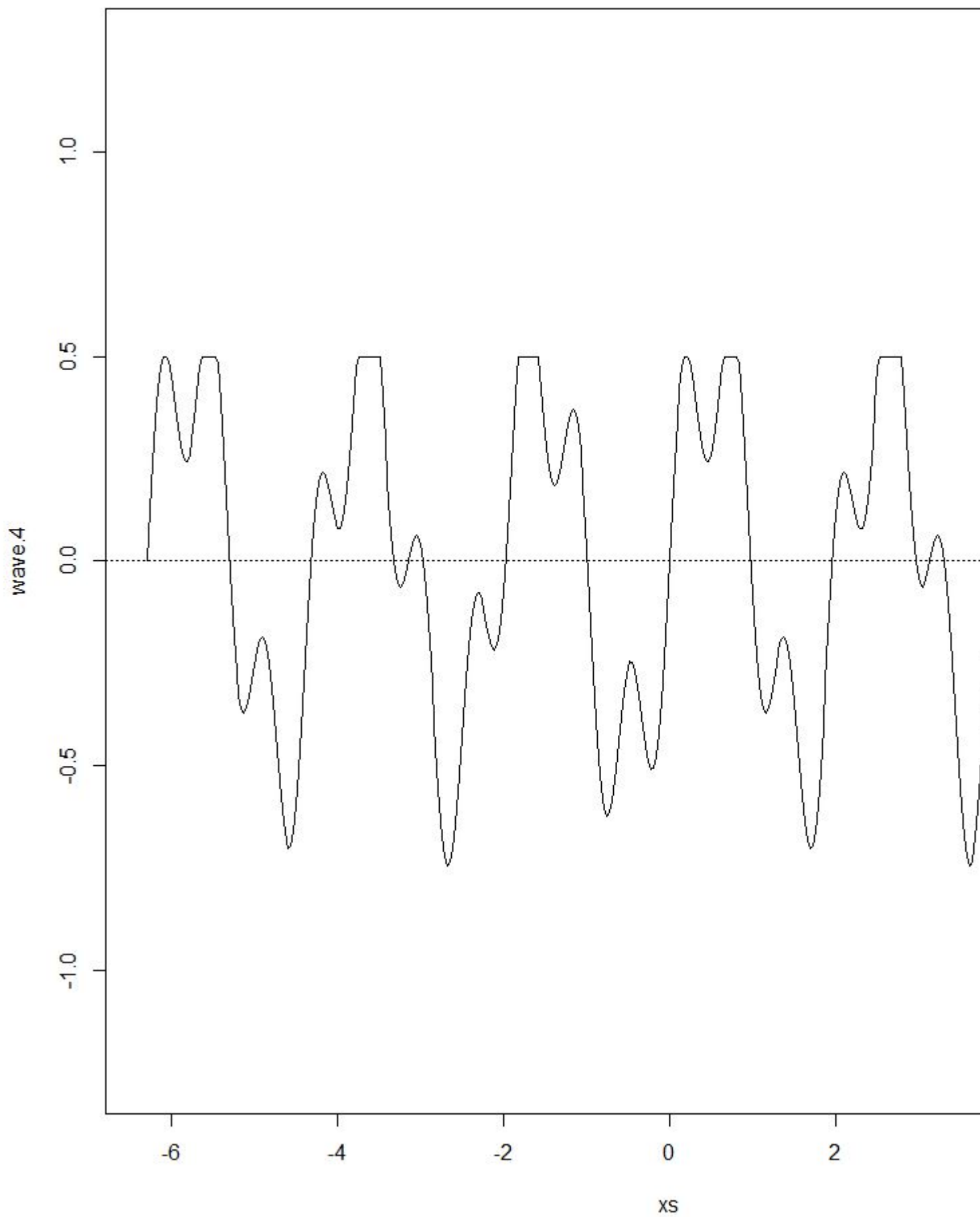
Complex Wave
wave.3 <- 0.5 * wave.1 + 0.25 * wave.2
plot(xs, wave.3, type="l"); title("Eg complex wave"); abline(h=0, lty=3)
```

### Eg complex wave



```
wave.4 <- wave.3
wave.4[wave.3>0.5] <- 0.5
plot(xs, wave.4, type="l", ylim=c(-1.25, 1.25))
title("overflowed, non-linear complex wave")
abline(h=0, lty=3)
```

### overflowed, non-linear complex wave



また、フーリエは、がであるにのみされ、すなわち、それらはりしパターンをするでないはフーリエによってされる、。なは、しパターンによってされる $f$ と $\lambda$ は、のめとわりのののであり、 $\lambda = v / f_0$ であり、 $v$ はのであるをする。にはまたはがありません。

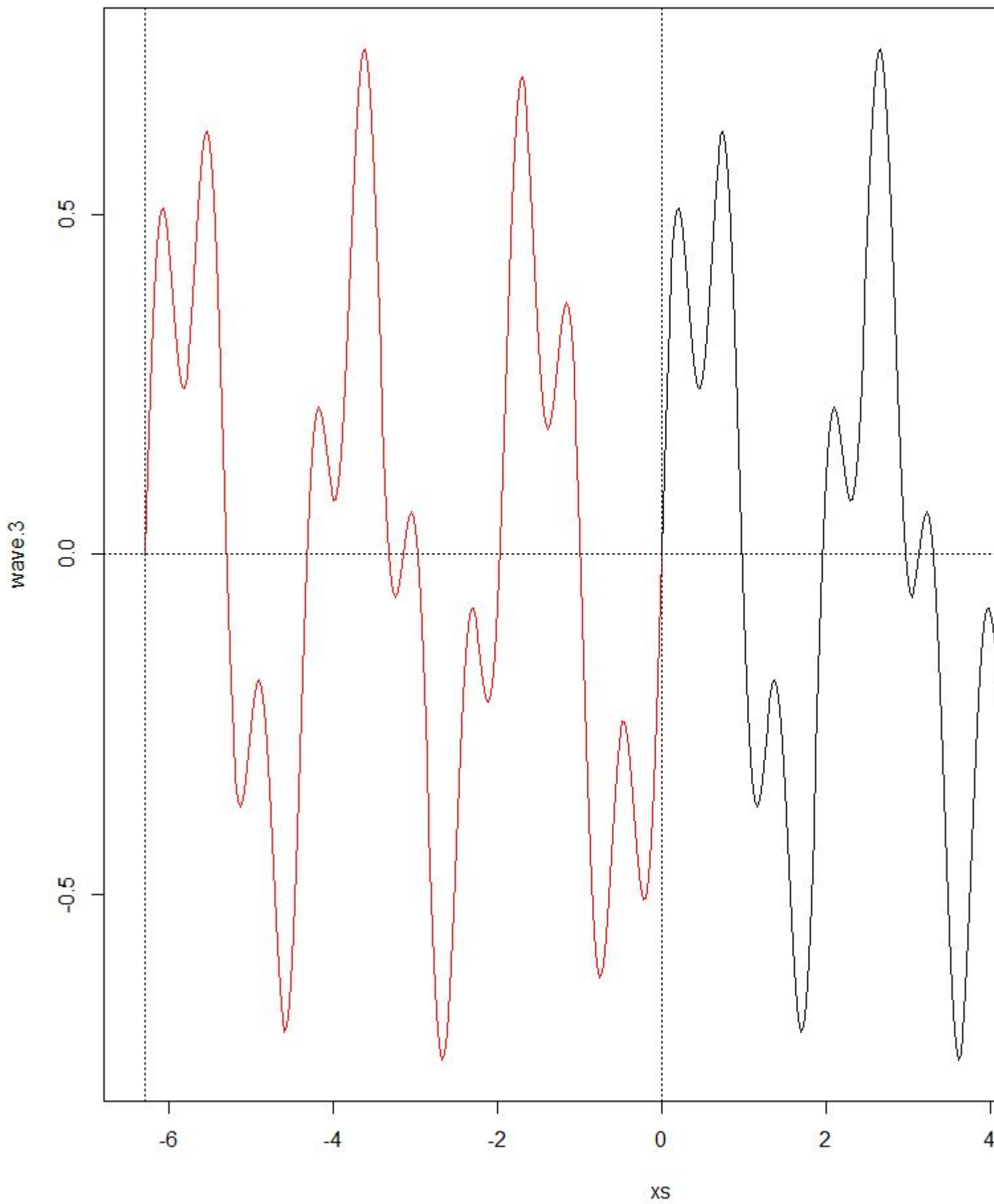
いくつかの

- $T$ は、されたてのサンプルの、のサンプルとのサンプルとのの
- サンプリングレート  $sr$ は、ともばれますにされたサンプルのです。にするために、サンプルのをしくします。このはサンプル  $si$  とばれ、をサンプル  $N$  でったものです。したがって、 $si = TN$
- $f_0$ は $1/T$ です。は、りしパターンのまたはのさである。のでは、は $12\pi$ であった。のは、のでなければならない。  $f_0$ は1とばれ、2は $2 * f_0$ であり、3は $3 * f_0$ である。

```
repeat.xs <- seq(-2*pi,0,pi/100)
wave.3.repeat <- 0.5*sin(3*repeat.xs) + 0.25*sin(10*repeat.xs)
plot(xs,wave.3,type="l")

title("Repeating pattern")
points(repeat.xs,wave.3.repeat,type="l",col="red");
abline(h=0,v=c(-2*pi,0),lty=3)
```

## Repeating pattern



フーリエをえられたをプロットするためのRです

```
plot.fourier <- function(fourier.series, f.0, ts) {
 w <- 2*pi*f.0 trajectory <- sapply(ts, function(t)
fourier.series(t,w))
 plot(ts, trajectory, type="l", xlab="time", ylab="f(t)");
 abline(h=0,lty=3) }
```

オンラインでフーリエとをむ <https://riptutorial.com/ja/r/topic/4139/フーリエと>

## 66: フォールトトレラント/コード

### パラメーター

| パラメーター       |                                                                                                                                                                                     |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| expr         | "try part"がにした、tryCatchはにされたをします。したがって、すべてがうまくいって、がないつまり、またはエラーにreadLinesりがされるのがされます。あなたがにりをexplicitlyするはないことにしてくださいreturn「してみてください」にコードとしては、ラップされていませんとのエラーのためのハンドラのそれとはってをinsided |
| /エラー/<br>etc | にするすべてのにしてハンドラを/します。AFAIUでは、それぞれのハンドラのがそれぞれののクラスにする、あらゆるタイプのとエラーだけでなく、カスタムもsimpleCondition、simpleConditionとフレンドをにハンドラをできますtryCatchのドキュメントの。                                         |
| に            | "try part"のがしたかどうか、またはががあったかどうかにかかわらず、にするがすべてがここにあります。のをするは、でむがあります。そうでなければ、finally = <expression> つまり、「try part」とじロジックとくことができます。                                                   |

#### tryCatch

tryCatchは、がないりexprにするをします。またはエラーです。その、それぞれのにするハンドラをすることで、のりreturn(NA)をすることができます?tryCatchwarningとerror。これらはすでにしますが、tryCatchですることでもありますように。

#### ハンドラののりをする事の

「し」にエラーがあったにNAをすようにしたので、yの3のはNAです。りとしてNULLをした、lapplyはNULLりをに「/」するため、yのさは3ではなく2になりNULL。また、あなたがしてなりをしなごreturn、ハンドラがすNULLすなわち、エラーまたはの。

#### 「ましくない」メッセージ

たちの3のときurlsベクトルは、のを、々はエラーがしているというにえて、のをるreadLinesそれがのエラーでするにをしたをくことができないとをいす

```
Warning message:
In file(con, "r") : cannot open file 'I'm no URL': No such file or directory
```



エラーはよりも「っている」ので、このケースではににありません。したがって、`readLines` `warn = FALSE`を `warn = FALSE`が、それはがないようです。をするのは、

```
suppressWarnings(readLines(con = url))
```

のわりに

```
readLines(con = url, warn = FALSE)
```

## Examples

`tryCatch`をすると、

えられたURLからHTMLコードをみむなバージョンのをしています。たちは、かがっているエラーか、それほとしていないするにしたいというです。エラーとのなはです

`tryCatch`をした

```
readUrl <- function(url) {
 out <- tryCatch(

#####
Try part: define the expression(s) you want to "try"
#####

{
 # Just to highlight:
 # If you want to use more than one R expression in the "try part"
 # then you'll have to use curly brackets.
 # Otherwise, just write the single expression you want to try and

 message("This is the 'try' part")
 readLines(con = url, warn = FALSE)
},

#####
Condition handler part: define how you want conditions to be handled
#####

Handler when a warning occurs:
warning = function(cond) {
 message(paste("Reading the URL caused a warning:", url))
 message("Here's the original warning message:")
 message(cond)

 # Choose a return value when such a type of condition occurs
 return(NULL)
},

Handler when an error occurs:
error = function(cond) {
 message(paste("This seems to be an invalid URL:", url))
 message("Here's the original error message:")
}
```

```

 message(cond)

 # Choose a return value when such a type of condition occurs
 return(NA)
 },

 #####
 # Final part: define what should happen AFTER #
 # everything has been tried and/or handled #
 #####

 finally = {
 message(paste("Processed URL:", url))
 message("Some message at the end\n")
 }
)
return(out)
}

```

## をテストする

1つのがなURLではないURLのベクトルをしましょう

```

urls <- c(
 "http://stat.ethz.ch/R-manual/R-devel/library/base/html/connections.html",
 "http://en.wikipedia.org/wiki/Xz",
 "I'm no URL"
)

```

そして、これをでしたへのかしてします

```

y <- lapply(urls, readUrl)
Processed URL: http://stat.ethz.ch/R-manual/R-devel/library/base/html/connections.html
Some message at the end
#
Processed URL: http://en.wikipedia.org/wiki/Xz
Some message at the end
#
URL does not seem to exist: I'm no URL
Here's the original error message:
cannot open the connection
Processed URL: I'm no URL
Some message at the end
#
Warning message:
In file(con, "r") : cannot open file 'I'm no URL': No such file or directory

```

## アウトプットの

```

length(y)
[1] 3

head(y[[1]])
[1] "<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">"

```

```
[2] "<html><head><title>R: Functions to Manipulate Connections</title>"
[3] "<meta http-equiv=\"Content-Type\" content=\"text/html; charset=utf-8\">"
[4] "<link rel=\"stylesheet\" type=\"text/css\" href=\"R.css\">"
[5] "</head><body>"
[6] ""

y[[3]]
[1] NA
```

オンラインでフォールトトレラント/コードをむ <https://riptutorial.com/ja/r/topic/4060/フォールトトレラント-コード>

## 67: ベースプロット

### パラメーター

| パラメータ |                                                                                                                                        |
|-------|----------------------------------------------------------------------------------------------------------------------------------------|
| x     | X° data\$variablexまたはdata[,x] いずれかを data\$variablex で data\$variablex data[,x]                                                         |
| y     | y° data\$variableyまたはdata[,y]                                                                                                          |
| main  | プロットのなタイトル                                                                                                                             |
| sub   | プロットの中のサブタイトル                                                                                                                          |
| xlab  | xのラベル                                                                                                                                  |
| ylab  | yのラベル                                                                                                                                  |
| pch   | プロットをすまたは                                                                                                                              |
| col   | をすまたは                                                                                                                                  |
| type  | プロットのタイプ。 "p"について、 "l"のため、 "b"のために、 "c"のみの "b"、 "o"の「overplotted」にする "h" 「のhistogram」は「」ライン、 "s"のステップのために、 "S"、のステップのために、 "n"ないプロットします |

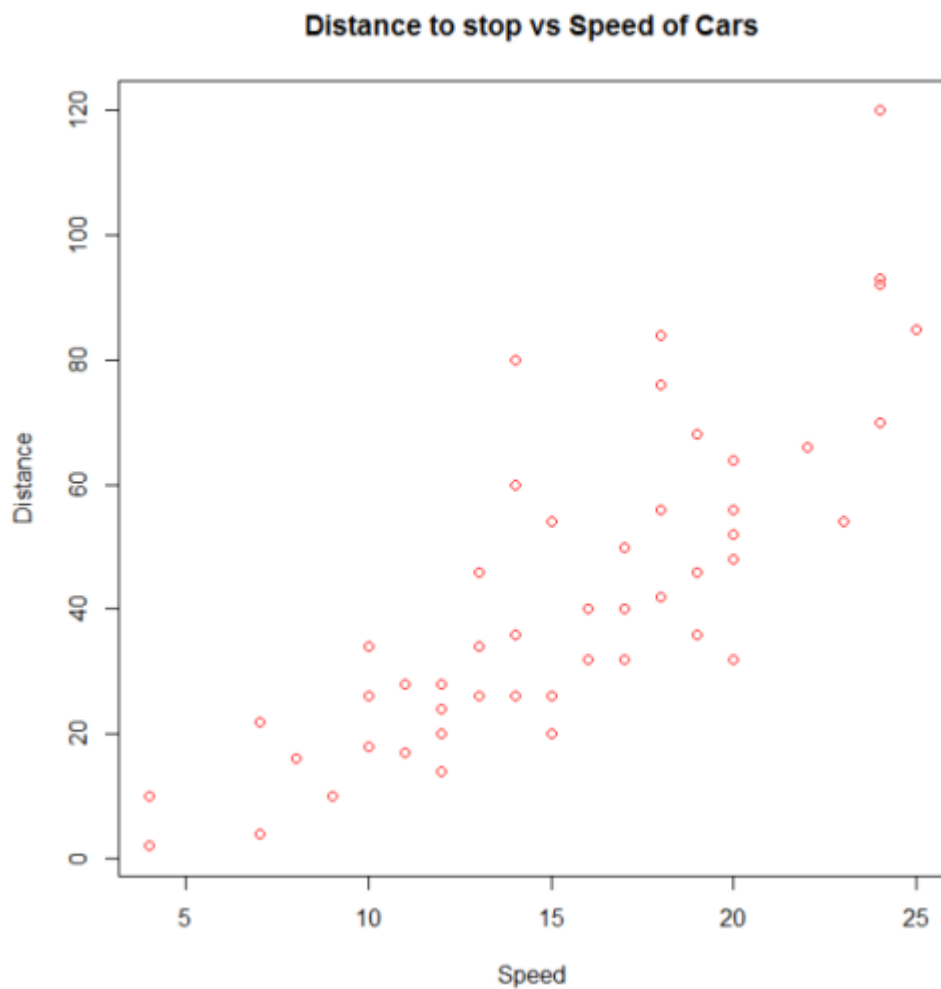
「パラメータ」セクションにリストされているは、`par`によってまたはできるなパラメータのほんのです。よりなリストについては、`par`をしてください。さらに、システムのインタラクティブグラフィックデバイスをむすべてのグラフィックデバイスは、をカスタマイズすることができるのパラメータをする。

## Examples

### プロット

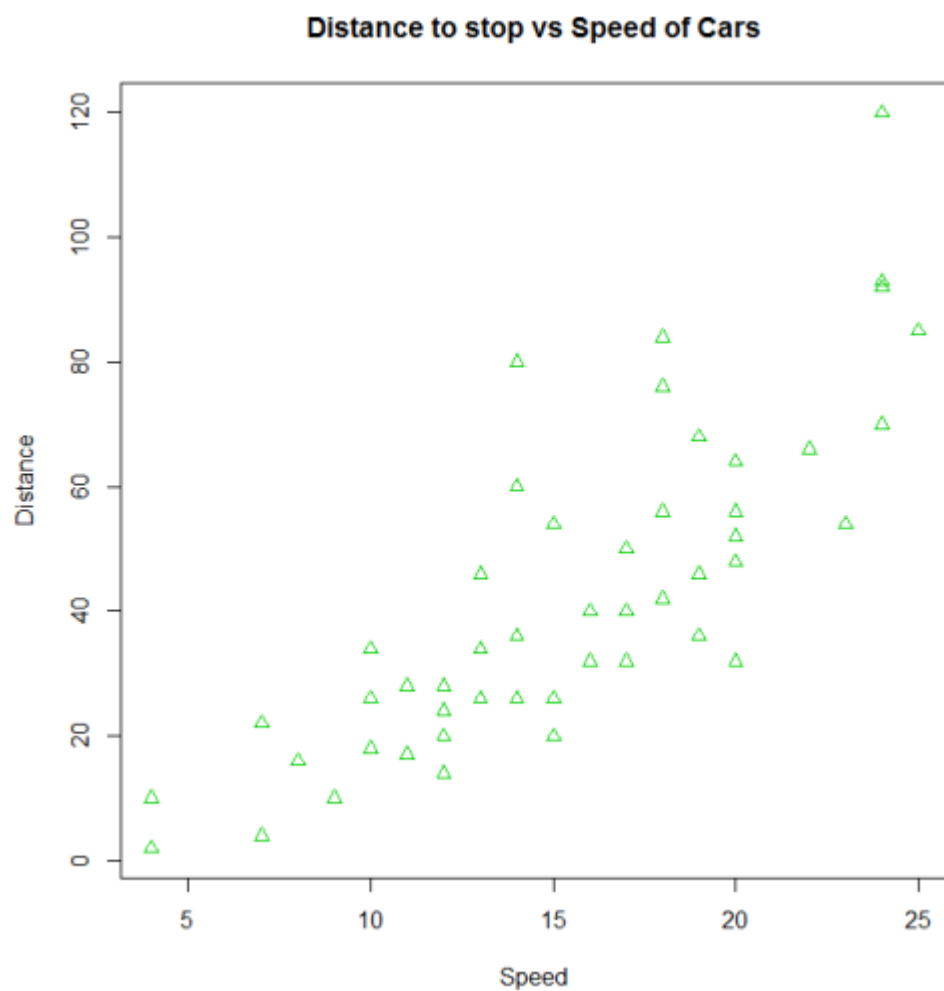
`plot()`びすと、なプロットがされます。ここでは、`cars`のと1920にしたをむビルトイン`cars`データフレームをします。データセットのについては、`helpcars`をしてください。

```
plot(x = cars$speed, y = cars$dist, pch = 1, col = 1,
 main = "Distance vs Speed of Cars",
 xlab = "Speed", ylab = "Distance")
```



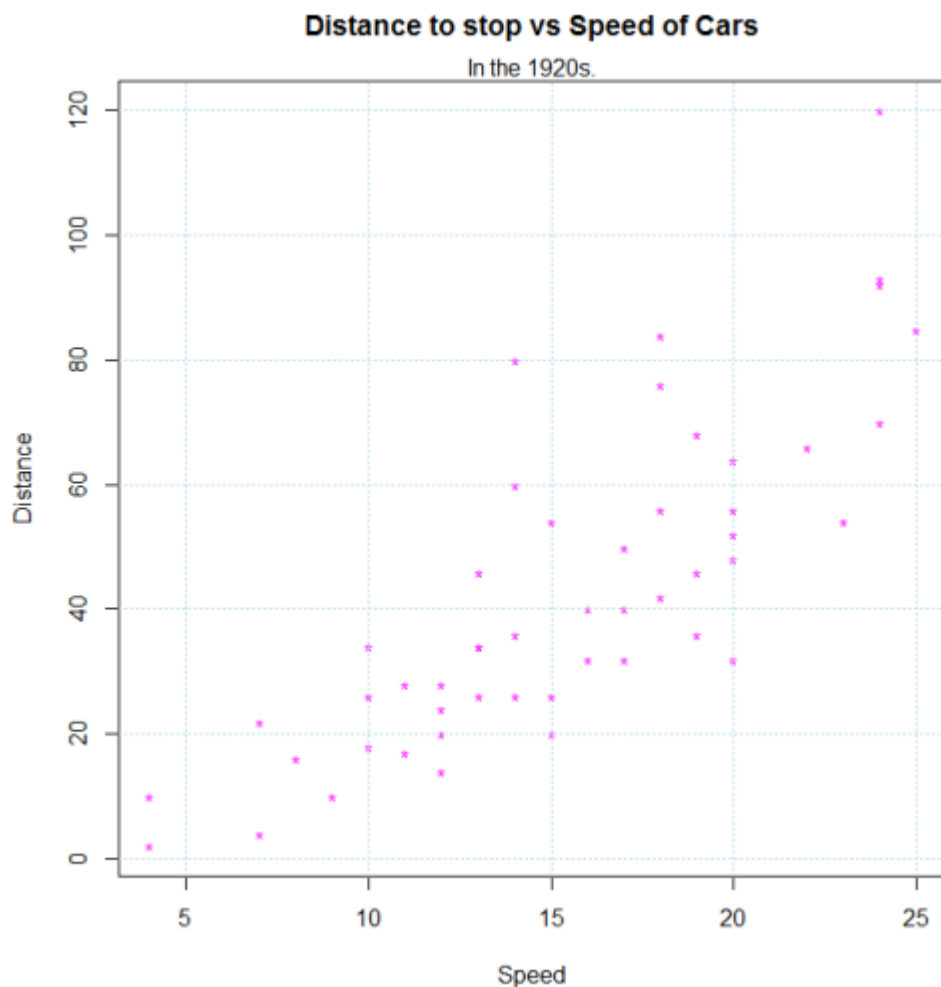
じをするために、コードののさまざまなバリエーションをすることができます。また、なるをするためにパラメータをすることもできます。

```
with(cars, plot(dist~speed, pch = 2, col = 3,
 main = "Distance to stop vs Speed of Cars",
 xlab = "Speed", ylab = "Distance"))
```



`points()`、`text()`、`mtext()`、`lines()`、`grid()`などをびすことで、このプロットにをできます。

```
plot(dist~speed, pch = "*", col = "magenta", data=cars,
 main = "Distance to stop vs Speed of Cars",
 xlab = "Speed", ylab = "Distance")
mtext("In the 1920s.")
grid(col="lightblue")
```



## Matplot

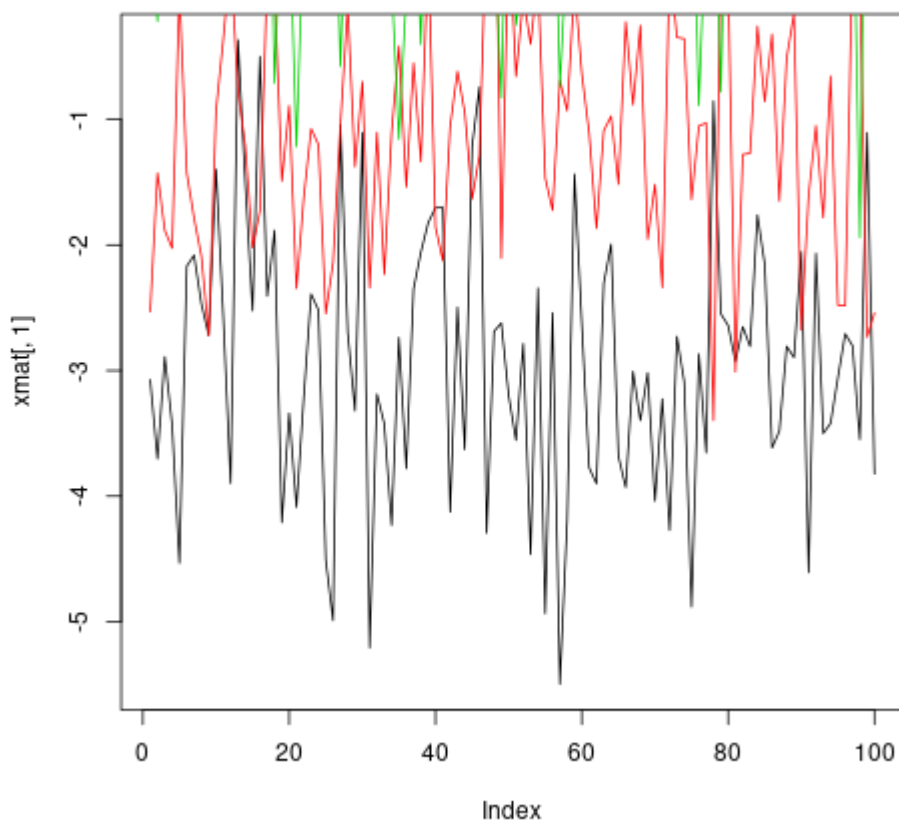
`matplot`は、のオブジェクト、にからののセットをじグラフにすばやくプロットするのにです。

ここでは、ランダムなの4つのセットをむのをします。

```
xmat <- cbind(rnorm(100, -3), rnorm(100, -1), rnorm(100, 1), rnorm(100, 3))
head(xmat)
[,1] [,2] [,3] [,4]
[1,] -3.072793 -2.53111494 0.6168063 3.780465
[2,] -3.702545 -1.42789347 -0.2197196 2.478416
[3,] -2.890698 -1.88476126 1.9586467 5.268474
[4,] -3.431133 -2.02626870 1.1153643 3.170689
[5,] -4.532925 0.02164187 0.9783948 3.162121
[6,] -2.169391 -1.42699116 0.3214854 4.480305
```

これらのをすべてじグラフにプロットする1つののは、1つの`plot`コールのにさらに3つの`points`または`lines`コールをすることです。

```
plot(xmat[,1], type = 'l')
lines(xmat[,2], col = 'red')
lines(xmat[,3], col = 'green')
lines(xmat[,4], col = 'blue')
```

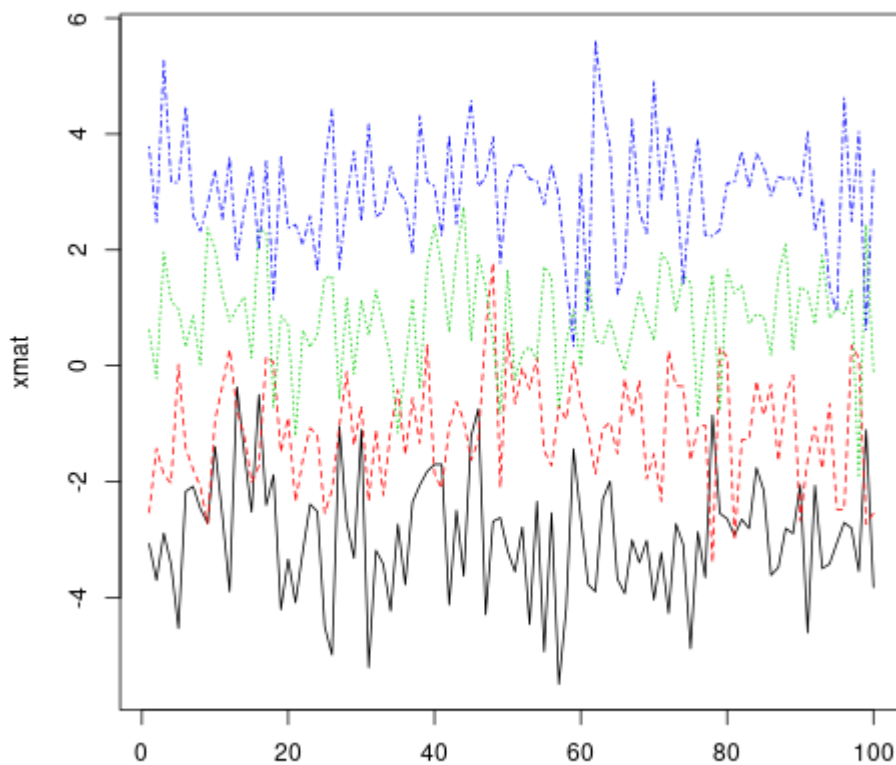


しかし、これはともであり、とりわけ、デフォルトでは、`plot`のみにフィットするようにされているため、をきこします。

このようなでは、`matplot`をする`matplot`がはるかにです。これは1のびしで、のをにし、の`matplot`をえてできます。

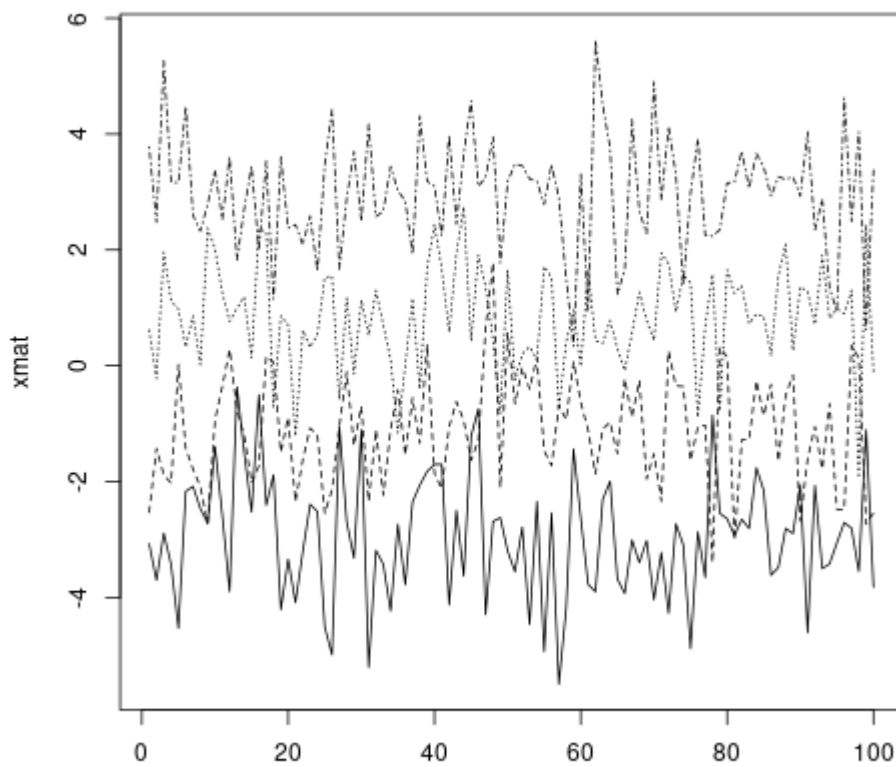
```
matplot(xmat, type = 'l')
```





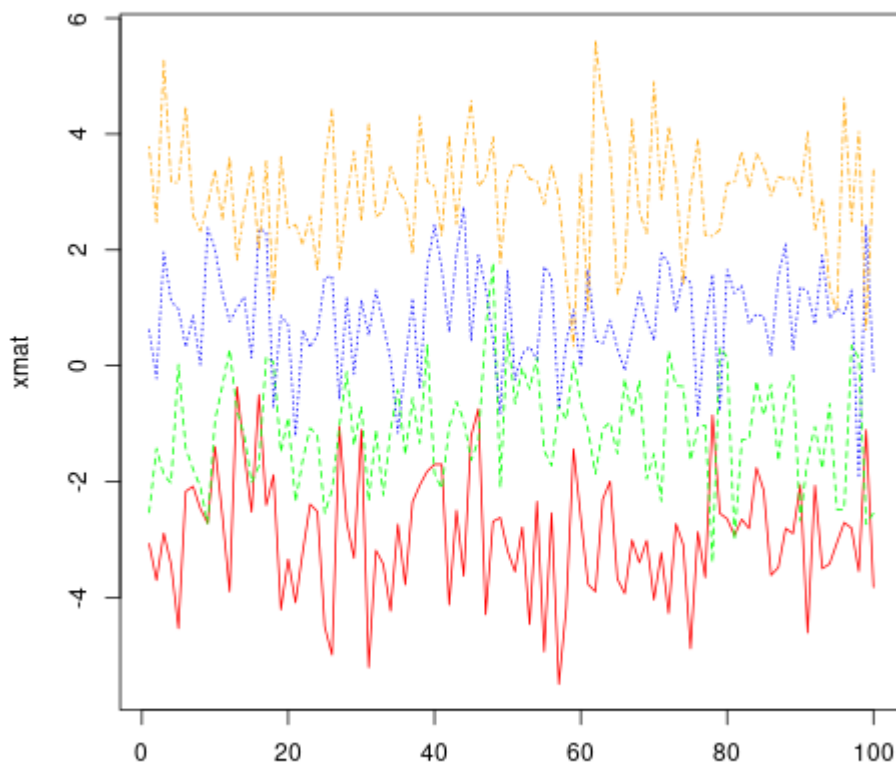
デフォルトでは、`matplot`は`col col`と`lty lty`のがするので、りされるになみわせのがえるのでしてください。しかし、これらののいずれかまたはは、のにすることができます...

```
matplot(xmat, type = 'l', col = 'black')
```



...またはカスタムベクトルなRベクトルリサイクルについてののにリサイクルされます。

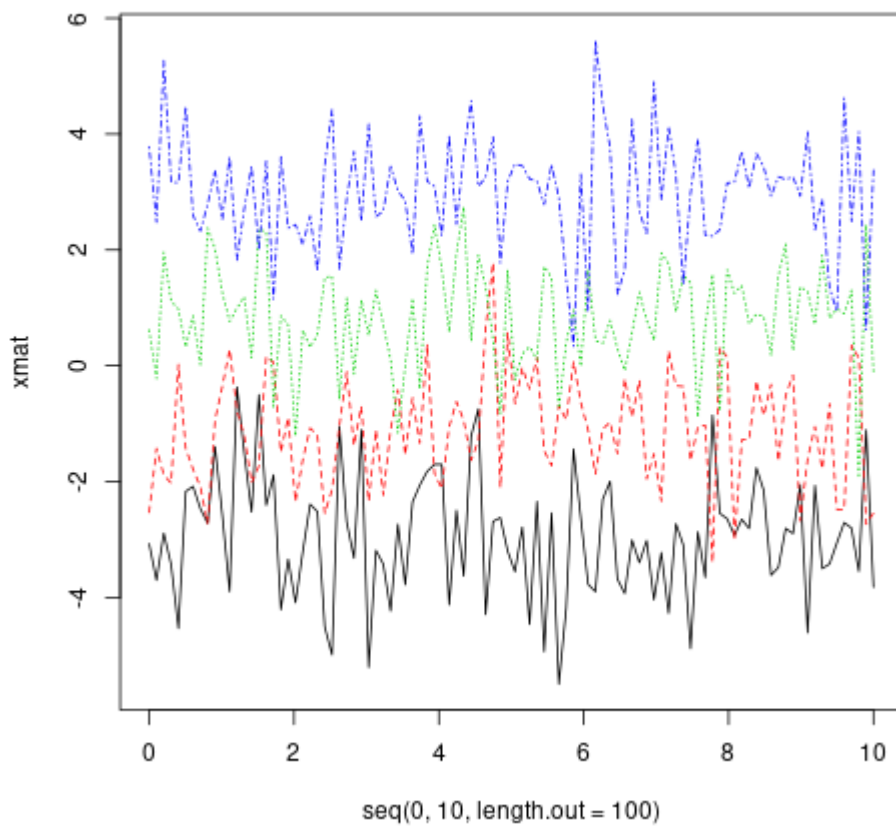
```
matplot(xmat, type = 'l', col = c('red', 'green', 'blue', 'orange'))
```



`main`、`xlab`、`xmin`などのグラフィックパラメータは、`plot`とくじようにし`plot`。については、`?par`してください。

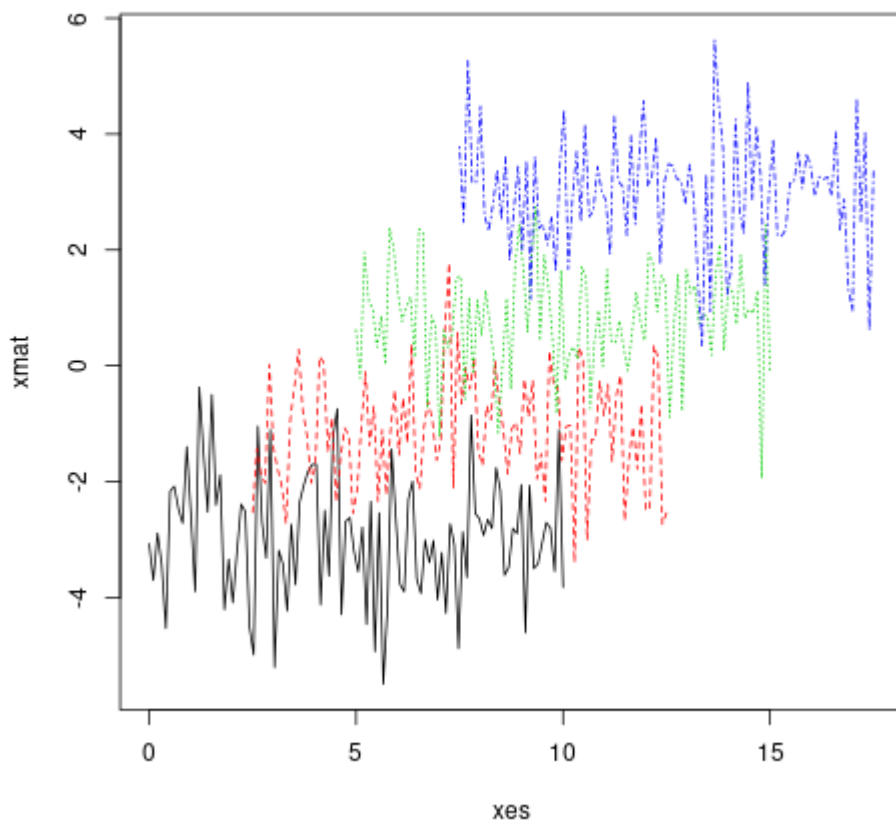
`plot`、1つのオブジェクトのみがえられた、`matplot`はそれが`y`であると`matplot`、`x`のインデックスをします。ただし、`x`と`y`はにできます。

```
matplot(x = seq(0, 10, length.out = 100), y = xmat, type='l')
```



、 $x$ と $y$ はになります。

```
xes <- cbind(seq(0, 10, length.out = 100),
 seq(2.5, 12.5, length.out = 100),
 seq(5, 15, length.out = 100),
 seq(7.5, 17.5, length.out = 100))
matplot(x = xes, y = xmat, type = 'l')
```

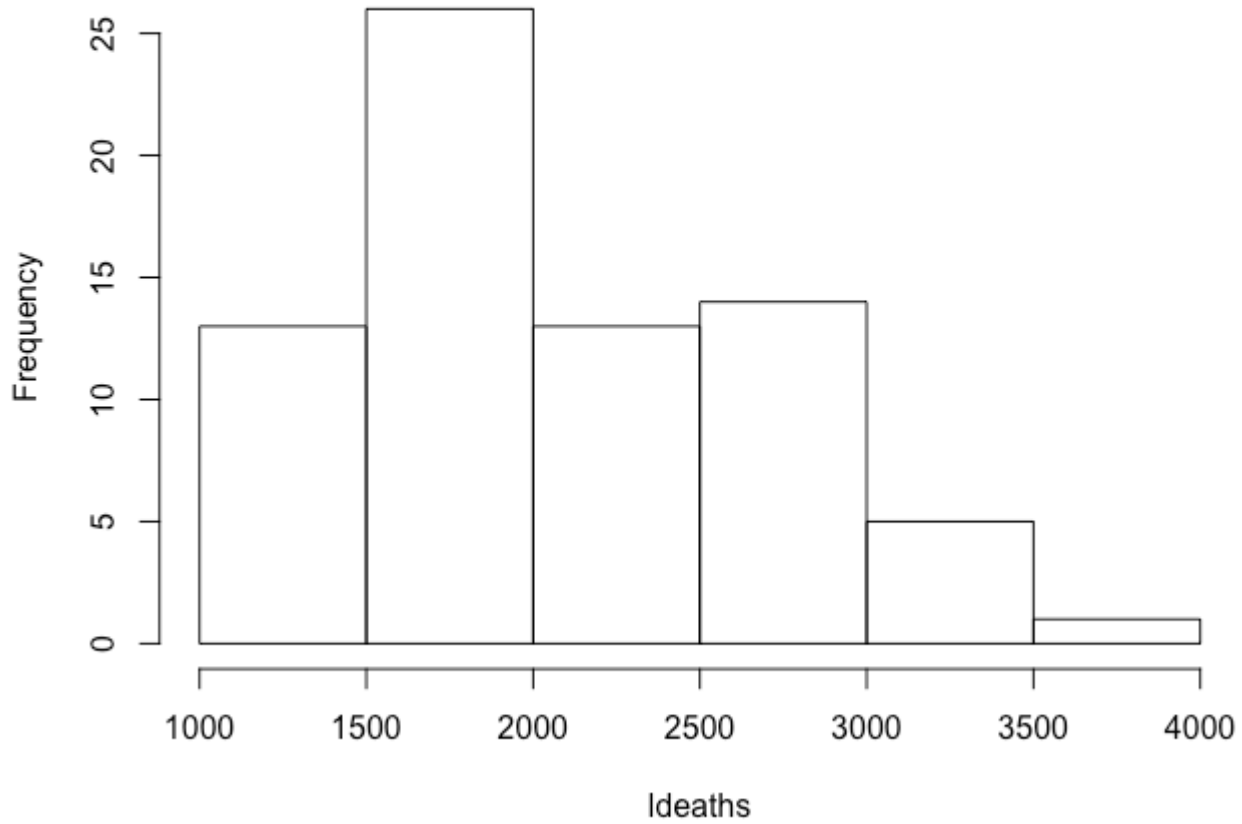


## ヒストグラム

ヒストグラムは、データのとなるのプロットをにする。

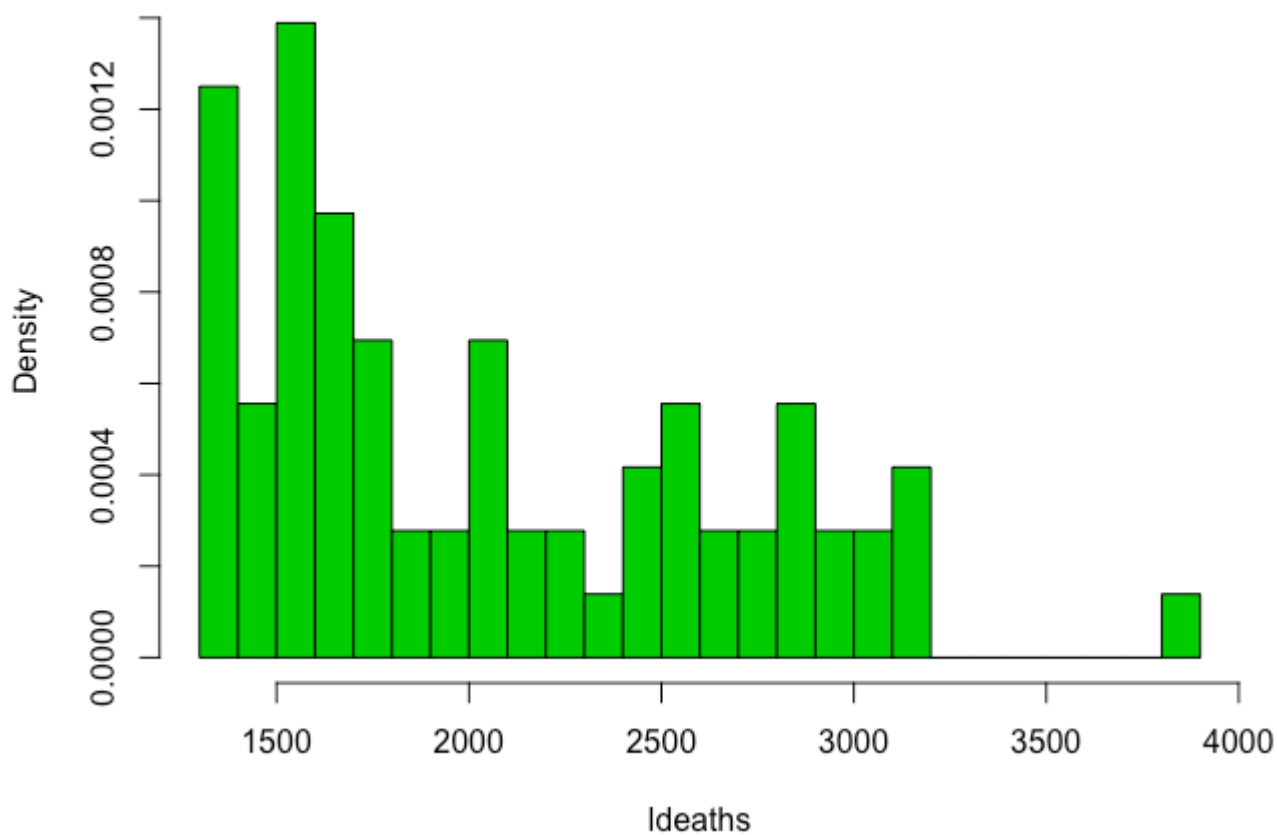
```
hist(ldeaths)
```

### Histogram of Ideaths



```
hist(ideaths, breaks = 20, freq = F, col = 3)
```

## Histogram of Ideaths



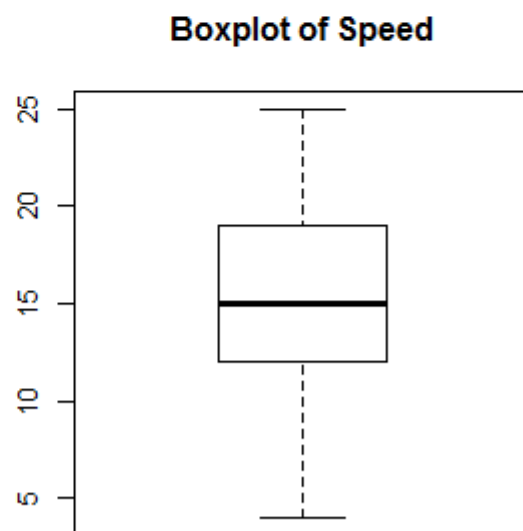
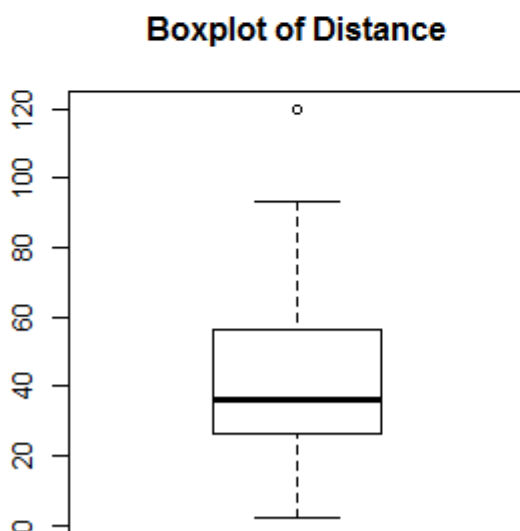
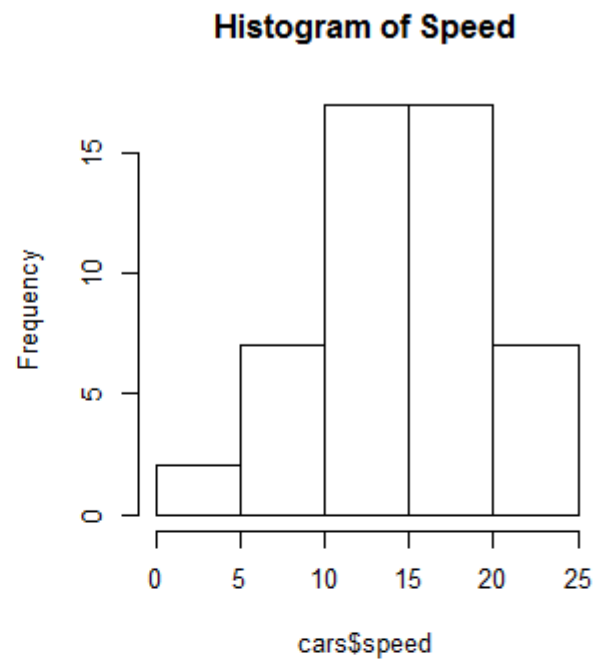
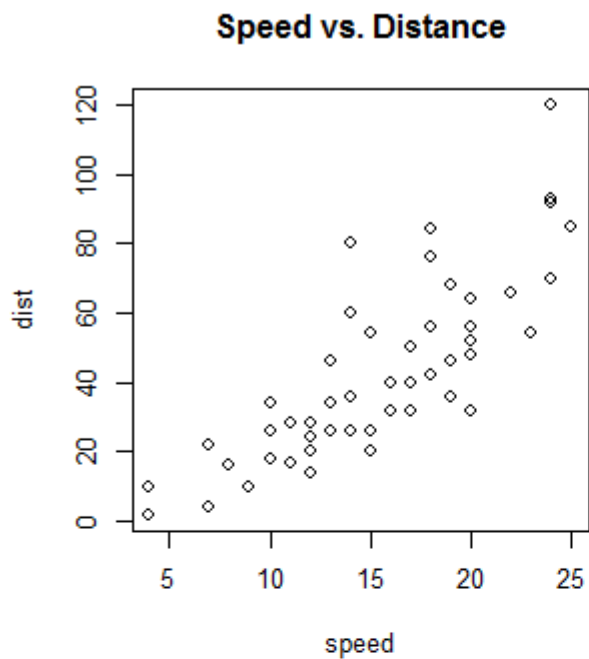
### プロットの

1つのグラフえは、のにあるBarplotなどでのプロットタイプをみわせるとなことがよくあります。  
。 Rは、`par()` および `layout()` のけをりてこれをにします。

#### `par()`

`par` は `mfrow` または `mfcol` をして、プロットのグリッドとしてする `nrows` および `ncols` の `c(nrows, ncols)` をします。のは、1つのグラフで4つのプロットをするをしています。

```
par(mfrow=c(2,2))
plot(cars, main="Speed vs. Distance")
hist(cars$speed, main="Histogram of Speed")
boxplot(cars$dist, main="Boxplot of Distance")
boxplot(cars$speed, main="Boxplot of Speed")
```

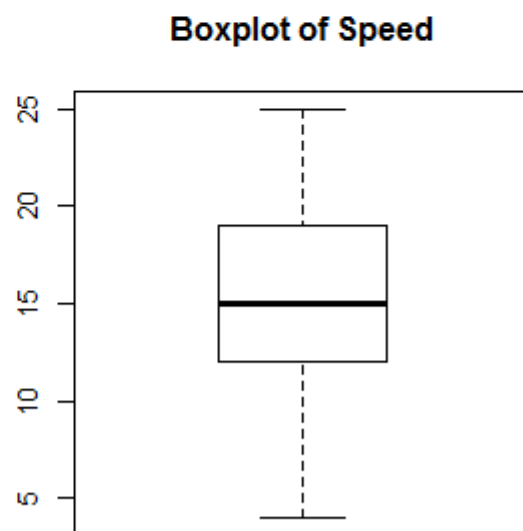
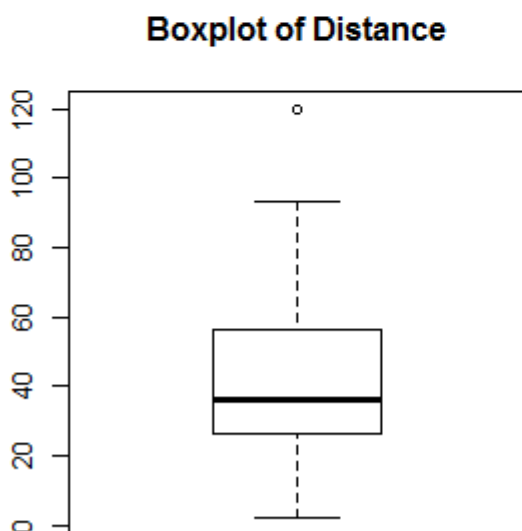
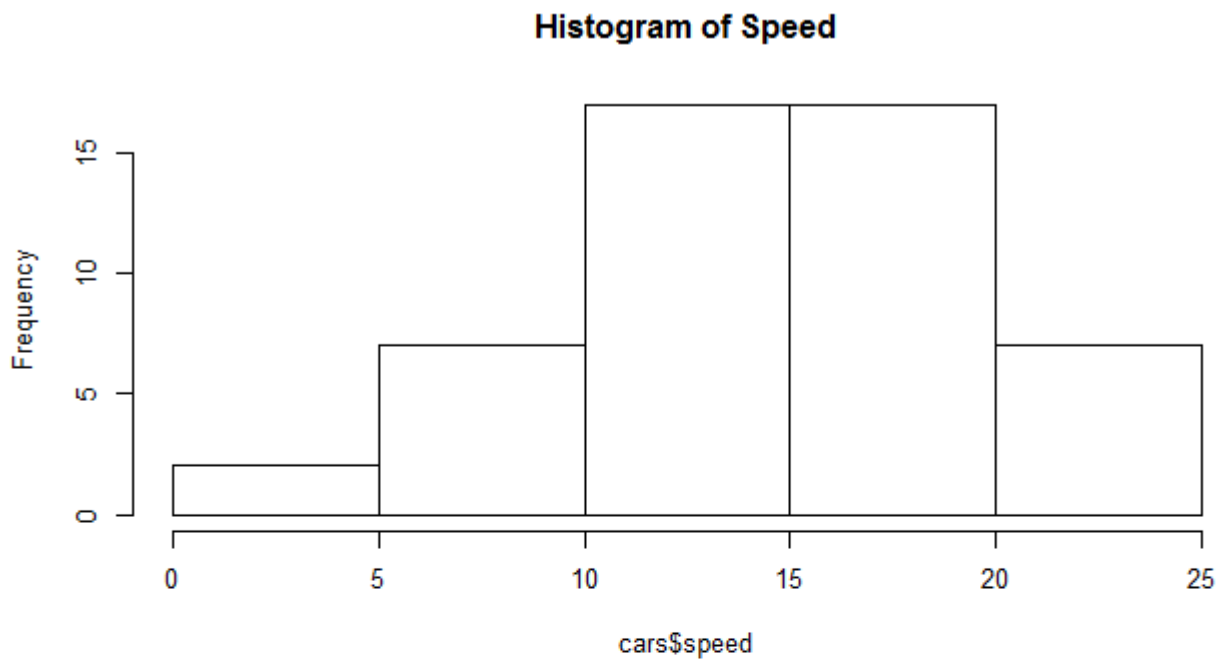


`layout()`

`layout()` はよりで、なグラフのプロットのとをすることができます。これはオブジェクトをととしてしています

```
layout(matrix(c(1,1,2,3), 2,2, byrow=T))
hist(cars$speed, main="Histogram of Speed")
boxplot(cars$dist, main="Boxplot of Distance")
boxplot(cars$speed, main="Boxplot of Speed")
```



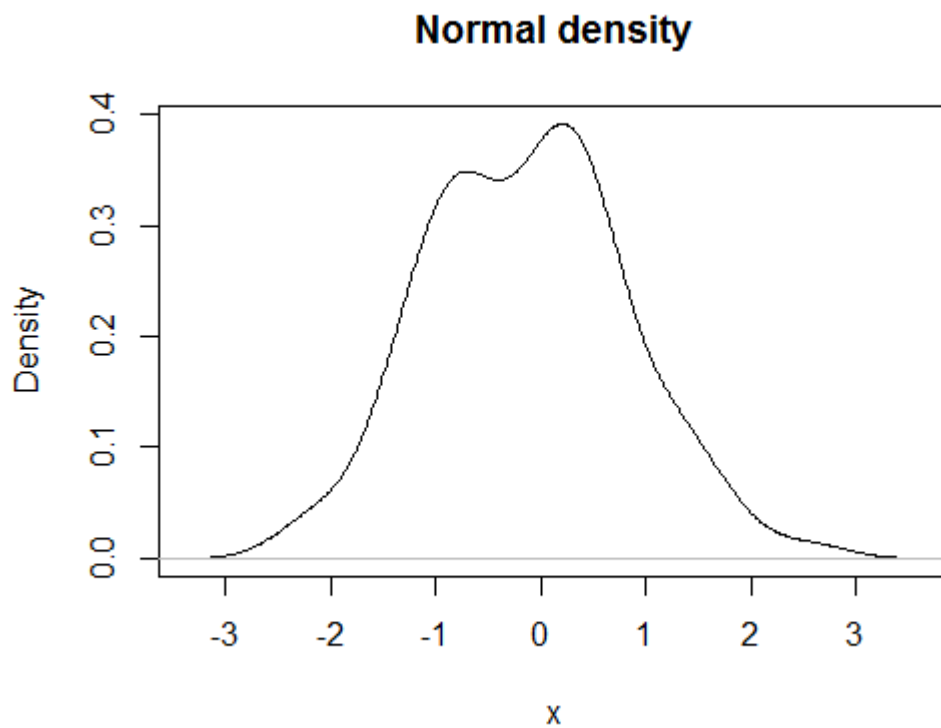


## プロット

ヒストグラムにするにでなフォローアップは、ランダムのをされたをプロットすることです。コマンドによってされるプロット

```
plot(density(rnorm(100)),main="Normal density",xlab="x")
```

のようになります

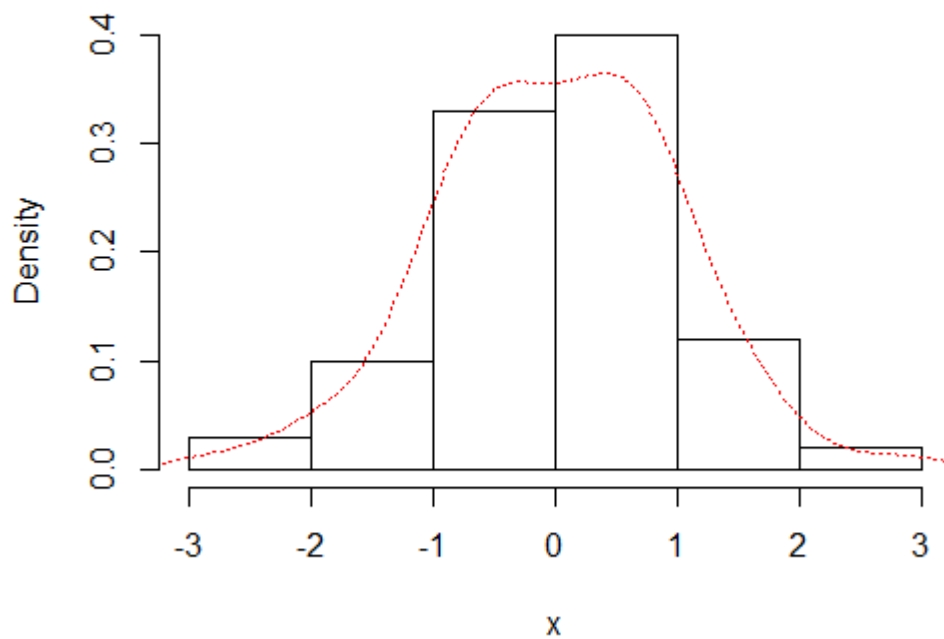


ヒストグラムとをねわせることができます。

```
x=rnorm(100)
hist(x,prob=TRUE,main="Normal density + histogram")
lines(density(x),lty="dotted",col="red")
```

それはえる

## Normal density + histogram

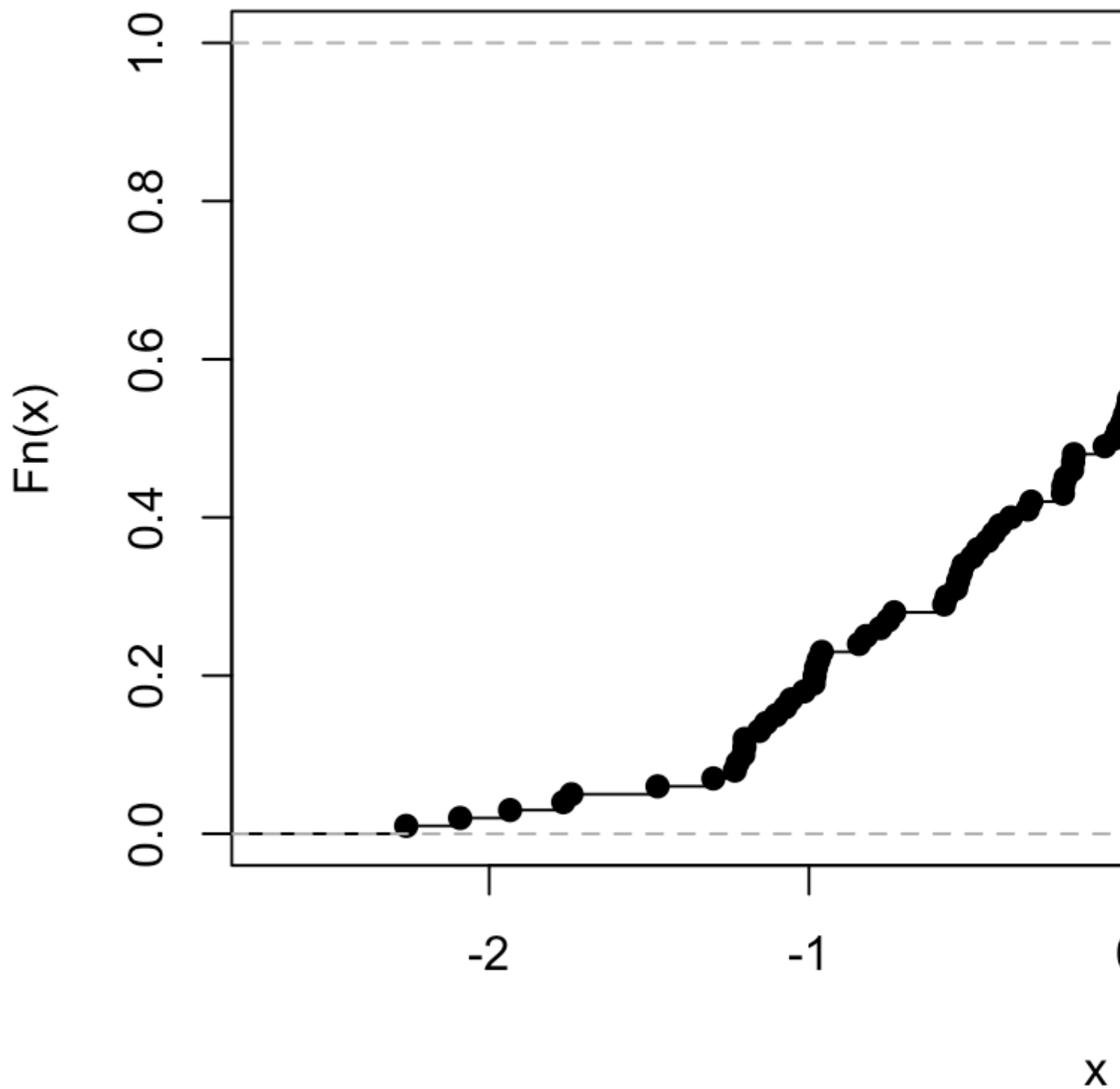


ヒストグラムとプロットにするにでなフォローアップは、です。こののために `ecdf()` をうことができます。コマンドによってされるプロット

```
plot(ecdf(rnorm(100)),main="Cumulative distribution",xlab="x")
```

のようになります

## Cumulative d



R\_Plots

•

2つのベクトルがあり、それらをプロットするがあります。

```
x_values <- rnorm(n = 20 , mean = 5 , sd = 8) #20 values generated from Normal(5,8)
y_values <- rbeta(n = 20 , shape1 = 500 , shape2 = 10) #20 values generated from Beta(500,10)
```

に `x_values` 、 に `y_values` をつプロットをするには、のコマンドをできます。

```
plot(x = x_values, y = y_values, type = "p") #standard scatter-plot
plot(x = x_values, y = y_values, type = "l") # plot with lines
plot(x = x_values, y = y_values, type = "n") # empty plot
```

コンソールに `?plot()` すると、さらにく のオプションがされます。

- **Boxplot**

いくつかのがあり、そのをべたい

```
#boxplot is an easy way to see if we have some outliers in the data.

z<- rbeta(20 , 500 , 10) #generating values from beta distribution
z[c(19 , 20)] <- c(0.97 , 1.05) # replace the two last values with outliers
boxplot(z) # the two points are the outliers of variable z.
```

- ヒストグラム

ヒストグラムをくな

```
hist(x = x_values) # Histogram for x vector
hist(x = x_values, breaks = 3) #use breaks to set the numbers of bars you want
```

- グラフ

のをしたいは、にグラフをきます

まず、のようなでデータをするがあります。

```
P <- c(rep('A' , 3) , rep('B' , 10) , rep('C' , 7))
t <- table(P) # this is a frequency matrix of variable P
pie(t) # And this is a visual version of the matrix above
```

オンラインでベースプロットをむ <https://riptutorial.com/ja/r/topic/1377/ベースプロット>

## 68: ベクトルの

### Examples

のシーケンス

:をして、コードのきなをベクトルするなど、のをします。

```
x <- 1:5
x
[1] 1 2 3 4 5
```

これはのでします

```
10:4
[1] 10 9 8 7 6 5 4
```

であっても

```
1.25:5
[1] 1.25 2.25 3.25 4.25
```

またはネガティブ

```
-4:4
[1] -4 -3 -2 -1 0 1 2 3 4
```

### seq

seqは、:が1のステップをできるようにするよりなです。

このは、start デフォルトは1からそのをむまでのシーケンスをします。

end to パラメータのみをできます

```
seq(5)
[1] 1 2 3 4 5
```

とに

```
seq(2, 5) # or seq(from=2, to=5)
[1] 2 3 4 5
```

そしてに、ステップ by

```
seq(2, 5, 0.5) # or seq(from=2, to=5, by=0.5)
[1] 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

`seq`は、ののさ `length.out` がにされるにがいたステップをにすることができる

```
seq(2,5, length.out = 10)
[1] 2.0 2.3 2.6 2.9 3.2 3.5 3.8 4.1 4.4 4.7 5.0
```

シーケンスがのベクトルとじさをつがあるは、 `length.out = length(x)` として `along.with` をできます。

```
x = 1:8
seq(2,5,along.with = x)
[1] 2.000000 2.428571 2.857143 3.285714 3.714286 4.142857 4.571429 5.000000
```

で2つのなされたがある `seq` `seq_along`、 `seq_len`、 および `seq.int`。 `seq_along` と `seq_len` は、1からNのをします。Nはの、 `seq_along` をつベクトルまたはリストのさ、 `seq_along` ので `seq_len` ます。

```
seq_along(x)
[1] 1 2 3 4 5 6 7 8
```

`seq_along` は、のオブジェクトのインデックスをします。

```
counting numbers 1 through 10
seq_len(10)
[1] 1 2 3 4 5 6 7 8 9 10
indices of existing vector (or list) with seq_along
letters[1:10]
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j"
seq_along(letters[1:10])
[1] 1 2 3 4 5 6 7 8 9 10
```

`seq.int` はののためにされる `seq` とじです。

また、のからシーケンスのベクトルをするい `sequence` もあります。

```
sequence(4)
[1] 1 2 3 4
sequence(c(3, 2))
[1] 1 2 3 1 2
sequence(c(3, 2, 5))
[1] 1 2 3 1 2 1 2 3 4 5
```

## ベクトル

Rのベクトルは、なるタイプえば、、、をつことができます。ベクトルをするもなは、`vector()` をうことです。

```
vector('integer',2) # creates a vector of integers of size 2.
vector('character',2) # creates a vector of characters of size 2.
```

```
vector('logical',2) # creates a vector of logicals of size 2.
```

しかし、Rでは、にはがです。

```
integer(2) # is the same as vector('integer',2) and creates an integer vector with two elements
character(2) # is the same as vector('character',2) and creates an character vector with two elements
logical(2) # is the same as vector('logical',2) and creates an logical vector with two elements
```

デフォルトのをつベクトルをすることもできます。くの、`c()`がこれにされます。cはまたはのです。

```
c(1, 2) # creates a integer vector of two elements: 1 and 2.
c('a', 'b') # creates a character vector of two elements: a and b.
c(T,F) # creates a logical vector of two elements: TRUE and FALSE.
```

すべきなは、Rはのえは1をサイズ1のベクトルとしてすることである。じことがえは1.1、えはTまたはF、またはえは'a'にもてはまります。したがって、にベクトルをみわせることになります。これはベクトルです。

じようなベクトルをみわせるがあることにしてください。そのの、Rはじタイプのベクトルのベクトルをしようとします。

```
c(1,1.1,'a',T) # all types (integer, numeric, character and logical) are converted to the 'lowest' type which is character.
```

ベクトルのをつけることは、`[`をってうことができます。

```
vec_int <- c(1,2,3)
vec_char <- c('a','b','c')
vec_int[2] # accessing the second element will return 2
vec_char[2] # accessing the second element will return 'b'
```

これはのにもできます

```
vec_int[2] <- 5 # change the second value from 2 to 5
vec_int # returns [1] 1 5 3
```

に、`:seq()`をして、のベクトルをくすることができます。

```
vec_int <- 1:10
vec_int # returns [1] 1 2 3 4 5 6 7 8 9 10
```

これは、ベクトルをよりなサブセットからよりなサブセットに

```
vec_char <- c('a','b','c','d','e')
vec_char[2:4] # returns [1] "b" "c" "d"
```



```
vec_char[c(1,3,5)] # returns [1] "a" "c" "e"
```

## きベクトルの

きベクトルは、いくつかのことができます。 `c`

```
xc <- c('a' = 5, 'b' = 6, 'c' = 7, 'd' = 8)
```

は

```
> xc
a b c d
5 6 7 8
```

list

```
x1 <- list('a' = 5, 'b' = 6, 'c' = 7, 'd' = 8)
```

は

```
> x1
$a
[1] 5

$b
[1] 6

$c
[1] 7

$d
[1] 8
```

`setNames` をすると、じさの2つのベクトルをしてきベクトルをできます。

```
x <- 5:8
y <- letters[1:4]

xy <- setNames(x, y)
```

その、きベクトルがされます。

```
> xy
a b c d
5 6 7 8
```

てかるように、これは`c`とじをえる。

また、じをるために`names` をうこともできます

```
xy <- 5:8
names(xy) <- letters[1:4]
```

このようなベクトルでは、をですることもです

```
> xy["c"]
c
7
```

このにより、そのようなベクトルをベクトル/テーブルとしてして、をデータフレームののベクトルまたはのにさせることができます。のデータフレームをしてください。

```
mydf <- data.frame(let = c('c','a','b','d'))

> mydf
 let
1 c
2 a
3 b
4 d
```

`mydf` `xy` からのしいで `num` とばれる `mydf` データフレームにしいをします。 `match` をする `match`、`xy` のなをできます。

```
mydf$num <- xy[match(mydf$let, names(xy))]
```

は

```
> mydf
 let num
1 c 7
2 a 5
3 b 6
4 d 8
```

## rep をってベクトルをする

`rep` は、かなりなでベクトルをりすためにできます。

```
repeat counting numbers, 1 through 5 twice
rep(1:5, 2)
[1] 1 2 3 4 5 1 2 3 4 5

repeat vector with incomplete recycling
rep(1:5, 2, length.out=7)
[1] 1 2 3 4 5 1 2
```

は、/のベクトルを、これらのをりして `data.frame` のベクトルにするにです。

```
same except repeat each integer next to each other
rep(1:5, each=2)
```

```
[1] 1 1 2 2 3 3 4 4 5 5
```

このようなデータへのをむ`rep`のれたは、さをベクトルのをりすできえることによって、パネルへのベクトルのをできることです。

```
automated length repetition
rep(1:5, 1:5)
[1] 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5
hand-fed repetition length vector
rep(1:5, c(1,1,1,2,2))
[1] 1 2 3 4 4 5 5
```

これは、データにじてするベクトルをにするために、が`rep`の2のをえることをにするをします。

`seq`とに、よりくされた`rep`バージョンは`rep_len`と`rep.int`です。これらは、`rep`するいくつかのをとします。したがって、がであり、りされるベクトルののがなでもながあります。

```
repeat counting numbers, 1 through 5 twice
rep.int(1:5, 2)
[1] 1 2 3 4 5 1 2 3 4 5

repeat vector with incomplete recycling
rep_len(1:5, length.out=7)
[1] 1 2 3 4 5 1 2
```

## ビルドからのベクトルとのシーケンス

`R`にはのがあります。のをできます。

- `LETTERS` ローマの26
- `letters` ローマの26
- `month.abb` のの3の
- `month.name` そののの
- `pi` とそのの

とのから、ベクトルをすることができます。

### 1のシーケンス

```
> letters
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t" "u" "v"
"w" "x" "y" "z"

> LETTERS[7:9]
[1] "G" "H" "I"

> letters[c(1,5,3,2,4)]
[1] "a" "e" "c" "b" "d"
```

## 2のまたはのシーケンス

```
> month.abb
[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "Dec"

> month.name[1:4]
[1] "January" "February" "March" "April"

> month.abb[c(3,6,9,12)]
[1] "Mar" "Jun" "Sep" "Dec"
```

オンラインでベクトルのをむ <https://riptutorial.com/ja/r/topic/1088/ベクトルの>

## 69: ボックスプロット

- `boxplotx`、...
- `boxplot`、データ = NULL、...、サブセット、`na.action = NULL`##クラス 'formula'のS3メソッド
- `boxplotx`、...、`range = 1.5`、`width = NULL`、`varwidth = FALSE`、`notch = FALSE`、`outline = TRUE`、`plot = TRUE`、`par "fg"`、`col = NULL`、`log = ""`、デフォルトのS3メソッドデフォルトでは" false "、デフォルトはfalse

### パラメーター

| パラメーター                 | ソースRドキュメント                                                                                  |
|------------------------|---------------------------------------------------------------------------------------------|
|                        | <code>ygrp</code> のような。ここで、 <code>y</code> は、グループ <code>grp</code> はによってグループにされるデータのベクトルです。 |
| データ                    | のをべき <code>data.frame</code> またはリスト。                                                        |
| サブセット                  | プロットにされるのサブセットをするオプションのベクトル。                                                                |
| <code>na.action</code> | データがNAをむとときにがこるべきかをす。デフォルトでは、レスポンスまたはグループのはされます。                                            |
| ボックスワックス               | すべてのボックスにされる。のグループしかないは、ボックスのをくすることでプロットのをできます。                                             |
| プロット                   | TRUEデフォルトの、ボックスプロットがされます。そうでないは、ボックスプロットがついているがされます。                                        |
|                        | <code>col</code> がnullでないは、ボックスプロットのボディをけするためにされるがまれているとみなされます。デフォルトではです。                   |

## Examples

`boxplot{graphics}`でbox-and-whiskerプロットをする

このでは、デフォルトの`boxplot()`と`iris`データフレームをしています。

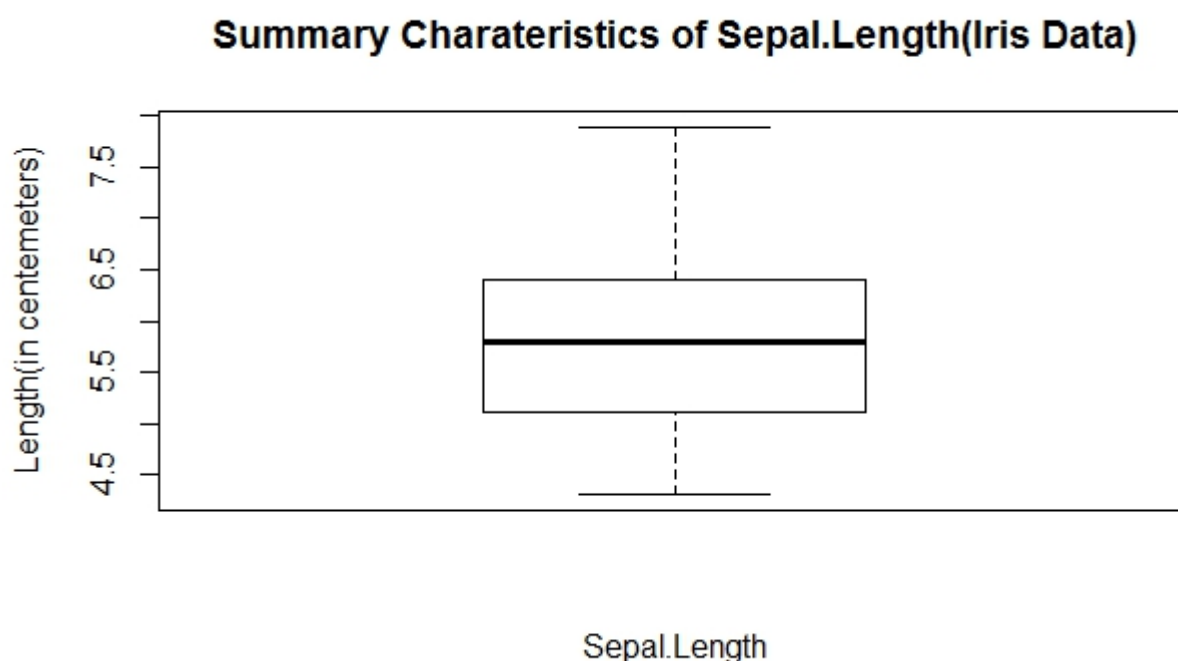
```
> head(iris)
 Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1 5.1 3.5 1.4 0.2 setosa
2 4.9 3.0 1.4 0.2 setosa
```

|   |     |     |     |     |        |
|---|-----|-----|-----|-----|--------|
| 3 | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4 | 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5 | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 6 | 5.4 | 3.9 | 1.7 | 0.4 | setosa |

## なboxplotSepal.Length

のボックスウィスカグラフをする

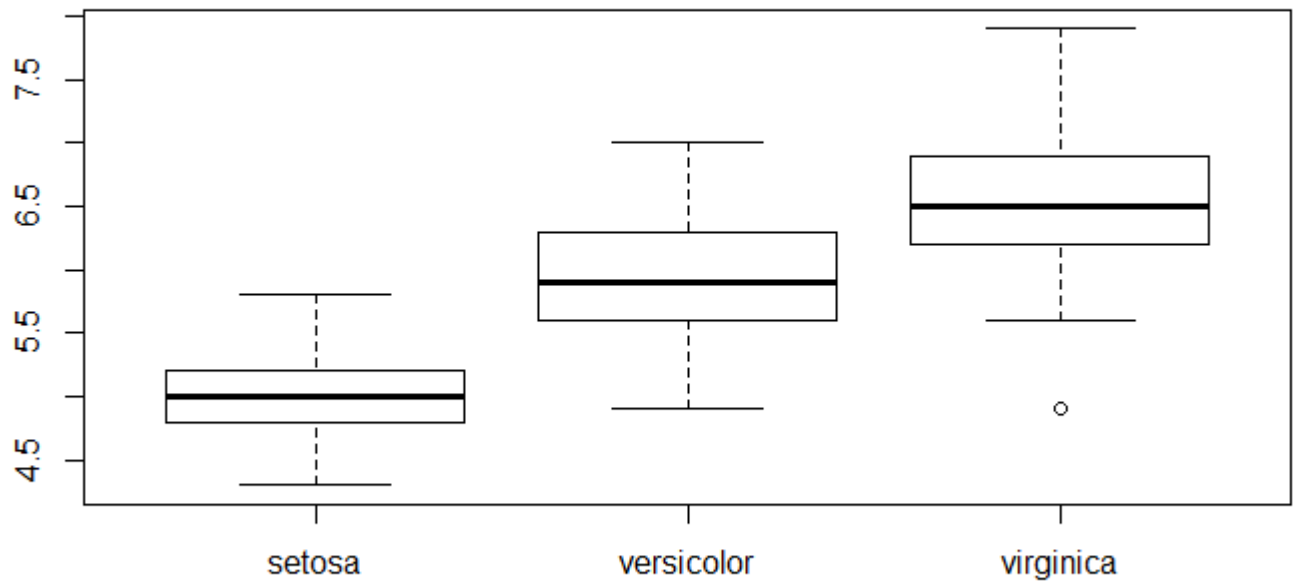
```
boxplot(iris[,1],xlab="Sepal.Length",ylab="Length(in centimeters)",
 main="Summary Charateristics of Sepal.Length(Iris Data)")
```



## によってグループけされたセパールのボックス プロット

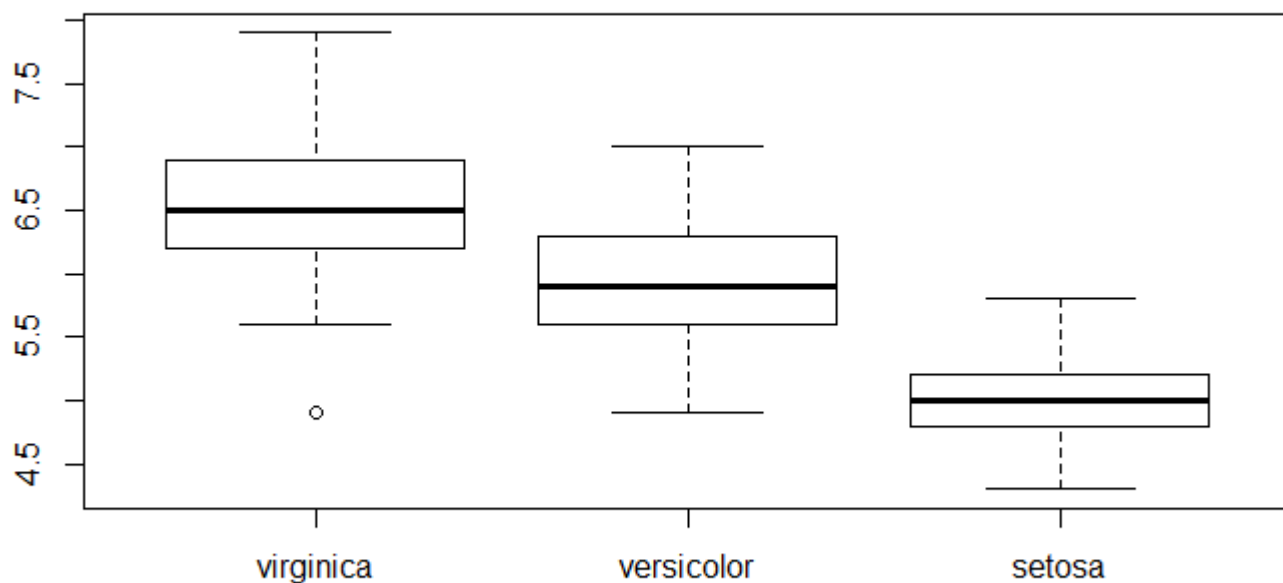
カテゴリでグループされたのボックスプロットをする

```
boxplot(Sepal.Length~Species,data = iris)
```



プロットのボックスのをするには、カテゴリのレベルのをするがあります。  
たとえば、もしたちがをしたいならば、 `virginica - versicolor - setosa`

```
newSpeciesOrder <- factor(iris$Species, levels=c("virginica","versicolor","setosa"))
boxplot(Sepal.Length~newSpeciesOrder,data = iris)
```

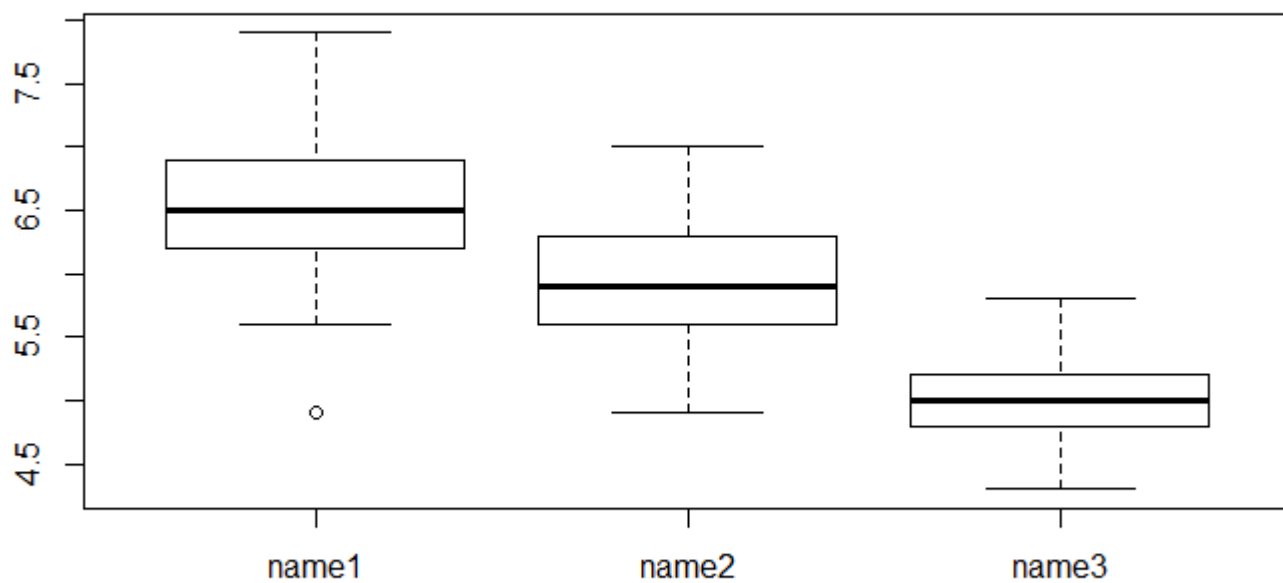


## グループをする

あなたのグループのよりいをしたい、`Names`パラメータをすることができます。それは、カテゴリのレベルのきさのベクトルをとる

```
boxplot(Sepal.Length~newSpeciesOrder,data = iris,names= c("name1","name2","name3"))
```

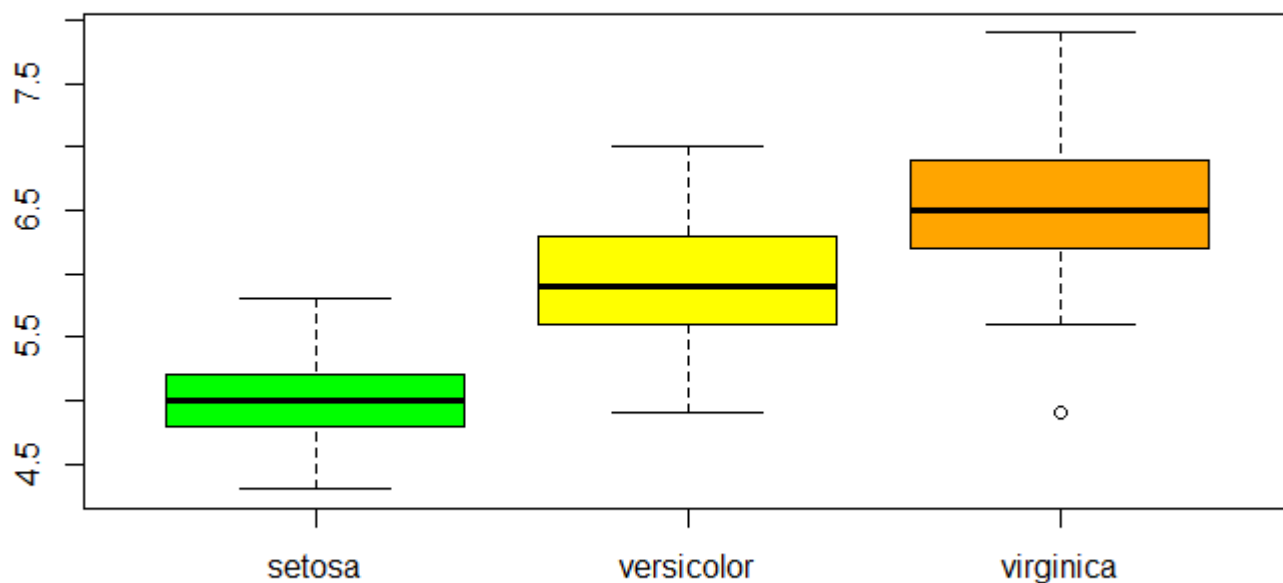




さな

col カテゴリのレベルのきさのベクトルをする

```
boxplot(Sepal.Length~Species,data = iris,col=c("green","yellow","orange"))
```

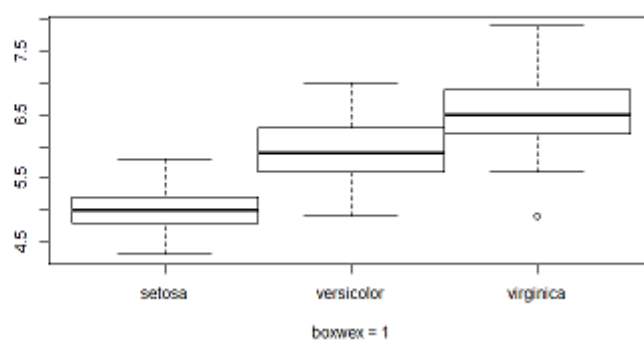
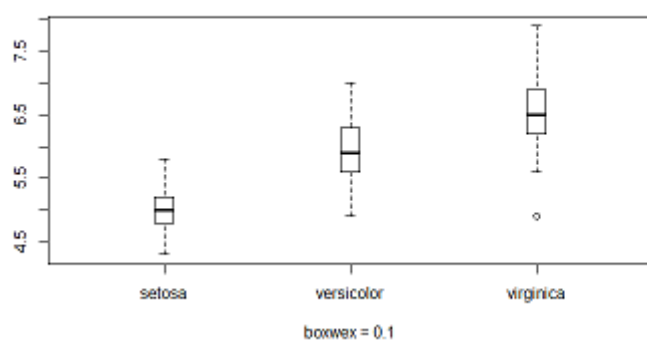


## ボックスの

`boxwex` ボックスのマージンをします。

```
boxplot(Sepal.Length~Species,data = iris,boxwex = 0.1)
```

```
boxplot(Sepal.Length~Species,data = iris,boxwex = 1)
```



## boxplots についたを `plot=FALSE`

をるには、パラメタ `plot` を `FALSE` するがあり `FALSE`。  
々ながえられている

```
> boxplot(Sepal.Length~newSpeciesOrder,data = iris,plot=FALSE)
$stats #summary of the numerical variable for the 3 groups
[,1] [,2] [,3]
```

```

[1,] 5.6 4.9 4.3 # extreme value
[2,] 6.2 5.6 4.8 # first quartile limit
[3,] 6.5 5.9 5.0 # median limit
[4,] 6.9 6.3 5.2 # third quartile limit
[5,] 7.9 7.0 5.8 # extreme value

$n #number of observations in each groups
[1] 50 50 50

$conf #extreme value of the notchs
 [,1] [,2] [,3]
[1,] 6.343588 5.743588 4.910622
[2,] 6.656412 6.056412 5.089378

$out #extreme value
[1] 4.9

$group #group in which are the extreme value
[1] 1

$names #groups names
[1] "virginica" "versicolor" "setosa"

```

のボックスプロットスタイルパラメータ。

## ボックス

- boxlty - ボックスラインタイプ
- boxlwd - ボックスの
- boxcol - ボックスの
- boxfill - ボックスのりつぶしの
- medlty - メジアンタイプがないは ""
- medlwd - メジアンラインwidht
- medcol - メジアンの
- medpch - メジアンポイントのはNA
- medcex - メジアンポイントサイズ
- medbg - メジアンポイントの

---

## ウィスカー

- ウィスキーウィスカーラインタイプ
- whisklwd - ウィスカー
- whiskcol - ウィスカーの

---

## ステープル

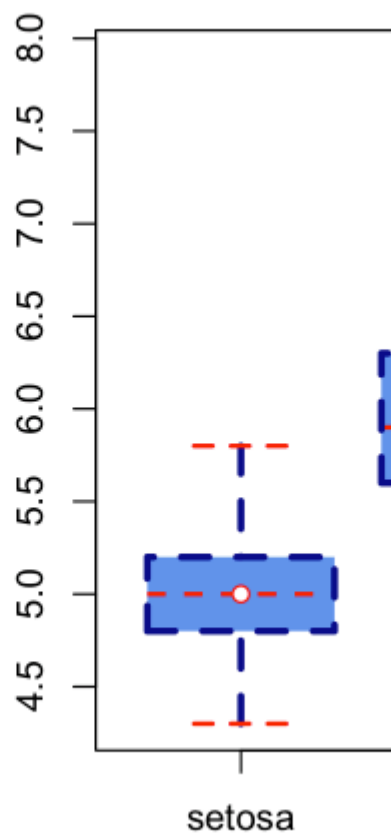
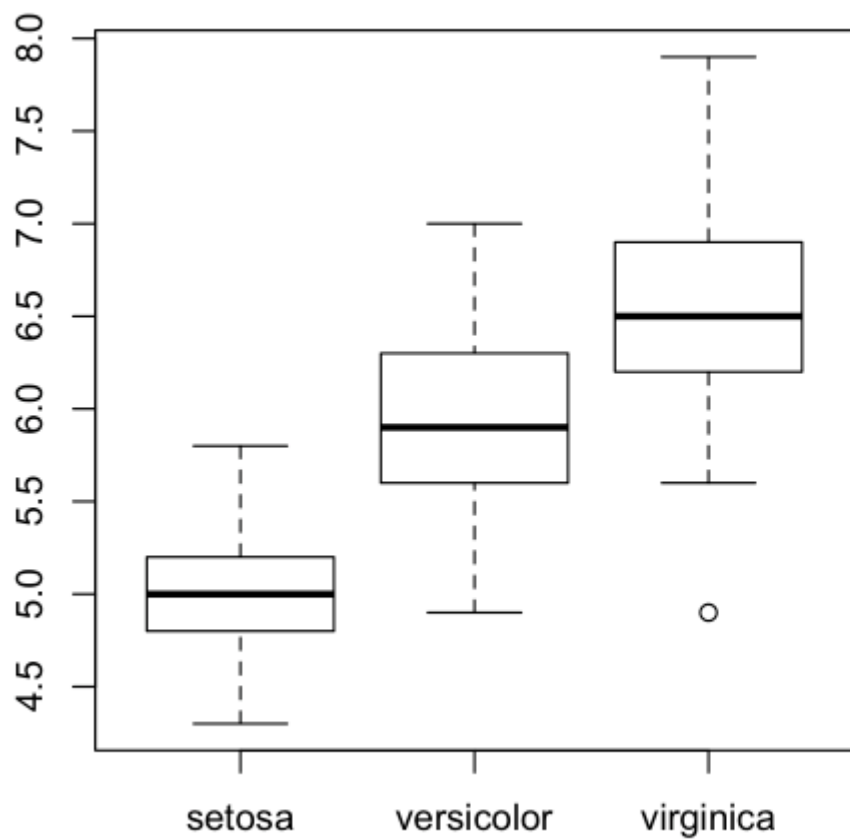
- ステープルめ - ステープルラインタイプ
- staplelwd - ステープルの
- ステープルコール - ステープルラインの

- 
- アウトアウト - アウトライアラインタイプラインがないは "ブランク"
  - outlwd - アウトライア
  - outcol - アウトライアの
  - アウトプット - アウトライアポイントタイプシンボルなしのはNA
  - アウトセックス - アウトリライアポイントサイズ
  - - れポイントの

---

およびきくされたプロットをべて

```
par(mfrow=c(1,2))
Default
boxplot(Sepal.Length ~ Species, data=iris)
Modified
boxplot(Sepal.Length ~ Species, data=iris,
 boxlty=2, boxlwd=3, boxfill="cornflowerblue", boxcol="darkblue",
 medlty=2, medlwd=2, medcol="red", medpch=21, medcex=1, medbg="white",
 whisklty=2, whisklwd=3, whiskcol="darkblue",
 staplelty=2, staplelwd=2, staplecol="red",
 outlty=3, outlwd=3, outcol="grey", outpch=NA
)
```



オンラインでボックスプロットをむ <https://riptutorial.com/ja/r/topic/1005/ボックスプロット>

---

## 70: メタドキュメンテーションのガイドライン

Rタグのドキュメントのについてするには、[Rチャット](#)をご覧ください。

### Examples

いをる

QAのいをするためののほとんどは、ドキュメントにきがれています。

- にして、ポイントにしてください。やはです。
- コードとそれをするのをめる。どちらもではではありません。
- データのソースにしないでください。であれば、データをするか、データセットライブラリをします。

```
library(help = "datasets")
```

ドキュメントのコンテキストには、いくつかのがあります。

- するはいつでも`?data.frame`のようなみみのドキュメントをしてください。SO Docsは、みみのをきえるみではありません。しいRユーザーには、みみのドキュメントがすることと、それらのドキュメントをつけるをっておくことができます。
- のにされるコンテンツを[]セクションにします。

スタイル

---

## プロンプト

コードをコピー・ペーストにするは、のめに`R>`、`>`、`+`などのプロンプトをします。ドキュメントのには、にコピーりをしたくないものもあります。これはです。

---

## コンソール

コンソールのはコードとはっきりとするがあります。なアプローチはのとおりです。

- にプロンプトをしますコンソールのにされます。
- `#`または`##`をして、すべてのをコメントアウトします。
- のままし、`[1]`をして、をからたせます。
- コードとコンソールのにをします。

---

## りて

=と<-は、Rオブジェクトのりてにはありません。 `x<-1` `x <- 1`と`x < -1`にあいまいなど、がなコードをくのをけるために、をにしてください。

---

## コードコメント

コードのとについてしてください。このがかコードコメントであるべきかどうかについては、く  
ていはありません。はみやすくなり、がくなりますが、コードコメントはにコピーしてりけるこ  
とができます。のオプションをにいてください。

---

## セクション

くのはセクションをとしないくらいですが、するはH1からめます。

オンラインでメタドキュメンテーションのガイドラインをむ [https://riptutorial.com/ja/r/topic/5410/  
メタ-ドキュメンテーションのガイドライン](https://riptutorial.com/ja/r/topic/5410/メタ-ドキュメンテーションのガイドライン)

## 71: ユーザーをする

- `<- readline(prompt = "のメッセージ")`
- `name <- readline(prompt = "あなたのはですか")`

### Examples

#### Rのユーザ

RとRをえるようにされたスワールパッケージがユーザーとプログラムののにクロストークをつのはいことがあります。

`readline` コマンドをしてユーザーのをめることができます。

```
name <- readline(prompt = "What is your name?")
```

ユーザーは、、、ベクトルなどののをえることができ、をスキャンすることは、ユーザーがなえをえていることをするためにここにあります。えば

```
result <- readline(prompt = "What is the result of 1+1?")
while(result!=2){
 readline(prompt = "Wrong answer. What is the result of 1+1?")
}
```

しかし、ユーザがとしてされるので、このコードはわりのないループにまらないことにしてください。

`as.numeric` をってに `as.numeric` ます

```
result <- as.numeric(readline(prompt = "What is the result of 1+1?"))
while(result!=2){
 readline(prompt = "Wrong answer. What is the result of 1+1?")
}
```

オンラインでユーザーをするをむ <https://riptutorial.com/ja/r/topic/5098/ユーザーをする>



## 72: ラスターイメージのI/O

き

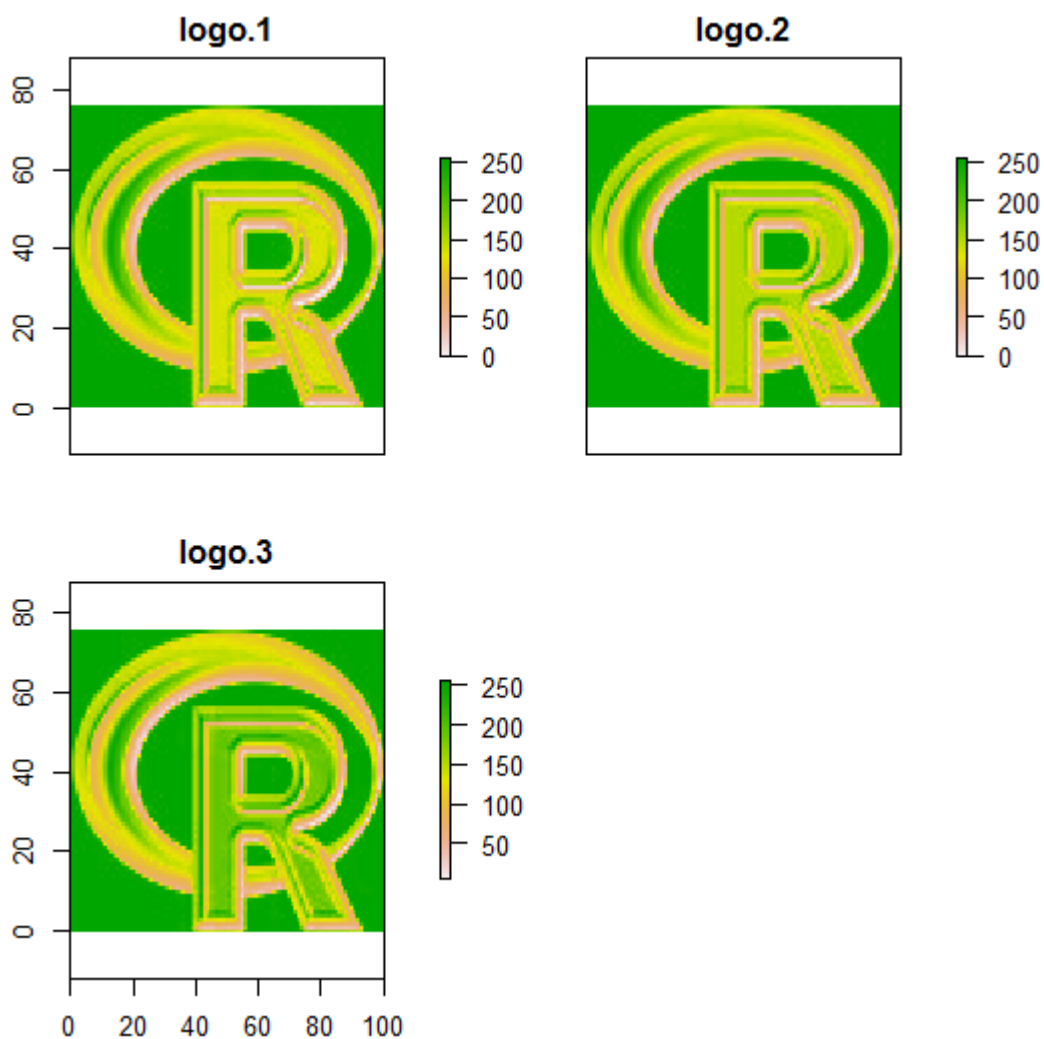
ラスタとイメージのとともにしてください。

### Examples

ラスタをみる

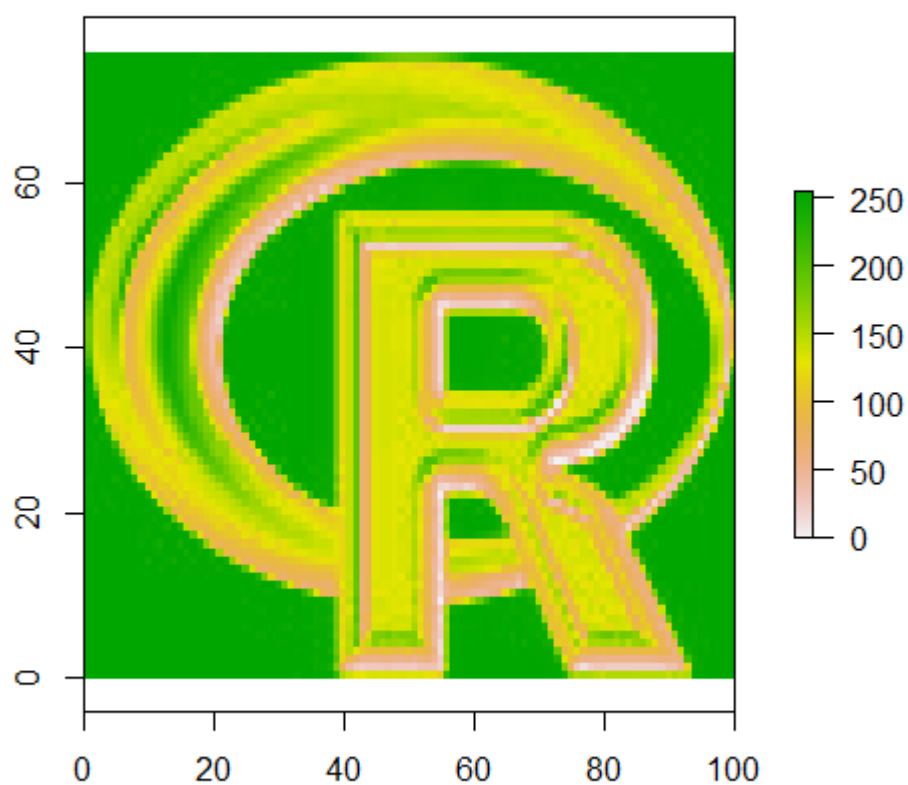
R-Logoはラスタファイル、、です。

```
library(raster)
r <- stack("C:/Program Files/R/R-3.2.3/doc/html/logo.jpg")
plot(r)
```



RasterStack オブジェクトの々のレイヤーは、 [[。

```
plot(r[[1]])
```



オンラインでラスターイメージのI/Oをむ <https://riptutorial.com/ja/r/topic/5539/ラスターイメージのi-o>

## 73: ラスタと

き

ラスタのI/Oもしてください。

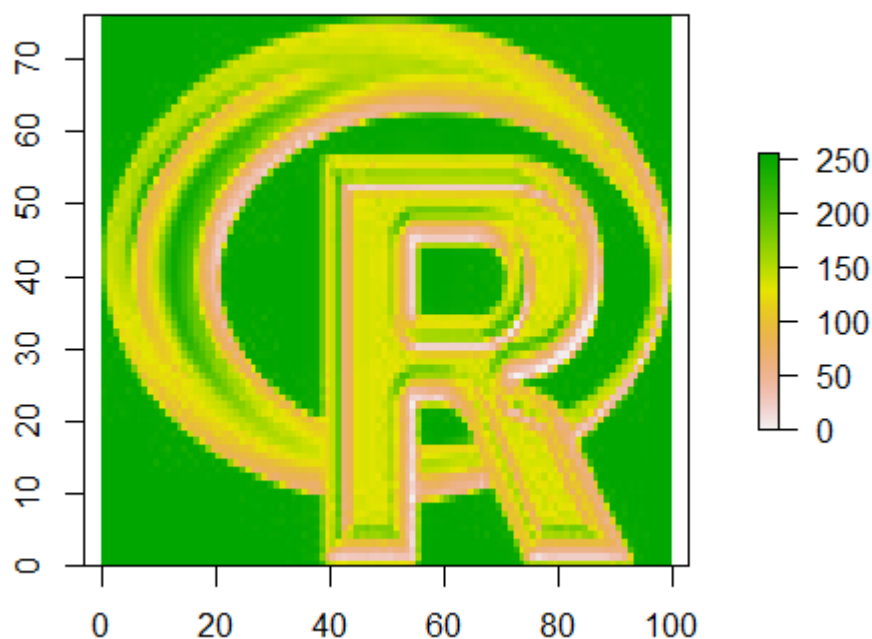
### Examples

#### GLCM テクスチャの

グレースケール Haralick et al. 1973は、のためのなである。 `glcm` パッケージは、Rの `RasterLayer` オブジェクトのそのようなをするためのしやすいを `RasterLayer` ます。

```
library(glcm)
library(raster)

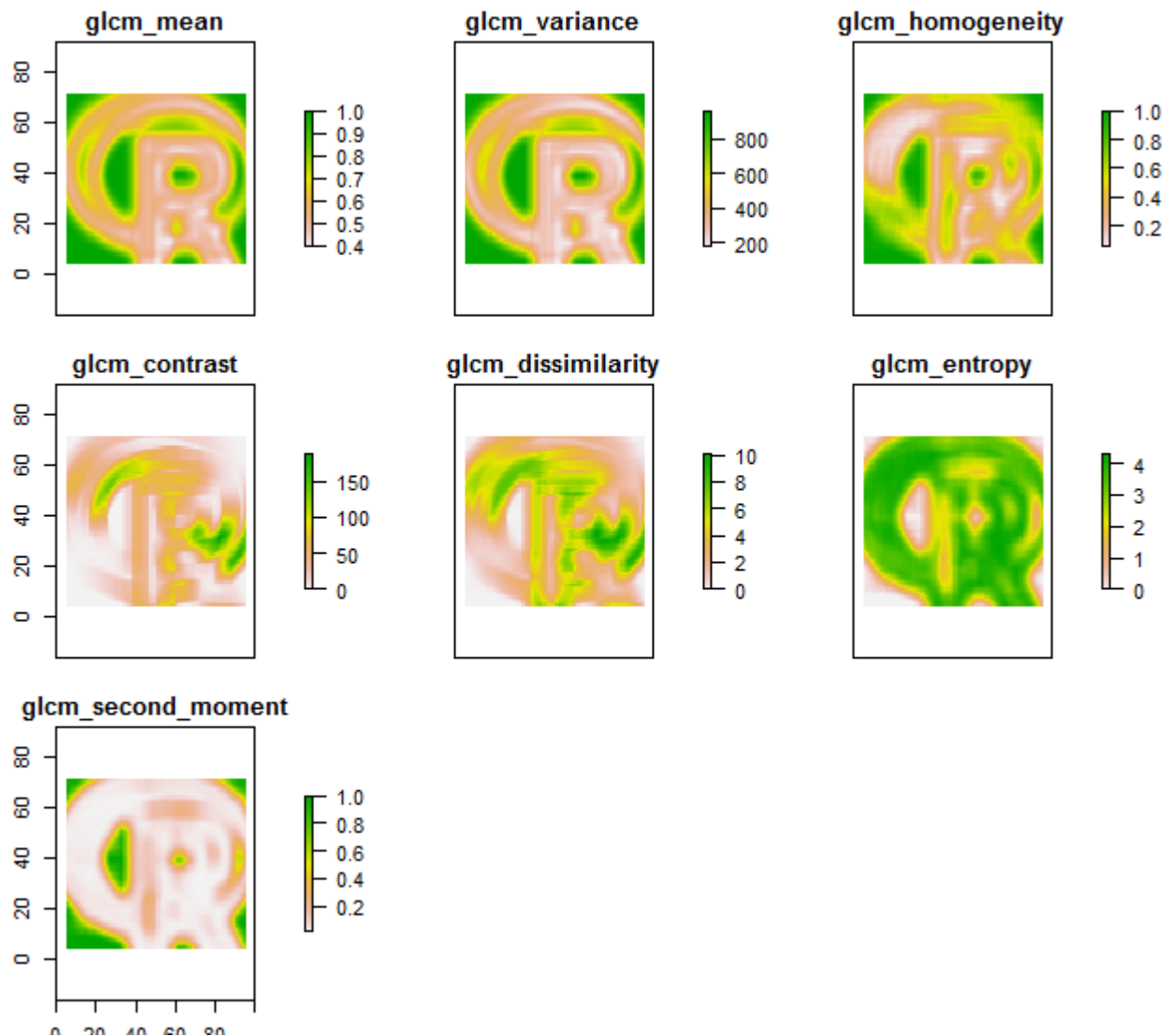
r <- raster("C:/Program Files/R/R-3.2.3/doc/html/logo.jpg")
plot(r)
```



#### のGLCM テクスチャの

```
rglcm <- glcm(r,
 window = c(9,9),
 shift = c(1,1),
 statistics = c("mean", "variance", "homogeneity", "contrast",
 "dissimilarity", "entropy", "second_moment")
)
```

```
plot(rglcm)
```

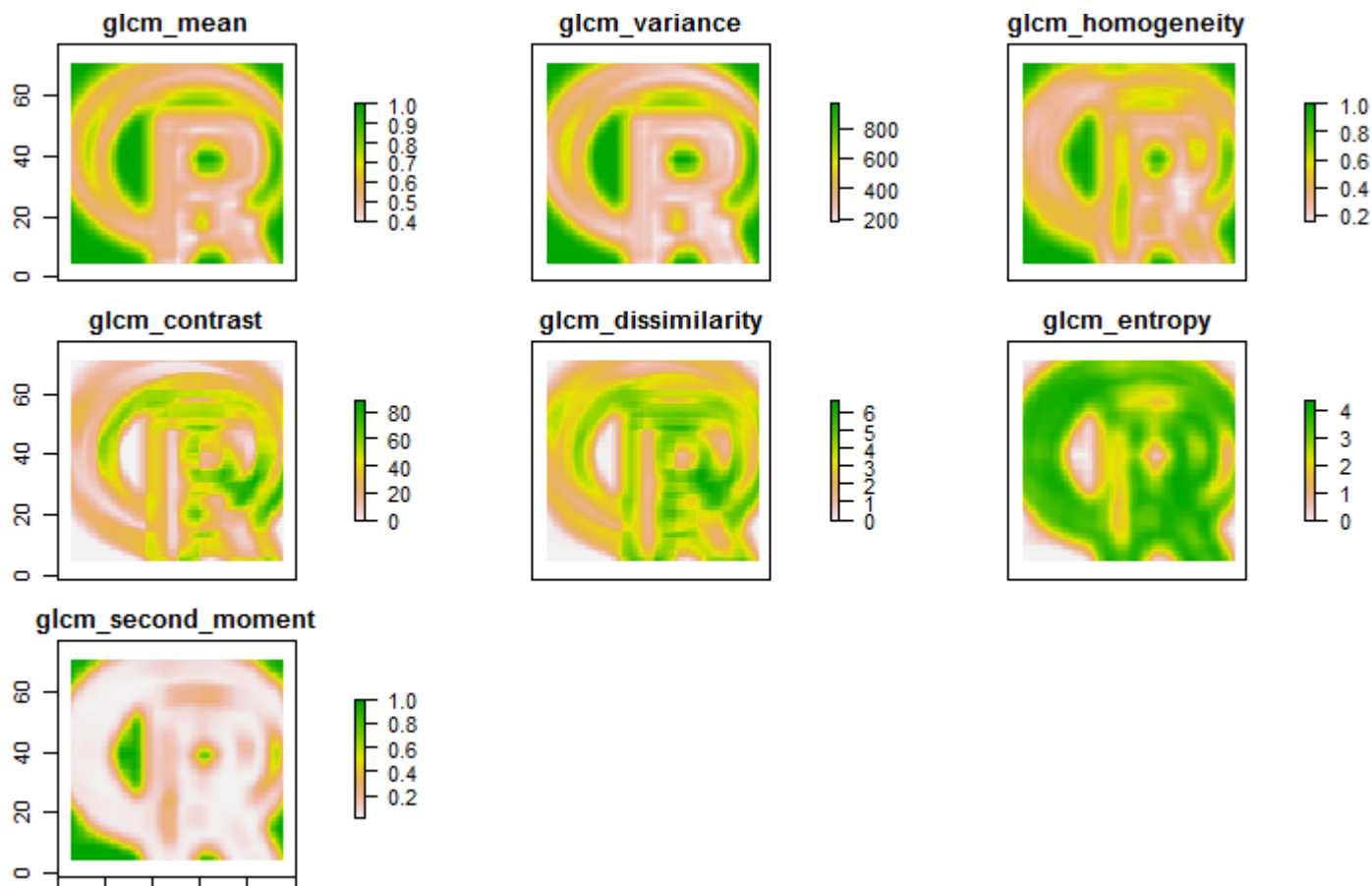


テクスチャの

テクスチャのは、4つの0°、45°、90°、135°でし、の1つのテクスチャにすることもできます。  
 このキーは`shift`パラメータです

```
rglcm1 <- glcm(r,
 window = c(9,9),
 shift=list(c(0,1), c(1,1), c(1,0), c(1,-1)),
 statistics = c("mean", "variance", "homogeneity", "contrast",
 "dissimilarity", "entropy", "second_moment")
)

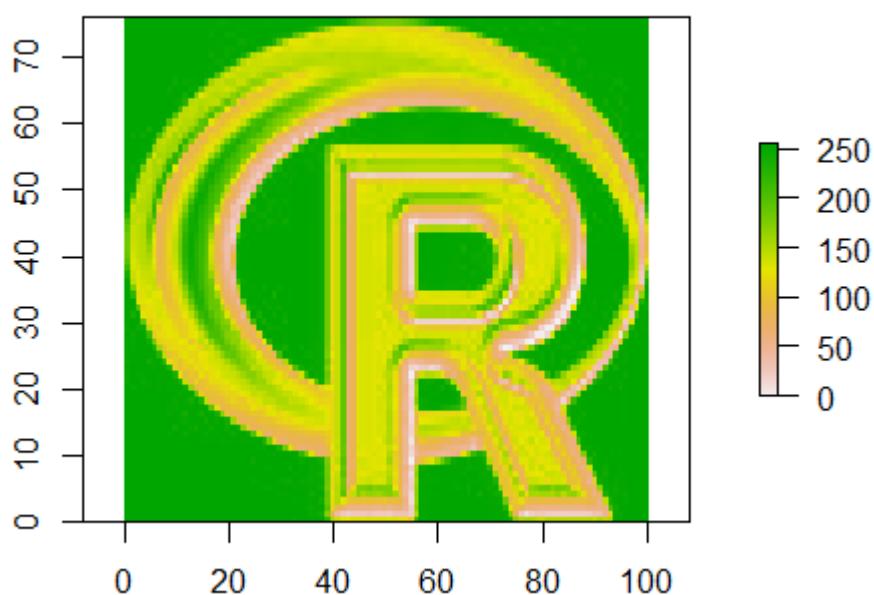
plot(rglcm1)
```



パッケージ `mmmand` は、`n` のののためのをします。ちょっとしたをって、ラスタースタイルのもです。

```
library(raster)
library(mmmand)

r <- raster("C:/Program Files/R/R-3.2.3/doc/html/logo.jpg")
plot(r)
```



に、カーネルウィンドウは、サイズえば、9x9およびタイプえば、 `disc`、 `box`または`diamond` されなければならない、

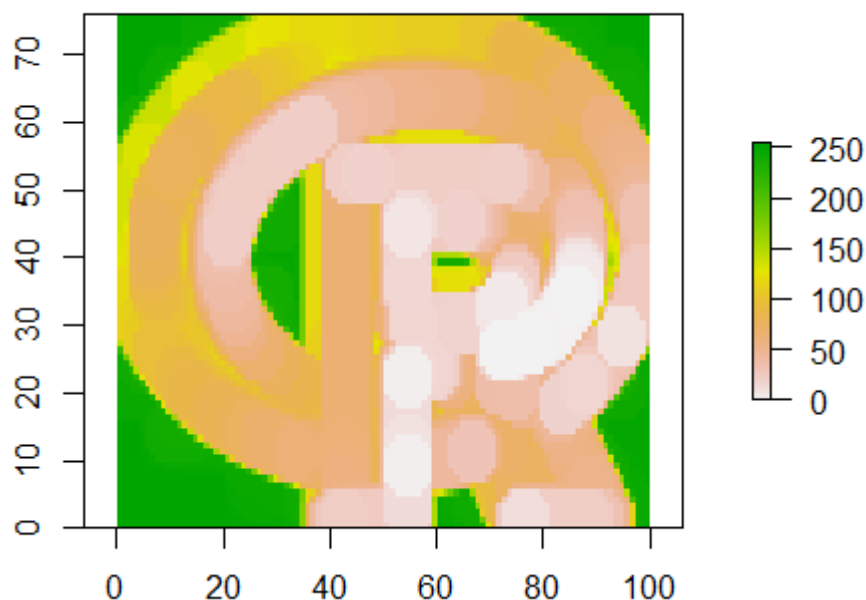
```
sk <- shapeKernel(c(9,9), type="disc")
```

その、ラスタは、 `erode()` のとしてされるにすることがあります。

```
rArr <- as.array(r, transpose = TRUE)
rErode <- erode(rArr, sk)
rErode <- setValues(r, as.vector(aperm(rErode)))
```

`erode()` に、 `dilate()`、 `opening()`、 `closing()` をこのようにすることができます。

```
plot(rErode)
```



オンラインでラスタとをむ <https://riptutorial.com/ja/r/topic/3726/ラスタと>

## 74: ランダムフォレストアルゴリズム

き

RandomForestは、データのオーバーフィットのをらすまたはのアンサンブルです。こののは、[Wikipediaのランダムフォレストのをしてください](#)。RのなはrandomForestパッケージにあります、のもあります。 [のCRANビューを](#)してください。

### Examples

な - と

```
Used for both Classification and Regression examples
library(randomForest)
library(car) ## For the Soils data
data(Soils)

#####
RF Classification Example
set.seed(656) ## for reproducibility
S_RF_Class = randomForest(Gp ~ ., data=Soils[,c(4,6:14)])
Gp_RF = predict(S_RF_Class, Soils[,6:14])
length(which(Gp_RF != Soils$Gp)) ## No Errors

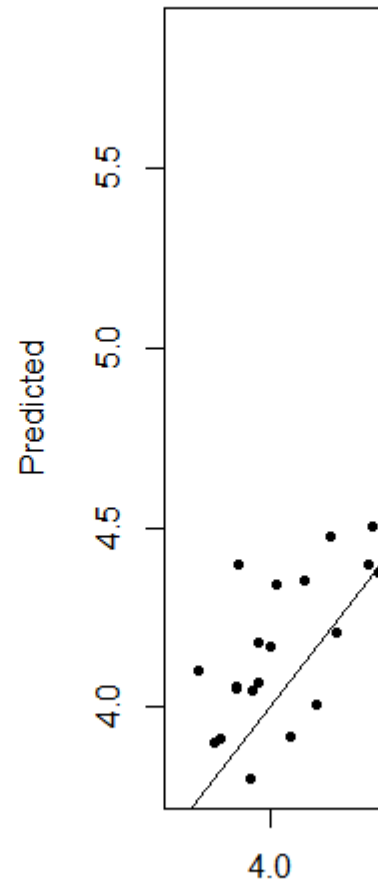
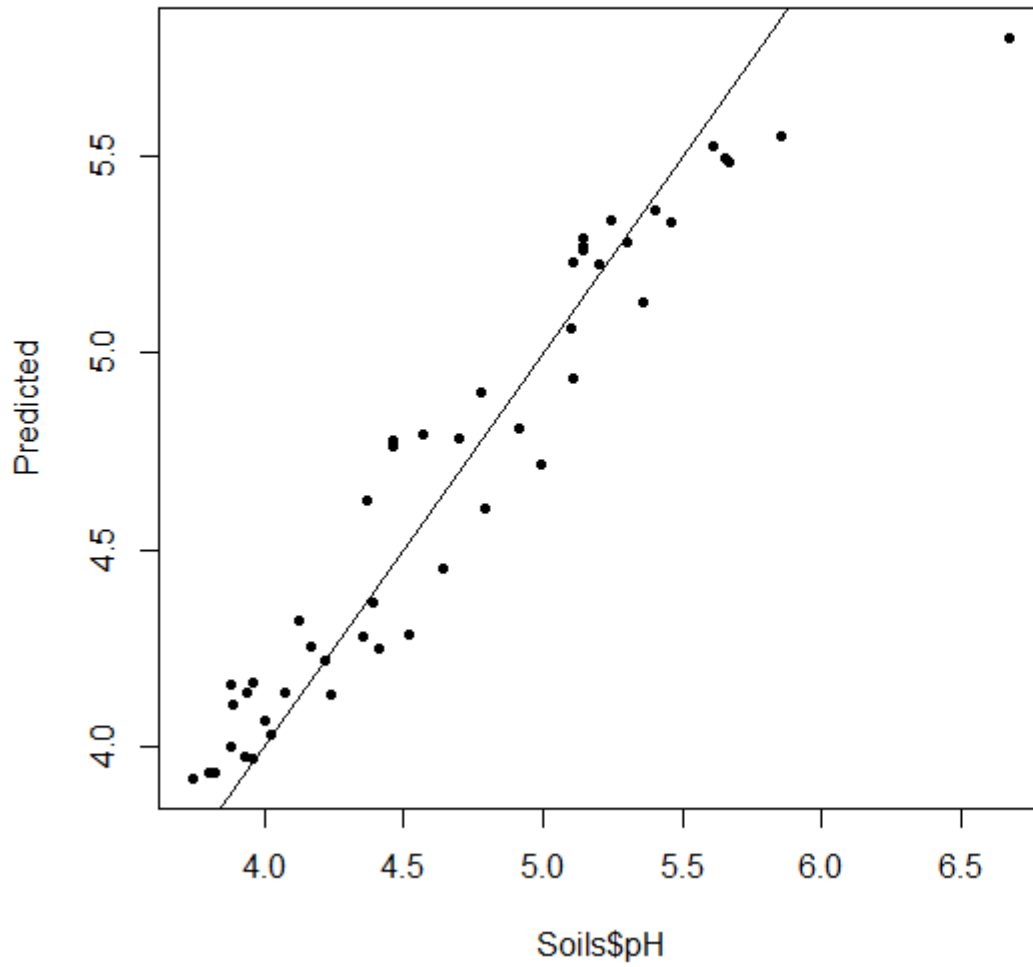
Naive Bayes for comparison
library(e1071)
S_NB = naiveBayes(Soils[,6:14], Soils[,4])
Gp_NB = predict(S_NB, Soils[,6:14], type="class")
length(which(Gp_NB != Soils$Gp)) ## 6 Errors
```

このではトレーニングデータをテストしましたが、RFがにれたモデルをできることをしています。

```
#####
RF Regression Example
set.seed(656) ## for reproducibility
S_RF_Reg = randomForest(pH ~ ., data=Soils[,6:14])
pH_RF = predict(S_RF_Reg, Soils[,6:14])

Compare Predictions with Actual values for RF and Linear Model
S_LM = lm(pH ~ ., data=Soils[,6:14])
pH_LM = predict(S_LM, Soils[,6:14])
par(mfrow=c(1,2))
plot(Soils$pH, pH_RF, pch=20, ylab="Predicted", main="Random Forest")
abline(0,1)
plot(Soils$pH, pH_LM, pch=20, ylab="Predicted", main="Linear Model")
abline(0,1)
```

## Random Forest



オンラインでランダムフォレストアルゴリズムをむ <https://riptutorial.com/ja/r/topic/8088/ランダムフォレストアルゴリズム>



## 75: ランダム

き

にはRがにされます。したがって、ランダムのためのなオプションセットがまれています。からのサンプリングにするなについては、 [の](#)ドキュメントをしてください。

のプログラミングからているユーザーは、したこととの`rand`がないためにすることがあります。なをしてわれる`r*`のためののファミリーをのリンクをします。あるからにされたは、「の」のために`runif`をしてできます。これは "する"のようにわしいようにもえるので、しいRユーザーのためにすることはしばしばです。

## Examples

ランダムと

`sample` コマンドをして、のにかかわらず、からのやランダムするなど、なのをシミュレートすることが出来ます。

このでは、`set.seed`をして、サンプルコードをにできるようにしています。ただし、`sample`は`set.seed`をにびさなくても`set.seed`ます。

## ランダム

もなでは、`sample`はベクトルのランダムをします。これはのでできます。

```
set.seed(1251)
sample(x = 10)

[1] 7 1 4 8 6 3 10 5 2 9
```

のがえられていない、`sample`は1から`x`までのベクトルのランダムをします。これは、データフレームののをランダムしようとするときにです。これは、トライアルのランダムテーブルをする、またはのためにランダムなサブセットのをするのなです。

```
library(datasets)
set.seed(1171)
iris_rand <- iris[sample(x = 1:nrow(iris)),]

> head(iris)
 Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1 5.1 3.5 1.4 0.2 setosa
2 4.9 3.0 1.4 0.2 setosa
3 4.7 3.2 1.3 0.2 setosa
4 4.6 3.1 1.5 0.2 setosa
```

```

5 5.0 3.6 1.4 0.2 setosa
6 5.4 3.9 1.7 0.4 setosa

> head(iris_rand)
 Sepal.Length Sepal.Width Petal.Length Petal.Width Species
145 6.7 3.3 5.7 2.5 virginica
5 5.0 3.6 1.4 0.2 setosa
85 5.4 3.0 4.5 1.5 versicolor
137 6.3 3.4 5.6 2.4 virginica
128 6.1 3.0 4.9 1.8 virginica
105 6.5 3.0 5.8 2.2 virginica

```

## えなしでく

`sample`をして、のにかかわらずセットからをシミュレートすることもできます。きえずにサンプリングするにはデフォルト、サンプルをするセットとでするがあります。されるセットはベクトルとしてえられます。

```

set.seed(7043)
sample(x = LETTERS,size = 7)

[1] "S" "P" "J" "F" "Z" "G" "R"

```

`size`のが`x`ののさと同じであれば、ランダムをしていることにしてください。また、せずにサンプリングをうときは、`x`のさよりもきなサイズをすることはできません。

```

set.seed(7305)
sample(x = letters,size = 26)

[1] "x" "z" "y" "i" "k" "f" "d" "s" "g" "v" "j" "o" "e" "c" "m" "n" "h" "u" "a" "b" "l" "r"
"w" "t" "q" "p"

sample(x = letters,size = 30)
Error in sample.int(length(x), size, replace, prob) :
 cannot take a sample larger than the population when 'replace = FALSE'

```

これはでにちむ。

## でく

きえてセットからランダムにするには、`replace`をして`sample`ます。デフォルトでは、`replace`は`FALSE`です。`TRUE`すると、されるセットののがにされることがあります。

```

set.seed(5062)
sample(x = c("A","B","C","D"),size = 8,replace = TRUE)

[1] "D" "C" "D" "B" "A" "A" "A" "A"

```

# をえる

デフォルトでは、`sample` をすると、をえるはじであるとみなされます。それをな "urn" とえてください。のコードは、からされたを20き、をきめてから、をのにすのとじです。には、、のがそれぞれ1つずつつっています。つまり、をくは1/3です。

```
set.seed(6472)
sample(x = c("Red", "Blue", "Green"),
 size = 20,
 replace = TRUE)
```

わりにじタスクをしたいとしますが、たちのには2つのい、1つのい、1つののがまれています。1つのオプションは、`x` にするをして、の`Red` をすることです。ただし、よりいは、`sample prob` をすることです。

`prob` は、をえるをつベクトルをけります。のでは、のをくは1/2になりますが、またはのをくは1/4になります。

```
set.seed(28432)
sample(x = c("Red", "Blue", "Green"),
 size = 20,
 replace = TRUE,
 prob = c(0.50, 0.25, 0.25))
```

には、`prob` えられたは1になるはありません。Rはえられたをに1になるにします。えば、のでは2 Red、1 Blue、1 Greenをえてみましょう。これらのをって、のコードとじをることができま

```
set.seed(28432)
frac_prob_example <- sample(x = c("Red", "Blue", "Green"),
 size = 200,
 replace = TRUE,
 prob = c(0.50, 0.25, 0.25))

set.seed(28432)
numeric_prob_example <- sample(x = c("Red", "Blue", "Green"),
 size = 200,
 replace = TRUE,
 prob = c(2, 1, 1))

> identical(frac_prob_example, numeric_prob_example)
[1] TRUE
```

きなは、すべてのをゼロにすることはできず、ゼロのもゼロにすることはできないということです。

`replace` がFALSE されているFALSE `prob` することもできFALSE。そのでは、がかれた、りのの`prob` ののがのをえる。このでは、するサンプルの`size` にするのになゼロをっていなければなりません。えば

```
set.seed(21741)
sample(x = c("Red", "Blue", "Green"),
 size = 2,
 replace = FALSE,
 prob = c(0.8, 0.19, 0.01))
```

このでは、ののとしてでがされます。がされるは80、がされるは19、がされるは1です。

のドローでは、レッドはもはやのにいません。りののは20は19、は1です。このドローでは、アイテムが19/20になるは95、1/20になるは5です。

をする

`set.seed`は、すべてのランダムシードをするためにされます。Rをしてできるランダムをするは、に`set.seed`をするがあります。

```
set.seed(1643)
samp1 <- sample(x = 1:5, size = 200, replace = TRUE)

set.seed(1643)
samp2 <- sample(x = 1:5, size = 200, replace = TRUE)

> identical(x = samp1, y = samp2)
[1] TRUE
```

にはランダムシードのながであることにしてください。

オンラインでランダムをむ <https://riptutorial.com/ja/r/topic/9574/ランダム>

## 76: ランレンジス

ランとは、したまたはのしたシーケンスです。りされるの、Rの「ランレンジス」は、そのにしてベクトルをにします。してください

```
dat <- c(1, 2, 2, 2, 3, 1, 4, 4, 1, 1)
```

たちは、1のさ1のランをとっています。に2のさ3ラン。3のさ1ラン。々。Rのランレンジスエンコーディングは、ベクトルのランのさとをすべてキャプチャします。

ランはまた、データのしたをすることもできます。Rにはこれらをエンコードするなはありませんが、[data.table](#)パッケージは[デッドエンドリンク](#)から`rleid`でできます。

### Examples

`'rle'`によるランレンジス

ランレンジスは、ベクトルのするのランのさをキャプチャします。のベクトルをえてみましょう

```
dat <- c(1, 2, 2, 2, 3, 1, 4, 4, 1, 1)
```

`rle`はとそのさをします

```
r <- rle(dat)
r
Run Length Encoding
lengths: int [1:6] 1 3 1 1 2 2
values : num [1:6] 1 2 3 1 4 1
```

のは`r$values`で`r$values`され`r$values`。

```
r$values
[1] 1 2 3 1 4 1
```

これは、に1のランをた、2のランをした、3をランし、に1をランしたことなどをキャプチャします。

ランのさは`r$lengths`キャプチャされます

```
r$lengths
[1] 1 3 1 1 2 2
```

々は、1ののランはさ1であり、いて2のランはさ3であった、というである。

## Rでのによるとグループ

のによってデータをグループして、らかのをすることができます。のなデータセットをえてみましょう。

```
(dat <- data.frame(x = c(1, 1, 2, 2, 2, 1), y = 1:6))
x y
1 1 1
2 1 2
3 2 3
4 2 4
5 2 5
6 1 6
```

$x$ は3つのランがあります。さ2の1、さ3の2、さ1の1のランです。 $y$ を $x$ これらののは1.5,4、および6です。

Rでは、に`rle`をとって $x$ のランレングスをします。

```
(r <- rle(dat$x))
Run Length Encoding
lengths: int [1:3] 2 3 1
values : num [1:3] 1 2 1
```

のステップは、データセットののをすることです。ランのは`length(r$lengths)`であり、ランの`r$lengths`は`r$lengths` `length`であることがわかっているので、ランのを`rep`ですることができます。

```
(run.id <- rep(seq_along(r$lengths), r$lengths))
[1] 1 1 2 2 2 3
```

は、`tapply`をして、IDをグループしての $y$ をすることができます。

```
data.frame(x=r$values, meanY=tapply(dat$y, run.id, mean))
x meanY
1 1 1.5
2 2 4.0
3 1 6.0
```

## data.tableでのによるとグループ

`data.table`パッケージは、データをランでグループするなをします。のデータをえてみましょう。

```
library(data.table)
(DT <- data.table(x = c(1, 1, 2, 2, 2, 1), y = 1:6))
x y
1: 1 1
2: 1 2
3: 2 3
4: 2 4
5: 2 5
6: 1 6
```

$x$ は3つのランがあります。さ2の1、さ3の2、さ1の1のランです。 $y$ を $x$ のこれらのは1.5,4、および6です。

`data.table` `rleid`は、ベクトルのIDをすIDをします。

```
rleid(DT$x)
[1] 1 1 2 2 2 3
```

このIDをにグループし、 $y$ データをすることができます。

```
DT[,mean(y),by=.(x, rleid(x))]
x rleid V1
1: 1 1 1.5
2: 2 2 4.0
3: 1 3 6.0
```

ベクトルをおよびするランレングス

じのいランをついベクトルは、ランレングスのエンコーディングランのとそのがりされるにすることです。として、1のなと0のわずかなをつ1000のさのベクトルをえてみましょう

```
set.seed(144)
dat <- sample(rep(0:1, c(1, 1e5)), 1e7, replace=TRUE)
table(dat)
0 1
103 9999897
```

1000エントリをするにはかなりのスペースがですが、わりにこのベクトルのランレングスエンコーディングをしてデータフレームをすることができます。

```
rle.df <- with(rle(dat), data.frame(values, lengths))
dim(rle.df)
[1] 207 2
head(rle.df)
values lengths
1 1 52818
2 0 1
3 1 219329
4 0 1
5 1 318306
6 0 1
```

ランレングスから、ベクトルの52,818のは1であり、それに1つの0、219,329のする1、いて0などがあることがわかります。ランレングスエンコーディングは207エントリしかなく、1000のわりに414のをするがあります。`rle.df`はデータフレームな`write.csv`ようなをとてすることができます。

ランレングスでベクトルをすることは、2つのですることができる。のは、にびすことである`rep`し、`values`のとしてランレングスのを`lengths`としてランレングスの

```
decompressed <- rep(rle.df$values, rle.df$lengths)
```

されたデータはのデータとであることができます。

```
identical(decompressed, dat)
[1] TRUE
```

のは、することであるRのみみinverse.rleにrleインスタンスのオブジェクト。

```
rle.obj <- rle(dat) # create a rle object here
class(rle.obj)
[1] "rle"

dat.inv <- inverse.rle(rle.obj) # apply the inverse.rle on the rle object
```

々は、これはにすることをすることができdat

```
identical(dat.inv, dat)
[1] TRUE
```

オンラインでランレングスをむ <https://riptutorial.com/ja/r/topic/1133/ランレングス>



## 77: リサイクル

Rのリサイクルはですか

リサイクルは、のでオブジェクトがにされて、のいオブジェクトのさにするです。

えは、ベクトルされたのはのようになります。

```
c(1,2,3) + c(1,2,3,4,5,6)
[1] 2 4 6 5 7 9
```

リサイクルのために、にこったは

```
c(1,2,3,1,2,3) + c(1,2,3,4,5,6)
```

よりいオブジェクトがいオブジェクトのでない、メッセージがされます。

```
c(1,2,3) + c(1,2,3,4,5,6,7)
[1] 2 4 6 5 7 9 8
Warning message:
In c(1, 2, 3) + c(1, 2, 3, 4, 5, 6, 7) :
 longer object length is not a multiple of shorter object length
```

リサイクルのもうつの

```
matrix(nrow =5, ncol = 2, 1:5)
 [,1] [,2]
[1,] 1 1
[2,] 2 2
[3,] 3 3
[4,] 4 4
[5,] 5 5
```

## Examples

サブセッティングでのリサイクル

リサイクルは、コードをするためのなでできます。

サブセッティング

ベクトルの3のをすべてしたい、のことができます

```
my_vec <- c(1,2,3,4,5,6,7,8,9,10)
my_vec[c(TRUE, FALSE)]

[1] 1 3 5 7 9
```

ここではベクトルのさにされました。

また、リサイクルをしてをうこともできます。

```
my_vec <- c("foo", "bar", "soap", "mix")
my_vec == "bar"
```

```
[1] FALSE TRUE FALSE FALSE
```

ここで「バー」はリサイクルされます。

オンラインでリサイクルをむ <https://riptutorial.com/ja/r/topic/5649/リサイクル>

## 78: リスト

### Examples

リストへのな

に、ユーザーとしてするオブジェクトのはベクトルになりがちです。例えば、ベクトル、ベクトル。これらのオブジェクトは、1つのタイプののみをすることができますベクトルはにをつことができます。

リストにはのをすることができ、なののをできるオブジェクトにすることができます。

リストのの

```
exampleList1 <- list('a', 'b')
exampleList2 <- list(1, 2)
exampleList3 <- list('a', 1, 2)
```

リストにされているデータをするために、strをすることができます。

```
str(exampleList1)
str(exampleList2)
str(exampleList3)
```

リストのサブセットは、リストのスライスをすること、すなわち、のリストののサブセットをむリストをすること、およびのをすることをする。[ベクトルににされるをすると、しいリストがされます。

```
Returns List
exampleList3[1]
exampleList3[1:2]
```

のをるには、[[

```
Returns Character
exampleList3[[1]]
```

リストのはのとおりで。

```
exampleList4 <- list(
 num = 1:3,
 numeric = 0.5,
 char = c('a', 'b')
)
```

きリストのエントリは、インデックスではなくでアクセスできます。

```
exampleList4[['char']]
```

わりに、`$`をしてきにアクセスすることもできます。

```
exampleList4$num
```

これは、タイプするがく、みやすくなるというがありますが、なとしにつくことができます。`$`はをしてするリストをし、しないをするがあります。

```
exampleList5 <- exampleList4[2:3]
```

```
exampleList4$num
```

```
c(1, 2, 3)
```

```
exampleList5$num
```

```
0.5
```

```
exampleList5[['num']]
```

```
NULL
```

リストは、さまざまなオブジェクトやさまざまなクラスのオブジェクトをできるため、にです。

```
Numeric vector
exampleVector1 <- c(12, 13, 14)
Character vector
exampleVector2 <- c("a", "b", "c", "d", "e", "f")
Matrix
exampleMatrix1 <- matrix(rnorm(4), ncol = 2, nrow = 2)
List
exampleList3 <- list('a', 1, 2)
```

```
exampleList6 <- list(
 num = exampleVector1,
 char = exampleVector2,
 mat = exampleMatrix1,
 list = exampleList3
)
exampleList6
#$num
#[1] 12 13 14
#
#$char
#[1] "a" "b" "c" "d" "e" "f"
#
#$mat
[,1] [,2]
#[1,] 0.5013050 -1.88801542
#[2,] 0.4295266 0.09751379
#
#$list
#$list[[1]]
#[1] "a"
#
#$list[[2]]
#[1] 1
```

```
#
#$list[[3]]
#[1] 2
```

## リストの

リストをすると、のベクトルやなどをオブジェクトのにできます。 `list` をして `list` をすることができます

```
l1 <- list(c(1, 2, 3), c("a", "b", "c"))
l1
[[1]]
[1] 1 2 3
##
[[2]]
[1] "a" "b" "c"
```

のリストをするベクトルはなるクラスであることにしてください。リストをすると、なるクラスのをグループできます。リストのにはをけることもできます。リストは `names` によってアクセスされ、じでされ、とのがにりてられます。

```
names(l1)
NULL
names(l1) <- c("vector1", "vector2")
l1
$vector1
[1] 1 2 3
##
$vector2
[1] "a" "b" "c"
```

リストオブジェクトのにリストをするがです。

```
l2 <- list(vec = c(1, 3, 5, 7, 9),
 mat = matrix(data = c(1, 2, 3), nrow = 3))
l2
$vec
[1] 1 3 5 7 9
##
$mat
[,1]
[1,] 1
[2,] 2
[3,] 3
names(l2)
[1] "vec" "mat"
```

リストのには、"vec"と "mat"という2つのがあり、ベクトルとがされています。

## リストをする

Rユーザにとって、リストは、するよりなデータの1つであるようにえるかもしれない。そののす

べてのがじであるというはありません。リストがどれほど/であるかのされたはありませんリストのはリストになるがあります

しかし、なの1つは、リストをしてでパラメーターをすときです。

```
Function example which returns a single element numeric vector
exampleFunction1 <- function(num1, num2){
 result <- num1 + num2
 return(result)
}

Using example function 1
exampleFunction1(1, 2)

Function example which returns a simple numeric vector
exampleFunction2 <- function(num1, num2, multiplier){
 tempResult1 <- num1 + num2
 tempResult2 <- tempResult1 * multiplier
 result <- c(tempResult1, tempResult2)
 return(result)
}

Using example function 2
exampleFunction2(1, 2, 4)
```

のでは、されるはなるなベクトルです。このようななベクトルをすはありません。

これで、Rはに1つのしかさないことにしてくださいなるをすがあるにできます。しかし、パラメータのセットをとり、ベクトルやデータフレームなどのいくつかのタイプののをすをするは、これらののをすべてリストにダンプするがありますそれをすに。

```
We will be using mtcars dataset here
Function which returns a result that is supposed to contain multiple type of results
This can be solved by putting the results into a list
exampleFunction3 <- function(dataframe, removeColumn, sumColumn){
 resultDataFrame <- dataframe[, -removeColumn]
 resultSum <- sum(dataframe[, sumColumn])
 resultList <- list(resultDataFrame, resultSum)
 return(resultList)
}

Using example function 3
exampleResult <- exampleFunction3(mtcars, 2, 4)
exampleResult[[1]]
exampleResult[[2]]
```

のリストをしなからリストをベクトルにする

リストをvectorまたはdata.frameオブジェクトにする、、のはされます。

これは、いくつかののえは、nのをつリストがmxn、data.frame、またはdata.tableにされるようにされたリストで、のさのリストがされるというがあります。のをしながら、リストをベクトルにですすることはです。

```
res <- list(character(0), c("Luzhuang", "Laisu", "Peihui"), character(0),
 c("Anjiangping", "Xinzhai", "Yongfeng"), character(0), character(0),
 c("Puji", "Gaotun", "Banjingcun"), character(0), character(0),
 character(0))
res
```

```
[[1]]
character(0)

[[2]]
[1] "Luzhuang" "Laisu" "Peihui"

[[3]]
character(0)

[[4]]
[1] "Anjiangping" "Xinzhai" "Yongfeng"

[[5]]
character(0)

[[6]]
character(0)

[[7]]
[1] "Puji" "Gaotun" "Banjingcun"

[[8]]
character(0)

[[9]]
character(0)

[[10]]
character(0)
```

```
res <- sapply(res, function(s) if (length(s) == 0) NA_character_ else paste(s, collapse = "
"))
res
```

```
[1] NA "Luzhuang Laisu Peihui" NA
"Anjiangping Xinzhai Yongfeng" NA

[6] NA "Puji Gaotun Banjingcun" NA
NA NA
```

## シリアライゼーションリストをとってをす

なるのデータをまとめるがあるがあります。たとえば、Azure MLでは、Rスクリプトモジュールからのを、データフレームだけをしてのモジュールにすがあります。データフレームとがあるとします。

```
> df
 name height team fun_index title age desc Y
1 Andrea 195 Lazio 97 6 33 eccellente 1
```

```

2 Paja 165 Fiorentina 87 6 31 deciso 1
3 Roro 190 Lazio 65 6 28 strano 0
4 Gioele 70 Lazio 100 0 2 simpatico 1
5 Cacio 170 Juventus 81 3 33 duro 0
6 Edola 171 Lazio 72 5 32 svampito 1
7 Salami 175 Inter 75 3 30 doppiopasso 1
8 Braugo 180 Inter 79 5 32 gjn 0
9 Benna 158 Juventus 80 6 28 esaurito 0
10 Riggio 182 Lazio 92 5 31 certezza 1
11 Giordano 185 Roma 79 5 29 buono 1

> number <- "42"

```

私たちはこのにアクセスすることができます

```

> paste(df$name[4], "is a", df3$team[4], "supporter.")
[1] "Gioele is a Lazio supporter."
> paste("The answer to THE question is", number)
[1] "The answer to THE question is 42"

```

なるタイプのデータをデータフレームに入れるためには、リストオブジェクトとをすることがあります。には、データをリストにのしてから、リストをのデータフレームにするがあります。

```

l <- list(df, number)
dataframe_container <- data.frame(out2 = as.integer(serialize(l, connection=NULL)))

```

をデータフレームにしたら、そのをデシリアライズしてすることがあります。

```

#----- unserialize -----+
unser_obj <- unserialize(as.raw(dataframe_container$out2))
#----- taking back the elements-----+
df_mod <- unser_obj[1][[1]]
number_mod <- unser_obj[2][[1]]

```

に、データがしくされたことをできます。

```

> paste(df_mod$name[4], "is a", df_mod$team[4], "supporter.")
[1] "Gioele is a Lazio supporter."
> paste("The answer to THE question is", number_mod)
[1] "The answer to THE question is 42"

```

オンラインでリストをむ <https://riptutorial.com/ja/r/topic/1365/リスト>



## 79: ロキシーゲン2

### パラメーター

| パラメータ    |                                  |
|----------|----------------------------------|
|          | パッケージの                           |
|          | のは、されたのです                        |
| する       | をエクスポートする - つまり、パッケージのユーザーがびしにする |
| インポート    | インポートするパッケージ                     |
| からのインポート | パッケージからインポートするリストの               |
| パラメータ    | するのパラメータ                         |

## Examples

roxygen2でパッケージをする

## roxygen2でく

roxygen2は、Hadley Wickhamによってされたパッケージです。

これは、#'にRスクリプトのドキュメントをめることをにします。ドキュメントにされるさまざまなパラメータは@であります。えは、パッケージのはのようにかれます

```
#' @author The Author
```

たとえば、のをしたいとします。

```
mean<-function(x) sum(x)/length(x)
```

このにさをいて、のようにパラメータをしたいといますについてはでします。

```
#' Mean
#
#' A function to compute the mean of a vector
#' @param x A numeric vector
#' @keyword mean
#' @importFrom base sum
```

```
#' @export
#' @examples
#' mean(1:3)
#' \dontrun{ mean(1:1e99) }
mean<-function(x) sum(x)/length(x)
```

- の `#' Mean` はドキュメントのタイトルです。のがコーパスをします。
- のパラメータは、する `@param` をしてされなければならない。 `@export` は、このをエクスポートするがあることをしているため、パッケージがロードされたときにびすことができます。
- `@keyword` はヘルプをすにするキーワードをします
- `@importFrom` は、このまたはあなたのパッケージでされるパッケージからインポートするすべてのをリストします。パッケージのなをインポートするには `@import` を `@import` ことができます
- は `@example` タグのにされています。
  - のパッケージは、パッケージのビルドにされます。
  - 2のコマンドは `\dontrun` コマンドのために - いをぐ - はありません。

---

## ドキュメントの

ドキュメントは、 `devtools::document()` をしてできます。また、 `devtools::check()` はにドキュメントをし、のドキュメントでしているをとしてすることにしてください。

オンラインでロキシーゲン2をむ <https://riptutorial.com/ja/r/topic/5171/ロキシーゲン2>

## 80: モデル

### Examples

タイタニックデータセットのロジスティック

ロジスティックは、されたモデルのなケースであり、をモデルするためにされます プロビットおよびログモデルはにしています。

このは、されているリンク、すなわち $\text{logit}$ または $\text{log-odds}$ にします。ロジットのはロジスティックとばれ、のようにえられます。

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

このは、 $-\text{Inf}$ ;  $+\text{Inf}$  [ののをり、 $0$ と $1$ ののをします。ロジスティックはをとり、をす。

ロジスティックは、オプション `family = binomial` ショートカット;  $\text{logit}$ はのデフォルトリンク `family = binomial(link="logit")` の `glm` をしてできます。

このでは、RMS Titanicにしているのをしようとしています。

データをむ

```
url <- "http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/titanic.txt"
titanic <- read.csv(file = url, stringsAsFactors = FALSE)
```

をする

この、をできえます。

```
titanic$age[is.na(titanic$age)] <- mean(titanic$age, na.rm = TRUE)
```

モデルをする

```
titanic.train <- glm(survived ~ pclass + sex + age,
 family = binomial, data = titanic)
```

モデルの

```
summary(titanic.train)
```

```
Call:
glm(formula = survived ~ pclass + sex + age, family = binomial, data = titanic)
```

```
Deviance Residuals:
```

```

 Min 1Q Median 3Q Max
-2.6452 -0.6641 -0.3679 0.6123 2.5615

Coefficients:
 Estimate Std. Error z value Pr(>|z|)
(Intercept) 3.552261 0.342188 10.381 < 2e-16 ***
pclass2nd -1.170777 0.211559 -5.534 3.13e-08 ***
pclass3rd -2.430672 0.195157 -12.455 < 2e-16 ***
sexmale -2.463377 0.154587 -15.935 < 2e-16 ***
age -0.042235 0.007415 -5.696 1.23e-08 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

 Null deviance: 1686.8 on 1312 degrees of freedom
Residual deviance: 1165.7 on 1308 degrees of freedom
AIC: 1175.7

Number of Fisher Scoring iterations: 5

```

- にされるのはコールです。これはモデルとされたオプションをいさせるものです。
- に、モデルのフィットのであるを。こののは、モデルでされた々のケースののをします。
- ののは、 $z$ -にはWald  $z$ -とばれる、およびする $p$ をしています。
  - は「ダミー」されている。モダリティはとみなされます。モダリティは、で $I$ ですることできます。
  - 4つのはすべて0.1レベルでにである。
  - ロジスティックは、の1のにするのオッズのをえる。
  - オッズのあたりののをするには、パラメータをします。
  - パラメータのCIをするには、`confint`し`confint`。
- のには、ヌルとのをむと、モデルのにできるAICがあります。
  - じデータにでフィッティングされたモデルをすると、AICがさければさいほどがなくなります。
  - モデルの1つのは、モデルの。このテストでは、をむモデルが、のみをつモデルつまり、ヌルモデルよりもはるかにれているかどうかをします。

オッズの

```

exp(coef(titanic.train)[3])

pclass3rd
0.08797765

```

このモデルでは、1クラスとして、3クラスのはの10の1をとっています。

パラメータの

```
confint(titanic.train)

Waiting for profiling to be done...
 2.5 % 97.5 %
(Intercept) 2.89486872 4.23734280
pclass2nd -1.58986065 -0.75987230
pclass3rd -2.81987935 -2.05419500
sexmale -2.77180962 -2.16528316
age -0.05695894 -0.02786211
```

モデルのをする

テストは、カレント・モデルとヌル・モデルののすなわち、モデルののにしいでカイします。

```
with(titanic.train, pchisq(null.deviance - deviance, df.null - df.residual
, lower.tail = FALSE))
[1] 1.892539e-111
```

pは0にく、になモデルをしています。

オンラインでモデルをむ <https://riptutorial.com/ja/r/topic/2892/モデル>

## 81:

リモートマシンでのでは、マシンにライブラリをダウンロードするがあります。

`package::function()` がきです。いくつかのパッケージには、`caret`、`pls`、`plyr`などのがネイティブにみまれてい`caret`。

**Microsoft R Open** Revolution Rは、くのをにするマルチスレッドBLAS / LAPACKライブラリもします。

## Examples

### foreachパッケージによる

`foreach`パッケージはをRにもたらしませんが、マルチコアCPUをするにマルチコアクラスタをりてるがあります。`doSNOW`パッケージは1つのがあります。

`foreach`ループのなはいは、と1から100000までのすべてののをすることです。

```
library(foreach)
library(doSNOW)

cl <- makeCluster(5, type = "SOCK")
registerDoSNOW(cl)

f <- foreach(i = 1:100000, .combine = c, .inorder = F) %dopar% {
 k <- i ** 2 + sqrt(i)
 k
}
```

`foreach`ののは、`.combine`によってされます。デフォルトのはリストです。のコードでは、わりに`c`をしてベクトルをします。`"+"`などのまたはをしてをし、さらにされたオブジェクトをすこともできます。

それぞれの`foreach-loop`ののがのびしであることにすることはです。したがって、このでは、`k`がにされます。

| パラメータ                 |                                                                                                                                  |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------|
| <code>.combine</code> | をする。ループのをどのようにみわせるかをします。なは <code>c</code> 、 <code>cbind</code> 、 <code>rbind</code> 、 <code>"+"</code> 、 <code>"*"</code> ...です。 |
| 。に                    | <code>TRUE</code> の、はなここでは <code>i</code> につてべられます。 <code>FALSE</code> 、はけされません。これは、にをえるがあります。                                   |
| パッケー                  | <code>base</code> のパッケージ、えは <code>mass</code> 、 <code>randomForest</code> などでされるの、これらのパッケー                                       |

## パラメータ

シには `c("mass", "randomForest")` をするがあります

パッケージによる

パッケージ `parallel` は、フォーク、ソケット、およびによるをにします。

ローカルホストにするコアのをします。

```
parallel::detectCores(all.tests = FALSE, logical = TRUE)
```

`localhost` にコアのクラスタをします。

```
parallelCluster <- parallel::makeCluster(parallel::detectCores())
```

まず、にしたをするがあります。 `mtcars` データセットをえてみましょう。 `cyl` レベルについてのモデルをすることで、 `mpg` のをできます。

```
data <- mtcars
yfactor <- 'cyl'
zlevels <- sort(unique(data[[yfactor]]))
datay <- data[,1]
dataz <- data[,2]
datax <- data[,3:11]

fitmodel <- function(zlevel, datax, datay, dataz) {
 glm.fit(x = datax[dataz == zlevel,], y = datay[dataz == zlevel])
}
```

`zlevels` すべてののをループすることができるをします。これはまだシリアルでわれていますが、されるなプロセスをするなステップです。

```
fitmodel <- function(zlevel, datax, datay, dataz) {
 glm.fit(x = datax[dataz == zlevel,], y = datay[dataz == zlevel])
}

for (zlevel in zlevels) {
 print("*****")
 print(zlevel)
 print(fitmodel(zlevel, datax, datay, dataz))
}
```

カレーこの

```
worker <- function(zlevel) {
 fitmodel(zlevel, datax, datay, dataz)
}
```

parallelをparallelコンピューティングは、にはアクセスできません。なことに、はローカルのparallelアクセスをします。ラッパーをすると、がになります。されるものにすることがあります。

```
wrapper <- function(datax, datay, dataz) {
 # force evaluation of all paramters not supplied by parallelization apply
 force(datax)
 force(datay)
 force(dataz)
 # these variables are now in an enviroment accessible by parallel function

 # function to be applied also in the environment
 fitmodel <- function(zlevel, datax, datay, dataz) {
 glm.fit(x = datax[dataz == zlevel,], y = datay[dataz == zlevel])
 }

 # calling in this environment iterating over single parameter zlevel
 worker <- function(zlevel) {
 fitmodel(zlevel,datax, datay, dataz)
 }
 return(worker)
}
```

すぐクラスターをし、ラッパーをします。

```
parallelcluster <- parallel::makeCluster(parallel::detectCores())
models <- parallel::parLapply(parallelcluster,zlevels,
 wrapper(datax, datay, dataz))
```

したら、ずクラスタをしてください。

```
parallel::stopCluster(parallelcluster)
```

parallelパッケージはがま apply() がいい、 par。

のきなは、RNGをシードとしてすることです。によるは、セッションのまたはのset.seed() いくれかからののによってされます。はじからするため、のシードをすることができ、じがじるがあります。コールは、さまざまなコアでシリアルにされ、をしません。

のシードをしてプロセスにするがあります。これはいくつかのパッケージ parallel、 snowなどでにわれますが、のパッケージではにするがあります。

```
s <- seed
for (i in 1:numofcores) {
 s <- nextRNGStream(s)
 # send s to worker i as .Random.seed
}
```

をのためにすることもできます。

```
clusterSetRNGStream(cl = parallelcluster, iseed)
```



## mcparallelDo

mcparallelDoパッケージをすると、UNIXLinuxやMacOSXなどのオペレーティングシステムでRコードをにすることができます。パッケージのにあるえは、コーディングではなく、データのニーズにわせています。をコーディングするには、 [future](#) パッケージをしてください。

### データをする

```
data(ToothGrowth)
```

### フォークでをするためにmcparallelDoをトリガする

```
mcparallelDo({glm(len ~ supp * dose, data=ToothGrowth)},"interactionPredictorModel")
```

### のこと、えは

```
binaryPredictorModel <- glm(len ~ supp, data=ToothGrowth)
gaussianPredictorModel <- glm(len ~ dose, data=ToothGrowth)
```

mcparallelDoのはtargetEnvironmentなどでされます.GlobalEnv。メッセージがするとデフォルトでは

```
summary(interactionPredictorModel)
```

## そのの

```
Example of not returning a value until we return to the top level
for (i in 1:10) {
 if (i == 1) {
 mcparallelDo({2+2}, targetValue = "output")
 }
 if (exists("output")) print(i)
}

Example of getting a value without returning to the top level
for (i in 1:10) {
 if (i == 1) {
 mcparallelDo({2+2}, targetValue = "output")
 }
 mcparallelDoCheck()
 if (exists("output")) print(i)
}
```

オンラインでをむ <https://riptutorial.com/ja/r/topic/1677/>

## 82:

### Examples

ランダムな

5つのランダムをするには

```
sample(5)
[1] 4 5 3 1 2
```

のベクトルのランダムをするには

```
sample(10:15)
[1] 11 15 12 10 14 13
```

また、パッケージの`pracma`することもできます

```
randperm(a, k)
Generates one random permutation of k of the elements a, if a is a vector,
or of 1:a if a is a single integer.
a: integer or numeric vector of some length n.
k: integer, smaller as a or length(a).

Examples
library(pracma)
randperm(1:10, 3)
[1] 3 7 9

randperm(10, 10)
[1] 4 5 10 8 2 7 6 9 3 1

randperm(seq(2, 10, by=2))
[1] 6 4 10 2 8
```

の

かがランダムなをつRコードをすることをする、`set.seed()`はにになります。たとえば、これら2つのはになるをしますこれはジェネレータのです。

```
> sample(1:10,5)
[1] 6 9 2 7 10
> sample(1:10,5)
[1] 7 6 1 2 10
```

これらの2つはなるもします。

```
> rnorm(5)
[1] 0.4874291 0.7383247 0.5757814 -0.3053884 1.5117812
```

```
> rnorm(5)
[1] 0.38984324 -0.62124058 -2.21469989 1.12493092 -0.04493361
```

しかし、シードをのケースでじものにとするとシンプルさのためにほとんどのが1をします、2つのサンプルがられます。

```
> set.seed(1)
> sample(letters, 2)
[1] "g" "j"
> set.seed(1)
> sample(letters, 2)
[1] "g" "j"
```

`rexp()` はののようにします。

```
> set.seed(1)
> rexp(5)
[1] 0.7551818 1.1816428 0.1457067 0.1397953 0.4360686
> set.seed(1)
> rexp(5)
[1] 0.7551818 1.1816428 0.1457067 0.1397953 0.4360686
```

さまざまなをとてをする

に、さまざまなをとて5つのをするをします。

## 0と10の

```
runif(5, min=0, max=10)
[1] 2.1724399 8.9209930 6.1969249 9.3303321 2.4054102
```

## 0とか1の

```
rnorm(5, mean=0, sd=1)
[1] -0.97414402 -0.85722281 -0.08555494 -0.37444299 1.20032409
```

## 10のと0.5の

```
rbinom(5, size=10, prob=0.5)
[1] 4 3 5 2 3
```

## 0.2の

```
rgeom(5, prob=0.2)
[1] 14 8 11 1 3
```

3つのいボール、10のいボール、5つのドローをむ

```
rhyper(5, m=3, n=10, k=5)
[1] 2 0 1 1 1
```

10のと0.8のの

```
rnbinom(5, size=10, prob=0.8)
[1] 3 1 3 4 2
```

とラムダが2のポアソン

```
rpois(5, lambda=2)
[1] 2 1 2 3 4
```

1.5の

```
rexp(5, rate=1.5)
[1] 1.8993303 0.4799358 0.5578280 1.5630711 0.6228000
```

0のと1ののロジスティック

```
rlogis(5, location=0, scale=1)
[1] 0.9498992 -1.0287433 -0.4192311 0.7028510 -1.2095458
```

15のカイ2

```
rchisq(5, df=15)
[1] 14.89209 19.36947 10.27745 19.48376 23.32898
```

パラメータ  $a = 1$  および  $b = 0.5$  のベータ

```
rbeta(5, shape1=1, shape2=0.5)
[1] 0.1670306 0.5321586 0.9869520 0.9548993 0.9999737
```

パラメータ3およびスケール = 0.5 のガンマ

```
rgamma(5, shape=3, scale=0.5)
[1] 2.2445984 0.7934152 3.2366673 2.2897537 0.8573059
```

## 0のと1のスケールでのコーシー

```
rcauchy(5, location=0, scale=1)
[1] -0.01285116 -0.38918446 8.71016696 10.60293284 -0.68017185
```

## 0および1りの

```
rlnorm(5, meanlog=0, sdlog=1)
[1] 0.8725009 2.9433779 0.3329107 2.5976206 2.8171894
```

## パラメータ0.5およびスケール1のワイブル

```
rweibull(5, shape=0.5, scale=1)
[1] 0.337599112 1.307774557 7.233985075 5.840429942 0.005751181
```

## のでは10、では20のWilcoxonをした。

```
rwilcox(5, 10, 20)
[1] 111 88 93 100 124
```

## されたをして5つのオブジェクトと3つのボックスをつ

```
rmultinom(5, size=5, prob=c(0.1,0.1,0.8))
 [,1] [,2] [,3] [,4] [,5]
[1,] 0 0 1 1 0
[2,] 2 0 1 1 0
[3,] 3 5 3 3 5
```

オンラインでをむ <https://riptutorial.com/ja/r/topic/1578/>

## 83: でする

き

`sub`と`gsub`はパターンをしてをするためにされます。 [パターンの](#)については、[との](#)については[パターンマッチング](#)とをしてください。

### Examples

グループをしてをべる

のをするは、`pattern`のカッコをしてのをグループできます。これらのグループは、`replacement`でしたをしてアドレスできます。

のは、`"`、`"`のののベクトルを`"`の`"`というのベクトルにべるをしてしています。

```
library(randomNames)
set.seed(1)

strings <- randomNames(5)
strings
[1] "Sigg, Zachary" "Holt, Jake" "Ortega, Sandra" "De La Torre,
[5] "Perkins, Donovan"

sub("^(.+),\\s(.+)$", "\\2 \\1", strings)
[1] "Zachary Sigg" "Jake Holt" "Sandra Ortega" "Nichole De La Torre"
[5] "Donovon Perkins"
```

だけがなは、ののペアをすることができます。

```
sub("^(.+),\\s(.+)", "\\1", strings)
[1] "Sigg" "Holt" "Ortega" "De La Torre" "Perkins"
```

したしたをする

からしたサブシーケンスをしたいとしましょうのがあります。えは

```
2,14,14,14,19
```

それをのようにします

```
2,14,19
```

`gsub`をして、これをできます

```
gsub("(\\d+)(,\\1)+", "\\1", "2,14,14,14,19")
[1] "2,14,19"
```

のなるりしにもできます。たとえば、のようになります。

```
> gsub("(\\d+)(,\\1)+", "\\1", "2,14,14,14,19,19,20,21")
[1] "2,14,19,20,21"
```

についてしましょう

1. `(\\d+)` `(\\d+)` られ、のなくとも1つをするグループ1。の、バックスラッシュはリテラルリ  
\ "または\ のなエスケープをします。 `\\d` は `[0-9]` とであるため、 `\\` をするがあります。
2. , , スペースやのりをむことができます
3. `\\1` グループ1とじ、つまりりし。それがこらない、パターンはしません。

のをしてみましようしたりしをする

```
one,two,two,three,four,four,five,six
```

その、にきえる `\\d` で `\\w` ところ、 `\\w` の、またはアンダースコアをむのにします。これは `[a-zA-Z0-9_]`

```
> gsub("(\\w+)(,\\1)+", "\\1", "one,two,two,three,four,four,five,six")
[1] "one,two,three,four,five,six"
>
```

に、のパターンは、のとしてしたのをむ。

オンラインででをするをむ <https://riptutorial.com/ja/r/topic/9219/> でをする

---

## 84: によるRのメモ

き

このトピックは、テキストなしのRにするであり、わかりやすいです。

それぞれのはできるだけとしてすることをしてしています。

### Examples

データ

---

## ベクトル

```
a <- c(1, 2, 3)
b <- c(4, 5, 6)
mean_ab <- (a + b) / 2

d <- c(1, 0, 1)
only_1_3 <- a[d == 1]
```

---

```
mat <- matrix(c(1,2,3,4), nrow = 2, ncol = 2)
dimnames(mat) <- list(c(), c("a", "b", "c"))
mat[,] == mat
```

---

## データフレーム

```
df <- data.frame(qualifiers = c("Buy", "Sell", "Sell"),
 symbols = c("AAPL", "MSFT", "GOOGL"),
 values = c(326.0, 598.3, 201.5))
df$symbols == df[[2]]
df$symbols == df[["symbols"]]
df[[2, 1]] == "AAPL"
```

---

## リスト

```
l <- list(a = 500, "aaa", 98.2)
length(l) == 3
class(l[1]) == "list"
class(l[[1]]) == "numeric"
class(l$a) == "numeric"
```



```

env <- new.env()
env[["foo"]] = "bar"
env2 <- env
env2[["foo"]] = "BAR"

env[["foo"]] == "BAR"
get("foo", envir = env) == "BAR"
rm("foo", envir = env)
env[["foo"]] == NULL

```

## プロットプロットを

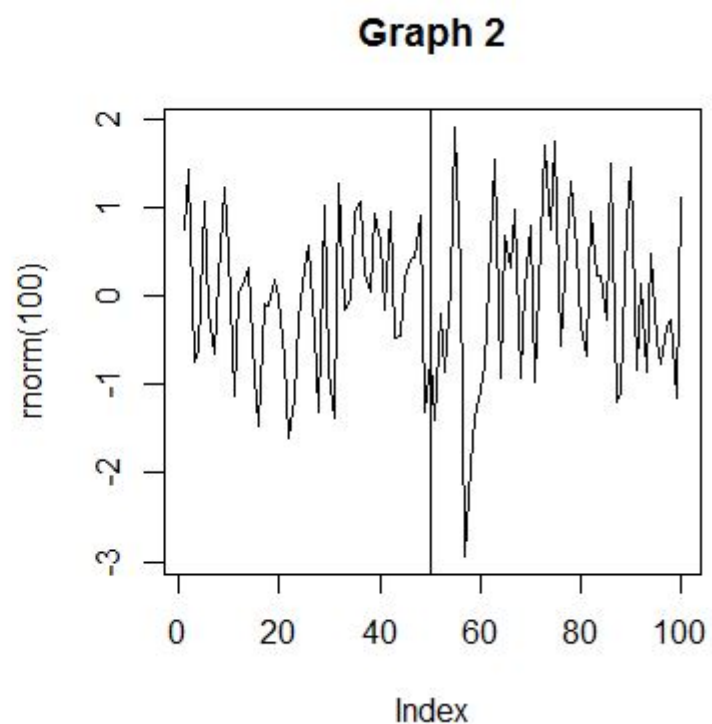
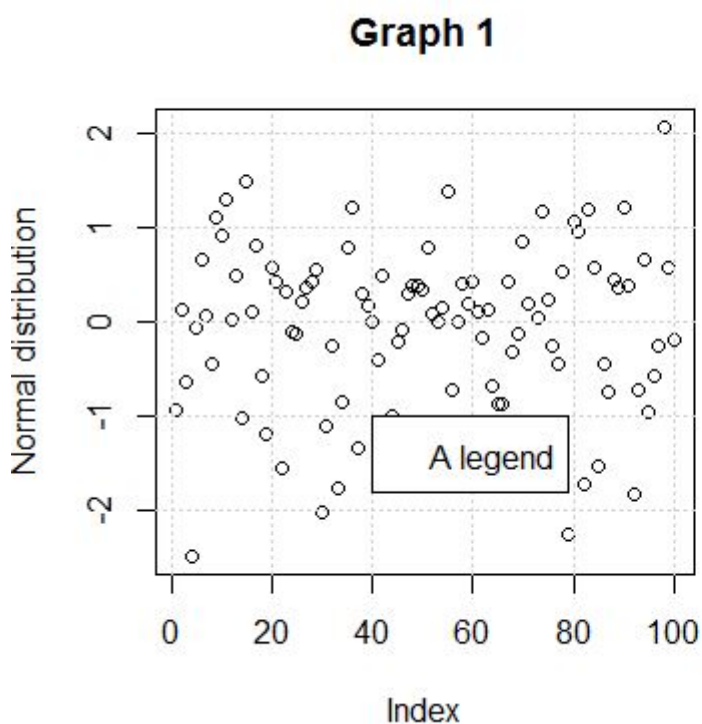
```

Creates a 1 row - 2 columns format
par(mfrow=c(1,2))

plot(rnorm(100), main = "Graph 1", ylab = "Normal distribution")
grid()
legend(x = 40, y = -1, legend = "A legend")

plot(rnorm(100), main = "Graph 2", type = "l")
abline(v = 50)

```



## よくわれる

```

Create 100 standard normals in a vector
x <- rnorm(100, mean = 0, sd = 1)

Find the length of a vector
length(x)

Compute the mean

```

```
mean(x)

Compute the standard deviation
sd(x)

Compute the median value
median(x)

Compute the range (min, max)
range(x)

Sum an iterable
sum(x)

Cumulative sum (x[1], x[1]+x[2], ...)
cumsum(x)

Display the first 3 elements
head(3, x)

Display min, 1st quartile, median, mean, 3rd quartile, max
summary(x)

Compute successive difference between elements
diff(x)

Create a range from 1 to 10 step 1
1:10

Create a range from 1 to 10 step 0.1
seq(1, 10, 0.1)

Print a string
print("hello world")
```

オンラインでによるRのメモをむ <https://riptutorial.com/ja/r/topic/10827/>によるrのメモ

## 85: ありません

き

がるがわからないときは、そのが`NA`でされていないといいます。

は`NA not available`でされます。なえば、`sqrt(-1)`は、`NaN`ではないでされます。

## Examples

しているデータをべる

`anyNA`は、がするかどうかをします。 `is.na`は`is.na`をします。

```
vec <- c(1, 2, 3, NA, 5)

anyNA(vec)
[1] TRUE
is.na(vec)
[1] FALSE FALSE FALSE TRUE FALSE
```

`is.na`は、`FALSE = 0`、`TRUE = 1`でにされるベクトルをします。これをしてのをべることができます。

```
sum(is.na(vec))
[1] 1
```

このアプローチをすると、[データフレーム](#)で`colSums`と`is.na`をして、カラムごとに`NA`sをカウントできます。

```
colSums(is.na(airquality))
Ozone Solar.R Wind Temp Month Day
37 7 0 0 0 0
```

[naniar](#)パッケージ [github](#)にはしますがCRANにはありませんは、しているをべるためのツールをします。

**NA**によるデータのみき

`read.*`でのデータセットをむとき、Rはに`"NA"`ようなをします。ただし、はに`NA`されるわけではありません。ドット`.`、ハイフン`-`またはたとえば`empty`は、が`NA`であることをします。 `read.*`の`na.strings`パラメータは、どのシンボル/キャラクタを`NA`としてうがあるかをRにえるためにできます。

```
read.csv("name_of_csv_file.csv", na.strings = "-")
```

のシンボルを`NA`としてみるがあることをすることもできます。

```
read.csv('missing.csv', na.strings = c('.', '-'))
```

に、`NA`sはカスタマイズされた`write.csv` `na`をしてすることができます。 [テーブルをみきするのためのツール](#)にもオプションがあります。

## なるクラスの`NA`sの

`NA`は、`logical`をします。

```
class(NA)
[1] "logical"
```

これは、のアトミックベクトルににすることができ、はなの`NA`ため、です。

```
x <- c(1, NA, 1)
class(x[2])
[1] "numeric"
```

のタイプのの`NA`がなは、`NA_character_`、`NA_integer_`、`NA_real_`または`NA_complex_`ます。ファンシークラスのについては、`NA_integer_`でサブセットします。たとえば、をするDate

```
class(Sys.Date() [NA_integer_])
[1] "Date"
```

## `TRUE` / `FALSE`および/または`NA`

`NA`とのである`NA`ります `NA`があいまいである。は、`NA OR TRUE`とされ`TRUE`なくともがあるとするので`TRUE`、しかし`NA OR FALSE`をす`NA`々がいるかどうかわからないので`NA`あったであろう`TRUE`または`FALSE`

```
NA | TRUE
[1] TRUE
TRUE | TRUE is TRUE and FALSE | TRUE is also TRUE.

NA | FALSE
[1] NA
TRUE | FALSE is TRUE but FALSE | FALSE is FALSE.

NA & TRUE
[1] NA
TRUE & TRUE is TRUE but FALSE & TRUE is FALSE.

NA & FALSE
[1] FALSE
TRUE & FALSE is FALSE and FALSE & FALSE is also FALSE.
```

これらのプロパティは、`NA`をむいくつかのについてデータ・セットをサブセットするにちます。

```
df <- data.frame(v1=0:9,
 v2=c(rep(1:2, each=4), NA, NA),
 v3=c(NA, letters[2:10]))

df[df$v2 == 1 & !is.na(df$v2),]
v1 v2 v3
#1 0 1 <NA>
#2 1 1 b
#3 2 1 c
#4 3 1 d

df[df$v2 == 1,]
 v1 v2 v3
#1 0 1 <NA>
#2 1 1 b
#3 2 1 c
#4 3 1 d
#NA NA NA <NA>
#NA.1 NA NA <NA>
```

のまたは

の

、しているデータはデータセットで`NA`としてコードされていません。えは、SPSSでは、は`99`されることがい。

```
num.vec <- c(1, 2, 3, 99, 5)
num.vec
[1] 1 2 3 99 5
```

サブセッティングをして`NA`をりてることがです

```
num.vec[num.vec == 99] <- NA
```

しかし、のように`is.na<-`をするのがましいです。ヘルプファイル `?is.na` の

`is.na<-`は、けをするなをするかもしれません。それは、えは、にしてなるをす。

```
is.na(num.vec) <- num.vec == 99
```

のメソッドがります

```
num.vec
[1] 1 2 3 NA 5
```

の

は、ベクトルからいくつかのことができます。

```
num.vec[!is.na(num.vec)]
num.vec[complete.cases(num.vec)]
na.omit(num.vec)
[1] 1 2 3 5
```

## からをする

をつベクトルにしてをすると、がされます。

```
mean(num.vec) # returns: [1] NA
```

`na.rm`パラメータは、に`NA`をからするようにします。

```
mean(num.vec, na.rm = TRUE) # returns: [1] 2.75

an alternative to using 'na.rm = TRUE':
mean(num.vec[!is.na(num.vec)]) # returns: [1] 2.75
```

`lm`のようないくつかのRには、`na.action`パラメータがあります。このデフォルトは`na.omit`ですが、`options(na.action = 'na.exclude')`ではRのデフォルトのことができます。

デフォルトをするはないが、のではの`na.action`がなは、`na.action`パラメータをびしにめるがあります。

```
lm(y2 ~ y1, data = anscombe, na.action = 'na.exclude')
```

オンラインでありませんをむ <https://riptutorial.com/ja/r/topic/3388>/ありません

---

## 86:

ファイルパスをするには、みきのために `file.path` し `file.path`。

`dir` をして、ディレクトリのファイルをしします。

## Examples

データフレームのみき

データフレームは、Rののデータです。それらは、さまざまなできむか、みむことができます。

このは、なをしています。のリソースについては、のリンクをしてください。

---

## きみ

のサンプルデータをするに、きみのフォルダにすることをしてください。 `getwd()` をしてフォルダをし、フォルダをするがあるは `?setwd` をんでください。

```
set.seed(1)
for (i in 1:3)
 write.table(
 data.frame(id = 1:2, v = sample(letters, 2)),
 file = sprintf("file201%s.csv", i)
)
```

は、にフォーマットされた3つのCSVファイルがディスクにあります。

---

これらのファイルはしているので、みんなにまとめてするがあり `list`。

```
file_names = c("file2011.csv", "file2012.csv", "file2013.csv")
file_contents = lapply(setNames(file_names, file_names), read.table)

$file2011.csv
id v
1 1 g
2 2 j
#
$file2012.csv
id v
1 1 o
2 2 w
#
$file2013.csv
id v
1 1 f
2 2 w
```

このファイルのリストをするには、に`str(file_contents)`でをべてから、`?rbind`リストをみねる  
`?rbind`、`?rbind`リストをすることについてんで`?lapply`。

---

## さらなるリソース

このをするには`?read.table`と`?write.table`を`?write.table`してください。また、

- [Rバイナリおよびそのオブジェクト](#)
- [プレーンテキストの](#)
  - コンマりのCSV
  - タブりのTSV
  - フォーマット
- [にしないバイナリテーブル](#)
  - フェザー
- [とスプレッドシート](#)
  - SAS
  - SPSS
  - スタータ
  - Excel
- [リレーショナルデータベースのテーブル](#)
  - MySQL
  - SQLite
  - PostgreSQL

オンラインでをむ <https://riptutorial.com/ja/r/topic/5543/>



## 87: なR

き

「」とは、のかおそらくあなたのが、したステップをりしてじをることをします。 [なりサーチタスクビュー](#)をしてください。

のあるをするには、すべてのをするがあります。たとえば、ジェネレータをする、じをするは、シードをするがあります。バリエーションをらすもう1つのは、テキストとをじですることです。

- Peng、RD2011。におけるな。 Science、3346060、1226-1227。  
<http://doi.org/10.1126/science.1213847>
- Peng、Roger D. R. Leanpub、2015のデータのためのレポート。 <https://leanpub.com/reportwriting>。

## Examples

データ

dput () と dget ()

ましいさなデータフレームをするもなは、dput () をうことです。 Rオブジェクトをプレーンテキストでエクスポートします。

ののデータをするに、きみなのフォルダにいることをしてください。 getwd () をし、フォルダをするがあるは ?setwd をみみます。

```
dput(mtcars, file = 'df.txt')
```

そしてでもdget () をってなRオブジェクトをGlobalEnvironmentにロードできます。

```
df <- dget('df.txt')
```

きなRオブジェクトの、それらをにするにはいくつかのがあります。 [と](#)をしてください。

パッケージの

パッケージのは、いくつかのRコードをするのになです。さまざまなパッケージがされると、それらののいくつかのがれることがあります。このにするなは、コードがかれたにRコードライターのマシンのイメージをコンピューターにすることです。ここにcheckpointパッケージがあります

。

パッケージのは、2014-09-17、CRANパッケージリポジトリをのミラーリポジトリである Microsoft R Archived Networkにコピーします。したがって、なRプロジェクトをするときに、パッケージののをするためには、がです。

1. すべてのパッケージおよびRバージョンがのものであることをしてください。
2. コードに`checkpoint::checkpoint('YYYY-MM-DD')`をめます。

`checkpoint`はあなたのR\_homeディレクトリ `~/` にディレクトリ `.checkpoint` をします。このディレクトリには、プロジェクトでされているすべてのパッケージがインストールされます。つまり、`checkpoint`は、プロジェクトディレクトリのすべての`.R`ファイルをべて、すべての`library()`または`require()`びしをし、されたにCRANにしていたでなパッケージをすべてインストールします

。

**PRO**パッケージののからされました。

**CONTRA**のごとに、のプロジェクトでするすべてのパッケージをダウンロードしてインストールするがあります。それはかなりがかかるかもしれません。

オンラインでなRをむ <https://riptutorial.com/ja/r/topic/4087/なr>

---

## 88:

き

のためにRコード、テーブル、グラフをフォーマットするには、く のがあります。

Rユーザーは、やをのがあるでしたいことがよくあります。については、「[なR](#)」をしてください。

## Examples

の

ここで、「テーブル」とはに `data.frame`、`data.frame`、`table`、

---

## プレーンテキストへの

プレーンテキストをモノスペースフォントにするには、コンソールにされているようにすればです。

ののデータをするに、きみなのフォルダにすることをしてください。 `getwd()` をし、フォルダをするがあるは `?setwd` をみみます。

```
..w = options()$width
options(width = 500) # reduce text wrapping
sink(file = "mytab.txt")
 summary(mtcars)
sink()
options(width = ..w)
rm(..w)
```

---

## りきの

CSVまたはのなにきんだ、スプレッドシートエディタでき、げのタッチをすることもできます。

ののデータをするに、きみなのフォルダにすることをしてください。 `getwd()` をし、フォルダをするがあるは `?setwd` をみみます。

```
write.csv(mtcars, file="mytab.csv")
```

---

## さらなるリソース

- `knitr::kable`
- 
- `tables::tabular`
- テックスレグ
- `xtable`

ドキュメントの

`utils` パッケージから `Sweave` すると、コード、、グラフ、およびテーブルをLaTeXドキュメントでフォーマットすることができます。

---

## そのリソース

- KnitrとRMarkdown

オンラインでをむ <https://riptutorial.com/ja/r/topic/9039/>

## Examples

スプリットのしない

`split`は、ファクタ/グループにして、ベクトルまたは`data.frame`をバケットに`split`することをにします。バケットにこのは、いで、グループごとのをするためにすることができるリストの、とるforループまたは`lapply / sapply`。

のは、ベクトルの`split`のいをしています。

のベクトルをのようになてください。

```
testdata <- c("e", "o", "r", "g", "a", "y", "w", "q", "i", "s", "b", "v", "x", "h", "u")
```

は、これらのを`vowels`と`consonants`にすることです。つまり、それをタイプにします。

にグループベクトルをしましょう

```
vowels <- c('a','e','i','o','u','y')
letter_type <- ifelse(testdata %in% vowels, "vowels", "consonants")
```

`letter_type`はvector `testdata`とじさであることにしてください。は、このテストデータを`vowels`と`consonants` 2つのグループに`split`ことができ`consonants`。

```
split(testdata, letter_type)
#$consonants
#[1] "r" "g" "w" "q" "s" "b" "v" "x" "h"

#$vowels
#[1] "e" "o" "a" "y" "i" "u"
```

したがって、は、がグループベクトル/`letter_type`からるリスト`letter_type`。

`split`は`data.frames`をするメソッドもあります。

たとえば、`iris`データをえてみましょう。

```
data(iris)
```

`split`をすることで、1つの`data.frame``iris` `specievariableSpecies`をむリストをできます。

```
> liris <- split(iris, iris$Species)
> names(liris)
[1] "setosa" "versicolor" "virginica"
> head(liris$setosa)
```

|   | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|---|--------------|-------------|--------------|-------------|---------|
| 1 | 5.1          | 3.5         | 1.4          | 0.2         | setosa  |
| 2 | 4.9          | 3.0         | 1.4          | 0.2         | setosa  |
| 3 | 4.7          | 3.2         | 1.3          | 0.2         | setosa  |
| 4 | 4.6          | 3.1         | 1.5          | 0.2         | setosa  |
| 5 | 5.0          | 3.6         | 1.4          | 0.2         | setosa  |
| 6 | 5.4          | 3.9         | 1.7          | 0.4         | setosa  |

セットアサグループのデータのみをむ。

1つのなは、のごとのをすることである。 `lapply`をうでしよう

```
> (lcor <- lapply(liris, FUN=function(df) cor(df[,1:4])))
```

| \$setosa     |              |             |              |             |  |
|--------------|--------------|-------------|--------------|-------------|--|
|              | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width |  |
| Sepal.Length | 1.0000000    | 0.7425467   | 0.2671758    | 0.2780984   |  |
| Sepal.Width  | 0.7425467    | 1.0000000   | 0.1777000    | 0.2327520   |  |
| Petal.Length | 0.2671758    | 0.1777000   | 1.0000000    | 0.3316300   |  |
| Petal.Width  | 0.2780984    | 0.2327520   | 0.3316300    | 1.0000000   |  |

| \$versicolor |              |             |              |             |  |
|--------------|--------------|-------------|--------------|-------------|--|
|              | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width |  |
| Sepal.Length | 1.0000000    | 0.5259107   | 0.7540490    | 0.5464611   |  |
| Sepal.Width  | 0.5259107    | 1.0000000   | 0.5605221    | 0.6639987   |  |
| Petal.Length | 0.7540490    | 0.5605221   | 1.0000000    | 0.7866681   |  |
| Petal.Width  | 0.5464611    | 0.6639987   | 0.7866681    | 1.0000000   |  |

| \$virginica  |              |             |              |             |  |
|--------------|--------------|-------------|--------------|-------------|--|
|              | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width |  |
| Sepal.Length | 1.0000000    | 0.4572278   | 0.8642247    | 0.2811077   |  |
| Sepal.Width  | 0.4572278    | 1.0000000   | 0.4010446    | 0.5377280   |  |
| Petal.Length | 0.8642247    | 0.4010446   | 1.0000000    | 0.3221082   |  |
| Petal.Width  | 0.2811077    | 0.5377280   | 0.3221082    | 1.0000000   |  |

に、グループごとにののをりすことができますは/け、はフィルタリングされ、のレコードのがされます

```
> library(reshape)
> (topcor <- lapply(lcor, FUN=function(cormat){
 correlations <- melt(cormat,variable_name="correlation");
 filtered <- correlations[correlations$X1 != correlations$X2,];
 filtered[which.max(filtered$correlation),]
}))
```

| \$setosa |             |              |             |
|----------|-------------|--------------|-------------|
|          | X1          | X2           | correlation |
| 2        | Sepal.Width | Sepal.Length | 0.7425467   |

| \$versicolor |             |              |             |
|--------------|-------------|--------------|-------------|
|              | X1          | X2           | correlation |
| 12           | Petal.Width | Petal.Length | 0.7866681   |

| \$virginica |              |              |             |
|-------------|--------------|--------------|-------------|
|             | X1           | X2           | correlation |
| 3           | Petal.Length | Sepal.Length | 0.8642247   |

そのようなグループのレベルで1つのがされることにしてください。をみねることにがあります。

```
> (result <- do.call("rbind", topcor))

 X1 X2 correlation
setosa Sepal.Width Sepal.Length 0.7425467
versicolor Petal.Width Petal.Length 0.7866681
virginica Petal.Length Sepal.Length 0.8642247
```

## split-apply-combineパラダイムでのsplitの

データののは、[コンバインスプリット](#)、あなたがグループにデータをした、グループののいくつかのべえをし、をします。

みみのmtcarsデータセットのシリンダcylにしてガロンたりのなマイルmpgで2のをしたいのデータをえてみましょう。まず、mtcarsデータフレームをシリンダでします。

```
(spl <- split(mtcars, mtcars$cyl))
$`4`
mpg cyl disp hp drat wt qsec vs am gear carb
Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1
Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2
Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2
Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1
...
#
$`6`
mpg cyl disp hp drat wt qsec vs am gear carb
Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4
Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4
Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1
Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1
...
#
$`8`
mpg cyl disp hp drat wt qsec vs am gear carb
Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2
Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4
Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3
Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3
...
```

これにより、シリンダに1つのデータフレームのリストがされました。によってされるように、私たちはとするデータフレームることができたspl\$`4`、 spl\$`6`、 およびspl\$`8` はそれをよりにするためにつけるかもしれないspl\$"4"またはspl[["4"]]。

は、lapplyをとって、リストのから2 mpgのをつをするをして、このリストをループすることができす

```
(best2 <- lapply(spl, function(x) tail(x[order(x$mpg),], 2)))
$`4`
mpg cyl disp hp drat wt qsec vs am gear carb
Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1
Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1
```

```
#
$`6`
mpg cyl disp hp drat wt qsec vs am gear carb
Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4
Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1
#
$`8`
mpg cyl disp hp drat wt qsec vs am gear carb
Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2
Pontiac Firebird 19.2 8 400 175 3.08 3.845 17.05 0 0 3 2
```

に、`rbind`をしてすべてをみわせることができます。 `rbind(best2[["4"]], best2[["6"]], best2[["8"]])`をびしたいが、なリストをとってれば、これはだろう。その、をします。

```
do.call(rbind, best2)
mpg cyl disp hp drat wt qsec vs am gear carb
4.Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1
4.Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1
6.Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4
6.Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1
8.Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2
8.Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2
```

これはとしてされる`best2` 2、リストのすべてのをつ`rbind` 1、のをします。

このようななでは、`split-apply-combine`を1のコードでするために、よりコンパクトにすることができます。

```
do.call(rbind, lapply(split(mtcars, mtcars$cyl), function(x) tail(x[order(x$mpg),], 2)))
```

`lapply(split(x,f), FUN)` みわせは`?by?by`をとってフレームすることもできます

```
by(mtcars, mtcars$cyl, function(x) tail(x[order(x$mpg),], 2))
do.call(rbind, by(mtcars, mtcars$cyl, function(x) tail(x[order(x$mpg),], 2)))
```

オンラインでをむ <https://riptutorial.com/ja/r/topic/1073/>



## き

Rにはをうためのみみがあり、ドキュメントは?Distributionsからまってい?Distributions。

に4つのプレフィックスがあります。

- **d** - えられたにする
- **p** -
- **q** - えられたにするをる
- **r** - ランダムサンプルをる

Rのインストールにみまれているディストリビューションについては、「?Distributions」をしてください。

## Examples

として\*normを試みましょう。ドキュメントから

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

ですから、のを0でりたければ、

```
dnorm(0)
```

たちに0.3989423のなえがえられます。

に、 pnorm(0) は .5 pnorm(0) ます。ここでも、のは0にあるので、これはをなさない。

qnormはにpnormのをしpnorm。 qnorm(.5) は0 qnorm(.5) 0。

に、 rnormがあります

```
rnorm(10)
```

から10サンプルをします。

あなたがえられたのパラメータをしたいのであれば、そのようにするだけです

```
rnorm(10, mean=4, sd= 3)
```

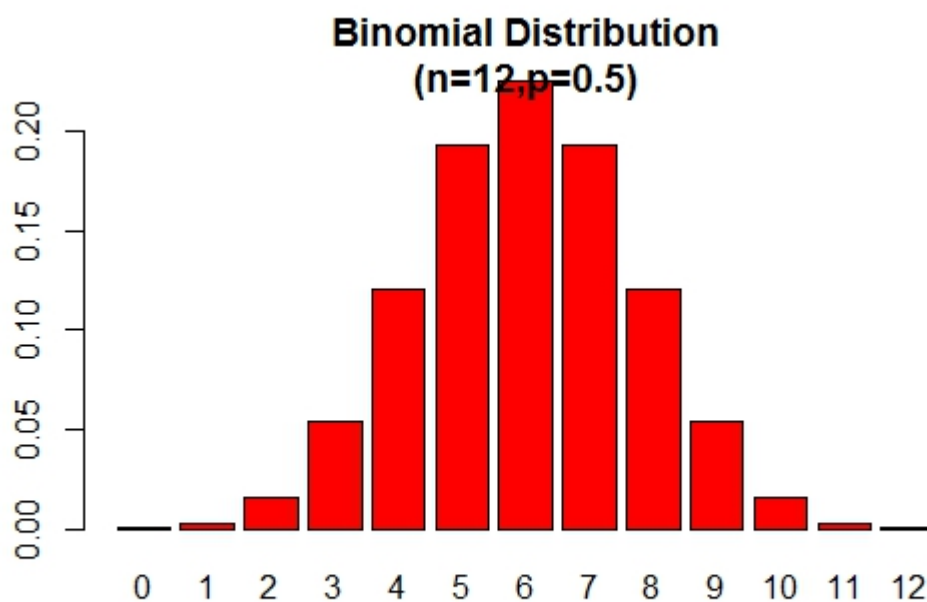
ここでは、 のためにされたdbinom、 pbinom、 qbinom、 およびrbinomについてします。

`dbinom()` は、のさまざまなのをします。には3つのがです。このの1は、のベクトル $x$ なでなければなりません。2と3はのdefining parametersで、 $n$  したと $p$  ことのです。えは、 $n = 5$ 、 $p = 0.5$ の、 $x$ のなは0,1,2,3,4,5です。つまり、`dbinom(x,n,p)` は $x = 0, 1, 2, 3, 4, 5$ の $P(X = x)$ をえます。

```
#Binom(n = 5, p = 0.5) probabilities
> n <- 5; p<- 0.5; x <- 0:n
> dbinom(x,n,p)
[1] 0.03125 0.15625 0.31250 0.31250 0.15625 0.03125
#To verify the total probability is 1
> sum(dbinom(x,n,p))
[1] 1
>
```

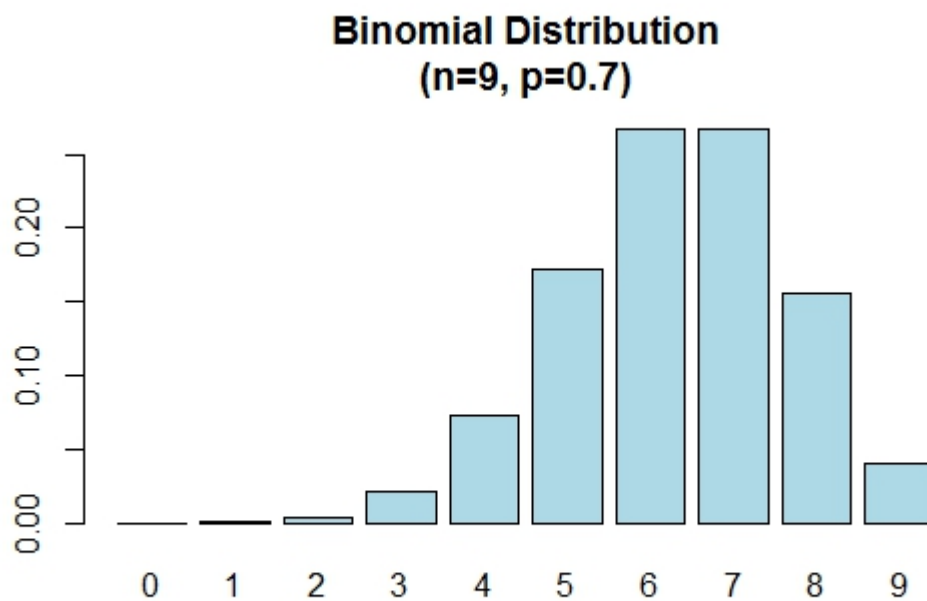
プロットは、ののようにできます。

```
> x <- 0:12
> prob <- dbinom(x,12,.5)
> barplot(prob,col = "red",ylim = c(0,.2),names.arg=x,
 main="Binomial Distribution\n(n=12,p=0.5)")
```



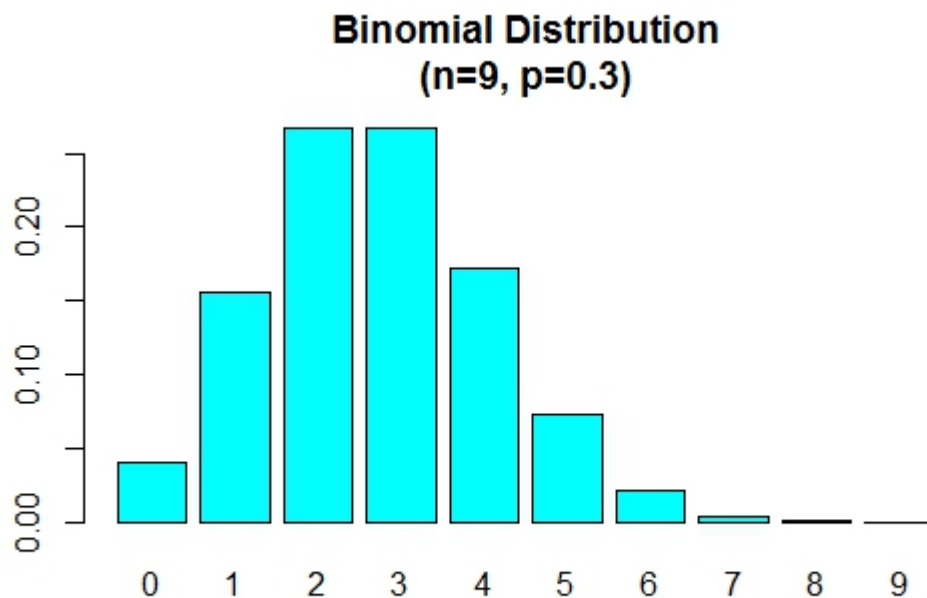
$p = 0.5$ とき、はであることにしてください。 $p$ が0.5よりきいに、がにっていることをするには、のをしてください。

```
> n=9; p=.7; x=0:n; prob=dbinom(x,n,p);
> barplot(prob,names.arg = x,main="Binomial Distribution\n(n=9, p=0.7)",col="lightblue")
```



$p$ が0.5よりさい、はのようになんでいます。

```
> n=9; p=.3; x=0:n; prob=dbinom(x,n,p);
> barplot(prob,names.arg = x,main="Binomial Distribution\n(n=9, p=0.3)",col="cyan")
```



ここで、`pbinom()`のをします。これは、 $P(X \leq x)$ などのをするためにできます。このの1は、のベクトル $x$ のです。

```
Calculating Probabilities
P(X <= 2) in a Bin(n=5,p=0.5) distribution
```

```
> pbinom(2,5,0.5)
[1] 0.5
```

のは、のようにもられる。

```
P(X <= 2) = P(X=0) + P(X=1) + P(X=2)
> sum(dbinom(0:2,5,0.5))
[1] 0.5
```

するために、タイプの  $P(a \leq X \leq b)$

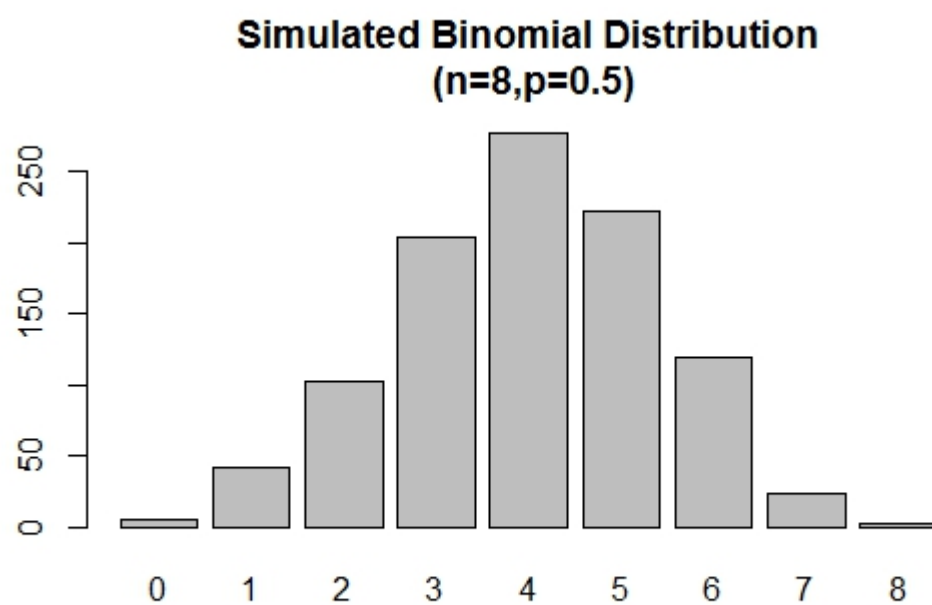
```
P(3 <= X <= 5) = P(X=3) + P(X=4) + P(X=5) in a Bin(n=9,p=0.6) dist
> sum(dbinom(c(3,4,5),9,0.6))
[1] 0.4923556
>
```

テーブルのでをする

```
> n = 10; p = 0.4; x = 0:n;
> prob = dbinom(x,n,p)
> cdf = pbinom(x,n,p)
> distTable = cbind(x,prob,cdf)
> distTable
 x prob cdf
[1,] 0 0.0060466176 0.006046618
[2,] 1 0.0403107840 0.046357402
[3,] 2 0.1209323520 0.167289754
[4,] 3 0.2149908480 0.382280602
[5,] 4 0.2508226560 0.633103258
[6,] 5 0.2006581248 0.833761382
[7,] 6 0.1114767360 0.945238118
[8,] 7 0.0424673280 0.987705446
[9,] 8 0.0106168320 0.998322278
[10,] 9 0.0015728640 0.999895142
[11,] 10 0.0001048576 1.000000000
>
```

`rbinom()` は、されたパラメータをつされたサイズのランダムサンプルをするためにされます。

```
Simulation
> xVal<-names(table(rbinom(1000,8,.5)))
> barplot(as.vector(table(rbinom(1000,8,.5))),names.arg =xVal,
 main="Simulated Binomial Distribution\n (n=8,p=0.5)")
```



オンラインでをむ <https://riptutorial.com/ja/r/topic/1885/>

## 91: の

### Examples

の

データセットのの`sum`をするがありますとします

```
set.seed(20)
df1 <- data.frame(ID = rep(c("A", "B", "C"), each = 3), V1 = rnorm(9), V2 = rnorm(9))
m1 <- as.matrix(df1[-1])
```

これをうにはくのがあります。 `base R` をすると、のオプションは `colSums`

```
colSums(df1[-1], na.rm = TRUE)
```

ここでは、のをしてし、の`sum`をい、 `na.rm = TRUE` データセットに`NA`sがあるをして、  
これは`matrix` もします

```
colSums(m1, na.rm = TRUE)
```

これはループで `lapply/sapply/vapply` を `lapply/sapply/vapply` うことができます

```
lapply(df1[-1], sum, na.rm = TRUE)
```

は`list` であることにすべきである。 `vector` がな

```
sapply(df1[-1], sum, na.rm = TRUE)
```

または

```
vapply(df1[-1], sum, na.rm = TRUE, numeric(1))
```

の、をループする `apply`、 `MARGIN = 1` `apply` をし `apply`

```
apply(m1, 2, FUN = sum, na.rm = TRUE)
```

`dplyr` や `data.table` などのパッケージでこれをうがあります

```
library(dplyr)
df1 %>%
 summarise_at(vars(matches("^V\\d+")), sum, na.rm = TRUE)
```

ここでは、`sum`を`summarise_at`するためになとするをしています。は、`v`でまり、そのに1つの`\\d+`がくすべてのとします。

`data.table`オプションはの`data.table`です。

```
library(data.table)
setDT(df1)[, lapply(.SD, sum, na.rm = TRUE), .SDcols = 2:ncol(df1)]
```

たちは、「`data.table`」に「`data.frame`」に`setDT(df1)`、`で`をするを`.SDcols` `Data.table`のサブセットをとって、ループ`.SD`と`get sum`。

---

でグループをするがあるは、グループを`/`ですることのでうことができます

```
df1 %>%
 group_by(ID) %>%
 summarise_at(vars(matches("^V\\d+")), sum, na.rm = TRUE)
```

すべてのの`sum`がなは、`summarise_each`わりに`summarise_at`できます

```
df1 %>%
 group_by(ID) %>%
 summarise_each(funs(sum(., na.rm = TRUE)))
```

`data.table`オプションはの`data.table`です。

```
setDT(df1)[, lapply(.SD, sum, na.rm = TRUE), by = ID]
```

オンラインでのをむ <https://riptutorial.com/ja/r/topic/2212/>の

## 92: フロー

forループは、ドメインのタスクまたはタスクのセットを繰り返すためのフローメソッドです。forループのコアはのとおりで。

```
for ([index] in [domain]){
 [body]
}
```

どこで

1. [index]はであり、ループの繰り返しごとに [domain] を1つだけとります。
2. [domain]は、するのベクトルです。
3. [body]は、でされるのです。

なとして、forループをしてのベクトルのをすることをえてみましょう。

```
x <- 1:4
cumulative_sum <- 0
for (i in x){
 cumulative_sum <- cumulative_sum + x[i]
}
cumulative_sum
```

## Forループのの

forループは、タスクを繰り返してしするのにです。ただし、にされていないは、applyファンクションファミリのにべてがなくなるがあります。それにもかかわらず、ループをするためにforループにめることができるいくつかのがあります。くの、forループをにすると、のににいられます。

「しくされた」forループは、コアにされ、ループのをキャプチャするオブジェクトをするステートメントをみます。このオブジェクトは、クラスとさのをするがあります。

```
[output] <- [vector_of_length]
for ([index] in [length_safe_domain]){
 [output][index] <- [body]
}
```

するために、ベクトルのを2するループをしてみましょうこれは、のためのなです。このタスクをするための「しい」は、`x_squared <- x^2`。

```
x <- 1:100
x_squared <- vector("numeric", length = length(x))
for (i in seq_along(x)){
 x_squared[i] <- x[i]^2
}
```



ここでも、に `x_squared` レセブタクルをし、`x` とじさのクラス "numeric" をえたことに `x_squared` 。さらに、`seq_along` をして「さのなドメイン」をしました。 `seq_along` は、for ループでのにしたオブジェクトのインデックスのベクトルをします。には `for (i in 1:length(x))` 、`x` さが0の、ループは `1:0` ドメインをしようとし `1:0` その、エラーがします0のインデックスはR。

レセブタクルオブジェクトとさのなドメインは、`apply` ファンクションファミリーによってにされ、ユーザーはforループのわりに `apply` アプローチをなりする `apply` がされます。しかし、にされていれば、forループはのをにえながらコードのをさせることがあります。

## ループのベクトル

Forループは、りしですがあるタスクをするになツールとなることがよくあります。ループがにされ、されると、ループをにすることにがあるかもしれません。

このでは、forループをして、`mtcars` データセットののをしますもう、`colMeans` をしてできるな。

```
column_mean_loop <- vector("numeric", length(mtcars))
for (k in seq_along(mtcars)){
 column_mean_loop[k] <- mean(mtcars[[k]])
}
```

ループのをとしてきえることにより、forループをにすることができます。

```
col_mean_fn <- function(x) mean(x)
column_mean_apply <- vapply(mtcars, col_mean_fn, numeric(1))
```

をするには

```
identical(column_mean_loop,
 unname(column_mean_apply)) #* vapply added names to the elements
 #* remove them for comparison
```

ベクトルされたのは、のコードをできることです。オブジェクトのさとをし、さのなをするみは、`apply` によってされます。さらに、はループよりしいです。のいは、とのさによってはのではできるほどです。

## Examples

なループ

このでは、データフレームのこのは `mtcars` の2をします。

オプションAインデックス

```
squared_deviance <- vector("list", length(mtcars))
```

```
for (i in seq_along(mtcars)){
 squared_deviance[[i]] <- (mtcars[[i]] - mean(mtcars[[i]]))^2
}
```

squared\_deviance は、りに11のリストです。

```
class(squared_deviance)
length(squared_deviance)
```

## オプションBインデックス

```
squared_deviance <- vector("list", length(mtcars))
Squared_deviance <- setNames(squared_deviance, names(mtcars))
for (k in names(mtcars)){
 squared_deviance[[k]] <- (mtcars[[k]] - mean(mtcars[[k]]))^2
}
```

として data.frame なはどうすればよいですかさて、リストをのオブジェクトにするためのくのオプションがあります。しかし、このはおそらくもシンプルなので、for を data.frame にすることになります。

```
squared_deviance <- mtcars #copy the original
squared_deviance[TRUE]<-NA #replace with NA or do squared_deviance[,]<-NA
for (i in seq_along(mtcars)){
 squared_deviance[[i]] <- (mtcars[[i]] - mean(mtcars[[i]]))^2
}
dim(squared_deviance)
[1] 32 11
```

オプションBをしても、はじイベントになります。

## Forループの

ループにいをするために、4つのなるでのをします。

1. されていないループをする
2. forループにされたものをする
3. \*apply ファミリをする
4. colMeans の

これらのオプションはコードでされます。オプションをするためののがされます。にのをいます。

## forループがされていない

```
column_mean_poor <- NULL
for (i in 1:length(mtcars)){
 column_mean_poor[i] <- mean(mtcars[[i]])
}
```

## よくされたループ

```
column_mean_optimal <- vector("numeric", length(mtcars))
for (i in seq_along(mtcars)){
 column_mean_optimal <- mean(mtcars[[i]])
}
```

### vapply

```
column_mean_vapply <- vapply(mtcars, mean, numeric(1))
```

### colMeans

```
column_mean_colMeans <- colMeans(mtcars)
```

の

これらの4つのアプローチをベンチマークしたのをにしますコードはされません

```
Unit: microseconds
 expr min lq mean median uq max neval cld
 poor 240.986 262.0820 287.1125 275.8160 307.2485 442.609 100 d
 optimal 220.313 237.4455 258.8426 247.0735 280.9130 362.469 100 c
 vapply 107.042 109.7320 124.4715 113.4130 132.6695 202.473 100 a
 colMeans 155.183 161.6955 180.2067 175.0045 194.2605 259.958 100 b
```

されたforループが、うまくされていないforループをしたことにしてください。なforループは、オブジェクトのさをえずさせており、さのに、Rはオブジェクトのクラスをしています。

このオーバーヘッドののは、ループをするにオブジェクトのタイプとそのさをするによって、されたforループによってされます。

しかし、このでは、vapplyをするとが2になります。そのは、がでなければならぬということです1つのがではないはエラーがされます。

colMeansは、よりタッチいvapply。このいは、colMeansでされたいくつかのエラーチェックと、vapplyではされなかったas.matrix(mtcars)はmtcarsであるdata.frameにvapplyます。

## のループwhileとrepeat

Rはながなでされるwhileとrepeat 2つのループをします。

### whileループ

whileループのなはのとおりです。

```
while (condition) {
 ## do something
 ## in loop body
}
```

`condition` はループに入るにされます。 `condition` が `TRUE` とされると、ループのコードがされ、`condition` が `FALSE` までこのプロセスが繰り返され、`FALSE` または `break` にした、を。 `for` ループとはなり、`while` ループがをしてをする、はにしてし、ループです。たとえば、のループはじタスクをします。

```
for (i in 0:4) {
 cat(i, "\n")
}
0
1
2
3
4

i <- 0
while (i < 5) {
 cat(i, "\n")
 i <- i + 1
}
0
1
2
3
4
```

の `while` ループでは、ループをぐために、 `i <- i + 1` がです。

さらに、 `while` ループをするには、ループのから `break` するためのびしがです。

```
iter <- 0
while (TRUE) {
 if (runif(1) < 0.25) {
 break
 } else {
 iter <- iter + 1
 }
}
iter
#[1] 4
```

このでは、 `condition` はに `TRUE` であるため、ループをさせるのは、ので `break` をびすことです。 `iter` のなは、このをするときの PRNG のにし、コードがされるたびになるにをするがあることにしてください。

## repeat ループ

repeat コンストラクトはに while (TRUE) { ## something } とじですが、のをとります

```
repeat ({
 ## do something
 ## in loop body
})
```

な {} はではありませんが、 () はです。 repeat をしてのをきし、

```
iter <- 0
repeat ({
 if (runif(1) < 0.25) {
 break
 } else {
 iter <- iter + 1
 }
})
iter
#[1] 2
```

---

## break

break は、すぐにむループをするだけであることにする break がです。つまり、はループです。

```
while (TRUE) {
 while (TRUE) {
 cat("inner loop\n")
 break
 }
 cat("outer loop\n")
}
```

しかし、しがあれば、れになったループからにれるがあります。たとえば、のでにループするのをえてみましょう。

```
while (TRUE) {
 cat("outer loop body\n")
 while (TRUE) {
 cat("inner loop body\n")
 x <- runif(1)
 if (x < .3) {
 break
 } else {
 cat(sprintf("x is %.5f\n", x))
 }
 }
}
```

1つののは、 break とはなり、 return にはレベルのみループにをすがあることをすることです。しかし、 return はでのみなので、に break をの return() きえることはできませんが、をとしてラップするがあります

```

(function() {
 while (TRUE) {
 cat("outer loop body\n")
 while (TRUE) {
 cat("inner loop body\n")
 x <- runif(1)
 if (x < .3) {
 return()
 } else {
 cat(sprintf("x is %.5f\n", x))
 }
 }
 }
})()

```

あるいは、のにダミー `exit` をし、がったらループから `<<-` でアクティブにすることができます

```

exit <- FALSE
while (TRUE) {
 cat("outer loop body\n")
 while (TRUE) {
 cat("inner loop body\n")
 x <- runif(1)
 if (x < .3) {
 exit <<- TRUE
 break
 } else {
 cat(sprintf("x is %.5f\n", x))
 }
 }
 if (exit) break
}

```

オンラインでフローをむ <https://riptutorial.com/ja/r/topic/2201/フロー>

## 93: しいからベクトルするコードの

### Examples

Rcppをしたループのタフなベクトルの

のがされ `first`、`x_i`が $\cos(x_{i-1} + 1)$  しいさ `len` ベクトルをするのtough-to-vectorize forループを  
えてみましょう。

```
repeatedCosPlusOne <- function(first, len) {
 x <- numeric(len)
 x[1] <- first
 for (i in 2:len) {
 x[i] <- cos(x[i-1] + 1)
 }
 return(x)
}
```

このコードでは、 $\cos(x[i-1]+1)$  したforループがされます。しかし、Rに "x + 1の"がないので、こ  
のをRでベクトルすることはではありません。

このをするとしては、RcppパッケージをしてC++でするがあります。

```
library(Rcpp)
cppFunction("NumericVector repeatedCosPlusOneRcpp(double first, int len) {
 NumericVector x(len);
 x[0] = first;
 for (int i=1; i < len; ++i) {
 x[i] = cos(x[i-1]+1);
 }
 return x;
}")
```

これはしばしばなにきなスピードアップをもたらし、まったくじをもたらします

```
all.equal(repeatedCosPlusOne(1, 1e6), repeatedCosPlusOneRcpp(1, 1e6))
[1] TRUE
system.time(repeatedCosPlusOne(1, 1e6))
user system elapsed
1.274 0.015 1.310
system.time(repeatedCosPlusOneRcpp(1, 1e6))
user system elapsed
0.028 0.001 0.030
```

この、Rcppコードは、Rのアプローチでは1.31ではなく、0.03で100のさのベクトルをします。

バイトコンパイルによるループののいベクトルの

のがされ `first`、`x_i`が $\cos(x_{i-1} + 1)$  しいさ `len` ベクトルをするのしいtoベクトルをえてみましょう

う  $\cos(x_{i-1} + 1)$

```
repeatedCosPlusOne <- function(first, len) {
 x <- numeric(len)
 x[1] <- first
 for (i in 2:len) {
 x[i] <- cos(x[i-1] + 1)
 }
 return(x)
}
```

1のコードをきさずにこのようなをするなの1つは、Rコンパイルパッケージをしてコードをバイトコンパイルすることです。

```
library(compiler)
repeatedCosPlusOneCompiled <- cmpfun(repeatedCosPlusOne)
```

のは、しばしばじをしながら、かなりくなります

```
all.equal(repeatedCosPlusOne(1, 1e6), repeatedCosPlusOneCompiled(1, 1e6))
[1] TRUE
system.time(repeatedCosPlusOne(1, 1e6))
user system elapsed
1.175 0.014 1.201
system.time(repeatedCosPlusOneCompiled(1, 1e6))
user system elapsed
0.339 0.002 0.341
```

この、バイトコンパイルは、1.20から0.34までのさ1のベクトルのしいto-vectorizeをスピードアップしました。

リマーク

repeatedCosPlusOneのは、ののなアプリケーションとして、Reduceよりになります。

```
iterFunc <- function(init, n, func) {
 funcs <- replicate(n, func)
 Reduce(function(., f) f(.), funcs, init = init, accumulate = TRUE)
}
repeatedCosPlusOne_vec <- function(first, len) {
 iterFunc(first, len - 1, function(.) cos(. + 1))
}
```

repeatedCosPlusOne\_vecは、repeatedCosPlusOne「ベクトル」とみなすことができます。しかし、それは2の1でくることがされる

```
library(microbenchmark)
microbenchmark(
 repeatedCosPlusOne(1, 1e4),
 repeatedCosPlusOne_vec(1, 1e4)
)
#> Unit: milliseconds
#> expr min lq mean median uq max
```



```
neval cld
#> repeatedCosPlusOne(1, 10000) 8.349261 9.216724 10.22715 10.23095 11.10817 14.33763
100 a
#> repeatedCosPlusOne_vec(1, 10000) 14.406291 16.236153 17.55571 17.22295 18.59085 24.37059
100 b
```

オンラインでいいからベクトルするコードのをむ <https://riptutorial.com/ja/r/topic/1203/> いいからベクトルするコードの

## 94: されたアーカイブのファイルのと

### Examples

**.zip** アーカイブからファイルをする

zipアーカイブをunzipするには、utilsパッケージベースRにまれていますのunzipをします。

```
unzip(zipfile = "bar.zip", exdir = "./foo")
```

これにより、"bar.zip"のすべてのファイルが"bar.zip" "foo"ディレクトリにされ、にじてされます。チルダのはディレクトリからにわれます。または、パスをzipファイルにすこともできます。

**.zip** アーカイブにファイルをリストする

zipアーカイブのファイルをするには、utilsパッケージベースRにまれていますのunzipをします。

```
unzip(zipfile = "bar.zip", list = TRUE)
```

これにより、"bar.zip"のすべてのファイルがリストされ、noneがされます。チルダのはディレクトリからにわれます。または、パスをzipファイルにすこともできます。

**.tar** アーカイブにファイルをリストする

tarアーカイブのファイルをリストするには、utilsパッケージベースRにまれていますのuntarをuntarます。

```
untar(zipfile = "bar.tar", list = TRUE)
```

これにより、"bar.tar"のすべてのファイルがリストされ、noneがされます。チルダのはディレクトリからにわれます。あるいは、パスをtarfileにすこともできます。

**.tar** アーカイブからのファイルの

tarアーカイブからファイルをするには、utilsパッケージベースRにまれていますのuntarをuntarます。

```
untar(tarfile = "bar.tar", exdir = "./foo")
```

これにより、"bar.tar"のすべてのファイルが"bar.tar" "foo"ディレクトリにされ、にじてされます。チルダのはディレクトリからにわれます。あるいは、パスをtarfileにすこともできます。

## ディレクトリのすべての.zipアーカイブをする

なforループをすると、ディレクトリのすべてのzipアーカイブをできます。

```
for (i in dir(pattern=".zip$"))
 unzip(i)
```

dirは、patternでされたパターンにするディレクトリのファイルのベクトルをしpattern。このベクトルは、zipアーカイブをunzipするためにunzipをして、インデックスiでループされます。

オンラインでされたアーカイブのファイルのとをむ <https://riptutorial.com/ja/r/topic/4323/されたアーカイブのファイルのと>

## 95: データファイルなどのI/O

き

のについてもしてください。

### Examples

シェイプファイルのインポートとエクスポート

`rgdal` パッケージをすると、Rをしてシェイプファイルをインポートおよびエクスポートすることができます。 `readOGR` をしてシェイプファイルをインポートできます。 `ArcGIS`などのファイルをインポートするは、の`dsn`はシェイプファイルをむフォルダへのパスです。 `layer`はファイルがわっていないシェープファイルのです `map`だけで、 `map.shp`ありませ `map.shp`。

```
library(rgdal)
readOGR(dsn = "path\\to\\the\\folder\\containing\\the\\shapefile", layer = "map")
```

シェイプファイルをエクスポートするには、 `writeOGR`をします。 のはR. `dsn`されたオブジェクトであり、 `layer`はとじです。 `obligatory 4`.は、シェイプファイルのにされるドライバです。 `ogrDrivers()` は、なすべてのドライバをリストします。 シェイプファイルを `ArcGis`または `QGis`にエクスポートするは、 `driver = "ESRI Shapefile"`できます。

```
writeOGR(Rmap, dsn = "path\\to\\the\\folder\\containing\\the\\shapefile", layer = "map",
 driver = "ESRI Shapefile")
```

`tmap` パッケージにはにな `read_shape()` があります。 これは `rgdal::readOGR()` ラッパーです。 `read_shape()` は、シェイプファイルをたくさんインポートするプロセスをします。 として、 `tmap` はにいです。

オンラインでデータファイルなどのI/Oをむ <https://riptutorial.com/ja/r/topic/5538/データ-ファイルなど-i-o>

## 96: の

き

データのI/Oも

## Examples

パッケージマッピングからの`map`によるなマッピング

パッケージ`maps`の`map()`は、Rをしてマッピングをするためのなをします。

なはのようにくことができます

```
require (maps)
map ()
```



アウトラインのは、カラーパラメータ`col`カラーのまたは16のいずれかにすることでできます。

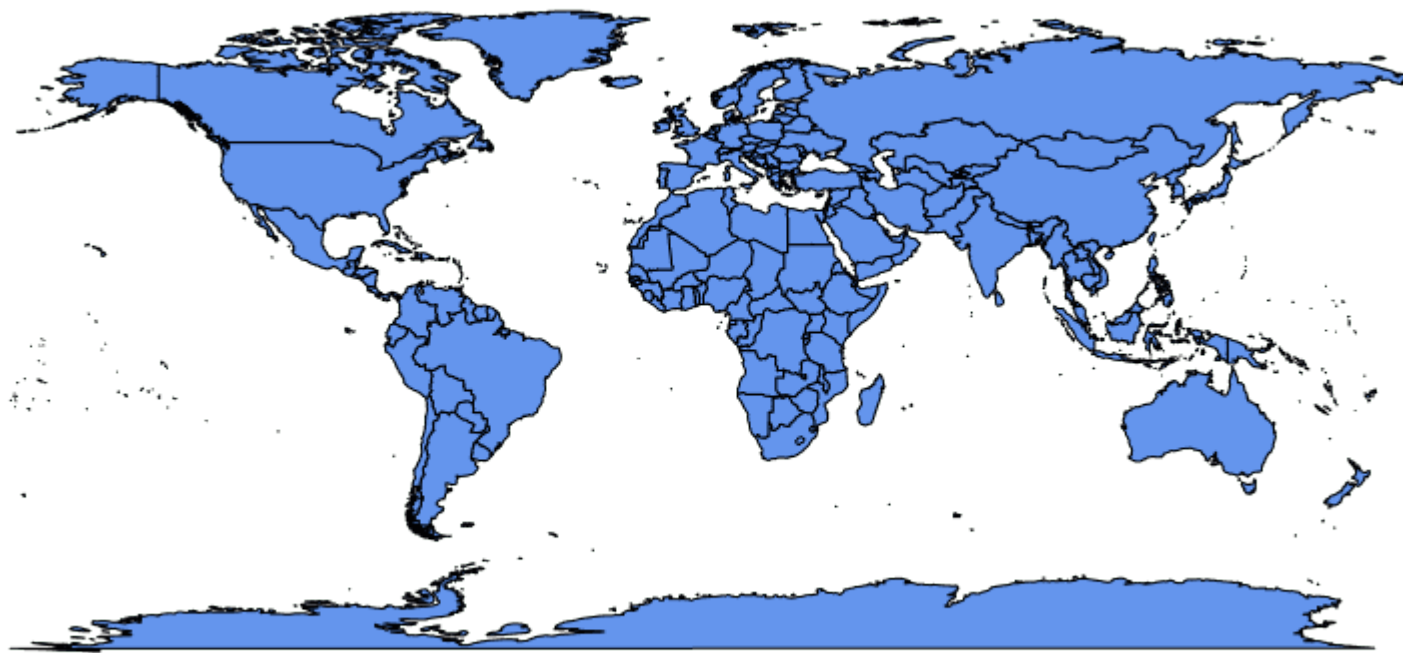
```
require (maps)
```

```
map(col = "cornflowerblue")
```



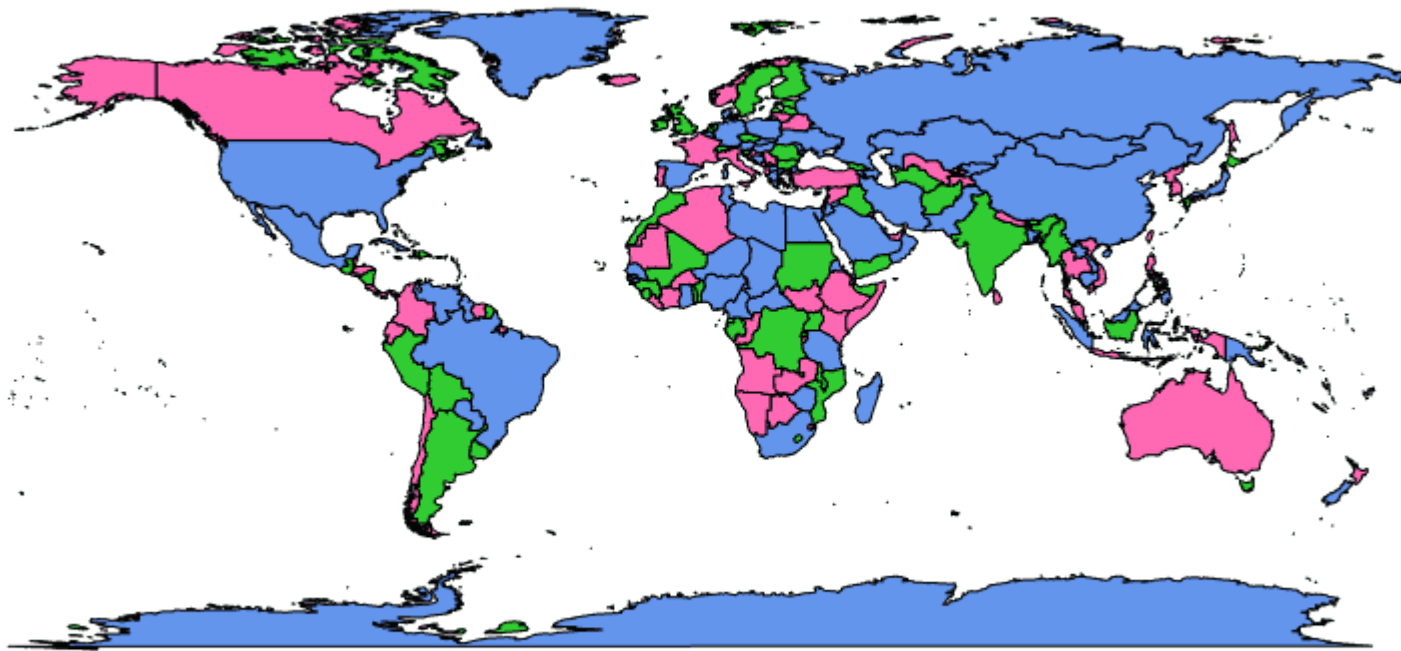
とをめるために`col`々がすることができます`fill = TRUE`

```
require(maps)
map(fill = TRUE, col = c("cornflowerblue"))
```



`fill = TRUE`もされている、のさのベクトルを`col`ことができます

```
require(maps)
map(fill = TRUE, col = c("cornflowerblue", "limegreen", "hotpink"))
```



のでは、`col`からの`col`は、をすマップのポリゴンににりてられ、ポリゴンよりもがないはがりサイクルされます。

カラーコードをして、ににできるをすこともできます。そのようにしてされたは、「choropleth」としてられています。

のchoroplethのでは、`database`が`"county"`と`"state"`の`map()`のを、`unemp`と`county.fips`データセットのデータをしてカラーコードのにしています。

```
require(maps)
if(require(mapproj)) { # mapproj is used for projection="polyconic"
 # color US county map by 2009 unemployment rate
 # match counties to map using FIPS county codes
 # Based on J's solution to the "Choropleth Challenge"
 # Code improvements by Hack-R (hack-r.github.io)

 # load data
 # unemp includes data for some counties not on the "lower 48 states" county
 # map, such as those in Alaska, Hawaii, Puerto Rico, and some tiny Virginia
 # cities
 data(unemp)
 data(county.fips)

 # define color buckets
 colors = c("paleturquoise", "skyblue", "cornflowerblue", "blueviolet", "hotpink",
 "darkgrey")
}
```



```

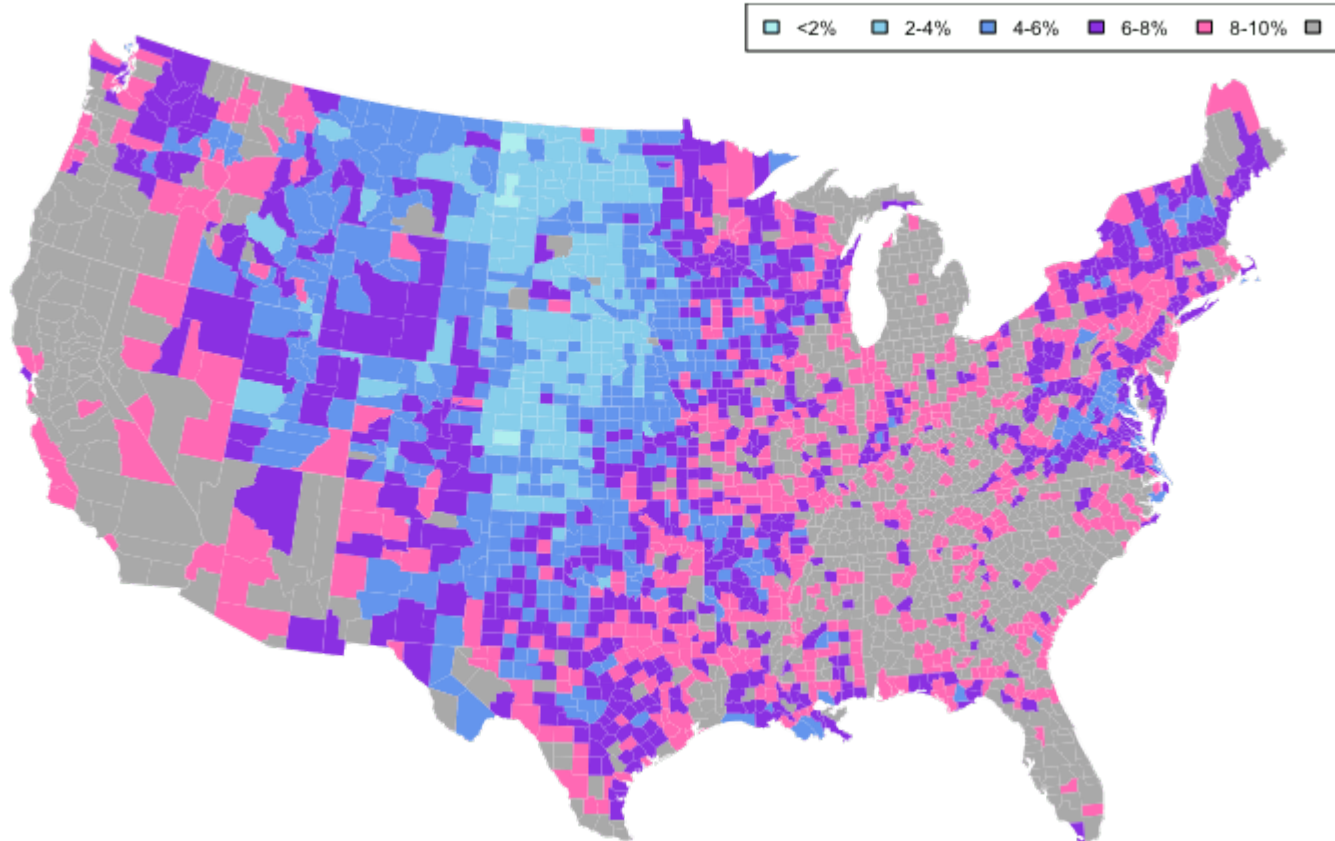
unemp$colorBuckets <- as.numeric(cut(unemp$unemp, c(0, 2, 4, 6, 8, 10, 100)))
leg.txt <- c("<2%", "2-4%", "4-6%", "6-8%", "8-10%", ">10%")

align data with map definitions by (partial) matching state, county
names, which include multiple polygons for some counties
cnty.fips <- county.fips$fips[match(map("county", plot=FALSE)$names,
 county.fips$polynome)]
colorsmatched <- unemp$colorBuckets[match(cnty.fips, unemp$fips)]

draw map
par(mar=c(1, 1, 2, 1) + 0.1)
map("county", col = colors[colorsmatched], fill = TRUE, resolution = 0,
 lty = 0, projection = "polyconic")
map("state", col = "white", fill = FALSE, add = TRUE, lty = 1, lwd = 0.1,
 projection="polyconic")
title("unemployment by county, 2009")
legend("topright", leg.txt, horiz = TRUE, fill = colors, cex=0.6)
}

```

## unemployment by county, 2009



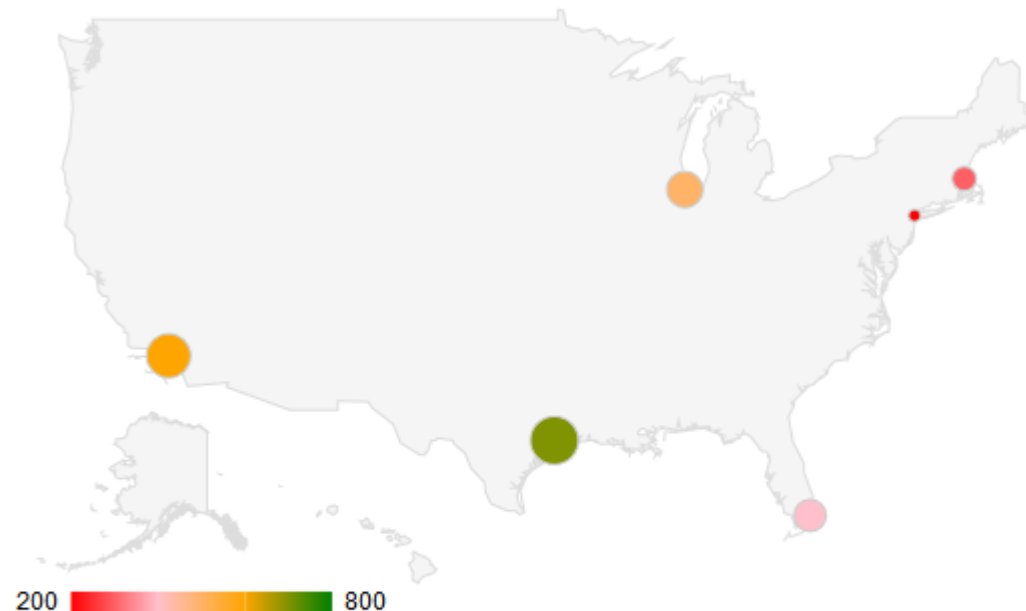
## Google Vizをった50となコレラ

なは、50のすべてのアメリカのアラスカとハワイがされているをしているのように、に々のをしにするです。

な50をすることは、Googleマップをするとです。GoogleのAPIへのインタフェースは、パッケージが `googleVis`、`ggmap`、および `RgoogleMaps`。

```
require(googleVis)

G4 <- gvisGeoChart(CityPopularity, locationvar='City', colorvar='Popularity',
 options=list(region='US', height=350,
 displayMode='markers',
 colorAxis="{values:[200,400,600,800],
 colors:['red', 'pink', 'orange','green']}"
)
plot(G4)
```



Data: CityPopularity • Chart ID: GeoChartID28504adb439a • googleVis-0.5.2  
R version 3.1.0 (2014-04-10) • [Google Terms of Use](#) • [Documentation and Data Policy](#)

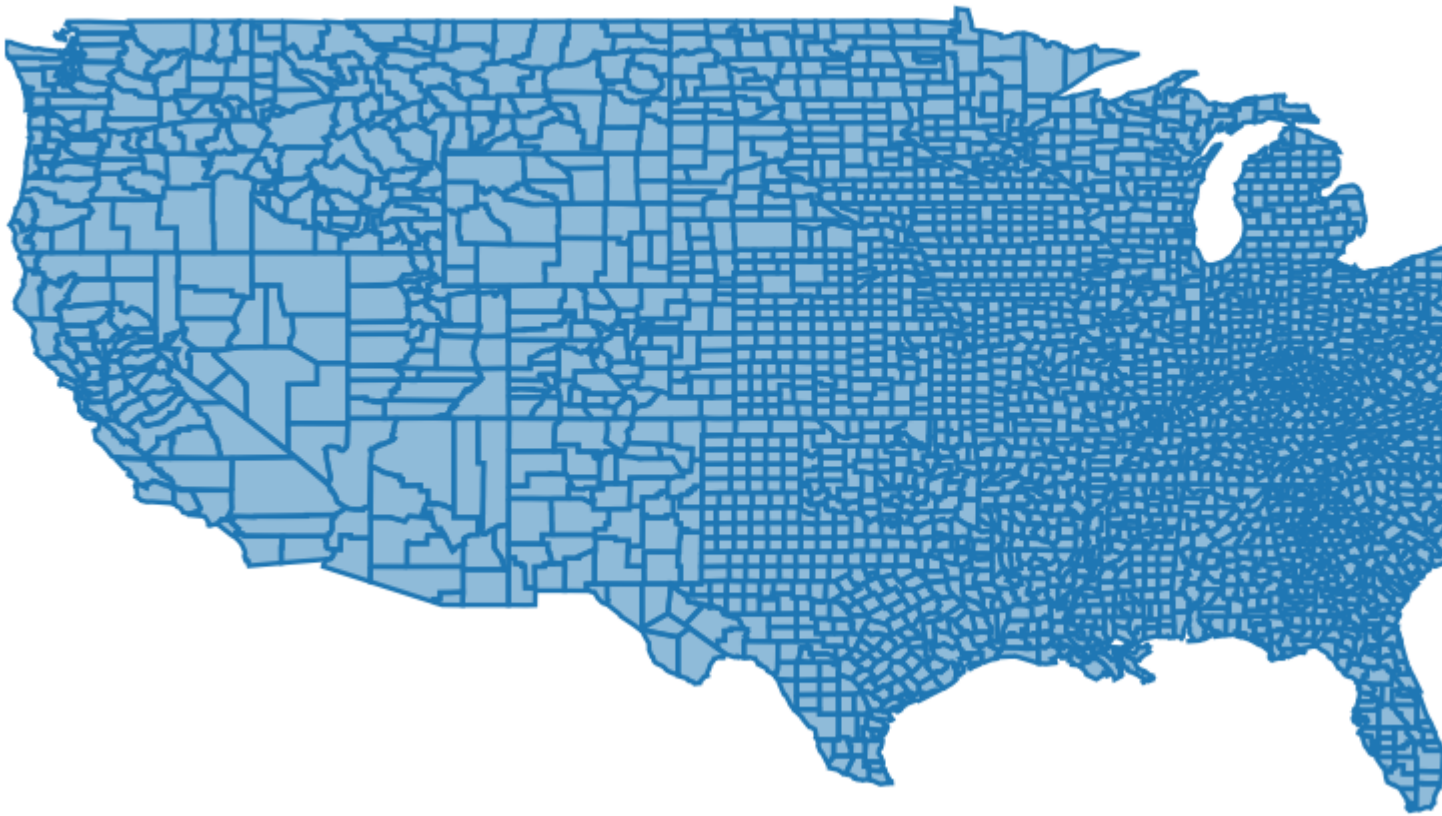
`gvisGeoChart()` は、パッケージ `maps` の `map()` などのマッピングメソッドとして、コレオレンスをするためにはあらかじめコーディングをとし `maps` の `colorvar` パラメータをすると、`locationvar` パラメータでされたレベルで `colorvar` ことができます。 `options` としてリストにされるさまざまなオプションは、サイズ `height`、`markers`、`colorAxis` および `colors` などのマップののカスタマイズをにします。

## インタラクティブプロットマップ

`plotly` パッケージは、などのインタラクティブプロット、くのことができます。 `plotly` をするはいくつかあります。どちらのマップデータを `plot_ly()` または `ggplotly()` の「ネイティブ」マッピングをする `plot_geo()` または `plot_mapbox()` の、あるいはみわせ。でマップをするはのようになります。

```
library(plotly)
map_data("county") %>%
 group_by(group) %>%
 plot_ly(x = ~long, y = ~lat) %>%
 add_polygons() %>%
```

```
layout(
 xaxis = list(title = "", showgrid = FALSE, showticklabels = FALSE),
 yaxis = list(title = "", showgrid = FALSE, showticklabels = FALSE)
)
```



のアプローチのみわけのために、スワップ `plot_ly()` のため `plot_geo()` または `plot_mapbox()` ので。よりのについては、[プロットの](#) をしてください。

のは、[layout.geo](#) をしてマップのしさとズームレベルをする「にネイティブ」なアプローチです。また、`maps` のデータベース `world.cities` をしてブラジルのをフィルタリングし、それらを「ネイティブ」マップのにプロットします。

な `poph` はとそのマウスのホバーにされるをつテキストです。 `q` は、のからのけられたです。 `ge` はマップのレイアウトにするがあります。については、[パッケージのドキュメント](#) をしてください。

```
library(maps)
dfb <- world.cities[world.cities$country.etc=="Brazil",]
library(plotly)
dfb$poph <- paste(dfb$name, "Pop", round(dfb$pop/1e6,2), " millions")
dfb$q <- with(dfb, cut(pop, quantile(pop), include.lowest = T))
levels(dfb$q) <- paste(c("1st", "2nd", "3rd", "4th"), "Quantile")
dfb$q <- as.ordered(dfb$q)

ge <- list(
```

```

scope = 'south america',
showland = TRUE,
landcolor = toRGB("gray85"),
subunitwidth = 1,
countrywidth = 1,
subunitcolor = toRGB("white"),
countrycolor = toRGB("white")
)

plot_geo(dfb, lon = ~long, lat = ~lat, text = ~poph,
 marker = ~list(size = sqrt(pop/10000) + 1, line = list(width = 0)),
 color = ~q, locationmode = 'country names') %>%
layout(geo = ge, title = 'Populations
(Click legend to toggle)')

```



リーフレットをった**HTML**マップの

リーフレットは、ウェブのダイナミックマップをするためのオープンソースのJavaScriptライブラリです。RStudioは、[htmlwidgets](#)でされた[leaflet](#)パッケージからなリーフレットのRバインディングを[htmlwidgets](#)。リーフレットマップは、[RMarkdown](#)と[Shiny](#)エコシステムとうまくされて



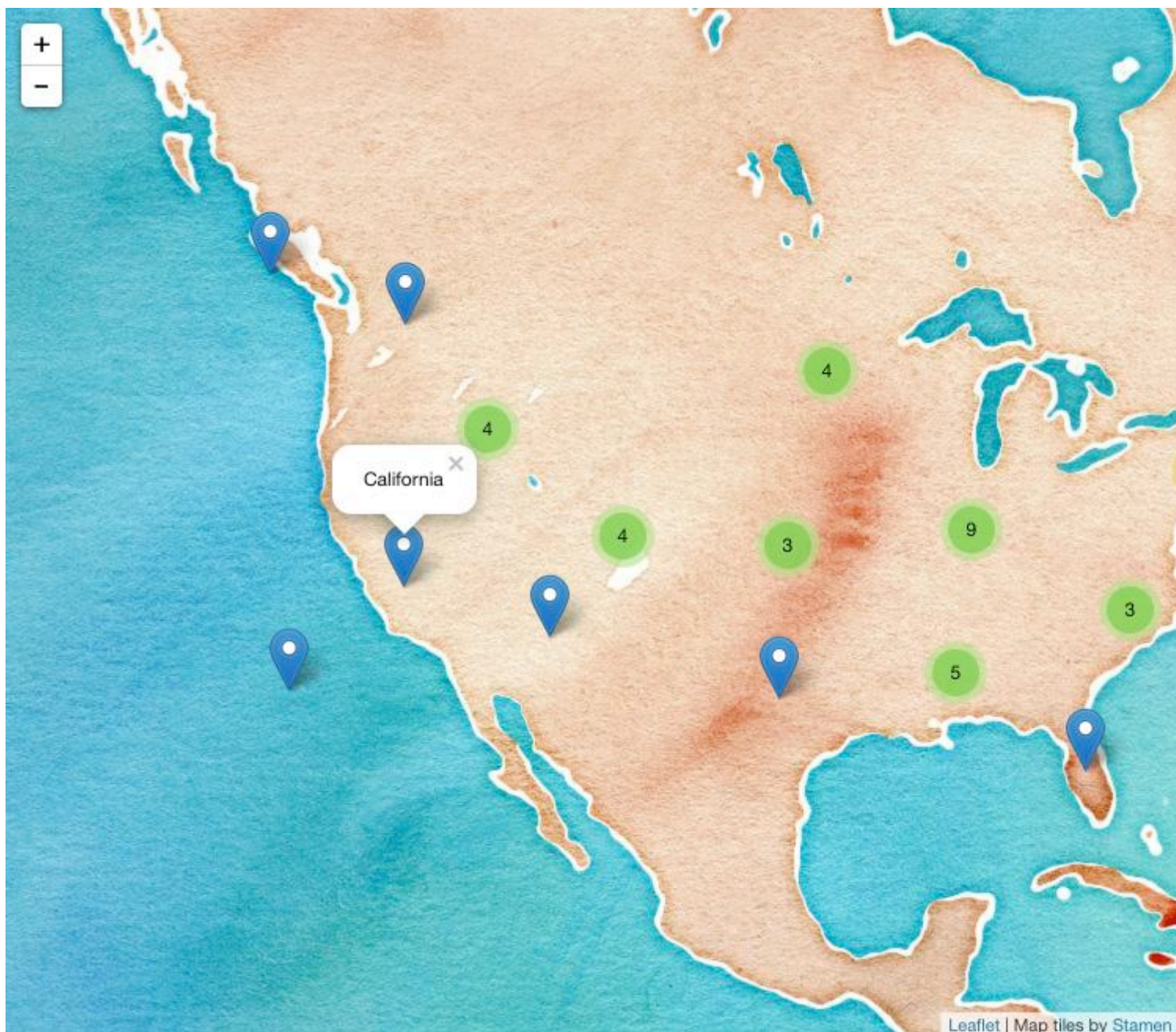
います。

マップは、`leaflet()` をしてマップをし、マップレイヤをまたはするをしてパイプされます。ポップアップをつマーカーからポリゴンまで、くののレイヤーができます。された`data.frame`の`leaflet()`スタイルをしてアクセスされている~。

`state.name`および`state.center`データセットをマップするには

```
library(leaflet)

data.frame(state.name, state.center) %>%
 leaflet() %>%
 addProviderTiles('Stamen.Watercolor') %>%
 addMarkers(lng = ~x, lat = ~y,
 popup = ~state.name,
 clusterOptions = markerClusterOptions())
```



スクリーンショットバージョンはクリックしてください

## Shinyアプリケーションのリーフレットマップ

リーフレットパッケージは、[シャイニー](#)と**になる**ようにされています

UIでは、あなたは、`leafletOutput()` し、サーバーにあなたが**びす** `renderLeaflet()`

```
library(shiny)
library(leaflet)

ui <- fluidPage(
 leafletOutput("my_leaf")
)

server <- function(input, output, session){

 output$my_leaf <- renderLeaflet({

 leaflet() %>%
 addProviderTiles('Hydda.Full') %>%
 setView(lat = -37.8, lng = 144.8, zoom = 10)

 })

}

shinyApp(ui, server)
```

ただし、`renderLeaflet`にするは、**がされるたびに**マップがされます。

したがって、にされているマップをするには、`leafletProxy()`をするがあります。

、`leaflet`をしてマップのなをし、`leafletProxy`をしてをします。たとえば、のようになります。

```
library(shiny)
library(leaflet)

ui <- fluidPage(
 sliderInput(inputId = "slider",
 label = "values",
 min = 0,
 max = 100,
 value = 0,
 step = 1),
 leafletOutput("my_leaf")
)

server <- function(input, output, session){
 set.seed(123456)
 df <- data.frame(latitude = sample(seq(-38.5, -37.5, by = 0.01), 100),
 longitude = sample(seq(144.0, 145.0, by = 0.01), 100),
 value = seq(1,100))

 ## create static element
 output$my_leaf <- renderLeaflet({

 leaflet() %>%
 addProviderTiles('Hydda.Full') %>%
```

```

 setView(lat = -37.8, lng = 144.8, zoom = 8)

 })

 ## filter data
 df_filtered <- reactive({
 df[df$value >= input$slider,]
 })

 ## respond to the filtered data
 observe({

 leafletProxy(mapId = "my_leaf", data = df_filtered()) %>%
 clearMarkers() %>% ## clear previous markers
 addMarkers()

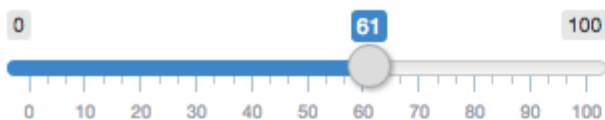
 })

}

shinyApp(ui, server)

```

values



オンラインでのをむ <https://riptutorial.com/ja/r/topic/1372/>の

## Examples

、データ、

Rでは、きデータをしてデータオブジェクトがされます。オブジェクトのは「」とばれることがあります、そののはRドキュメントではのをちません。Rのはとをし、`az`、`Az`、`0-9`、ドット/ピリオド`.`、アンダースコア`_`をむことがあります。データのをするには、のにわなければなりません。

- やアンダースコアえは`_1a`、またはな`.11`またはダッシュ`-`やスペースで`_1a`は、でんだときにのみできます。と``_11``はバックティックでされます

```
list('_11' = "a")
#$`_11`
#[1] "a"
```

- 、ドット、アンダースコアののすべてのみわせはにできます。バックティックのによるはじオブジェクトをします。
- はでまる`.`システムとみなされ、`ls()`をしてにされるわけではありません。

のにはありません。

なオブジェクトのをいくつかかけます `foobar`、`foo.bar`、`foo_bar`、`.foobar`

Rでは、には、`<-`をしてがりてられます。`=`はにをするためにもできますが、びしのでパラメータとをけるのがです。オペレータののスペースをすると、ユーザにがじるがあることにしてください。`a<-1`は、`a < -1`ではなく`a <- 1` `a <- 1`としてされます。

```
> foo <- 42
> fooEquals = 43
```

したがって、`foo`は42のがりてられます。コンソールで`foo`とすると42がされ、`fooEquals`とすると43がされます。

```
> foo
[1] 42
> fooEquals
[1] 43
```

のコマンドは、`x`というのにをりて、にをします。

```
> (x <- 5)
[1] 5
```



```
actually two function calls: first one to `<-`; second one to the `()`-function
> is.function(``)
[1] TRUE # Often used in R help page examples for its side-effect of printing.
```

->をとってにすることもできます。

```
> 5 -> x
> x
[1] 5
>
```

## データのタイプ

Rにスカラーデータはありません。さ1のベクトルはスカラーのようにいます。

- ベクトルベクトルは、じクラスのオブジェクトのシーケンスでなければなりません。のシーケンス、のシーケンス、のシーケンス。 `v <- c(2, 3, 7, 10)` `v2 <- c("a", "b", "c")`はともベクトルです。
- 、またはの。 `a <- matrix(data = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12), nrow = 4, ncol = 3, byrow = F)`。ベクトルとに、はじクラスのでられていなければなりません。からをするにはとをするがあります `a[1,2]`は、12のである `[1] 5`をします。
- リストなるの `mylist <- list(course = 'stat', date = '04/07/2009', num_isc = 7, num_cons = 6, num_mat = as.character(c(45020, 45679, 46789, 43126, 42345, 47568, 45674)), results = c(30, 19, 29, NA, 25, 26, 27))`。リストからをするには、リストのがいているまたはインデックスでをします。えられたでは、 `mylist$results`と `mylist[[6]]`はじをします。 `mylist[6]`をしても、Rはエラーをすことはありませんが、をリストとしてします。 `mylist[[6]][2]`はされていますが19がえられます、 `mylist[6][2]`はエラーを `mylist[6][2]`ます。
- `data.frame`じさのベクトルであるが、おそらくなるタイプのベクトルをつオブジェクト。それらはではありません。 `exam <- data.frame(matr = as.character(c(45020, 45679, 46789, 43126, 42345, 47568, 45674)), res_S = c(30, 19, 29, NA, 25, 26, 27), res_O = c(3, 3, 1, NA, 3, 2, NA), res_TOT = c(30, 22, 30, NA, 28, 28, 27))`。は、 `exam$matr`、 `exam[, 'matr']`またはインデックス `exam[1]`、 `exam[, 1]`でみることができます。は、 `exam['rowname', ]`またはインデックス `exam[1, ]`でもみることができます。データフレームにはの `rownames-attribute`としいさのをつなるリストです。

## なといくつかのいアドバイス

デフォルトのはごとにわれます。オペレータののの?Syntaxをしてください。ほとんどのおよびRののものは、のをするリサイクルをとっています。これらのオブジェクトがえられた

## オブジェクトの

```
> a <- 1
> b <- 2
```

```
> c <- c(2,3,4)
> d <- c(10,10,10)
> e <- c(1,2,3,4)
> f <- 1:6
> W <- cbind(1:4,5:8,9:12)
> Z <- rbind(rep(0,3),1:3,rep(10,3),c(4,7,1))
```

## いくつかのベクトル

```
> a+b # scalar + scalar
[1] 3
> c+d # vector + vector
[1] 12 13 14
> a*b # scalar * scalar
[1] 2
> c*d # vector * vector (componentwise!)
[1] 20 30 40
> c+a # vector + scalar
[1] 3 4 5
> c^2 #
[1] 4 9 16
> exp(c)
[1] 7.389056 20.085537 54.598150
```

## いくつかのベクトルの

```
> c+e # warning but.. no errors, since recycling is assumed to be desired.
[1] 3 5 7 6
Warning message:
In c + e : longer object length is not a multiple of shorter object length
```

Rはそれができるものをし、いベクトルをしてをめる...は、2つのベクトルのさがにはではないためにのみえられます。 `c + f`のありません。

## のマトリックス

```
> Z+W # matrix + matrix #(componentwise)
> Z*W # matrix* matrix#(Standard product is always componentwise)
```

をするには `V*W`

```
> W + a # matrix+ scalar is still componentwise
 [,1] [,2] [,3]
[1,] 2 6 10
[2,] 3 7 11
[3,] 4 8 12
[4,] 5 9 13
```

```
> W + c # matrix + vector... : no warnings and R does the operation in a column-wise manner
 [,1] [,2] [,3]
[1,] 3 8 13
[2,] 5 10 12
[3,] 7 9 14
[4,] 6 11 16
```

## "プライベート"

Rのまたはののドットは、またはがされることをすためによくされます。

したがって、のをしてください

```
> foo <- 'foo'
> .foo <- 'bar'
```

そして`ls`を使ってオブジェクトをリストすると、のオブジェクトだけがされます。

```
> ls()
[1] "foo"
```

しかし、に`all.names = TRUE`をすと、`'private'`がされます

```
> ls(all.names = TRUE)
[1] ".foo" "foo"
```

オンラインでをむ <https://riptutorial.com/ja/r/topic/9013/>

## 98: の

スコープをつもなとしは、でします。すべてのとは、スレッドでされるしいにされなければなりません。

## Examples

と

でされたは、そのでのみされないうりします。

```
x <- 1

foo <- function(x) {
 y <- 3
 z <- x + y
 return(z)
}

y
```

エラーオブジェクト 'y'が見つかりません

にされてりてされたはきされますが、でのみされます。

```
foo <- function(x) {
 x <- 2
 y <- 3
 z <- x + y
 return(z)
}

foo(1)
x
```

5

1

よりのにりてられたは、そのでされることなくします。

```
foo <- function() {
 y <- 3
 z <- x + y
 return(z)
}

foo()
```

4

## サブ

されることなくローカルでされたにアクセスするには、でびされたつまり、サブをそのでするがあります。

これはします

```
bar <- function() {
 z <- x + y
 return(z)
}

foo <- function() {
 y <- 3
 z <- bar()
 return(z)
}

foo()
```

barのエラーオブジェクト 'y'が見つかりません

これはします

```
foo <- function() {

 bar <- function() {
 z <- x + y
 return(z)
 }

 y <- 3
 z <- bar()
 return(z)
}

foo()
```

## 4

### グローバルアサインメント

は、`<<-`をしてのからグローバルにりてることができます。 `bar()` は `y` アクセスできるようになりました。

```
bar <- function() {
 z <- x + y
 return(z)
}

foo <- function() {
 y <<- 3
 z <- bar()
 return(z)
}
```

```
}

foo()
```

4

グローバルなりてはけてください。ラッパーをするか、のローカルからをにびすことがにましいです。

とのなりて

Rのは、にびしてをけることができます。は、にりてられ、それらのからびすことができます。

にされるは、 `package:base` または `package:base` のサブをむです。

```
e1 <- new.env(parent = baseenv())
e2 <- new.env(parent = e1)
```

は、にりてられ、それらのからびすことができます。

```
assign("a", 3, envir = e1)
get("a", envir = e1)
get("a", envir = e2)
```

3

3

`e2` は `e1` からするので、`e1` と `e2` で `a` は `3` です。ただし、`e2` に `a` しても `e1` の `a` のはされません。

```
assign("a", 2, envir = e2)
get("a", envir = e2)
get("a", envir = e1)
```

3

2

`on.exit()` は、グローバルをりてるがある、のクリーンアップにです。

のパラメータ、にグラフィックスのパラメータは、グローバルにしかできません。このさなは、よりなプロットをするによくされます。

```
new_plot <- function(...) {

 old_pars <- par(mar = c(5,4,4,2) + .1, mfrow = c(1,1))
 on.exit(par(old_pars))
 plot(...)
}
```

## パッケージとマスキング

なるパッケージのとオブジェクトはじをつことがあります。でみまれたパッケージはのパッケージをマスクし、メッセージがされます。でをびすと、にみまれたパッケージのがされます。のにはにアクセスできます。

```
library(plyr)
library(dplyr)
```

ファイル 'dplyr'

のオブジェクトは 'packageplyr'からマスクされています。

する、える、く、する、IDをする、をする、する、する

のオブジェクトは 'packagestats'からマスクされています

フィルタリング、

のオブジェクトは 'packagebase'からマスクされています

、setdiff、setequal、union

コードをくとき、にしてにをびすためのベストプラクティスである`package::function()`にこのをします。

オンラインでのをむ <https://riptutorial.com/ja/r/topic/3138/>の

## 99: テーブル Excel、SAS、SPSS、Stata の I / O

### Examples

リオでのデータのインポート

くのかなファイルからデータをインポートするなは、[rio](#)です。このパッケージは、くのにされるデータインポートをラップする `import()` をし、それによってインタフェースをします。ファイルや URL を `import()` すだけでにします

```
import("example.csv") # comma-separated values
import("example.tsv") # tab-separated values
import("example.dta") # Stata
import("example.sav") # SPSS
import("example.sas7bdat") # SAS
import("example.xlsx") # Excel
```

`import()` は、ディレクトリ、URL HTTP または HTTPS、およびクリップボードからみることもできます。サポートされているすべてのファイルのなリストは、[rio パッケージの github リポジトリ](#)にあります。

みみしようとしているのファイルにするいくつかのパラメータをし、`import()` ですこともできます。

```
import("example.csv", format = ",") #for csv file where comma is used as separator
import("example.csv", format = ";") #for csv file where semicolon is used as separator
```

### Excel ファイルのインポート

のにまとめられているように、それぞれなるまたはリソースをして Excel ファイルをみむいくつかの R パッケージがあります。

| R パッケージ   |      |
|-----------|------|
| xlsx      | Java |
| XLconnect | Java |
| openxlsx  | C ++ |
| readxl    | C ++ |
| RODBC     | ODBC |
| gdata     | Perl |



JavaやODBCをするパッケージでは、RのバージョンやOSによってのがじるがあるため、システムのをすることがです。たとえば、XLconnectビットをしている、xlsxまたはXLconnectをするにはJava 64ビットもXLconnect。

パッケージでのExcelファイルのみみのをにします。くのパッケージには、じまたはによくたがあります。したがって、package::functionのようにパッケージをするとです。openxlsxパッケージには、openxlsxインストールがです。

## xlsxパッケージをしてExcelのファイルをむ

```
library(xlsx)
```

インポートするには、シートのインデックスまたはがです。

```
xlsx::read.xlsx("Book1.xlsx", sheetIndex=1)

xlsx::read.xlsx("Book1.xlsx", sheetName="Sheet1")
```

## XLconnectパッケージでExcelファイルを読みむ

```
library(XLConnect)
wb <- XLConnect::loadWorkbook("Book1.xlsx")

Either, if Book1.xlsx has a sheet called "Sheet1":
sheet1 <- XLConnect::readWorksheet(wb, "Sheet1")
Or, more generally, just get the first sheet in Book1.xlsx:
sheet1 <- XLConnect::readWorksheet(wb, getSheets(wb)[1])
```

XLConnectは、Book1.xlsxめまれたのExcelセルスタイルをにインポートしBook1.xlsx。これは、ワークブックオブジェクトをし、にされたExcelドキュメントをきすにです。まず、あなたはでのセルのをするがありますBook1.xlsxとして、えば、そしてそれらをmyHeader、myBodyとmyPcts。に、ワークブックをRにロードした

```
Headerstyle <- XLConnect::getCellStyle(wb, "myHeader")
Bodystyle <- XLConnect::getCellStyle(wb, "myBody")
Pctsstyle <- XLConnect::getCellStyle(wb, "myPcts")
```

セルスタイルがRにされるようになりました。データののにセルスタイルをりてするには、をしてスタイルをりてるがあります。

```
Headerrange <- expand.grid(row = 1, col = 1:8)
Bodyrange <- expand.grid(row = 2:6, col = c(1:5, 8))
Pctrange <- expand.grid(row = 2:6, col = c(6, 7))

XLConnect::setCellStyle(wb, sheet = "sheet1", row = Headerrange$row,
 col = Headerrange$col, cellstyle = Headerstyle)
```

```
XLConnect::setCellStyle(wb, sheet = "sheet1", row = Bodyrange$row,
 col = Bodyrange$col, cellstyle = Bodystyle)
XLConnect::setCellStyle(wb, sheet = "sheet1", row = Pctrange$row,
 col = Pctrange$col, cellstyle = Pctsstyle)
```

`XLConnect` はですが、がにくることがあります。はるかにくてなフォーマットオプションが `openxlsx` によってされてい `openxlsx`。

## openxlsx パッケージをして Excel のファイルを む

Excel ファイルはパッケージ `openxlsx` でインポートできます

```
library(openxlsx)

openxlsx::read.xlsx("spreadsheet1.xlsx", colNames=TRUE, rowNames=TRUE)

#colNames: If TRUE, the first row of data will be used as column names.
#rowNames: If TRUE, first column of data will be used as row names.
```

R にみむ `sheet` は、 `sheet` にそのをすることです。

```
openxlsx::read.xlsx("spreadsheet1.xlsx", sheet = 1)
```

またはそのをすることによって

```
openxlsx::read.xlsx("spreadsheet1.xlsx", sheet = "Sheet1")
```

さらに、 `openxlsx` はみりシートのをできます。のをにするには、 `detectDates` を `TRUE` にするがあります。

```
openxlsx::read.xlsx("spreadsheet1.xlsx", sheet = "Sheet1", detectDates= TRUE)
```

## Reading は readxl パッケージをしてファイルを します

Excel ファイルは、 `readxl` パッケージをしてデータフレームとして R インポートできます。

```
library(readxl)
```

`.xls` ファイルと `.xlsx` ファイルのをみることができます。

```
readxl::read_excel("spreadsheet1.xls")
readxl::read_excel("spreadsheet2.xlsx")
```

インポートするシートは、またはでできます。

```
readxl::read_excel("spreadsheet.xls", sheet = 1)
readxl::read_excel("spreadsheet.xls", sheet = "summary")
```

`col_names = TRUE`は、のをとします。

```
readxl::read_excel("spreadsheet.xls", sheet = 1, col_names = TRUE)
```

`col_types`をして、データのをベクトルとしてできます。

```
readxl::read_excel("spreadsheet.xls", sheet = 1, col_names = TRUE,
 col_types = c("text", "date", "numeric", "numeric"))
```

## ReadingはRODBCパッケージでファイルをみます

Excelファイルは、WindowsのAccess Database EngineACEはJETとインターフェイスするODBC Excelドライバをしてみることができます。RODBCパッケージをすると、Rはこのドライバにしてワークブックにいわせることができます。ワークシートは、1のヘッダーを、のタイプののデータですとされています。このは、JET / ACEがインストールされている.dllファイルであり、のオペレーティングシステムではできないため、Windows / PCマシンにされています。

```
library(RODBC)

xlconn <- odbcDriverConnect('Driver={Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)};
 DBQ=C:\\Path\\To\\Workbook.xlsx')

df <- sqlQuery(xlconn, "SELECT * FROM [SheetName$]")
close(xlconn)
```

このアプローチでSQLエンジンとすると、`JOIN`や`UNION`をむデータベーステーブルとにExcelワークシートをできます。はJET / ACE SQLのにいます。データアクセスDMLステートメント、になクエリではなく、ワークブックで`SELECT`をできるのは、DMLステートメントのみです。

```
joindf <- sqlQuery(xlconn, "SELECT t1.*, t2.* FROM [Sheet1$] t1
 INNER JOIN [Sheet2$] t2
 ON t1.[ID] = t2.[ID]")

uniondf <- sqlQuery(xlconn, "SELECT * FROM [Sheet1$]
 UNION
 SELECT * FROM [Sheet2$]")
```

のワークブックをすじODBCチャネルからのワークブックをすることもできます。

```
otherwkbkdf <- sqlQuery(xlconn, "SELECT * FROM
 [Excel 12.0 Xml;HDR=Yes;
 Database=C:\\Path\\To\\Other\\Workbook.xlsx].[Sheet1$];")
```

## Readingはgdataパッケージでファイルをみみます

この

**Stata**、**SPSS**、および**SAS**ファイルのみき

パッケージ`foreign`と`haven`は、**Stata**、**SPSS**、**SAS**、およびソフトウェアのようなさまざまなパッケージからファイルをインポートおよびエクスポートするためにできます。サポートされているデータのファイルをインポートするための`read`があります。

```
loading the packages
library(foreign)
library(haven)
library(readstata13)
library(Hmisc)
```

もなデータのをいくつかします。

```
reading Stata files with `foreign`
read.dta("path\to\your\data")
reading Stata files with `haven`
read_dta("path\to\your\data")
```

`foreign`パッケージは、**Stata 7-12**のバージョンの`stata.dta`ファイルを見ることができます。ページによると、`read.dta`はかれなかれ`read.dta`されており、バージョン13ではみみのためにされません。よりのバージョンの**Stata**では、`readstata13`パッケージまたは`haven`できます。`readstata13`、ファイルは

```
reading recent Stata (13+) files with `readstata13`
read.dta13("path\to\your\data")
```

**SPSS**および**SAS**ファイルでのみみ

```
reading SPSS files with `foreign`
read.spss("path\to\your\data.sav", to.data.frame = TRUE)
reading SPSS files with `haven`
read_spss("path\to\your\data.sav")
read_sav("path\to\your\data.sav")
read_por("path\to\your\data.por")
```

```
reading SAS files with `foreign`
read.ssd("path\to\your\data")
reading SAS files with `haven`
read_sas("path\to\your\data")
reading native SAS files with `Hmisc`
sas.get("path\to\your\data") #requires access to saslib
Reading SA XPORT format (*.XPT) files
sasxport.get("path\to\your\data.xpt") # does not require access to SAS executable
```

SASciiパッケージは、SAS SETインポートコードをけれ、`read.fwf`でできるテキストファイルをするをします。なりリリースデータセットのインポートにしてにであることがされています。サポートは<https://github.com/ajdamico/SAScii>です。

データフレームをのパッケージにエクスポートするには、きみ`write.foreign()`します。これは2つのファイルをします。1つはデータをし、もう1つはデータを見るためになをんでいます。

```
writing to Stata, SPSS or SAS files with `foreign`
write.foreign(dataframe, datafile, codefile,
 package = c("SPSS", "Stata", "SAS"), ...)
write.foreign(dataframe, "path\to\data\file", "path\to\instruction\file", package = "Stata")

writing to Stata files with `foreign`
write.dta(dataframe, "file", version = 7L,
 convert.dates = TRUE, tz = "GMT",
 convert.factors = c("labels", "string", "numeric", "codes"))

writing to Stata files with `haven`
write_dta(dataframe, "path\to\your\data")

writing to Stata files with `readstatal3`
save.dta13(dataframe, file, data.label = NULL, time.stamp = TRUE,
 convert.factors = TRUE, convert.dates = TRUE, tz = "GMT",
 add.rownames = FALSE, compress = FALSE, version = 117,
 convert.underscore = FALSE)

writing to SPSS files with `haven`
write_sav(dataframe, "path\to\your\data")
```

SPSSによってされたファイルは、のように`read.spss`でみることもできます。

```
foreign::read.spss('data.sav', to.data.frame=TRUE, use.value.labels=FALSE,
 use.missings=TRUE, reencode='UTF-8')
to.data.frame if TRUE: return a data frame
use.value.labels if TRUE: convert variables with value labels into R factors with those levels
use.missings if TRUE: information on user-defined missing values will used to set the corresponding values to NA.
reencode character strings will be re-encoded to the current locale. The default, NA, means to do so in a UTF-8 locale, only.
```

フェザーファイルのインポートまたはエクスポート

Featherは、PythonとRののをめ、メタデータクラスなどをしながら、にとらわれないでデータフレームをするようにされたApache Arrowのです。フェザーファイルをむとのdata.frameではなく

```
library(feather)

path <- "filename.feather"
df <- mtcars

write_feather(df, path)

df2 <- read_feather(path)

head(df2)
A tibble: 6 x 11
mpg cyl disp hp drat wt qsec vs am gear carb
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4
2 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4
3 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1
4 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1
5 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2
6 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1

head(df)
mpg cyl disp hp drat wt qsec vs am gear carb
Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4
Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4
Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1
Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1
Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2
Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1
```

のドキュメントには、のがまれています。

ユーザーへのフェザーはアルファソフトウェアとしてうがあります。に、ファイルは  
のにするがいです。のデータにフェザーをしないでください。

オンラインでテーブルExcel、SAS、SPSS、StataのI/Oをむ

<https://riptutorial.com/ja/r/topic/5536/テーブル-excel-sas-spss-stata-のi---o>

# 100:

## Examples

の

Rのは、いわゆるWilkinson-Rogersの<sup>1</sup>をにします。

のためにlmのようなモデルをするときは、 formulaがです。このformulaは、どのをするかをします。

```
my_formula1 <- formula(mpg ~ wt)
class(my_formula1)
gives "formula"

mod1 <- lm(my_formula1, data = mtcars)
coef(mod1)
gives (Intercept) wt
37.285126 -5.344472
```

~ LHSのがされ、RHSにはがまれます。には、のformulaびしはです。なぜなら、チルダはクラスオブジェクトをすだからです。

```
form <- mpg ~ wt
class(form)
#[1] "formula"
```

formulaの~のは、をできることです。

```
form_mt <- formula(mpg ~ wt, env = mtcars)
```

この、は、デフォルトでパラメータとに、 wtがされることをす。インターセプトは、 formula 0または-1をめることによって、1に0にすることができます。

```
coef(lm(mpg ~ 0 + wt, data = mtcars))
coef(lm(mpg ~ wt -1, data = mtcars))
```

aとbとののa、 b a:bをのformulaめa:bできます。

```
coef(lm(mpg ~ wt:vs, data = mtcars))
```

なからは、がないモデルではをたないことがであるため、なアプローチはformulaをa + b + a:bにすることです。これはしますがa\*bくことでできます。\*は、の2つのまたはのまたはが「」のときのをします。

```
coef(lm(mpg ~ wt*vs, data = mtcars))
```

\*をすると、のようなすべてのをむがされます。

```
coef(lm(mpg ~ wt*vs*hp, data = mtcars))
```

にえて、7つのをえる。3インタラクション、3インタラクション、3つのメインエフェクト。

たとえば、3インタラクションをし、すべての2インタラクションをしたいは、2つのがあります。まず、-をして、のをすることができます。

```
coef(lm(mpg ~ wt*vs*hp - wt:vs:hp, data = mtcars))
```

あるいは、たちがとするのレベルをするために^をすることができます

```
coef(lm(mpg ~ (wt + vs + hp) ^ 2, data = mtcars))
```

これらの2つのは、じモデルをするがあります。

に、.なすべてのをなエフェクトとしてすることをしてしています。この、dataは、なLHSにないをするためにされます。したがって

```
coef(lm(mpg ~ ., data = mtcars))
```

と10ののをえる。このは、やにすべてのをするパッケージでにされます。してください.コンテキストにしますのでは?update.formulaをしてください。

1. GNウィルキンソンとCEロジャース。ロイヤル。シリーズC Vol. 22、No.31973、pp.392-399

、およびをする

$y \sim .$ ここに.モデルのフィッティングにされるデータフレームの $y$ をくすべてのとしてされます。これは、のとです。たとえば、 $y \sim \text{var1} + \text{var2} + \text{var3} + \dots + \text{var15}$

$y \sim .^2$ は、データフレームののすべてのとをえます。これは $y \sim \text{var1} + \text{var2} + \dots + \text{var15} + \text{var1}:\text{var2} + \text{var1}:\text{var3} + \text{var1}:\text{var4} \dots \text{and so on}$

$y \sim \text{var1} + \text{var2} + \dots + \text{var15} + \text{I}(\text{var1}^2) + \text{I}(\text{var2}^2) + \text{I}(\text{var3}^2) \dots + \text{I}(\text{var15}^2)$  ここで $\text{I}(\text{var}^2)$  データフレームの1つの2をします。

$y \sim \text{poly}(\text{var1}, \text{degree} = 2) + \text{poly}(\text{var2}, \text{degree} = 2) + \dots + \text{poly}(\text{var15}, \text{degree} = 2)$

または

$y \sim \text{poly}(\text{var1}, \text{var2}, \text{var3}, \dots, \text{var15}, \text{degree} = 2)$ はのとです。

$\text{poly}(\text{var1}, \text{degree} = 2)$ は $\text{var1} + \text{I}(\text{var1}^2)$ とです。



3をるには、`poly()`で`degree = 3`を`degree = 3`ます。

`polyI(var, 2)`にはがです。これはモデルをフィッティングしたで、それぞれなるをしますが、フィッティングされたはじモデルのなるパラメータをすためです。することをめします`I(var, 2)`のに`poly()`でられるをけるために`poly()`。

すると、、、のをるためには、のようがあります。

```
y ~ .^2 + I(var1^2) + I(var2^2)+...I(var15^2)
```

#### 4つのデモ

```
old <- reformulate('y ~ x1+x2+x3+x4')
new <- reformulate(" y ~ .^2 + I(x1^2) + I(x2^2) + I(x3^2) + I(x4^2) ")
tmp <- .Call(stats::C_updateform, old, new)
terms.formula(tmp, simplify = TRUE)

~y ~ x1 + x2 + x3 + x4 + I(x1^2) + I(x2^2) + I(x3^2) + I(x4^2) +
x1:x2 + x1:x3 + x1:x4 + x2:x3 + x2:x4 + x3:x4
attr("variables")
list(~y, x1, x2, x3, x4, I(x1^2), I(x2^2), I(x3^2), I(x4^2))
attr("factors")
x1 x2 x3 x4 I(x1^2) I(x2^2) I(x3^2) I(x4^2) x1:x2 x1:x3 x1:x4 x2:x3 x2:x4 x3:x4
~y 0 0 0 0 0 0 0 0 0 0 0 0 0 0
x1 1 0 0 0 0 0 0 0 1 1 1 0 0 0
x2 0 1 0 0 0 0 0 0 1 0 0 1 1 0
x3 0 0 1 0 0 0 0 0 0 1 0 1 0 1
x4 0 0 0 1 0 0 0 0 0 0 1 0 1 1
I(x1^2) 0 0 0 0 1 0 0 0 0 0 0 0 0 0
I(x2^2) 0 0 0 0 0 1 0 0 0 0 0 0 0 0
I(x3^2) 0 0 0 0 0 0 1 0 0 0 0 0 0 0
I(x4^2) 0 0 0 0 0 0 0 1 0 0 0 0 0 0
attr("term.labels")
[1] "x1" "x2" "x3" "x4" "I(x1^2)" "I(x2^2)" "I(x3^2)" "I(x4^2)"
[9] "x1:x2" "x1:x3" "x1:x4" "x2:x3" "x2:x4" "x3:x4"
attr("order")
[1] 1 1 1 1 1 1 1 1 2 2 2 2 2 2
attr("intercept")
[1] 1
attr("response")
[1] 1
attr("Environment")
<environment: R_GlobalEnv>
```

オンラインでをむ <https://riptutorial.com/ja/r/topic/1061/>

## 101: +

`parse`、テキストとファイルをにします。

`eval`はをします。

## Examples

でコードをする

このでは、でされたコードをします。

```
the string
str <- "1+1"

A string is not an expression.
is.expression(str)
[1] FALSE

eval(str)
[1] "1+1"

parse convert string into expressions
parsed.str <- parse(text="1+1")

is.expression(parsed.str)
[1] TRUE

eval(parsed.str)
[1] 2
```

オンラインで+をむ <https://riptutorial.com/ja/r/topic/5746/-plus>

## 102:

き

は、にまたは`as.numeric`、`as.data.frame`などのためのをして、にオブジェクトのがされたときにRでします。

## Examples

の

はRのデータでわれますが、くの、にわれるため、データはすべてののにできます。えば、

```
x = 1:3
x
[1] 1 2 3
typeof(x)
#[1] "integer"

x[2] = "hi"
x
#[1] "1" "hi" "3"
typeof(x)
#[1] "character"
```

まず、`x`は`integer`で`integer`ことにしてください。しかし、`x[2] = "hi"`とすると、Rのベクトルがのデータしかできないので、`x`すべてのが`character`にされました。

オンラインでをむ <https://riptutorial.com/ja/r/topic/9793/>

## 103: をする

セットには、々のコピーが1つだけまれています。のプログラミングとはなり、ベースRにはセットのデータがありません。そのわりに、Rは、そののだけをとって、のようなベクトルをいます。これは、`setdiff`、`intersect`、`union`、`setequal`、および`%in%`ます。ため`v %in% ss`ないベクトルが、セットとしてわれる`v`。

Rののセットデータの、Rcppパッケージにはいくつかのオプションがあります。

## Examples

ペアのをする

## セットの

Rでは、ベクトルにがまれることがあります。

```
v = "A"
w = c("A", "A")
```

ただし、セットにはのコピーが1つしかありません。Rはそののだけをとってのようなベクトルをうので、の2つのベクトルはじものとみなされます

```
setequal(v, w)
TRUE
```

## セットのみわせ

なはなをとっています

```
x = c(1, 2, 3)
y = c(2, 4)

union(x, y)
1 2 3 4

intersect(x, y)
2

setdiff(x, y)
1 3
```

これらはすべてじページの`?union`されています。

## ベクトルのメンバーシップをする

`%in%`は、ベクトルとセットをします。

```
v = "A"
w = c("A", "A")

w %in% v
TRUE TRUE

v %in% w
TRUE
```

のはにわれ、のベクトルにけられたセットのメンバーシップそのすべてののでされているについてテストされます。

テストとはなり、`%in%`にTRUEまたはFALSEしFALSE。

```
c(1, NA) %in% c(1, 2, 3, 4)
TRUE FALSE
```

ドキュメントは?`%in%`ます。

## ベクトルのデカルトまたは「クロス」

x、yというのベクトルをすべてつけるには、xをベクトルXからきし、yをYから`expand.grid`には`expand.grid`をい`expand.grid`。

```
X = c(1, 1, 2)
Y = c(4, 5)

expand.grid(X, Y)

Var1 Var2
1 1 4
2 1 4
3 2 4
4 1 5
5 1 5
6 2 5
```

はされたベクトルの1つのをつ`data.frame`です。くの、ベクトルの「グリッド」をするのではなく、セットのデカルトをることがです。たちは`unique` `lapply`と`do.call`うことができます

```
m = do.call(expand.grid, lapply(list(X, Y), unique))

Var1 Var2
1 1 4
2 2 4
3 1 5
4 2 5
```

# みわせにをする

としてられるみわせ $f(x,y)$ にをするは、のとしてすることができます。

```
m$p = with(m, Var1*Var2)
Var1 Var2 p
1 1 4 4
2 2 4 8
3 1 5 5
4 2 5 10
```

々がとして、このアプローチは、くのベクトルのためにしますが、2のなには、々ですることができるマトリックスにをって、よりいフィットです`outer`

```
uX = unique(X)
uY = unique(Y)

outer(setNames(uX, uX), setNames(uY, uY), `*`)

4 5
1 4 5
2 8 10
```

するとツールについては、コンビナトリアルのトピックをしてください。

ユニーク/ドロップのをする/ベクトルからのをする

のがであるように`unique`がするようにします1だけされます。

```
x = c(2, 1, 1, 2, 1)

unique(x)
2 1
```

は、にしたでされます。

`duplicated`タグには、

```
duplicated(x)
FALSE FALSE TRUE TRUE TRUE
```

`anyDuplicated(x) > 0`は、ベクトルにがまれているかどうかをくするです。

ベクトルのなり/ベンの

2つのセットののをカウントするには、カスタムをくことができます

```
xtab_set <- function(A, B){
 both <- union(A, B)
```

```
inA <- both %in% A
inB <- both %in% B
return(table(inA, inB))
}

A = 1:20
B = 10:30

xtab_set(A, B)

inB
inA FALSE TRUE
FALSE 0 10
TRUE 9 11
```

さまざまなパッケージによってされるVennダイアグラムをして、のセットにわたるオーバーラップをすることができます。

オンラインでをするをむ <https://riptutorial.com/ja/r/topic/1383/>をする

## 104: クラスとモード

### Examples

はとをし、のベクトルにりてられるデフォルトモードです。 `is.numeric()` は、ベクトルがかどうかをします。と `is.numeric()` をしますが、 `as.numeric()` はに **double** にしようとします。

```
x <- 12.3
y <- 12L

#confirm types
typeof(x)
[1] "double"
typeof(y)
[1] "integer"

confirm both numeric
is.numeric(x)
[1] TRUE
is.numeric(y)
[1] TRUE

logical to numeric
as.numeric(TRUE)
[1] 1

While TRUE == 1, it is a double and not an integer
is.integer(as.numeric(TRUE))
[1] FALSE
```

ダブルはRのデフォルトのです。これらはベクタです。つまり、ベクタのにして8バイトのメモリをします。 Rにはデータがないため、すべてののはでされます。

```
is.double(1)
TRUE
is.double(1.0)
TRUE
is.double(1L)
FALSE
```

は、コンポーネントなしでできます。は、にLがいたでされます。それがダブルとみなされたにLのないの。

```
typeof(1)
[1] "double"
class(1)
[1] "numeric"
typeof(1L)
[1] "integer"
class(1L)
[1] "integer"
```



ほとんどの、またはをしてもはありませんが、をにきえと、メモリとがなくなることがあります。ダブルベクトルはあたり8バイトをし、ベクトルはあたり4バイトのみをします。ベクトルのサイズが大きくなるにつれて、なをするとプロセスがにスピードアップするがあります。

```
test speed on lots of arithmetic
microbenchmark(
 for(i in 1:100000){
 2L * i
 10L + i
 },

 for(i in 1:100000){
 2.0 * i
 10.0 + i
 }
)
Unit: milliseconds
```

|                      |        |           | expr | min      | lq       | mean     | median   | uq       |
|----------------------|--------|-----------|------|----------|----------|----------|----------|----------|
| max neval            |        |           |      |          |          |          |          |          |
| for (i in 1:1e+05) { | 2L * i | 10L + i } |      | 40.74775 | 42.34747 | 50.70543 | 42.99120 | 65.46864 |
| 94.11804             | 100    |           |      |          |          |          |          |          |
| for (i in 1:1e+05) { | 2 * i  | 10 + i }  |      | 41.07807 | 42.38358 | 53.52588 | 44.26364 | 65.84971 |
| 83.00456             | 100    |           |      |          |          |          |          |          |

オンラインでクラスとモードをむ <https://riptutorial.com/ja/r/topic/9018/クラスとモード>

# 105:

## Examples

### tbl\_dfの

tbl\_df *tibble diff*とするは、なパッケージでよくされるデータフレームのバリエーションです。それはチブルパッケージにされています。

as\_data\_frameをして、データフレームをtbl\_dfにします。

```
library(tibble)
mtcars_tbl <- as_data_frame(mtcars)
```

data.framesとtbl\_dfsのもないの1つは、printというものです。

```
A tibble: 32 x 11
 mpg cyl disp hp drat wt qsec vs am gear carb
* <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4
2 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4
3 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1
4 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1
5 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2
6 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1
7 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4
8 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2
9 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2
10 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4
... with 22 more rows
```

- されたには、の 32 x 11
- これには、のタイプ dbl
- これは、られたのをします。 このoptions(tibble.print\_max = [number])をするにはoptions(tibble.print\_max = [number])。

dplyrパッケージのくのは、group\_by()などのtbl\_dfsでにします。

### tidyverse

tidyverse とは tidyverse ですか

tidyverseは、なRをHadley / Rstudioによってされたされたツールにえる、くエレガントなです。tidyverseにまれるすべてのパッケージのは、tidyverse ツールのにいます。しかし、まず、にらのをさせてください

tidyverseは、のデータとAPIをしているため、してするのパッケージです。 tidyverse パッケージは、1つのコマンドでにコアパッケージをインストールしてロードできるようにされています。

びったりののすべてのパッケージについてび、それらがどのようにフィットするかについてののは、 R for Data Scienceです。されたパッケージのウェブサイトをし、をにし、にデータにするのためののをするように、ケのになことについてもっとくことをしてください。

ソース

---

## どうやってうのですか

のRパッケージだけで、パッケージをインストールしてみむがあります。

```
install.package("tidyverse")
library("tidyverse")
```

いは、1つのコマンドでのパッケージがインストール/ロードされていることです。ボーナスとして、インストールされた/ロードされたすべてのパッケージがのあるバージョンであることをすることができます。

---

## これらのパッケージはですか

よくられてくわれているパッケージ

- [ggplot2](#) なデータの[SO\\_doc](#)
- [dplyr](#) い [Rcpp](#)データとしたアプローチ[SO\\_doc](#)
- [tidyr](#) データけのためのツール[SO\\_doc](#)
- [readr](#) データのインポート。
- [purrr](#) JSパッケージ[underscore.js](#)、[lodash](#)、[lazy.js](#)のスタイルで、ののなをえたRのプログラミングツールをさせることで、なをさせます。
- [tibble](#) データフレームのな。
- [magrittr](#) コードをみやすくするための[SO\\_doc](#)

のデータをするためのパッケージ

- [hms](#) にむ
- [stringr](#) をにできるようにされたしたをする
- [lubridate](#) な/[SO\\_doc](#)
- [forcats](#) によるな。

データのインポート

- [DBI](#) RとデータベースシステムDBMSのインタフェースをします。
- にSPSS、SASとStataのファイルのインポート [SO\\_doc](#)を
- [httr](#) [httr](#)のは、のWeb APIのにわせてカスタマイズされたカールパッケージのラッパーをすることです
- [jsonlite](#) データとWebにされたJSONパーサとジェネレータ
- パッケージ [SO\\_doc](#)をとせずに [readxl](#) read.xlsおよび.xlsxファイル
- [rvest](#) rvestはウェブページ [SO\\_doc](#)からをりすのにちます
- [xml2](#) XML

## モデリング

- [modelr](#) モデリングにエレガントなパイプラインをするのにつをします
- [ほうき](#) モデルをなデータにに

に、 [tidyverse](#)はをすることをします

- [knitr](#) くべき [リテラル](#)プログラミングエンジンです。なAPIをすることで、コードをせずにをにできます。 [SO\\_docs 1つ](#)、 [2つ](#)
- [rmarkdown](#) のあるプログラミングのためのRstudioのパッケージ。 [SO\\_docs 1つ](#)、 [2つ](#)、 [3つ](#)、 [4つ](#)

オンラインでをむ <https://riptutorial.com/ja/r/topic/1395/>

## 106: をってする

き

tidyrは、データをするための2つのツールがあります `gather` ワイドロングと `spread` ワイドまでのさ。

のオプションについては、 [データのを](#) してください。

## Examples

`spread` をしていからいへの

```
library(tidyr)

example data
set.seed(123)
df <- data.frame(
 name = rep(c("firstName", "secondName"), each=4),
 numbers = rep(1:4, 2),
 value = rnorm(8)
)
df
name numbers value
1 firstName 1 -0.56047565
2 firstName 2 -0.23017749
3 firstName 3 1.55870831
4 firstName 4 0.07050839
5 secondName 1 0.12928774
6 secondName 2 1.71506499
7 secondName 3 0.46091621
8 secondName 4 -1.26506123
```

「」を々のに「」することができます。

```
spread(data = df,
 key = numbers,
 value = value)
name 1 2 3 4
1 firstName -0.5604756 -0.2301775 1.5587083 0.07050839
2 secondName 0.1292877 1.7150650 0.4609162 -1.26506123
```

または、「」を々のにげます。

```
spread(data = df,
 key = name,
 value = value)
numbers firstName secondName
1 1 -0.56047565 0.1292877
2 2 -0.23017749 1.7150650
```

```
3 3 1.55870831 0.4609162
4 4 0.07050839 -1.2650612
```

**gather**をして、ワイドフォーマットからロングフォーマットへの

```
library(tidyr)

example data
df <- read.table(text = " numbers firstName secondName
1 1 1.5862639 0.4087477
2 2 0.1499581 0.9963923
3 3 0.4117353 0.3740009
4 4 -0.4926862 0.4437916", header = T)
df
numbers firstName secondName
1 1 1.5862639 0.4087477
2 2 0.1499581 0.9963923
3 3 0.4117353 0.3740009
4 4 -0.4926862 0.4437916
```

キーとして 'numbers' をしてをめることができます。

```
gather(data = df,
 key = numbers,
 value = myValue)
numbers numbers myValue
1 1 firstName 1.5862639
2 2 firstName 0.1499581
3 3 firstName 0.4117353
4 4 firstName -0.4926862
5 1 secondName 0.4087477
6 2 secondName 0.9963923
7 3 secondName 0.3740009
8 4 secondName 0.4437916
```

オンラインでをってするをむ <https://riptutorial.com/ja/r/topic/9195/>をってする

---

## 107: クラス

き

はのが'ベクトル'とぶものです。

---

## トピック

パターン

- 
- [パターンマッチングと](#)
- [strsplit](#)
- [のみき](#)

## Examples

がかどうかをチェックするには、`is.character()`をします。をにするには、`as.character()`をします。

```
x <- "The quick brown fox jumps over the lazy dog"
class(x)
[1] "character"
is.character(x)
[1] TRUE
```

はにすることができますが、をにしようとすると`NA`ます。

```
as.numeric("2")
[1] 2
as.numeric("fox")
[1] NA
Warning message:
NAs introduced by coercion
```

オンラインでクラスをむ <https://riptutorial.com/ja/r/topic/9017/クラス>

## 108: のみき

するドキュメント

- [ユーザーをする](#)

## Examples

のと

Rには、のやにできるみみがいくつかありますが、`print`と`cat`がもです。Rはインタープリターなので、Rコンソールですことができます

```
print("Hello World")
#[1] "Hello World"
cat("Hello World\n")
#Hello World
```

2つののとにしてください。は、のではありません`x`でした`x <- "Hello World"`。らはによってされ`print`に。

`cat`は、1つのベクトルをとしてとり、それらをコンソールにします。ベクトルのさが1よりきい、はスペースでられますデフォルト。

```
cat(c("hello", "world", "\n"))
#hello world
```

`\n`がなければ、はのようになります。

```
cat("Hello World")
#Hello World>
```

のコマンドのプロンプトがのにされます。RStudioなどのコンソールによっては、でわらないにをすることがあります。

`print`はされたののクラスがされ、クラスのメソッドがにされることをする「」のです。`"Hello World"`ようなベクトルの、は`cat`のにています。ただし、がされ、ベクトルのこののはのみをす`[1]`がされます。

```
print("Hello World")
#[1] "Hello World"
```

このデフォルトのは、にRにのをするときにはされるものです。`s`が`print(s)`または`print("Hello World")`をびすのと同じであることにしてください



```
s <- "Hello World"
s
#[1] "Hello World"
```

あるいはにもりてなくとも

```
"Hello World"
#[1] "Hello World"
```

々はして、ベクトルの2のとして、のをするは `c()` oncatenate **C**に、そのの `print()` のそれとはかなりなっている `cat`

```
print(c("Hello World", "Here I am. "))
#[1] "Hello World" "Here I am."
```

`c()` はをわないことにしてください。 そののために `paste` をするがあります。 Rは、ベクトルに2つのが々にされていることをしています。 のにまたがるベクトルがあれば、ののに `[1]` をするのにと、Rはをするのインデックスをします。

```
c("Hello World", "Here I am!", "This next string is really long.")
#[1] "Hello World" "Here I am!"
#[3] "This next string is really long."
```

`print` ののは、にされるオブジェクトのクラスによってなります。

「」や「」などのなるクラスのオブジェクトを `print` とびすと、クラスではないオブジェクトをとっていることをすために、からがされます。

```
print(1)
#[1] 1
print(TRUE)
#[1] TRUE
```

Factorオブジェクトは、コンソールをしてSOにオブジェクトをすると、しばしばあいまいさをじるとじでされます。 インタラクティブなをいて `cat` や `print` をすることはまれです。 に `print()` びすことは、コンソールでの `foo` が `print(foo)` ショートカットであるため、にですのをにするか、によって `invisible` としてされるオブジェクトをしないうり `print(foo)` 。 Rのコンソールは、REPL read-eval-print-loopとしてられています。 `cat` はなのためにするのがですオープンファイルへののきみなど。ときどきでされることがありますが `print()` へのびしはされます、で `cat()` をしてコンソールへののをするのはいいです。されるは、メッセージのは `message()` または `warning()` です。それらは `cat` とにしますが、オプションでエンドユーザーがすることができます。は、ユーザーがにじてするためになりてすることができます。

```
message("hello world")
#hello world
suppressMessages(message("hello world"))
```

## ファイルのみき

ずしもローカルシステムのパスからみきするはありません。えば、R code streaming map-reduce がファイルのみきするがあるなどです。ローカルシステムをえ、クラウドやきなデータのをまえたのシナリオもあります、これはますますになっています。これをうの1つは、なです。

`file()` コマンドでみむファイルをしします "r"はみみモードです

```
conn <- file("/path/example.data", "r") #when file is in local system
conn1 <- file("stdin", "r") #when just standard input/output for files are available
```

これによりファイルのみがされるため、のようにこれらのファイルからデータをみることができます。

```
line <- readLines(conn, n=1, warn=FALSE)
```

ここでは、ファイル `conn` からごとに `n=1` としてデータをみみます。みし1で10または20のブロックみしのためにデータブロックをみるための `n` たたとえば10,20などのをすることができます。1つのなファイルをむには、 `n=-1` します。

データ、またはモデル。ファイルにきむことができる `writeLines()`, `cat()` などのくのならるコマンドをしてをファイルにきすことができます。ただし、これらのコマンドはすべて、きみにされたファイルをしします。これは、 `file()` コマンドをのようにして `file()` できます。

```
conn2 <- file("/path/result.data", "w") #when file is in local system
conn3 <- file("stdout", "w") #when just standard input/output for files are available
```

に、のようにデータをきみます。

```
writeLines("text", conn2, sep = "\n")
```

オペレーティングシステムコマンドのをキャプチャする

## ベクトルをす

ベース<sub>R</sub>は、システムコマンドをびすための2つのがあります。とも、システムコマンドのをキャプチャするためにのパラメータがです。

```
system("top -a -b -n 1", intern = TRUE)
system2("top", "-a -b -n 1", stdout = TRUE)
```

どちらもベクトルをしします。

```
[1] "top - 08:52:03 up 70 days, 15:09, 0 users, load average: 0.00, 0.00, 0.00"
```

```
[2] "Tasks: 125 total, 1 running, 124 sleeping, 0 stopped, 0 zombie"
[3] "Cpu(s): 0.9%us, 0.3%sy, 0.0%ni, 98.7%id, 0.1%wa, 0.0%hi, 0.0%si, 0.0%st"
[4] "Mem: 12194312k total, 3613292k used, 8581020k free, 216940k buffers"
[5] "Swap: 12582908k total, 2334156k used, 10248752k free, 1682340k cached"
[6] ""
[7] " PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND "
```

|      | PID   | USER  | PR | NI | VIRT  | RES  | SHR  | S | %CPU | %MEM | TIME+     | COMMAND |
|------|-------|-------|----|----|-------|------|------|---|------|------|-----------|---------|
| [8]  | 11300 | root  | 20 | 0  | 1278m | 375m | 3696 | S | 0.0  | 3.2  | 124:40.92 | trala   |
| [9]  | 6093  | user1 | 20 | 0  | 1817m | 269m | 1888 | S | 0.0  | 2.3  | 12:17.96  | R       |
| [10] | 4949  | user2 | 20 | 0  | 1917m | 214m | 1888 | S | 0.0  | 1.8  | 11:16.73  | R       |

として、UNIXコマンド `top -a -b -n 1` がされています。これはOSのものであり、コンピュータでサンプルをするためにするがあるがあります。

パッケージ `devtools` は、システムコマンドをし、パラメータなしでをキャプチャするがあります。また、ベクトルをします。

```
devtools::system_output("top", "-a -b -n 1")
```

## データフレームをす

`data.table` パッケージの `fread` は、シェルコマンドをし、`read.table` ようなをみることをにします。それはし `data.table` または `data.frame` 。

```
fread("top -a -b -n 1", check.names = TRUE)
```

|    | PID   | USER  | PR | NI | VIRT  | RES  | SHR  | S | X.CPU | X.MEM | TIME.     | COMMAND |
|----|-------|-------|----|----|-------|------|------|---|-------|-------|-----------|---------|
| 1: | 11300 | root  | 20 | 0  | 1278m | 375m | 3696 | S | 0     | 3.2   | 124:40.92 | trala   |
| 2: | 6093  | user1 | 20 | 0  | 1817m | 269m | 1888 | S | 0     | 2.3   | 12:18.56  | R       |
| 3: | 4949  | user2 | 20 | 0  | 1917m | 214m | 1888 | S | 0     | 1.8   | 11:17.33  | R       |
| 4: | 7922  | user3 | 20 | 0  | 3094m | 131m | 1892 | S | 0     | 1.1   | 21:04.95  | R       |

その `fread` に6のヘッダをスキップしています。

ここでは、`%CPU`、`%MEN`、および `TIME+` をににするために、`check.names = TRUE` パラメータがされました。

オンラインでのみきをむ <https://riptutorial.com/ja/r/topic/5541/> のみき

## 109:

き

Rには、`POSIXct`のクラスがしています。`?Dates`、`?DateTimeClasses`、`?difftime`し、さらにのためのこれらのドキュメントの「」セクションにしてください。するドキュメントと [-クラス](#)。

## クラス

- [POSIXct](#)

date-timeクラスPOSIXctは、1970-01-01 00:00:00 UTC1970-01-01 00:00:00 UTC UNIXエポックのをでし1970-01-01 00:00:00 UTC。これは、`Sys.time()`を`Sys.time()`するときにされるです。

- [POSIXlt](#)

クラス、、、などのリストをします。これは`strptime`によってされるです。

- [Date](#)のクラスで、をとします。

## フォーマットの

UNIXのtidyverseとでは、POSIXctがのす。これは、POSIXltよりもく、メモリをしません。

```
origin = as.POSIXct("1970-01-01 00:00:00", format = "%Y-%m-%d %H:%M:%S", tz = "UTC")

origin
[1] "1970-01-01 UTC"

origin + 47
[1] "1970-01-01 00:00:47 UTC"

as.numeric(origin) # At epoch
0

as.numeric(Sys.time()) # Right now (output as of July 21, 2016 at 11:47:37 EDT)
1469116057

posixlt = as.POSIXlt(Sys.time(), format = "%Y-%m-%d %H:%M:%S", tz = "America/Chicago")

Conversion to POSIXct
posixct = as.POSIXct(posixlt)
posixct

Accessing components
posixlt$sec # Seconds 0-61
posixlt$min # Minutes 0-59
posixlt$hour # Hour 0-23
```

```

posixlt$yday # Day of the Month 1-31
posixlt$mon # Months after the first of the year 0-11
posixlt$year # Years since 1900.

ct = as.POSIXct("2015-05-25")
lt = as.POSIXlt("2015-05-25")

object.size(ct)
520 bytes
object.size(lt)
1816 bytes

```

## パッケージ

- どんなときも
- data.table IDateとITime
- ファストタイム
- 
- ナノタイム

## Examples

のと

Rはの、タイムゾーンにアクセスできます

```

Sys.Date() # Returns date as a Date object

[1] "2016-07-21"

Sys.time() # Returns date & time at current locale as a POSIXct object

[1] "2016-07-21 10:04:39 CDT"

as.numeric(Sys.time()) # Seconds from UNIX Epoch (1970-01-01 00:00:00 UTC)

[1] 1469113479

Sys.timezone() # Time zone at current location

[1] "Australia/Melbourne"

```

OlsonNames() をして、のシステムのOlson / IANAデータベースのタイムゾーンをします。

```

str(OlsonNames())
chr [1:589] "Africa/Abidjan" "Africa/Accra" "Africa/Addis_Ababa" "Africa/Algiers"
"Africa/Asmara" "Africa/Asmera" "Africa/Bamako" ...

```

のわりにく

ののにしたいとしましょう。このがちます

```
eom <- function(x, p=as.POSIXlt(x)) as.Date(modifyList(p, list(mon=p$mon + 1, mday=0)))
```

テスト

```
x <- seq(as.POSIXct("2000-12-10"), as.POSIXct("2001-05-10"), by="months")
> data.frame(before=x, after=eom(x))
 before after
1 2000-12-10 2000-12-31
2 2001-01-10 2001-01-31
3 2001-02-10 2001-02-28
4 2001-03-10 2001-03-31
5 2001-04-10 2001-04-30
6 2001-05-10 2001-05-31
>
```

をでする

```
> eom('2000-01-01')
[1] "2000-01-31"
```

ののへ

あるののにきたいとしましょう

```
date <- as.Date("2017-01-20")
> as.POSIXlt(cut(date, "month"))
[1] "2017-01-01 EST"
```

ケもカもしてをする

々は、したにしたいとしましょう`num`ケのを。 `mondate`パッケージをするのをできます。

```
moveNumOfMonths <- function(date, num) {
 as.Date(mondate(date) + num)
}
```

がののをするは、ののをしてし、をします。

えば

1ケまで

```
> moveNumOfMonths("2017-10-30", -1)
[1] "2017-09-30"
```

る2ケ

```
> moveNumOfMonths("2017-10-30",-2)
[1] "2017-08-30"
```

2ケに

```
> moveNumOfMonths("2017-02-28", 2)
[1] "2017-04-30"
```

2のから2ケ、つまり4のにします。

なのである、それがおよびのためにどのようにするかを試みましょう

```
> moveNumOfMonths("2016-11-30", 2)
[1] "2017-01-31"
> moveNumOfMonths("2017-01-31", -2)
[1] "2016-11-30"
```

11は30なので、でじがられますが、

```
> moveNumOfMonths("2017-01-30", -2)
[1] "2016-11-30"
> moveNumOfMonths("2016-11-30", 2)
[1] "2017-01-31"
```

1は31なので、11から2ケすると1のになります。

オンラインでをむ <https://riptutorial.com/ja/r/topic/1157/>

---

## 110: クラスPOSIXctおよびPOSIXlt

き

Rには、POSIXctとPOSIXltという2つのクラスがあり`?DateTimeClasses`。「`?DateTimeClasses`」をしてください。

---

とし

POSIXctでは、はとのみがされますが、フルタイムはまだされています。

---

トピック

- 

---

パッケージ

- 

## Examples

オブジェクトのと

```
test date-time object
options(digits.secs = 3)
d = as.POSIXct("2016-08-30 14:18:30.58", tz = "UTC")

format(d,"%S") # 00-61 Second as integer
[1] "30"

format(d,"%OS") # 00-60.99... Second as fractional
[1] "30.579"

format(d,"%M") # 00-59 Minute
[1] "18"

format(d,"%H") # 00-23 Hours
[1] "14"

format(d,"%I") # 01-12 Hours
[1] "02"

format(d,"%p") # AM/PM Indicator
[1] "PM"
```



```
format(d,"%z") # Signed offset
[1] "+0000"

format(d,"%Z") # Time Zone Abbreviation
[1] "UTC"
```

フォーマットのやのフォーマットについては`?strptime`をしてください。

をオブジェクトにする

を`POSIXct`と`POSIXlt`にするは、のパラメータをり、のをしますが、そののにはいがあります。「」をしてください。

```
as.POSIXct("11:38", # time string
 format = "%H:%M") # formatting string
[1] "2016-07-21 11:38:00 CDT"
strptime("11:38", # identical, but makes a POSIXlt object
 format = "%H:%M")
[1] "2016-07-21 11:38:00 CDT"

as.POSIXct("11 AM",
 format = "%I %p")
[1] "2016-07-21 11:00:00 CDT"
```

とタイムゾーンはされることにしてください。

```
as.POSIXct("11:38:22", # time string without timezone
 format = "%H:%M:%S",
 tz = "America/New_York") # set time zone
[1] "2016-07-21 11:38:22 EDT"

as.POSIXct("2016-07-21 00:00:00",
 format = "%F %T") # shortcut tokens for "%Y-%m-%d" and "%H:%M:%S"
```

ここでののは`?strptime`をしてください。

## ノート

している

- がされていないは、のからのがされます。
- timeがされていないは、から0がされます。
- stringまたは`tz`パラメータのいずれかにタイムゾーンがされていないは、ローカルタイムゾーンがされます。
- けられる`tz`はによってなります。
  - CSTは"CST6CDT"または"America/Chicago"えられます"America/Chicago"

- サポートされているタイムゾーンでは、
  - `R OlsonNames()`
  - あるいは、`R system("cat $R_HOME/share/zoneinfo/zone.tab")` でしてください。
- これらは、[IANA Internet Assigned Numbers Authority](#)
  - [tzデータベースのタイムゾーンのリスト Wikipedia](#)
  - [IANA TZデータ 2016e](#)

を/するには、をでするPOSIXctをします

```
adding/subtracting times - 60 seconds
as.POSIXct("2016-01-01") + 60
[1] "2016-01-01 00:01:00 AEDT"

adding 3 hours, 14 minutes, 15 seconds
as.POSIXct("2016-01-01") + ((3 * 60 * 60) + (14 * 60) + 15)
[1] "2016-01-01 03:14:15 AEDT"
```

よりには、`as.diffftime`をして、またはdatetimeオブジェクトにするをできます。例えば

```
as.POSIXct("2016-01-01") +
 as.diffftime(3, units="hours") +
 as.diffftime(14, units="mins") +
 as.diffftime(15, units="secs")
[1] "2016-01-01 03:14:15 AEDT"
```

/のいをつけるには、`diffftime()`を、、、またはのにします。

```
using POSIXct objects
diffftime(
 as.POSIXct("2016-01-01 12:00:00"),
 as.POSIXct("2016-01-01 11:59:59"),
 unit = "secs")
Time difference of 1 secs
```

`seq.POSIXt()` またはに`seq`して、/のシーケンスをします。

オンラインでクラスPOSIXctおよびPOSIXltをむ <https://riptutorial.com/ja/r/topic/9027/クラス-posixctおよびposixlt->

## 111: と

とは、`glm()` やのなパッケージなど、`stats` パッケージのなでできます。の [CRANタスクビュー](#) には、なパッケージのなリストと、トピックごとのながされています。

## Examples

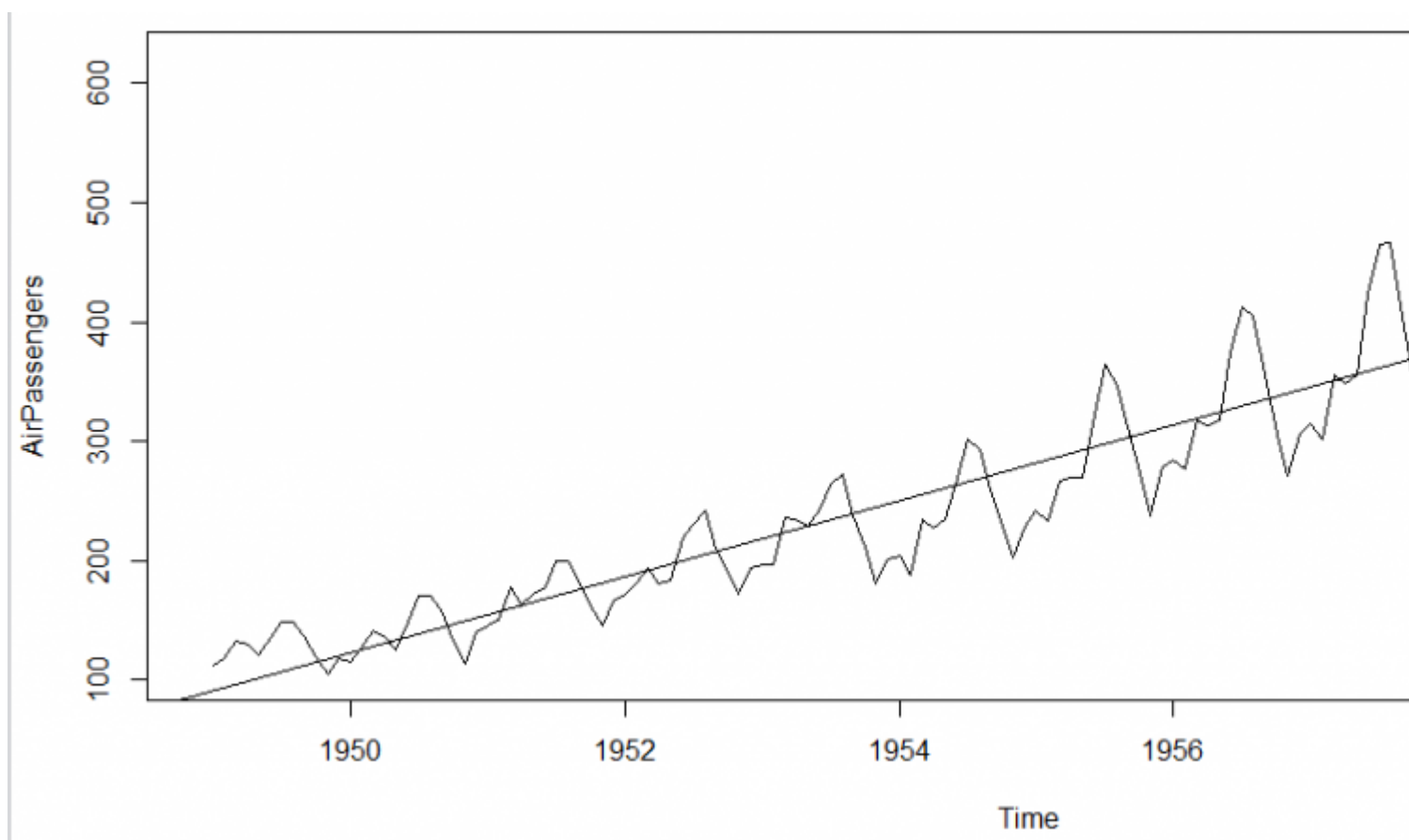
データをいたデータ

```
data(AirPassengers)
class(AirPassengers)
```

1 "ts"

Exploratory Data AnalysisEDAのでは、のステップとして、データのプロットをることができます。

```
plot(AirPassengers) # plot the raw data
abline(reg=lm(AirPassengers~time(AirPassengers))) # fit a trend line
```

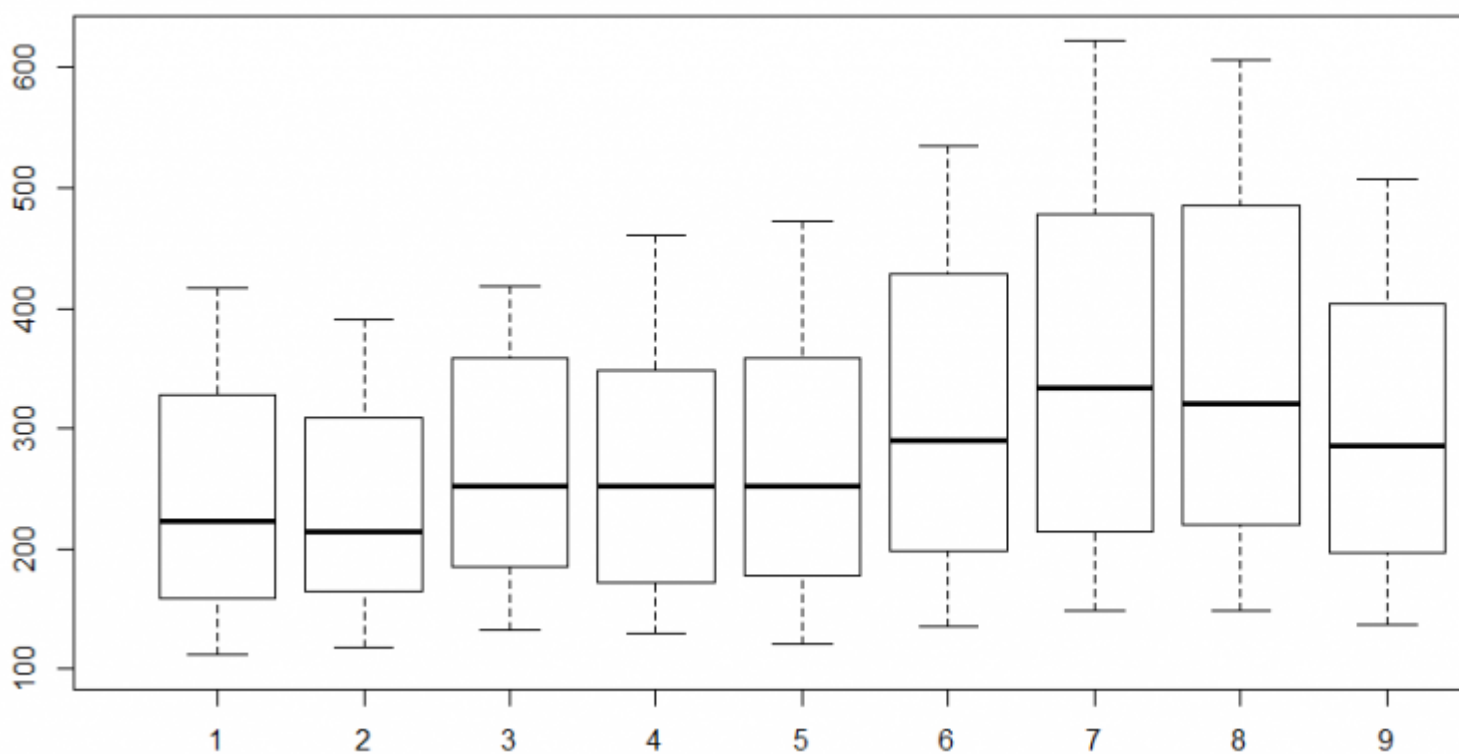


さらなるEDAのために、々はにわたるサイクルをべる

```
cycle(AirPassengers)
```

|      | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1949 | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 1950 | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 1951 | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 1952 | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 1953 | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 1954 | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 1955 | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 1956 | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 1957 | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 1958 | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 1959 | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 1960 | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  |

```
boxplot(AirPassengers~cycle(AirPassengers)) #Box plot across months to explore seasonal effects
```



**ts**オブジェクトの

データは`ts`オブジェクトとしてできます。`ts`オブジェクトには、ARIMAでされるなにするがまれています。また、`window`コマンドをして、でシリーズのをびすこともでき`window`。

```
#Create a dummy dataset of 100 observations
x <- rnorm(100)

#Convert this vector to a ts object with 100 annual observations
x <- ts(x, start = c(1900), freq = 1)

#Convert this vector to a ts object with 100 monthly observations starting in July
x <- ts(x, start = c(1900, 7), freq = 12)
```

```
#Alternatively, the starting observation can be a number:
x <- ts(x, start = 1900.5, freq = 12)

#Convert this vector to a ts object with 100 daily observations and weekly frequency starting
in the first week of 1900
x <- ts(x, start = c(1900, 1), freq = 7)

#The default plot for a ts object is a line plot
plot(x)

#The window function can call elements or sets of elements by date

#Call the first 4 weeks of 1900
window(x, start = c(1900, 1), end = (1900, 4))

#Call only the 10th week in 1900
window(x, start = c(1900, 10), end = (1900, 10))

#Call all weeks including and after the 10th week of 1900
window(x, start = c(1900, 10))
```

のシリーズで`ts`オブジェクトをすることはです

```
#Create a dummy matrix of 3 series with 100 observations each
x <- cbind(rnorm(100), rnorm(100), rnorm(100))

#Create a multi-series ts with annual observation starting in 1900
x <- ts(x, start = 1900, freq = 1)

#R will draw a plot for each series in the object
plot(x)
```

オンラインでとをむ <https://riptutorial.com/ja/r/topic/2701/>と

## 112: モデル

パッケージの`Arima`は、をどのようにうかをよりにしています。これは、ベースRの`arima`にべて、のユーザーにとってよになるがあります。

ARIMAは、にシリーズをして、データからモデルをしするためのなフレームワークです。このフレームワークのは、のをさせるために、およびのダイナミクスをシリーズすることです。よりには、ARIMAモデルは、システムへのショックがをじてどのようにわるかをするをします。

から、ARIMAは、をし、をするためにです。

### Examples

#### ArimaをいたAR1プロセスのモデリング

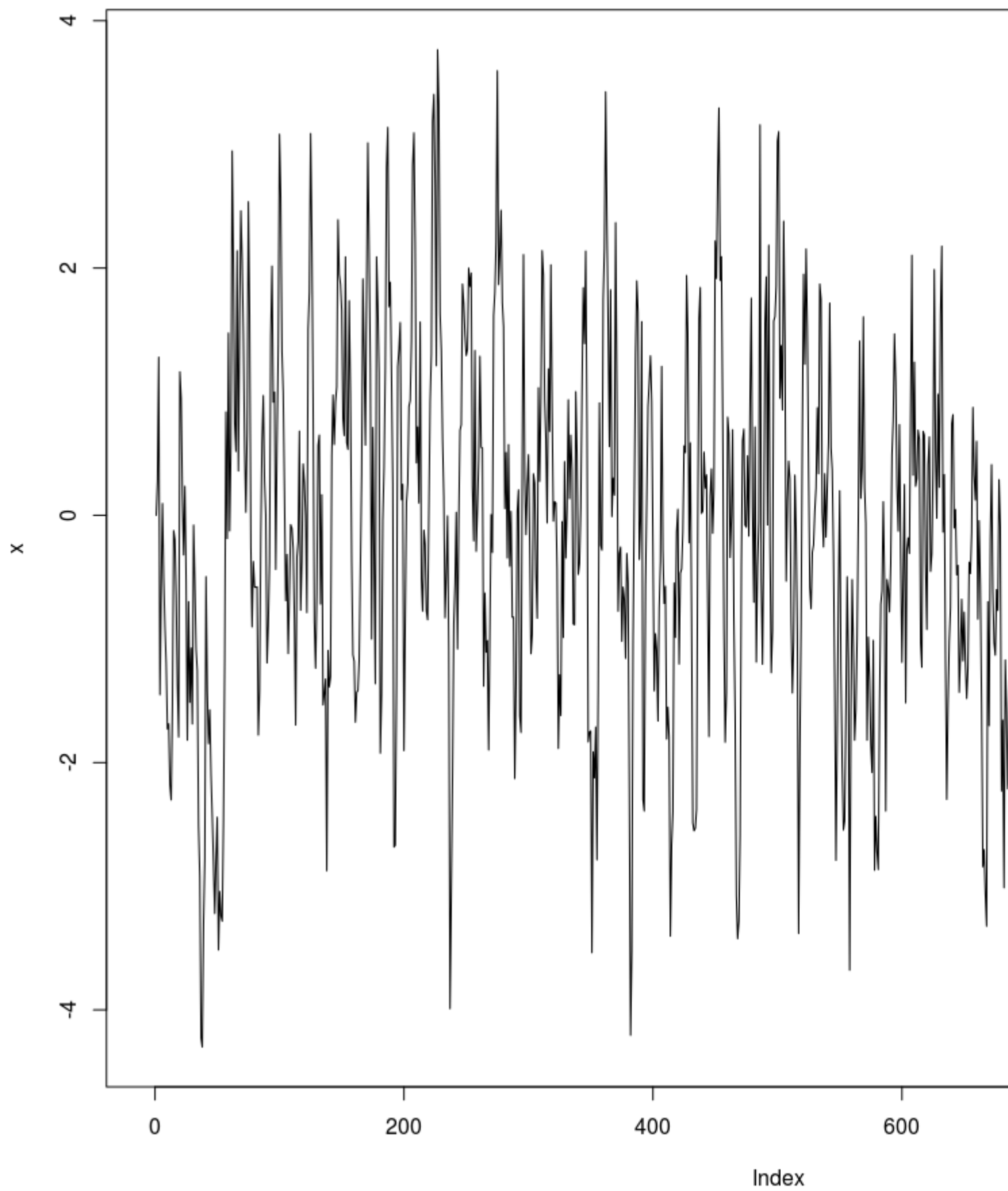
プロセスをモデルする

$$x_t = .7x_{t-1} + \epsilon \quad \epsilon \sim N(0, 1)$$

```
#Load the forecast package
library(forecast)

#Generate an AR1 process of length n (from Cowpertwait & Meltcalfe)
Set up variables
set.seed(1234)
n <- 1000
x <- matrix(0,1000,1)
w <- rnorm(n)

loop to create x
for (t in 2:n) x[t] <- 0.7 * x[t-1] + w[t]
plot(x,type='l')
```



々は、1、0、MAオーダー0のArimaモデルをさせる。

```
#Fit an AR1 model using Arima
fit <- Arima(x, order = c(1, 0, 0))
summary(fit)
Series: x
ARIMA(1,0,0) with non-zero mean
#
Coefficients:
ar1 intercept
0.7040 -0.0842
s.e. 0.0224 0.1062
#
sigma^2 estimated as 0.9923: log likelihood=-1415.39
AIC=2836.79 AICc=2836.81 BIC=2851.51
#
Training set error measures:
ME RMSE MAE MPE MAPE MASE ACF1
Training set -8.369365e-05 0.9961194 0.7835914 Inf Inf 0.91488 0.02263595
Verify that the model captured the true AR parameter
```

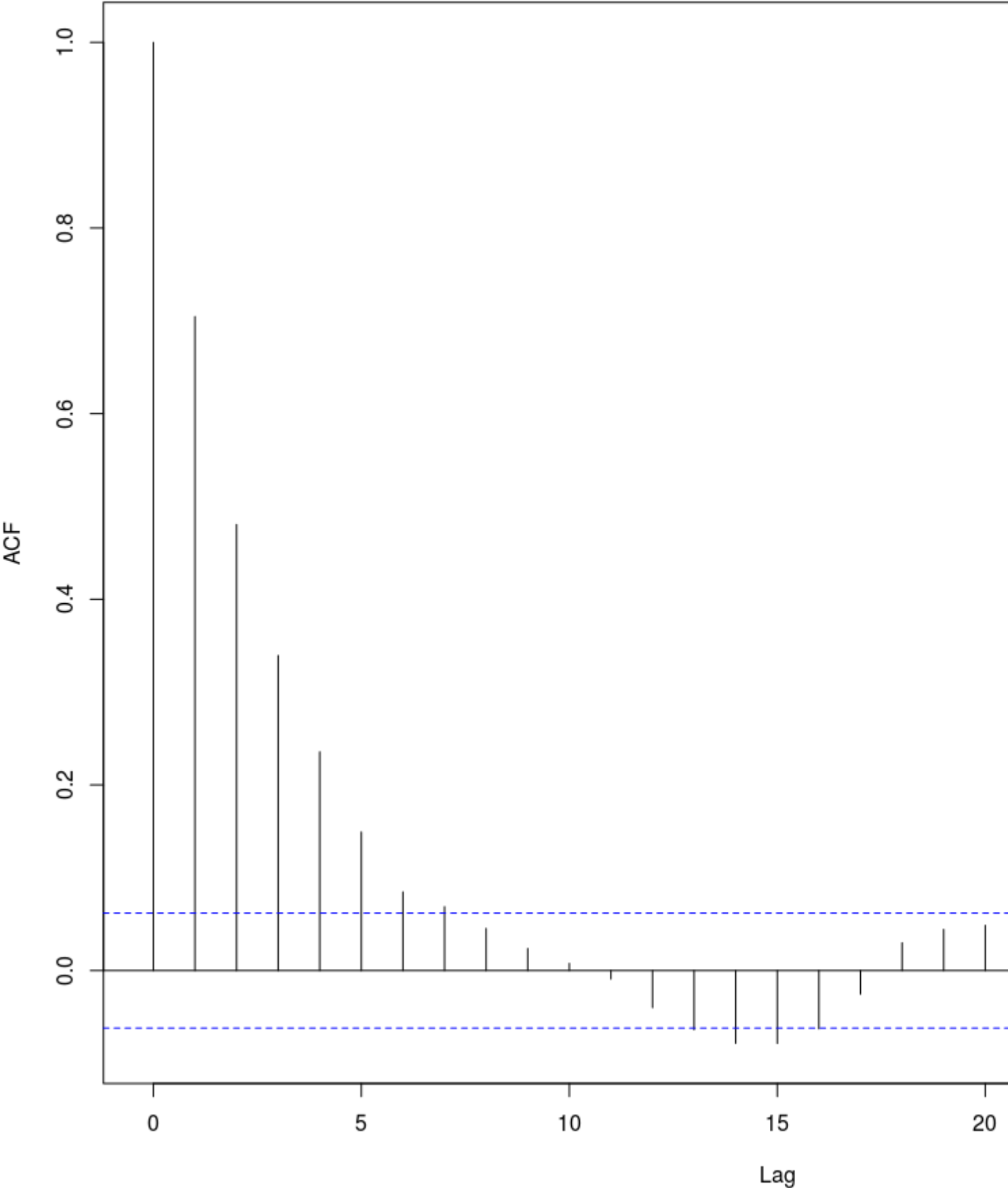
たちののは、されたデータからののにいことにしてください

```
fit$coef[1]
ar1
0.7040085

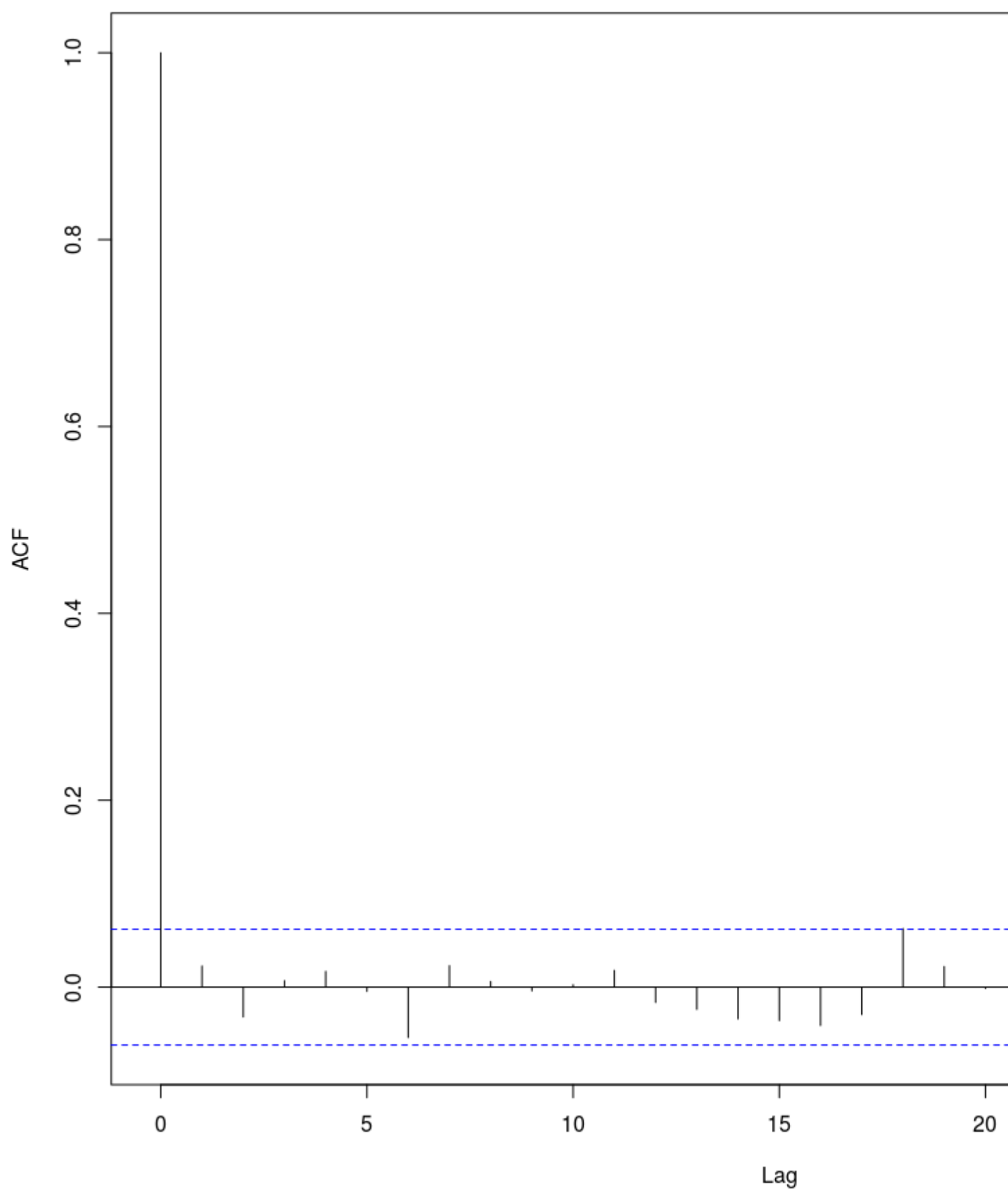
#Verify that the model eliminates the autocorrelation
acf(x)
```



Series 1



```
acf(fit$resid)
```



```

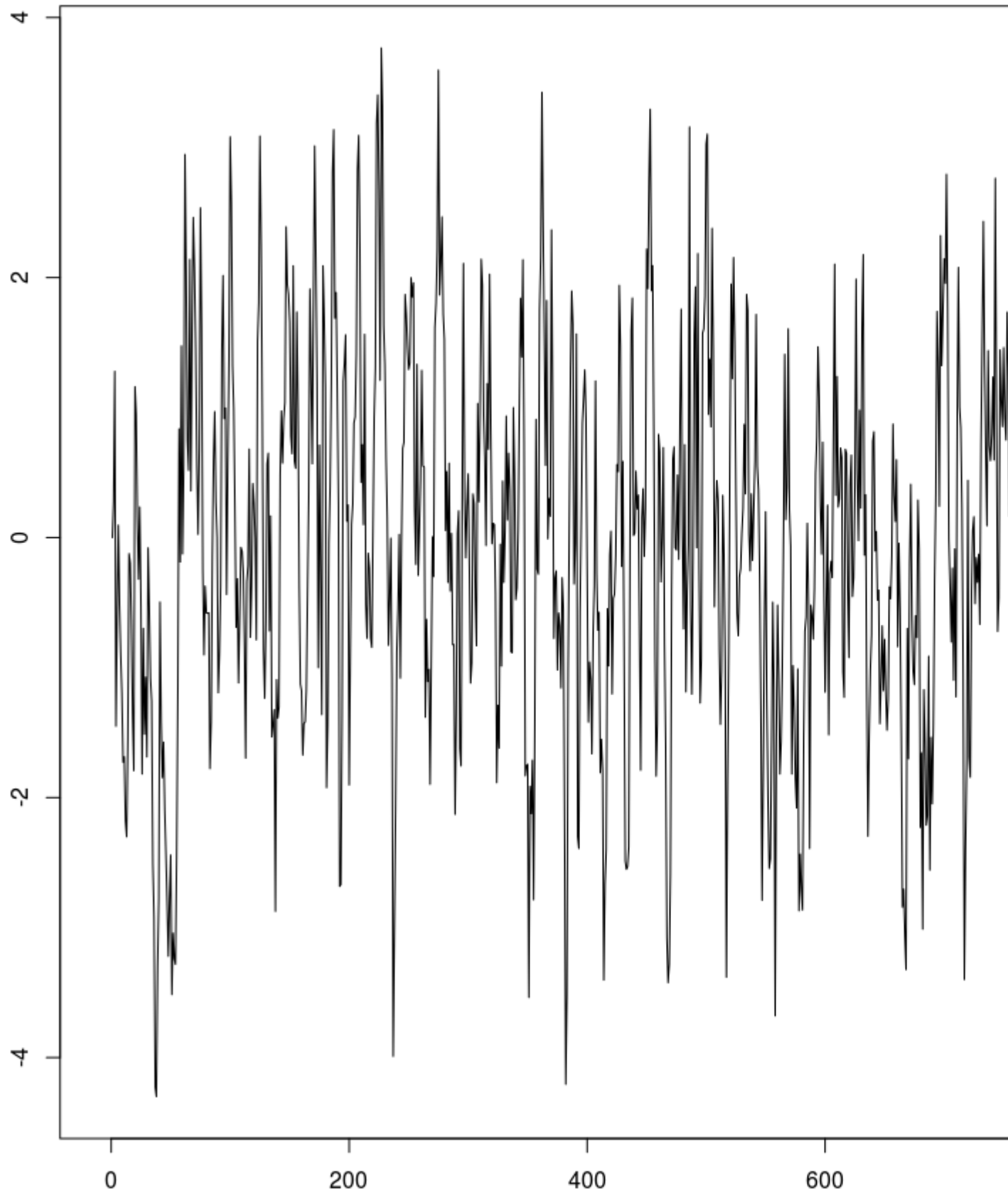
#Forecast 10 periods
fcst <- forecast(fit, h = 100)
fcst
 Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
1001 0.282529070 -0.9940493 1.559107 -1.669829 2.234887
1002 0.173976408 -1.3872262 1.735179 -2.213677 2.561630
1003 0.097554408 -1.5869850 1.782094 -2.478726 2.673835
1004 0.043752667 -1.6986831 1.786188 -2.621073 2.708578
1005 0.005875783 -1.7645535 1.776305 -2.701762 2.713514
...

#Call the point predictions
fcst$mean
Time Series:
Start = 1001
End = 1100
Frequency = 1
[1] 0.282529070 0.173976408 0.097554408 0.043752667 0.005875783 -0.020789866 -
0.039562711 -0.052778954
[9] -0.062083302
...

#Plot the forecast
plot(fcst)

```

## Forecasts from ARIMA(1,0,0) with non-zero



オンラインでモデルをむ <https://riptutorial.com/ja/r/topic/1725/モデル>

## 113: グラフ

き

グラフのは、のレベルのまたはをすることです。えは、グラフは、々なグループレベル - 、 、 に  
おけるのまたはをでするためにされます。そのようなプロットは、々なレベルののなをすること  
に

### Examples

#### barplot

バープロットでは、レベルがxにされ、さまざまなレベルのまたはがyでされます。レベルについ  
て、レベルのまたはにするさをするなの1つのがされる。

`barplot()` は、R's System Libraryのグラフィックパッケージにまれています。 `barplot()` にはなく  
とも1つのをするがあります。 Rヘルプはこれを `heights` とびます。これはベクトルまたはのいず  
れかでなければなりません。ベクトルの、メンバーはさまざまなレベルです。

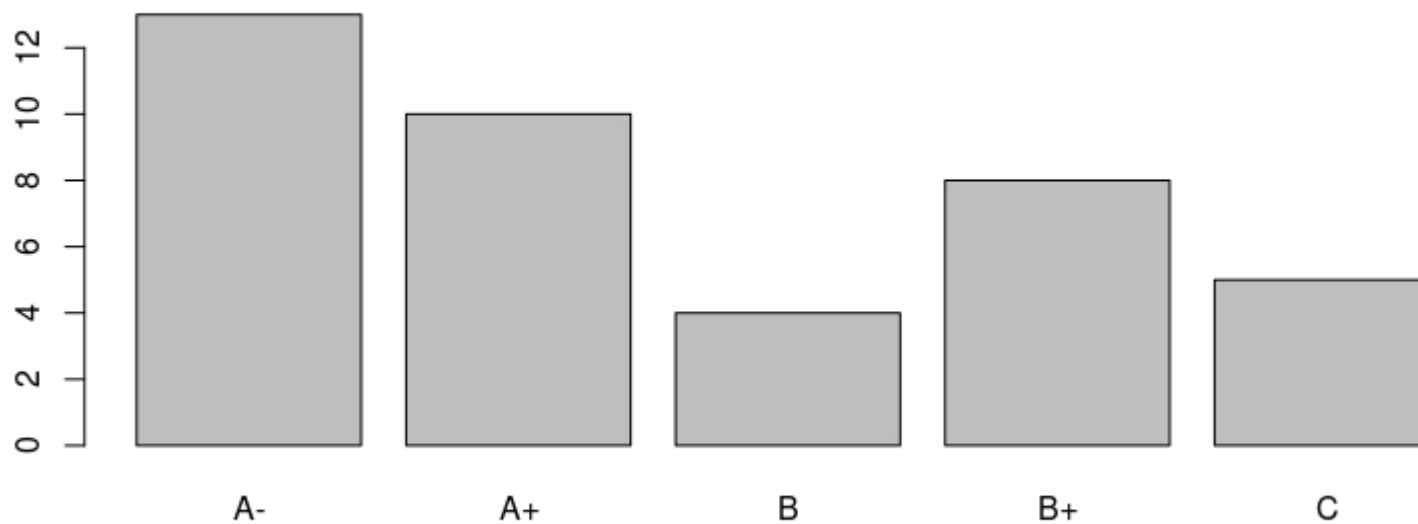
`barplot()` をするために、のデータをしてください。

```
> grades<-c("A+", "A-", "B+", "B", "C")
> Marks<-sample(grades,40,replace=T,prob=c(.2,.3,.25,.15,.1))
> Marks
[1] "A+" "A-" "B+" "A-" "A+" "B" "A+" "B+" "A-" "B" "A+" "A-"
[13] "A-" "B+" "A-" "A-" "A-" "A-" "A+" "A-" "A+" "A+" "C" "C"
[25] "B" "C" "B+" "C" "B+" "B+" "B+" "A+" "B+" "A-" "A+" "A-"
[37] "A-" "B" "C" "A+"
>
```

マークベクトルのグラフは、

```
> barplot(table(Marks),main="Mid-Marks in Algorithms")
```

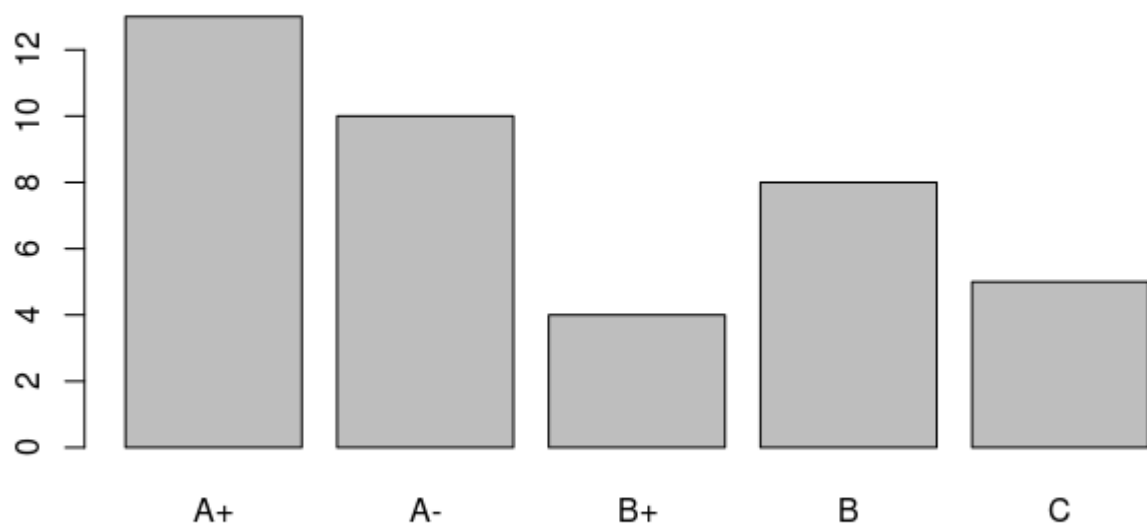
## Mid-Marks in Algorithms



`barplot`は、レベルをxのレベルのlexicographical orderにべることにしてください。パラメータ `names.arg` をすると、`plot`のグラフは、ベクトル、 にされているでできます。

```
plot to the desired horizontal axis labels
> barplot(table(Marks),names.arg=grades ,main="Mid-Marks in Algorithms")
```

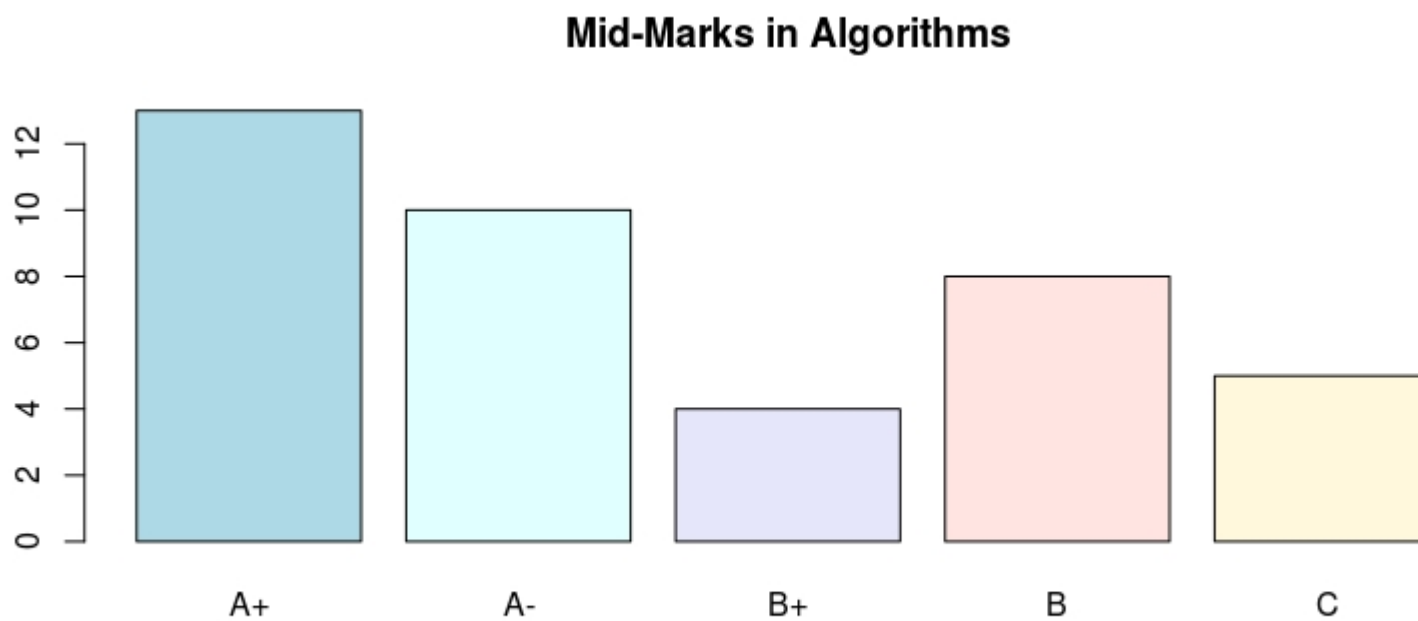
## Mid-Marks in Algorithms



カラーバーは、 `col`=パラメータをしてできます。

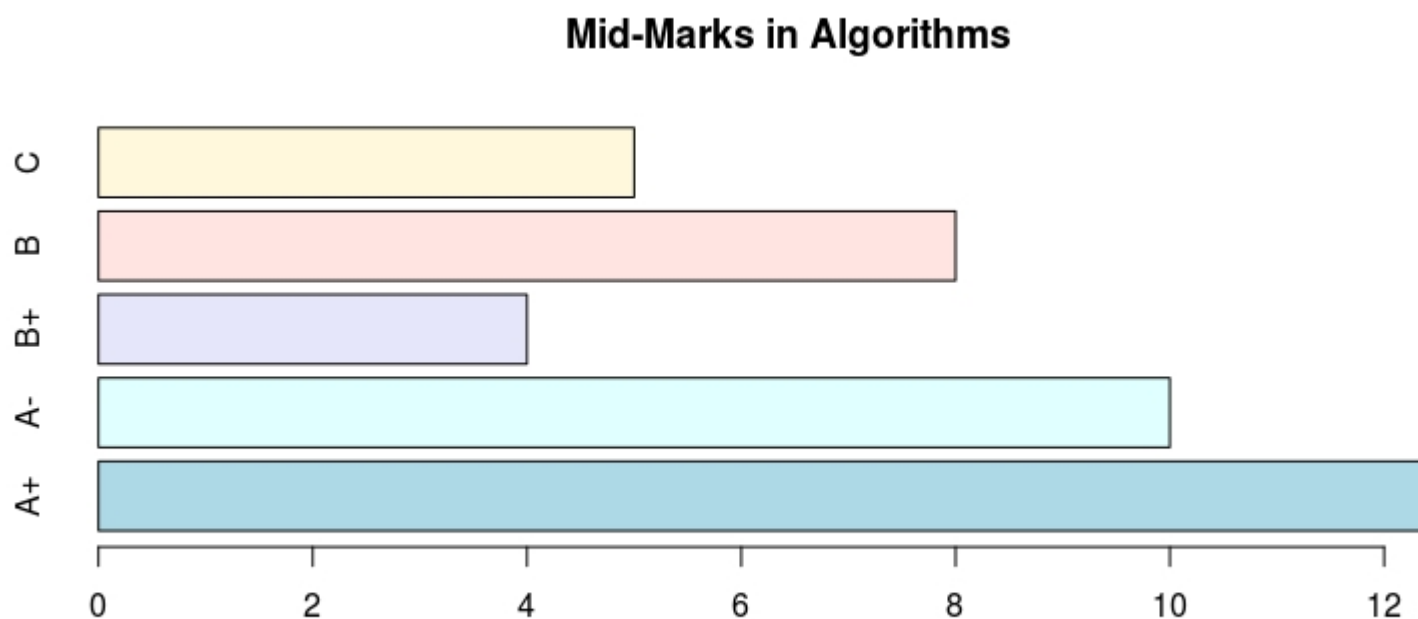


```
> barplot(table(Marks),names.arg=grades,col = c("lightblue",
 "lightcyan", "lavender", "mistyrose", "cornsilk"),
 main="Mid-Marks in Algorithms")
```



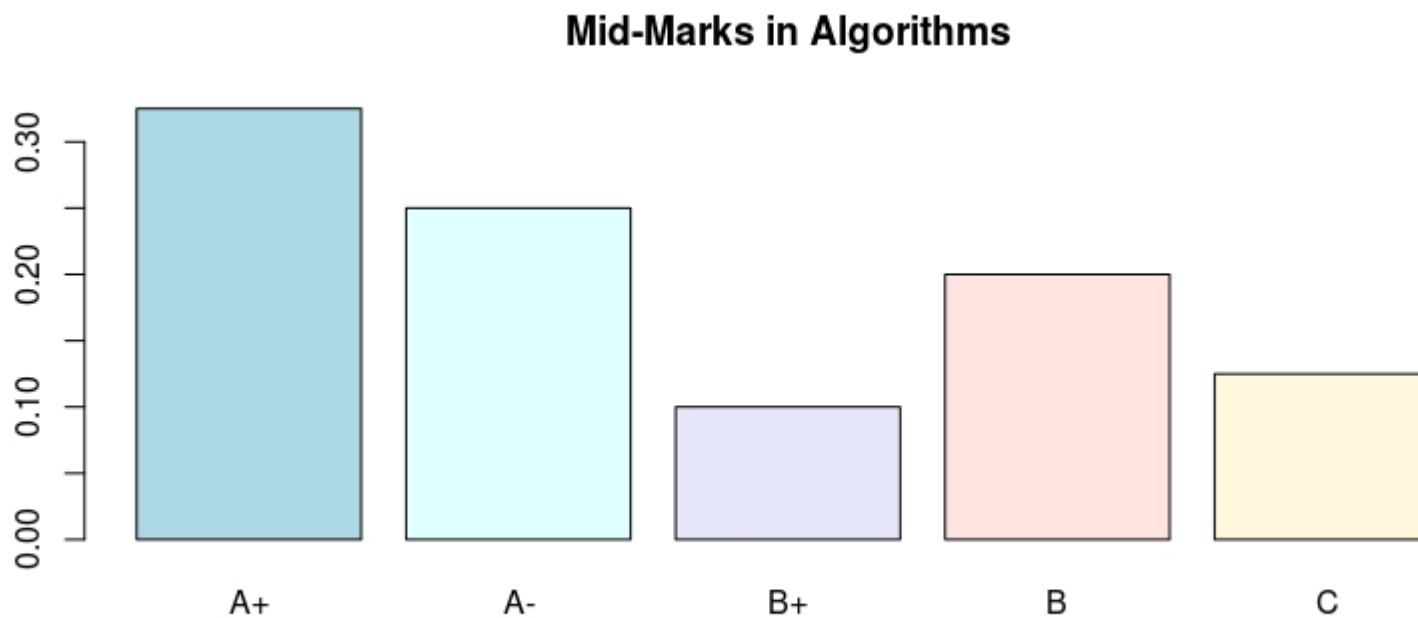
グラフは、のようになてできます。

```
> barplot(table(Marks),names.arg=grades,hORIZ=TRUE,col = c("lightblue",
 "lightcyan", "lavender", "mistyrose", "cornsilk"),
 main="Mid-Marks in Algorithms")
```



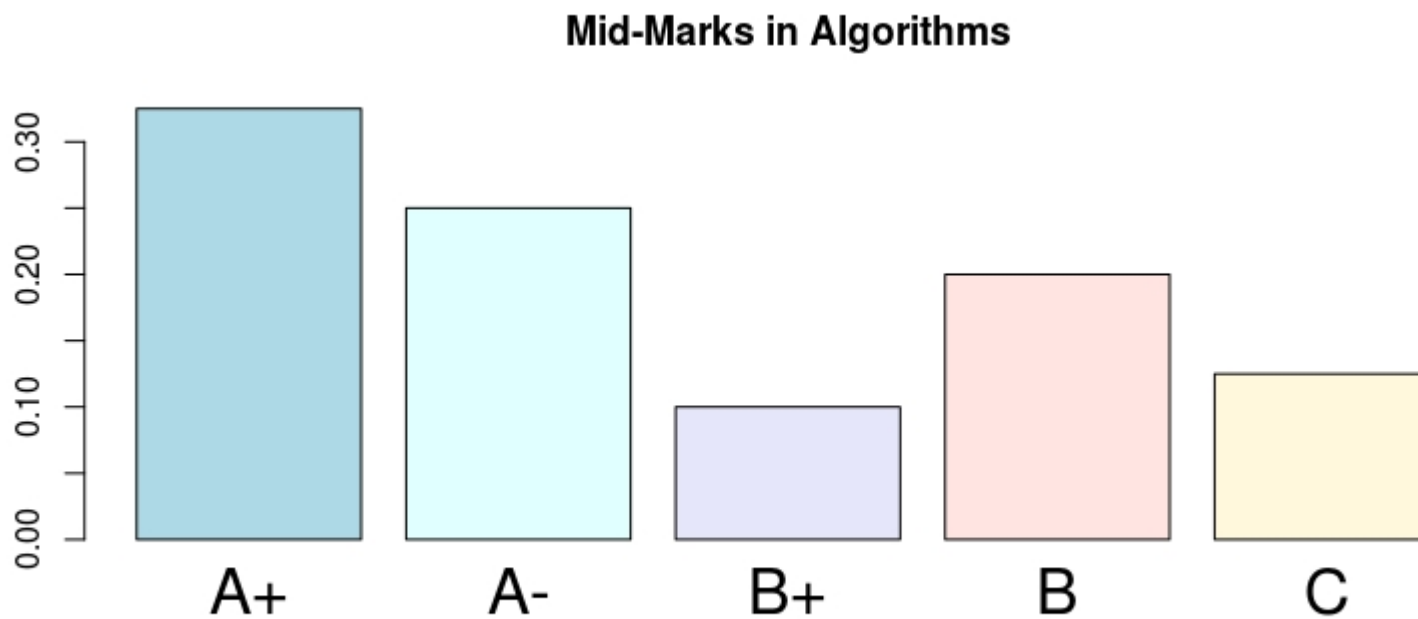
yのをするグラフは、のようになっていることができる。

```
> barplot(prop.table(table(Marks)), names.arg=grades, col = c("lightblue",
 "lightcyan", "lavender", "mistyrose", "cornsilk"),
 main="Mid-Marks in Algorithms")
```



xのレベルのサイズは、`cex.names`パラメータをしてさせることができます。

```
> barplot(prop.table(table(Marks)),names.arg=grades,col = c("lightblue",
 "lightcyan", "lavender", "mistyrose", "cornsilk"),
 main="Mid-Marks in Algorithms",cex.names=2)
```

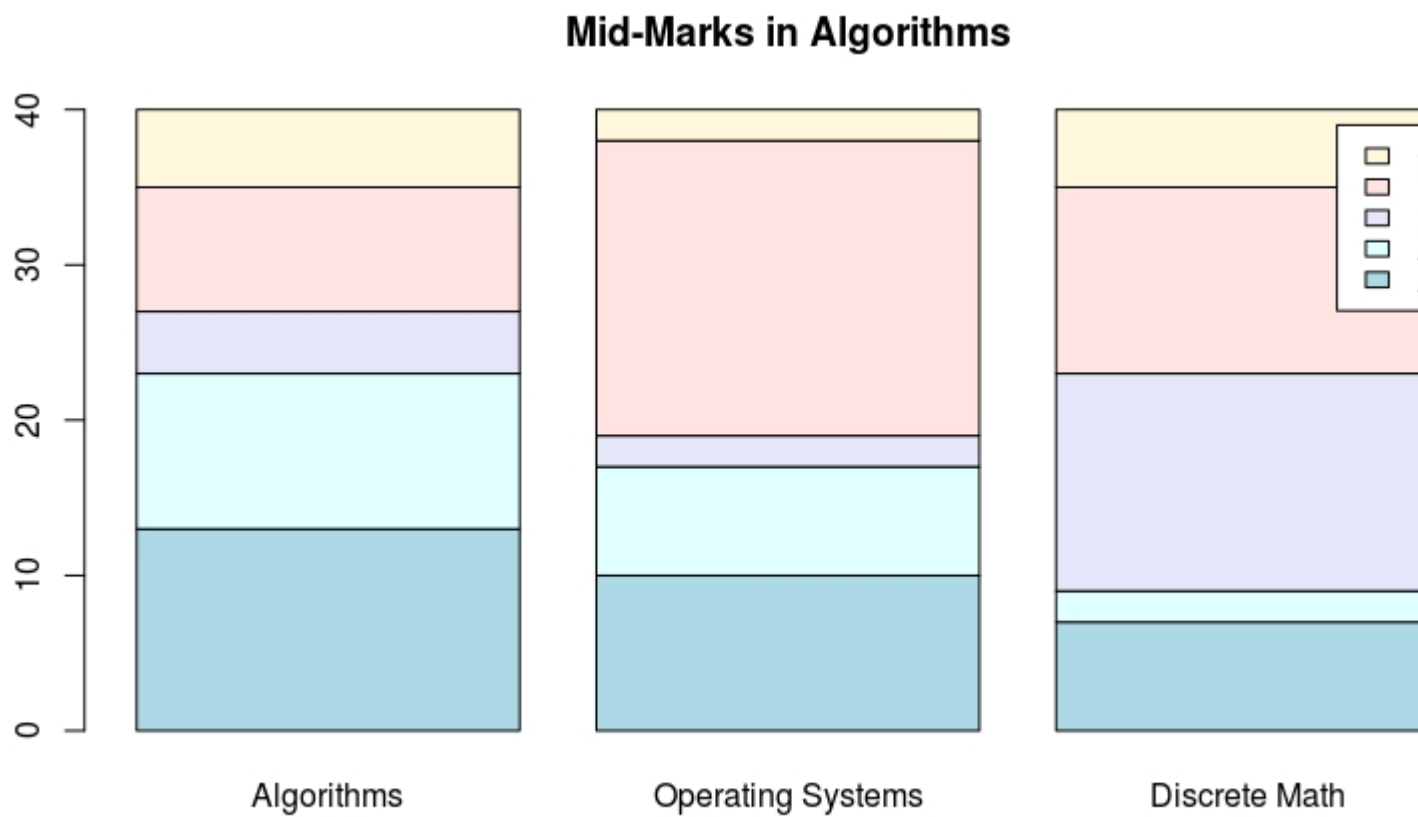


`barplot()` の `heights` パラメータはであるがあります。え、それはであってもよく、はコースでられた々なであり、はのラベルであってもよい。のをえてみましょう。

```
> gradTab
 Algorithms Operating Systems Discrete Math
A- 13 10 7
A+ 10 7 2
B 4 2 14
B+ 8 19 12
C 5 2 5
```

スタックされたバーをするには、のコマンドをします。

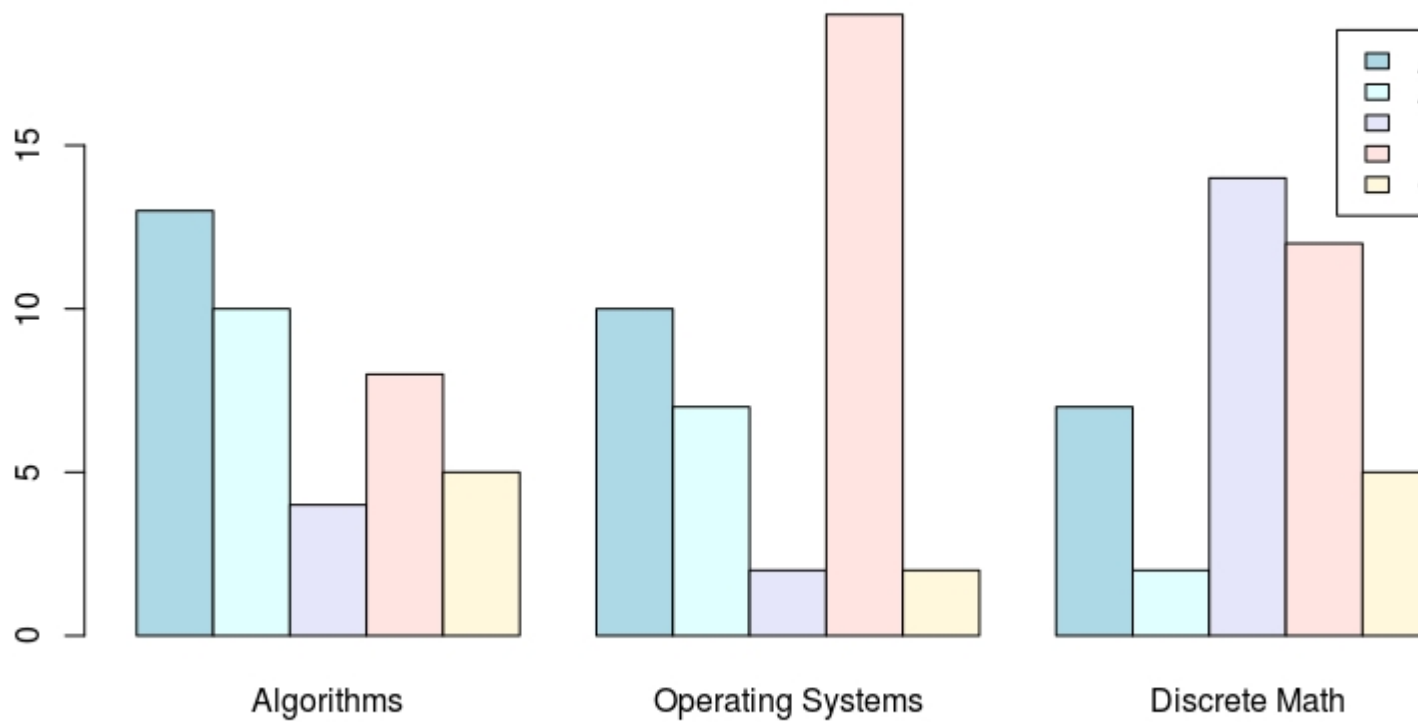
```
> barplot(gradTab,col = c("lightblue","lightcyan",
 "lavender", "mistyrose", "cornsilk"),legend.text = grades,
 main="Mid-Marks in Algorithms")
```



されたグラフをするには、のように `besides` パラメータをします。

```
> barplot(gradTab,beside = T,col = c("lightblue","lightcyan",
 "lavender", "mistyrose", "cornsilk"),legend.text = grades,
 main="Mid-Marks in Algorithms")
```

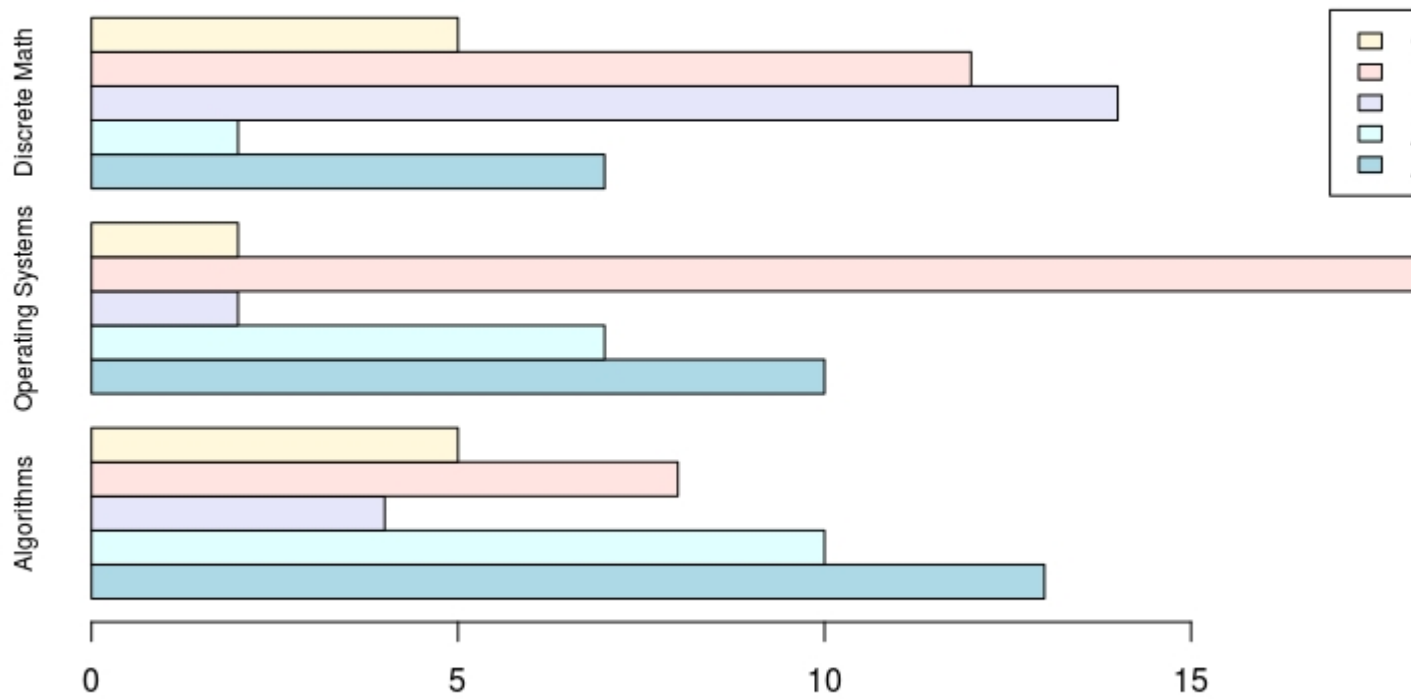
## Mid-Marks in Algorithms



グラフは、 `horiz=T` パラメータをしてできます。

```
> barplot(gradTab,beside = T,horiz=T,col = c("lightblue","lightcyan",
 "lavender", "mistyrose", "cornsilk"),legend.text = grades,
 cex.names=.75,main="Mid-Marks in Algorithms")
```

## Mid-Marks in Algorithms



オンラインでグラフをむ <https://riptutorial.com/ja/r/topic/8091/グラフ>

## 114:

### Examples

ランダムフォレストモデルの

アルゴリズムのは、ランダムフォレストアルゴリズムであるBreiman、L.2001、Random Forests、*Machine Learning* 455、p. 5-32。このアルゴリズムは、randomForestパッケージのBreimanのFortranによってRでされてrandomForestます。

クラスターをfactorとしてすることにより、Rでランダムフォレストのオブジェクトをすることができます。これは、すでにirisデータセットでわかです。したがって、のでランダムフォレストをできます。

```
library(randomForest)

rf <- randomForest(x = iris[, 1:4],
 y = iris$Species,
 ntree = 500,
 do.trace = 100)

rf

Call:
randomForest(x = iris[, 1:4], y = iris$Species, ntree = 500, do.trace = 100)
Type of random forest: classification
Number of trees: 500
No. of variables tried at each split: 2
#
OOB estimate of error rate: 4%
Confusion matrix:
setosa versicolor virginica class.error
setosa 50 0 0 0.00
versicolor 0 47 3 0.06
virginica 0 3 47 0.06
```

| パラメータ    |                                               |
|----------|-----------------------------------------------|
| バツ       | クラスのをするデータフレーム                                |
| y        | 々ののクラス。このベクトルがfactorは、モデルがされ、そうでないは、モデルがされます。 |
| ntree    | された々のCARTツリーの                                 |
| do.trace | iのステップごとに、なエラーとクラスのエラーがされます                   |

オンラインでをむ <https://riptutorial.com/ja/r/topic/8326/>

# 115: プログラミング

## Examples

ビルトイン

Rには、Map、Reduce、Filter、Find、Position、Negateがみまれています。

Mapは、されたをのリストにします。

```
words <- list("this", "is", "an", "example")
Map(toupper, words)
```

Reduceは、バイナリをにのリストににします。

```
Reduce(`*`, 1:10)
```

とのリストがされたFilter、がTRUEののみをむフィルターされたリストをします。

```
Filter(is.character, list(1,"a",2,"b",3,"c"))
```

えられたFind、のリストはがであるのをします。

```
Find(is.character, list(1,"a",2,"b",3,"c"))
```

Positionおよびのリストをえられたがであるリストのののをします。

```
Position(is.character, list(1,"a",2,"b",3,"c"))
```

Negateは、をさせ、TRUEをしたがFALSEをし、TRUEをしたをします。

```
is.noncharacter <- Negate(is.character)
is.noncharacter("a")
is.noncharacter(mean)
```

オンラインでプログラミングをむ <https://riptutorial.com/ja/r/topic/5050/プログラミング>



---

## 116:

き

「」または「」ともばれるは、とできるパターンをします。 `?regex` Rドキュメントのは `?regex` とし、については [Regex Docs](#) をしてください。 SOのトピックではれないもな「つかまえ」は、ほとんどのR-regexが `pattern` パラメータでエスケープするためにペアのバックスラッシュをするがあるということです。

---

## クラス

- `"[AB]"` はAまたはB
- `"[:alpha:]"` はの
- `"[:lower:]"` はをします。 `"[az]"` はいがしないことにしてください。 `ú`。
- `"[:upper:]"` はをします。 `"[AZ]"` はいがしないことにしてください。 `ú`。
- `"[:digit:]"` は0,1,2、...、または9ののをし、 `"[0-9]"` とです。

---

`+`、`*`、`?` いつものようにでしてください。 `-+` はなくとも1マッチする、`*0` マッチする、そして`?0` または1します。

---

## ラインインジケータのと

にのをすることができます

- `"^..."` はをにのにく
- `"...$"` はをのにします

---

## のとのい

Rのは、のでされているとはしなるがあります。

- Rは、ので、のバックスラッシュはエスケープがです `"\"` すでにRでにれることをする、そう、例えば、ほとんどのエンジン、いずれかのをキャプチャすることは、にするがあります `\"`、`\"` Rに。
- RのUTF-8はのUでエスケープするがあります。 `[\u{1F600}]` と `[\u1F600]` しますが、Rubyではこれはの `u` とします。

# そのリソース

のサイト[reg101](#)は、Rスクリプトをするにオンラインをチェックするのにしています。

[R Programming wikibook](#)には、をったがたくさんあるテキストのページがあります。

## Examples

をりく

```
string <- ' some text on line one;
and then some text on line two '
```

をする

"トリミング"は、からののをすることをします。これは、ののみをしてうことができます。`gsub`は、のマッチのにをするためにされます。

R 3.2.0より

```
gsub(pattern = "^(+| +$)",
 replacement = "",
 x = string)

[1] "some text on line one; \nand then some text on line two"
```

R 3.2.0

```
trimws(x = string)

[1] "some text on line one; \nand then some text on line two"
```

するをする

R 3.2.0より

```
sub(pattern = "^ +",
 replacement = "",
 x = string)

[1] "some text on line one; \nand then some text on line two "
```

R 3.2.0

```
trimws(x = string,
 which = "left")
```

```
[1] "some text on line one; \nand then some text on line two "
```

## のをする

### R 3.2.0より

```
sub(pattern = " +$",
 replacement = "",
 x = string)
```

```
[1] " some text on line one; \nand then some text on line two"
```

### R 3.2.0

```
trimws(x = string,
 which = "right")
```

```
[1] " some text on line one; \nand then some text on line two"
```

## すべてのをする

```
gsub(pattern = "\\s",
 replacement = "",
 x = string)
```

```
[1] "sometextonlineone;andthensometextonlinetwo"
```

タブ `\t`、`\r`と`\n`、などのもされることにしてください。

## を「YYYYMMDD」でする

をとしてするファイルには、`YYYYMMDD`でをけることが20170101\_results.csv。 20170101\_results.csv。このようなのは、のをしてできます。

```
\\d{4}(0[1-9]|1[012])(0[1-9]|[12][0-9]|3[01])
```

のは、0000-9999、01-12および01-31をします。

### えば

```
> grepl("\\d{4}(0[1-9]|1[012])(0[1-9]|[12][0-9]|3[01])", "20170101")
[1] TRUE
> grepl("\\d{4}(0[1-9]|1[012])(0[1-9]|[12][0-9]|3[01])", "20171206")
[1] TRUE
> grepl("\\d{4}(0[1-9]|1[012])(0[1-9]|[12][0-9]|3[01])", "29991231")
[1] TRUE
```

これはのをしますが、なた例えば、 20170229 2017はうるうではありませんでをえることがあります。

```
> grepl("\\d{4}(0[1-9]|1[012])(0[1-9]|[12][0-9]|3[01])", "20170229")
[1] TRUE
```

をするは、このユーザーをしてをできます。

```
is.Date <- function(x) {return(!is.na(as.Date(as.character(x), format = '%Y%m%d')))}
```

その、

```
> is.Date(c("20170229", "20170101", 20170101))
[1] FALSE TRUE TRUE
```

アメリカのを

のregexは50と Commonwealth / Territory [www.50states.com](http://www.50states.com)がまれます

```
regex <-
"(A[LKSZJR]|(C[AOT])|(D[EC])|(F[ML])|(G[AU])|(HI)|(I[DLNA])|(K[SY])|(LA)|(M[EHDAINSOT])|(N[EVHJMYCD])|(N
```

えば

```
> test <- c("AL", "AZ", "AR", "AJ", "AS", "DC", "FM", "GU", "PW", "FL", "AJ", "AP")
> grepl(us.states.pattern, test)
[1] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE
>
```

50のだけをしたいは、 state から state.abb というRデータセットをすることをおめします。

```
> data(state)
> test %in% state.abb
[1] TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE
```

AL, AZ, AR, FL 50-StatesについてのみTRUEをます。

のを

のはのとおりです。

```
us.phones.regex <- "^\\s*(\\+\\s*1(?:\\s+))*[0-9]{3}\\s*-?\\s*[0-9]{3}\\s*-?\\s*[0-9]{4}$"
```

のをします。 +1-xxx-xxx-xxxx グループのとにオプションの/ブランクをみますが、にはまれません +1-xxx-xxx-xx xxはです。 -りは、 xxx xxx xxx またはりなしのxxxxxxxxxxにきえることができます。 +1 プレフィックスはオプションです。

それをチェックしよう

```
us.phones.regex <- "^\\s*(\\+\\s*1(?:\\s*))?[0-9]{3}\\s*-?\\s*[0-9]{3}\\s*-?\\s*[0-9]{4}$"

phones.OK <- c("305-123-4567", "305 123 4567", "+1-786-123-4567",
 "+1 786 123 4567", "7861234567", "786 - 123 4567", "+ 1 786 - 123 4567")

phones.NOK <- c("124-456-78901", "124-456-789", "124-456-78 90",
 "124-45 6-7890", "12 4-456-7890")
```

## なケース

```
> grepl(us.phones.regex, phones.OK)
[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE
>
```

## なケース

```
> grepl(us.phones.regex, phones.NOK)
[1] FALSE FALSE FALSE FALSE FALSE
>
```

- `\\s`のスペース、タブ、またはとします。

## Rパターンのをエスケープする

Rとregexのがエスケープ`\\`しているので、`grep`、`sub`、`gsub`またはパターンをけれるののしいパターンをするには、しばしばバックスラッシュのペアリングがです。1つのにがあり、もう1つがで、もう1つがでない3つののベクトルをするは、またはタブを4つのスペースにえて、マッチングのためにバックスラッシュをしています

```
x <- c("a\\nb", "c\\td", "e f")
x # how it's stored
[1] "a\\nb" "c\\td" "e f"
cat(x) # how it will be seen with cat
#a
#b c d e f

gsub(patt="\\n|\\t", repl=" ", x)
#[1] "a b" "c d" "e f"
```

パターンのにれ、してスペルのみがなはオプションは、このまたはペアリングをとするのであることにしてください。は、エスケープするがあるのをとしません。すべてのと4スペースのがタブできえられるようにしたいは、のようになります

```
gsub("\\n| ", "\\t", x)
#[1] "a\\tb" "c\\td" "e\\tf"
```

## PerlとPOSIXのい

Rでされているのエンジンは、わずかになる2つあります。デフォルトはPOSIXとしています。Rのすべてのには、のをにするオプションもされています `perl = TRUE`。

## ルックアヘッド/ルックバック

`perl = TRUE`は、のルックアヘッドとルックバックをにします。

- `"(?<=A)B"`のとする<sub>B</sub>それががしていますにのみ、<sub>A</sub>、すなわち`"ABACADABRA"`にするだろうが、`"abacadabra"`と`"aBacadabra"`ではないだろうが。

オンラインでをむ <https://riptutorial.com/ja/r/topic/5748/-->

## 117:

- `ymd_hms...`、`quiet = FALSE`、`tz = "UTC"`、ロケール = `Sys.getlocale "LC_TIME"`
- `すく` `tzzone = ""`
- `、`、`tzzone = attr、 "tzzone"`
- `num = NULL`、`units = "seconds"`、...
- `num = NULL`、`units = "second"`、...

CRANからパッケージをインストールするには

```
install.packages("lubridate")
```

Githubからをインストールするには

```
library(devtools)
dev mode allows testing of development packages in a sandbox, without interfering
with the other packages you have installed.
dev_mode(on=T)
install_github("hadley/lubridate")
dev_mode(on=F)
```

lubridateパッケージのビネットをするには

```
vignette("lubridate")
```

`foo`にするけをるには

```
help(foo) # help about function foo
?foo # same thing

Example
help("is.period")
?is.period
```

`foo`をるには

```
example("foo")

Example
example("interval")
```

## Examples

きのからのとの

lubridateパッケージは、からオブジェクトとオブジェクトをするなをします。は

|        | する | ベースR              |
|--------|----|-------------------|
| y      |    | %y 、 %Y           |
| myとdをむ |    | %m 、 %b 、 %h 、 %B |
| d      |    | %d 、 %e           |
| h      |    | %H 、 %I%p         |
| mhとs   |    | %M                |
| s      |    | %S                |

、 、 、 、 などのでをするは、 `ymd()` をして、とのにをけてします `ymd_hms()` "2016-07-22"または `ymd_hms()` "2016-07-22 13:04:47"。

は、のなしで、ほとんどのセパレータ `/`、`-`、などをできます。また、のないセパレータでもします。

`date`は `Date` クラスのオブジェクトをします。

```
library(lubridate)

mdy(c(' 07/02/2016 ', ' 7 / 03 / 2016', ' 7 / 4 / 16 '))
[1] "2016-07-02" "2016-07-03" "2016-07-04"

ymd(c("20160724", "2016/07/23", "2016-07-25")) # inconsistent separators
[1] "2016-07-24" "2016-07-23" "2016-07-25"
```

## Datetimes

### ユーティリティ

をしてすることができる `ymd_hms` むバリエント `ymd_hm` と `ymd_h`。すべての `datetime` は、 `as.POSIXct` または `strptime` ような `tz` タイムゾーンをけることができますが、ローカルタイムゾーンではなく "UTC" にデフォルトされています。

`datetime` は、 `POSIXct` クラスのオブジェクトをします。

```
x <- c("20160724 130102", "2016/07/23 14:02:01", "2016-07-25 15:03:00")
ymd_hms(x, tz="EST")
[1] "2016-07-24 13:01:02 EST" "2016-07-23 14:02:01 EST"
[3] "2016-07-25 15:03:00 EST"
```



```
ymd_hms(x)
[1] "2016-07-24 13:01:02 UTC" "2016-07-23 14:02:01 UTC"
[3] "2016-07-25 15:03:00 UTC"
```

## パーサ

lubridateには、as.POSIXctやstrptimeのようなを使ってdatetimeをする3つのもまれています

|                  | クラス                                                 | けたの                                                                          |
|------------------|-----------------------------------------------------|------------------------------------------------------------------------------|
| parse_date_time  | POSIXct                                             | フレキシブル。けるstrptimeでスタイルを%またはlubridateのスタイル、たとえば"ymd hms"。データのベクトルをけれ、であるとします。 |
| parse_date_time2 | デフォルトのPOSIXct<br>。 lt = TRUE、 POSIXlt               | しいきセットからのstrptimeトークン %きまたはなしのみをくれます。                                        |
| fast_strptime    | デフォルトPOSIXlt; lt<br>= FALSE、 POSIXctの<br>lt = FALSE | しいのみけけ% -delimited strptime デリミタきトークン- /、 :されたセットから、など。                      |

```
x <- c('2016-07-22 13:04:47', '07/22/2016 1:04:47 pm')

parse_date_time(x, orders = c('mdy lmsp', 'ymd hms'))
[1] "2016-07-22 13:04:47 UTC" "2016-07-22 13:04:47 UTC"

x <- c('2016-07-22 13:04:47', '2016-07-22 14:47:58')

parse_date_time2(x, orders = 'Ymd HMS')
[1] "2016-07-22 13:04:47 UTC" "2016-07-22 14:47:58 UTC"

fast_strptime(x, format = '%Y-%m-%d %H:%M:%S')
[1] "2016-07-22 13:04:47 UTC" "2016-07-22 14:47:58 UTC"
```

parse\_date\_time2とfast\_strptimeはのためになCパーサをします。

トークンをフォーマットするには?parse\_date\_timeをしてください。

## lubridateのとの

Lubridateは、をにするためのymd() します。y、m、およびdは、の、、のにします。

```
mdy("07-21-2016") # Returns Date

[1] "2016-07-21"

mdy("07-21-2016", tz = "UTC") # Returns a vector of class POSIXt

"2016-07-21 UTC"
```

```
dmy("21-07-2016") # Returns Date

[1] "2016-07-21"

dmy(c("21.07.2016", "22.07.2016")) # Returns vector of class Date

[1] "2016-07-21" "2016-07-22"
```

## lubridateのとの

```
date <- now()
date
"2016-07-22 03:42:35 IST"

year(date)
2016

minute(date)
42

wday(date, label = T, abbr = T)
[1] Fri
Levels: Sun < Mon < Tues < Wed < Thurs < Fri < Sat

day(date) <- 31
"2016-07-31 03:42:35 IST"

If an element is set to a larger value than it supports, the difference
will roll over into the next higher element
day(date) <- 32
"2016-08-01 03:42:35 IST"
```

## インスタント

はのです。あるをするのオブジェクトはとしてされます。オブジェクトがかどうかをテストするには、`is.instant`し`is.instant`。

```
library(lubridate)

today_start <- dmy_hms("22.07.2016 12:00:00", tz = "IST") # default tz="UTC"
today_start
[1] "2016-07-22 12:00:00 IST"
is.instant(today_start)
[1] TRUE

now_dt <- ymd_hms(now(), tz="IST")
now_dt
[1] "2016-07-22 13:53:09 IST"
is.instant(now_dt)
[1] TRUE

is.instant("helloworld")
[1] FALSE
is.instant(60)
[1] FALSE
```

、および

インターバルは、のタイムパンをするもなです。は、2つのののにじるのスパンです。

```
create interval by subtracting two instants
today_start <- ymd_hms("2016-07-22 12:00:00", tz="IST")
today_start
[1] "2016-07-22 12:00:00 IST"
today_end <- ymd_hms("2016-07-22 23:59:59", tz="IST")
today_end
[1] "2016-07-22 23:59:59 IST"
span <- today_end - today_start
span
Time difference of 11.99972 hours
as.interval(span, today_start)
[1] 2016-07-22 12:00:00 IST--2016-07-22 23:59:59 IST

create interval using interval() function
span <- interval(today_start, today_end)
[1] 2016-07-22 12:00:00 IST--2016-07-22 23:59:59 IST
```

は、2つのにじるなをします。

```
duration(60, "seconds")
[1] "60s"

duration(2, "minutes")
[1] "120s (~2 minutes)"
```

をえるは、そののためにされません。

はしてすることができdseconds、dminutesおよびそののヘルパーを。  
なりストについては、?quick\_durationsをして?quick\_durations。

```
dseconds(60)
[1] "60s"

dhours(2)
[1] "7200s (~2 hours)"

dyears(1)
[1] "31536000s (~365 days)"
```

をしてにして、しいをることができます。

```
today_start + dhours(5)
[1] "2016-07-22 17:00:00 IST"

today_start + dhours(5) + dminutes(30) + dseconds(15)
[1] "2016-07-22 17:30:15 IST"
```

はでできます。

```
as.duration(span)
[1] "43199s (~12 hours)"
```

は、2つのインスタントでクロックのをします。

は、`period`、`seconds`、`hours`などのヘルパーをしてすることができます。ヘルパーのなリストをするには、`?quick_periods`し`?quick_periods`。

```
period(1, "hour")
[1] "1H 0M 0S"

hours(1)
[1] "1H 0M 0S"

period(6, "months")
[1] "6m 0d 0H 0M 0S"

months(6)
[1] "6m 0d 0H 0M 0S"

years(1)
[1] "1y 0m 0d 0H 0M 0S"
```

`is.period`をして、オブジェクトがピリオドであるかどうかをべることができます。

```
is.period(years(1))
[1] TRUE

is.period(dyears(1))
[1] FALSE
```

めの

```
now_dt <- ymd_hms(now(), tz="IST")
now_dt
[1] "2016-07-22 13:53:09 IST"
```

`round_date()`は、オブジェクトをし、されたのもいにめます。

```
round_date(now_dt, "minute")
[1] "2016-07-22 13:53:00 IST"

round_date(now_dt, "hour")
[1] "2016-07-22 14:00:00 IST"

round_date(now_dt, "year")
[1] "2017-01-01 IST"
```

`floor_date()`はオブジェクトをとり、されたのもいにめます。

```
floor_date(now_dt, "minute")
[1] "2016-07-22 13:53:00 IST"
```

```

floor_date(now_dt, "hour")
[1] "2016-07-22 13:00:00 IST"

floor_date(now_dt, "year")
[1] "2016-01-01 IST"

```

`ceiling_date()` は、オブジェクトをし、されたのもいにめます。

```

ceiling_date(now_dt, "minute")
[1] "2016-07-22 13:54:00 IST"

ceiling_date(now_dt, "hour")
[1] "2016-07-22 14:00:00 IST"

ceiling_date(now_dt, "year")
[1] "2017-01-01 IST"

```

との

とはなり、うるう、うるう、のなどのイベントがいつするかをらずに、をにモデルするためにをすることができます。

```

start_2012 <- ymd_hms("2012-01-01 12:00:00")
[1] "2012-01-01 12:00:00 UTC"

period() considers leap year calculations.
start_2012 + period(1, "years")
[1] "2013-01-01 12:00:00 UTC"

Here duration() doesn't consider leap year calculations.
start_2012 + duration(1)
[1] "2012-12-31 12:00:00 UTC"

```

`with_tz` は、なるタイムゾーンにされるをします。

```

nyc_time <- now("America/New_York")
nyc_time
[1] "2016-07-22 05:49:08 EDT"

corresponding Europe/Moscow time
with_tz(nyc_time, tzzone = "Europe/Moscow")
[1] "2016-07-22 12:49:08 MSK"

```

`force_tz` は、しいタイムゾーンのxとじをつをします。

```

nyc_time <- now("America/New_York")
nyc_time
[1] "2016-07-22 05:49:08 EDT"

force_tz(nyc_time, tzzone = "Europe/Moscow") # only timezone changes
[1] "2016-07-22 05:49:08 MSK"

```

オンラインでをむ <https://riptutorial.com/ja/r/topic/2496/>

## 118: のパッケージでパイプリテを<>どのように き

ユーザがしたパッケージでパイプをするには、インポートするの他に、NAMESPACEにパイプをリストするがあります。

### Examples

パイプをユーティリティファイルにれる

これをうための1つのオプションは、パッケージからパイプをエクスポートすることです。これは、  
、くのパッケージがパッケージのとしてエクスポートされていないなさなのためにする 'な' `zzz.R`  
または`utils.R`ファイルでうことができます。たとえば、のようになります。

```
#' Pipe operator
#'
#' @name %>%
#' @rdname pipe
#' @keywords internal
#' @export
#' @importFrom magrittr %>%
#' @usage lhs \%>\% rhs
NULL
```

オンラインでのパッケージでパイプリテを<>どのようにをむ

<https://riptutorial.com/ja/r/topic/10547/>のパッケージでパイプリテを--lt--gt---どのように-

## Examples

**randomForestSRC**をいたランダムフォレスト

ランダムフォレストアルゴリズムをおよびタスクにできるように、にもできます。

のでは、モデルが**CRANの** `randomForestSRC` パッケージをした、スコアリング、およびパフォーマンスにし、されます。

```
require(randomForestSRC)

set.seed(130948) #Other seeds give similar comparative results
x1 <- runif(1000)
y <- rnorm(1000, mean = x1, sd = .3)
data <- data.frame(x1 = x1, y = y)
head(data)
```

```
 x1 y
1 0.9604353 1.3549648
2 0.3771234 0.2961592
3 0.7844242 0.6942191
4 0.9860443 1.5348900
5 0.1942237 0.4629535
6 0.7442532 -0.0672639
```

```
(modRFSRC <- rfsrc(y ~ x1, data = data, ntree=500, nodesize = 5))
```

```

 Sample size: 1000
 Number of trees: 500
 Minimum terminal node size: 5
 Average no. of terminal nodes: 208.258
No. of variables tried at each split: 1
 Total no. of variables: 1
 Analysis: RF-R
 Family: regr
 Splitting rule: mse
 % variance explained: 32.08
 Error rate: 0.11
```

```
x1new <- runif(10000)
ynew <- rnorm(10000, mean = x1new, sd = .3)
newdata <- data.frame(x1 = x1new, y = ynew)

survival.results <- predict(modRFSRC, newdata = newdata)
survival.results
```

```

Sample size of test (predict) data: 10000
 Number of grow trees: 500
```

```
Average no. of grow terminal nodes: 208.258
Total no. of grow variables: 1
 Analysis: RF-R
 Family: regr
% variance explained: 34.97
Test set error rate: 0.11
```

はじめに、サバイバルパッケージによるパラメトリックモデルのなフィッティングとプロット

survivalRにおけるのためのものにされるパッケージで、lung々がモデルをてはめることによってをめることができ、データセット `survreg()` との、`survfit()` およびしいデータでこのパッケージの `predict` メソッドをびすことによって、サバイバルカーブをします。

のでは、2つのされたをプロットし、しいデータの2つのセットのの`sex`をえて、そのをします。

```
require(survival)
s <- with(lung, Surv(time, status))

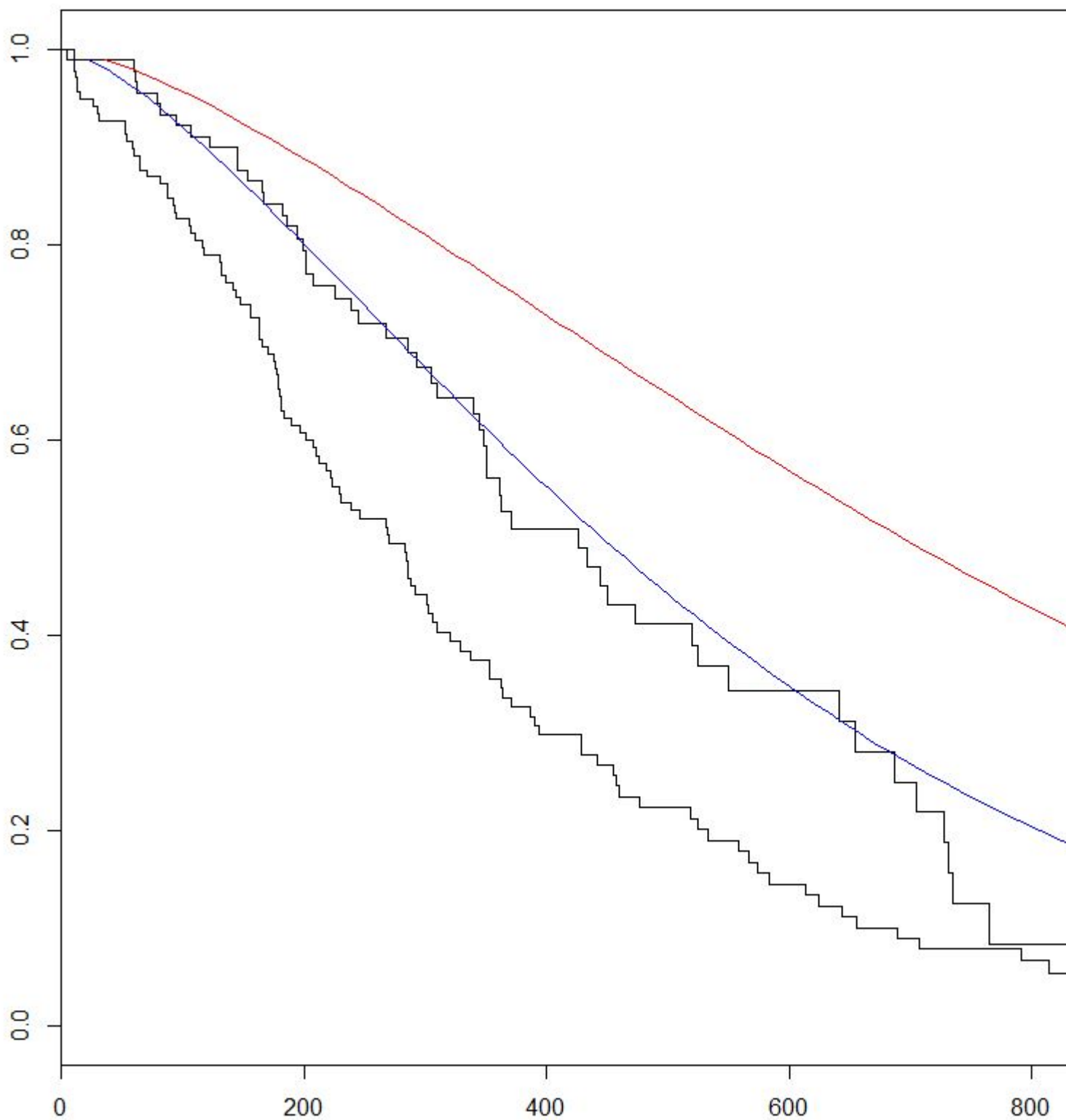
sWei <- survreg(s ~ as.factor(sex)+age+ph.ecog+wt.loss+ph.karno, dist='weibull', data=lung)

fitKM <- survfit(s ~ sex, data=lung)
plot(fitKM)

lines(predict(sWei, newdata = list(sex = 1,
 age = 1,
 ph.ecog = 1,
 ph.karno = 90,
 wt.loss = 2),
 type = "quantile",
 p = seq(.01, .99, by = .01)),
 seq(.99, .01, by = -.01),
 col = "blue")

lines(predict(sWei, newdata = list(sex = 2,
 age = 1,
 ph.ecog = 1,
 ph.karno = 90,
 wt.loss = 2),
 type = "quantile",
 p = seq(.01, .99, by = .01)),
 seq(.99, .01, by = -.01),
 col = "red")
```



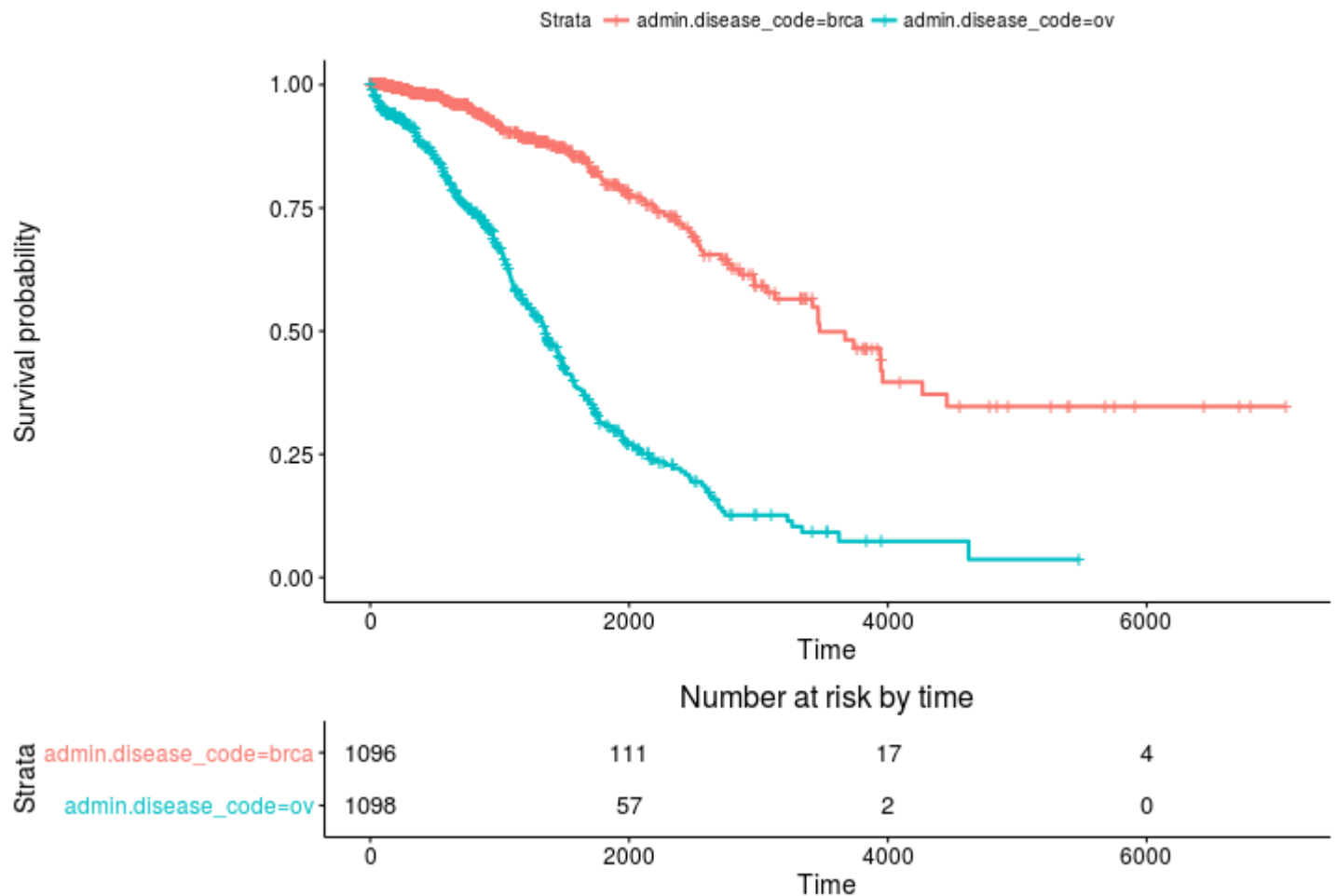


サバイバーとのリスクセットの**Kaplan Meier**

ベースプロット

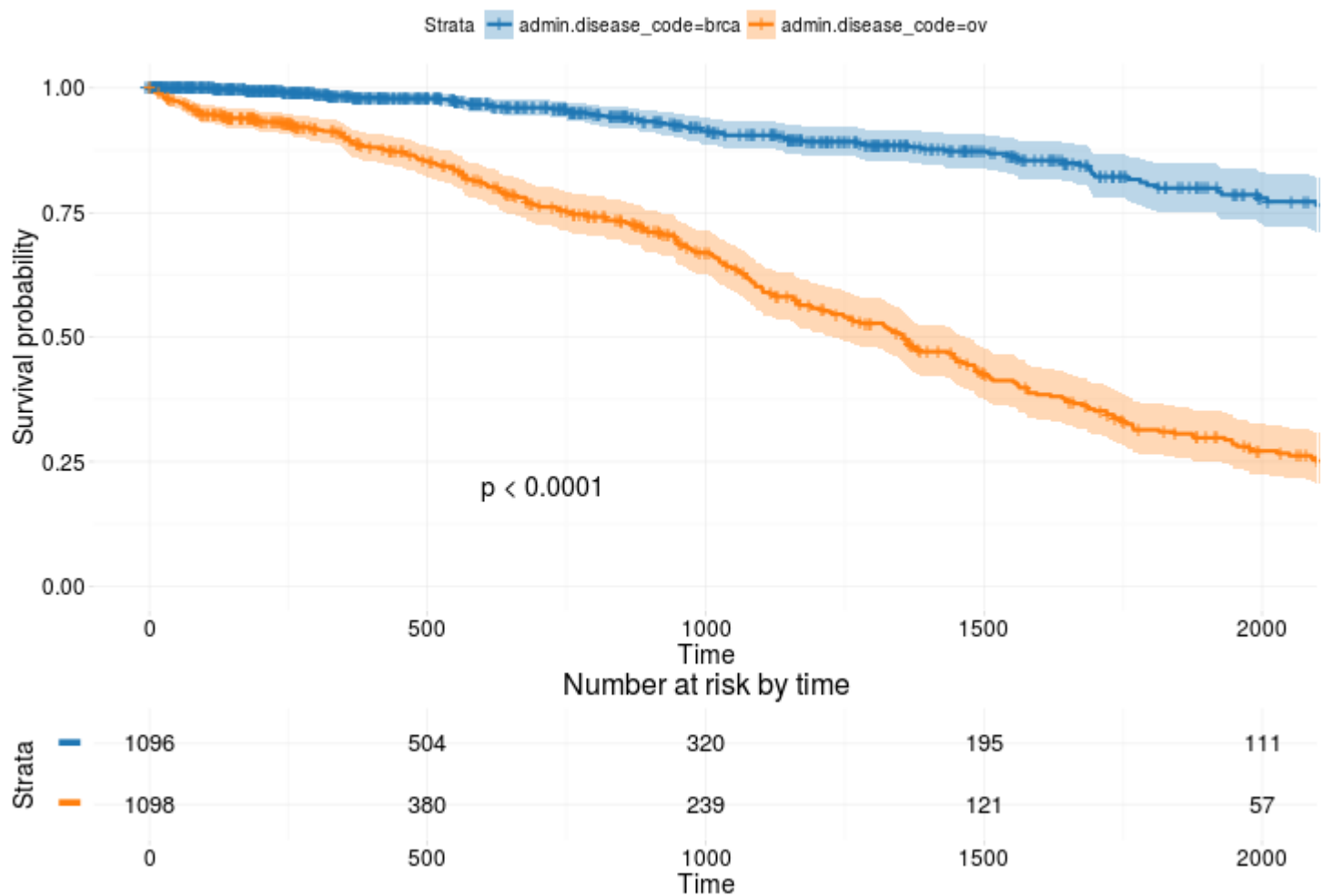
```
install.packages('survminer')
source("https://bioconductor.org/biocLite.R")
biocLite("RTCGA.clinical") # data for examples
```

```
library(RTCGA.clinical)
survivalTCGA(BRCA.clinical, OV.clinical,
 extract.cols = "admin.disease_code") -> BRCAOV.survInfo
library(survival)
fit <- survfit(Surv(times, patient.vital_status) ~ admin.disease_code,
 data = BRCAOV.survInfo)
library(survminer)
ggsurvplot(fit, risk.table = TRUE)
```



よりな

```
ggsurvplot(
 fit, # survfit object with calculated statistics.
 risk.table = TRUE, # show risk table.
 pval = TRUE, # show p-value of log-rank test.
 conf.int = TRUE, # show confidence intervals for
 # point estimates of survival curves.
 xlim = c(0,2000), # present narrower X axis, but not affect
 # survival estimates.
 break.time.by = 500, # break X axis in time intervals by 500.
 ggtheme = theme_RTCGA(), # customize plot and risk table with a theme.
 risk.table.y.text.col = T, # colour risk table text annotations.
 risk.table.y.text = FALSE # show bars instead of names in text annotations
 # in legend of risk table
)
```



につく

<http://r-addict.com/2016/05/23/Informative-Survival-Plots.html>

オンラインでをむ <https://riptutorial.com/ja/r/topic/3788/>

## Examples

### XYデータセットからをする

データにしては、Rはデータ、ののためのなツールであることをしています。

くの、データは、のXYデータとしてできます。このでは、XYデータセットからデータセットをするをします。

パッケージ `rgdal` と `sp` はなをします。Rのデータとしてすることができる `Spatial*DataFrame` \* することができる `Points`、`Lines` は `Polygons`。

このでは、[OpenGeocode](#) でダウンロードできるデータをしています。

に、ディレクトリをダウンロードしたCSVデータセットのフォルダにするがあります。さらに、パッケージ `rgdal` をロードするがあります。

```
setwd("D:/GeocodeExample/")
library(rgdal)
```

その、とそのをしているCSVファイルが `data.frame` としてRにロードされ `data.frame`

```
xy <- read.csv("worldcities.csv", stringsAsFactors = FALSE)
```

くの、データとそのカラム、データタイプなどをすることがです。

```
head(xy)
str(xy)
```

これは、"-33.532"のようなエントリをしているので、とのとはとしてされることをしています。データセットをするでされる `SpatialPointsDataFrame()` では、データ `numeric` が必要です。したがって、2つのをするがあります。

```
xy$latitude <- as.numeric(xy$latitude)
xy$longitude <- as.numeric(xy$longitude)
```

のうちのいくつかをデータにできないため、`NA` がされます。それらはするがあります。

```
xy <- xy[!is.na(xy$longitude),]
```

に、XYデータセットをデータセットにすることができます。これには、がされているCRSのとがです。

```
xySPoints <- SpatialPointsDataFrame(coords = c(xy[,c("longitude", "latitude")]),
proj4string = CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"),
data = xy
)
```

なプロットをすると、されたのピークをにきめることができます。

```
plot(xySPoints, pch = ".")
```



シェイプファイル **.shp** のインポート

## rgdal

ESRIシェイプファイルは、 `rgdal` パッケージから `readOGR()` をしてRにインポートできます。

```
library(rgdal)
shp <- readOGR(dsn = "/path/to/your/file", layer = "filename")
```

なのは、 `dsn` は / でわるべきではなく、 `layer` がファイルのをしていないことえ、 `.shp` です。

## ラスタ

シェイプファイルをインポートするもう1つのは、 `raster` ライブラリと `shapefile` です。

```
library(raster)
shp <- shapefile("path/to/your/file.shp")
```

パスがrgdal importとどのようなになるかにしてください。

## tmap

tmapパッケージは、 rgdal::readORGのらしいラッパーをします。

```
library(tmap)
sph <- read_shape("path/to/your/file.shp")
```

オンラインでをむ <https://riptutorial.com/ja/r/topic/2093/>

## 121:

Rのほぼすべてののがにです。たとえば、`+`は`function (e1, e2) .Primitive("+")`としてされた  
`function (e1, e2) .Primitive("+")`ここで、`e1`はの、`e2`はの。これは、ユーザでの`+`をマスキ  
ングすることによって、にするをすることがであることをする。

えば

```
`+` <- function(e1, e2) {e1-e2}

> 3+10
[1] -7
```

## Examples

と

にをするをえてみましょうたとえば、ループでうことができます。

```
3+1:5
```

える

```
[1] 4 5 6 7 8
```

がいるためです:よりいいがある`+`。

にがこるかはのとおりです。

- `3+1:5`
- `3+c(1, 2, 3, 4, 5)`レンジをしてのベクトルをる。
- `c(4, 5, 6, 7, 8)` `4,5,6,7,8` `c(4, 5, 6, 7, 8)` ベクターのメンバーへの3の。

このようなをけるためには、Rインタプリタに、のように`( )`ってをけするをえるがあります。

```
(3+1):5
```

Rはをするにカッコにあるものをし、のようになります。

```
[1] 4 5
```

しとき

なは、にまたはベクトルのリストでされます。

## 1. の

するには`+`でしたを、`-`をくには、

```
> 3 + 4.5
[1] 7.5
> 3 + 4.5 + 2
[1] 9.5
> 3 + 4.5 + 2 - 3.8
[1] 5.7
> 3 + NA
[1] NA
> NA + NA
[1] NA
> NA - NA
[1] NA
> NaN - NA
[1] NaN
> NaN + NA
[1] NaN
```

このはにをしてじをうことができます

```
> a <- 3; B <- 4.5; cc <- 2; Dd <- 3.8 ;na<-NA;nan<-NaN
> a + B
[1] 7.5
> a + B + cc
[1] 9.5
> a + B + cc - Dd
[1] 5.7
> B-nan
[1] NaN
> a+na-na
[1] NA
> a + na
[1] NA
> B-nan
[1] NaN
> a+na-na
[1] NA
```

## 2. ベクトルの

この、のベクトルをし、それらのベクトル、またはののみわせをしてをいます。この、はベクトルのをしてわれます。

```
> A <- c(3, 4.5, 2, -3.8);
> A
[1] 3.0 4.5 2.0 -3.8
> A + 2 # Adding a number
[1] 5.0 6.5 4.0 -1.8
> 8 - A # number less vector
[1] 5.0 3.5 6.0 11.8
> n <- length(A) #number of elements of vector A
> n
[1] 4
```



```
> A[-n] + A[n] # Add the last element to the same vector without the last element
[1] -0.8 0.7 -1.8
> A[1:2] + 3 # vector with the first two elements plus a number
[1] 6.0 7.5
> A[1:2] - A[3:4] # vector with the first two elements less the vector with elements 3 and 4
[1] 1.0 8.3
```

`sum`をして、ベクトルのすべてのをすることもできます。

```
> sum(A)
[1] 5.7
> sum(-A)
[1] -5.7
> sum(A[-n]) + A[n]
[1] 5.7
```

ベクトルのさがるをうとぎにこる<sub>R</sub>のの1つであるリサイクルにするがあります。のよりいベクトルは、もいベクトルのさとするまで、にじておそらくにされます。に、はにりされます。この、がされます。

```
> B <- c(3, 5, -3, 2.7, 1.8)
> B
[1] 3.0 5.0 -3.0 2.7 1.8
> A
[1] 3.0 4.5 2.0 -3.8
> A + B # the first element of A is repeated
[1] 6.0 9.5 -1.0 -1.1 4.8
Warning message:
In A + B : longer object length is not a multiple of shorter object length
> B - A # the first element of A is repeated
[1] 0.0 0.5 -5.0 6.5 -1.2
Warning message:
In B - A : longer object length is not a multiple of shorter object length
```

この、しいきは、いのベクトルのだけをするということです。

```
> B[1:n] + A
[1] 6.0 9.5 -1.0 -1.1
> B[1:n] - A
[1] 0.0 0.5 -5.0 6.5
```

`sum`をすると、のすべてのがびされます。

```
> sum(A, B)
[1] 15.2
> sum(A, -B)
[1] -3.8
> sum(A)+sum(B)
[1] 15.2
> sum(A)-sum(B)
[1] -3.8
```

オンラインでをむ <https://riptutorial.com/ja/r/topic/4389/>

## 122: モデル

- `lm`、データ、サブセット、み、`na.action`、メソッド = "qr"、モデル = TRUE、`x` = FALSE、`y` = FALSE、`qr` = TRUE、`singular.ok` = TRUE、コントラスト = NULL、オフセット、..。

### パラメーター

| パラメータ                    |                                                                                                                                                                                                          |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                          | ウィルキンソン=ロジャースの。 <code>response ~ ... where ...</code> は、のまたは <code>data</code> でされた <code>data</code> フレームにするがまれています                                                                                     |
| データ                      | とをむデータフレーム                                                                                                                                                                                               |
| サブセット                    | されるのサブセットをするベクトルは、 <code>data</code> ののからなステートメントとしてすることができ <code>data</code>                                                                                                                            |
| み                        | のを                                                                                                                                                                                                       |
| <code>na.action</code>   | NA の <code>?na.action</code>                                                                                                                                                                             |
|                          | フィッティングの。 は "qr" または "model.frame" は、モデルをフィットせずにモデルフレームをします。これは <code>model=TRUE</code> をするのと同じです                                                                                                        |
| モデル                      | フィットしたオブジェクトにモデルフレームをするかどうか                                                                                                                                                                              |
| バツ                       | したオブジェクトにモデルをするかどうか                                                                                                                                                                                      |
| <code>y</code>           | したオブジェクトにモデルをするかどうか                                                                                                                                                                                      |
| <code>qr</code>          | フィットしたオブジェクトにQRをするかどうか                                                                                                                                                                                   |
| <code>singular.ok</code> | フィットをにするかどうか、をつモデルこの、のサブセットはに NA にされます                                                                                                                                                                   |
|                          | モデルののにされるコントラストのリスト。 <code>?model.matrix.default contrasts.arg</code> をしてください。コントラストは、 <code>options() contrasts</code> をまたはある <code>contrast</code> をりてることによってすることもでき <code>options() ?contrasts</code> |
| オフセット                    | モデルのなのコンポーネントをするためにされます。のとしてすることもできます。 See <code>?model.offset</code>                                                                                                                                    |
| ...                      | フィッティング <code>lm.fit()</code> または <code>lm.wfit()</code> にされるの                                                                                                                                           |

# Examples

## mtcars データセットの

ビルトインのmtcars データフレームには、ガロンあたりのマイル、などの32のにするがまれていますについては、 `help(mtcars)` し `help(mtcars)` 。

`mpg` と `wt` のにがある、のようにこれらのをプロットすることがあります。

```
plot(mpg ~ wt, data = mtcars, col=2)
```

プロットはのをしています。をしてモデルのをめるは、 `lm` をします。

```
fit <- lm(mpg ~ wt, data = mtcars)
```

ここでの~は「する」をするので、 `mpg ~ wt` というのは `wt` でされる `mpg` をしていることをします。をするものはのとおりです。

```
summary(fit)
```

## それはをえる

```
Call:
lm(formula = mpg ~ wt, data = mtcars)

Residuals:
 Min 1Q Median 3Q Max
-4.5432 -2.3647 -0.1252 1.4096 6.8727

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 37.2851 1.8776 19.858 < 2e-16 ***
wt -5.3445 0.5591 -9.559 1.29e-10 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.046 on 30 degrees of freedom
Multiple R-squared: 0.7528, Adjusted R-squared: 0.7446
F-statistic: 91.38 on 1 and 30 DF, p-value: 1.294e-10
```

## これはのをします

- `mpg` のをする `wt` と `y` のは  $37.2851 + (-5.3445) * wt$
- とはおそらくによるものではないことをするの `p`
- $R^2$  やされた  $R^2$  などのフィットのなは、 `mpg` ののどれがモデルによってされるかをします

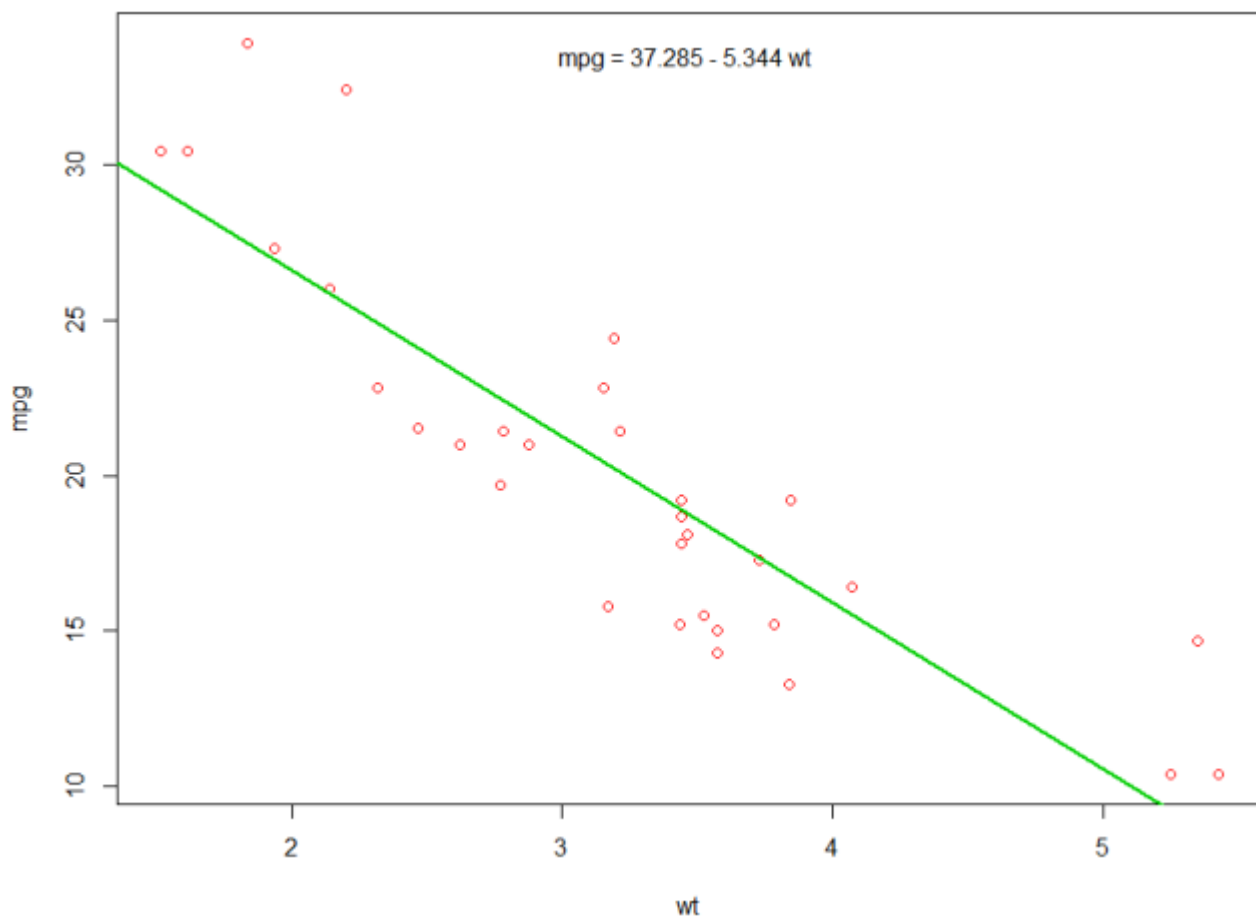
された `mpg` をするのプロットにラインをすることができます

```
abline(fit,col=3,lwd=2)
```

そのプロットにをすることもできます。まず、`coef`を`coef`ます。その、`paste0`をしてをなと`+/-`で  
`paste0`、をします。に、`mtext`をしてプロットにし`mtext`。

```
bs <- round(coef(fit), 3)
lmlab <- paste0("mpg = ", bs[1],
 ifelse(sign(bs[2])==1, " + ", " - "), abs(bs[2]), " wt ")
mtext(lmlab, 3, line=-2)
```

はのとおりで。



をプロットするベース

`mtcars`をけると、ここでは、にしたのプロットをするながあります。

にモデルをフィットさせ、

```
fit <- lm(mpg ~ wt, data = mtcars)
```

に、のある2つのをプロットし、ドメインにをします。

```
plot(mtcars$wt,mtcars$mpg,pch=18, xlab = 'wt',ylab = 'mpg')
lines(c(min(mtcars$wt),max(mtcars$wt)),
 as.numeric(predict(fit, data.frame(wt=c(min(mtcars$wt),max(mtcars$wt))))))
```

ほぼそこにのステップは、プロット、、rsquare、およびをすることです。これは、vectorをしてわれます。

```
rp = vector('expression',3)
rp[1] = substitute(expression(italic(y) == MYOTHERVALUE3 + MYOTHERVALUE4 %% x),
 list(MYOTHERVALUE3 = format(fit$coefficients[1], digits = 2),
 MYOTHERVALUE4 = format(fit$coefficients[2], digits = 2))) [2]
rp[2] = substitute(expression(italic(R)^2 == MYVALUE),
 list(MYVALUE = format(summary(fit)$adj.r.squared,dig=3))) [2]
rp[3] = substitute(expression(Pearson-R == MYOTHERVALUE2),
 list(MYOTHERVALUE2 = format(cor(mtcars$wt,mtcars$mpg), digits = 2))) [2]

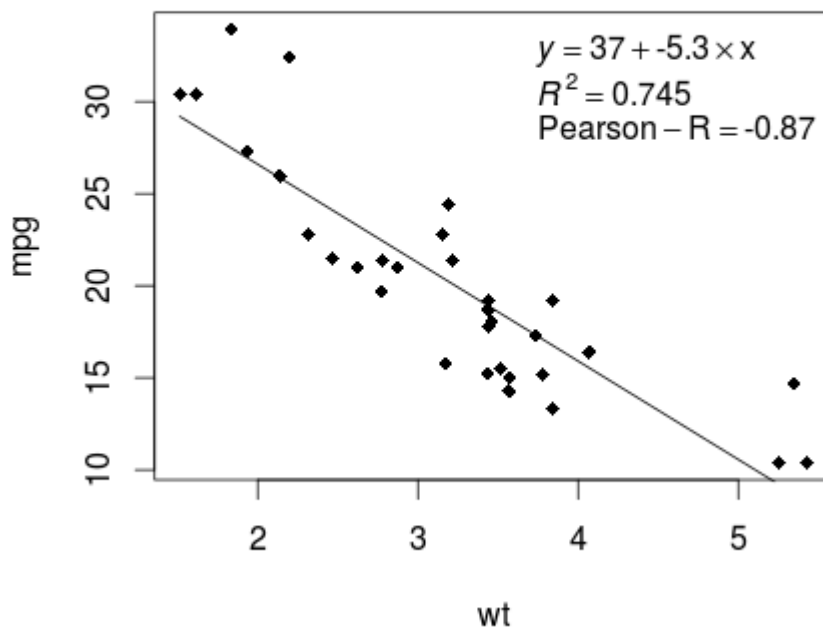
legend("topright", legend = rp, bty = 'n')
```

ベクトルをさせることによって、RMSEなどのパラメータをできます。10のがだとしてみてください。ベクトルはのようになります。

```
rp = vector('expression',10)
```

`r[1] .... r[10]` をするがあります `r[10]`

はのとおりで。



には、モデルがのものよりもいくつかのデータやにきなみをけることをみます。これは、モデルのにデータのみをすることになり。にしてなみをするシナリオには、に2あります。

- みなるのなるレベルをします。たとえば、がエリアからのであるデータをする、はのにします。データのをうに、をしてをするとです。 [ソース](#)

- ・ サンプルングウェイト Inverse Probability Weights - IPWデータがされたとはなるにしてされたをするための。なるおよびのをするデザインは、においてである。をつデータをうにです。  
。 [ソース](#)

`lm()` はなみけをいます。サンプルングウェイトの、`survey` パッケージをしてオブジェクトをし、`svyglm()` をします。デフォルトでは、`survey` パッケージはサンプルングをします。`lm()` および `svyglm()` と `gaussian()` どちらも、をすることにより、をくため、すべてが、じがされますらはどのようにされるかがなります。

## テストデータ

```
data <- structure(list(lexptot = c(9.1595012302023, 9.86330744180814,
8.92372556833205, 8.58202430280175, 10.1133857229336), progwillm = c(1L,
1L, 1L, 1L, 0L), sexhead = c(1L, 1L, 0L, 1L, 1L), agehead = c(79L,
43L, 52L, 48L, 35L), weight = c(1.04273509979248, 1.01139605045319,
1.01139605045319, 1.01139605045319, 0.76305216550827)), .Names = c("lexptot",
"progwillm", "sexhead", "agehead", "weight"), class = c("tbl_df",
"tbl", "data.frame"), row.names = c(NA, -5L))
```

## アナリティックウェイト

```
lm.analytic <- lm(lexptot ~ progwillm + sexhead + agehead,
 data = data, weight = weight)
summary(lm.analytic)
```

```
Call:
lm(formula = lexptot ~ progwillm + sexhead + agehead, data = data,
 weights = weight)
```

```
Weighted Residuals:
 1 2 3 4 5
9.249e-02 5.823e-01 0.000e+00 -6.762e-01 -1.527e-16
```

```
Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.016054 1.744293 5.742 0.110
progwillm -0.781204 1.344974 -0.581 0.665
sexhead 0.306742 1.040625 0.295 0.818
agehead -0.005983 0.032024 -0.187 0.882
```

```
Residual standard error: 0.8971 on 1 degrees of freedom
Multiple R-squared: 0.467, Adjusted R-squared: -1.132
F-statistic: 0.2921 on 3 and 1 DF, p-value: 0.8386
```

## サンプルングウェイト IPW

```
library(survey)
data$X <- 1:nrow(data) # Create unique id

Build survey design object with unique id, ipw, and data.frame
des1 <- svydesign(id = ~X, weights = ~weight, data = data)
```

```
Run glm with survey design object
prog.lm <- svyglm(lexptot ~ progwillm + sexhead + agehead, design=des1)
```

Call:

```
svyglm(formula = lexptot ~ progwillm + sexhead + agehead, design = des1)
```

Survey design:

```
svydesign(id = ~X, weights = ~weight, data = data)
```

Coefficients:

|             | Estimate  | Std. Error | t value | Pr(> t ) |
|-------------|-----------|------------|---------|----------|
| (Intercept) | 10.016054 | 0.183942   | 54.452  | 0.0117 * |
| progwillm   | -0.781204 | 0.640372   | -1.220  | 0.4371   |
| sexhead     | 0.306742  | 0.397089   | 0.772   | 0.5813   |
| agehead     | -0.005983 | 0.014747   | -0.406  | 0.7546   |

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.2078647)

Number of Fisher Scoring iterations: 2

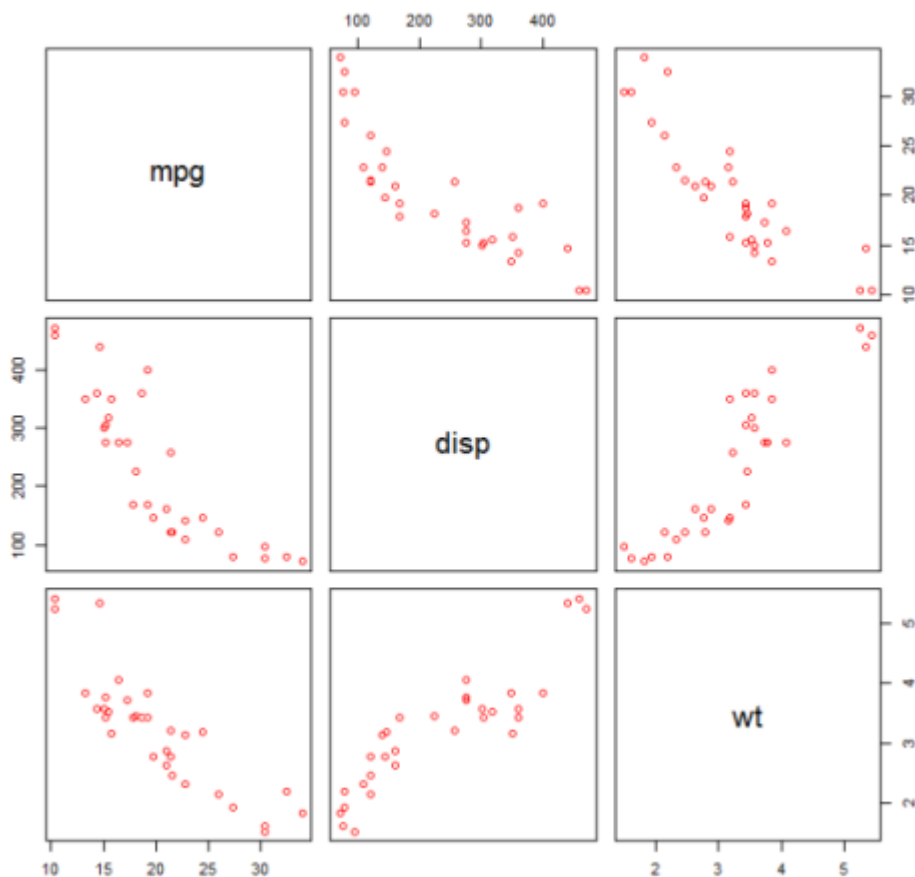
## によるのチェック

によっては、でするときに、データのをチェックするがあります。これをう1つの、モデルにさせ、モデルよりもデータにするかどうかをチェックすることです。がにであるとえられるため、2のモデルにすることをすなようなのがある。

`mtcars` データセットにモデルをさせましょう。モデルについては、[mtcarsデータセットのを](#)してください。

まず `mpg` Miles / gallon、 `disp` Displacementcu.in.、 `wt` Weight1000 lbsのをします。 `mpg`と `disp`のはにえます。

```
plot(mtcars[,c("mpg", "disp", "wt")])
```



は`disp`がでないことをす。

```
fit0 = lm(mpg ~ wt+disp, mtcars)
summary(fit0)

Coefficients:
Estimate Std. Error t value Pr(>|t|)
#(Intercept) 34.96055 2.16454 16.151 4.91e-16 ***
#wt -3.35082 1.16413 -2.878 0.00743 **
#disp -0.01773 0.00919 -1.929 0.06362 .
#---
#Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#Residual standard error: 2.917 on 29 degrees of freedom
#Multiple R-squared: 0.7809, Adjusted R-squared: 0.7658
```

に、2モデルのをるために、`I(disp^2)`をしました。しいモデルは $R^2$ をとよりよくえ、すべてのがです。

```
fit1 = lm(mpg ~ wt+disp+I(disp^2), mtcars)
summary(fit1)

Coefficients:
Estimate Std. Error t value Pr(>|t|)
#(Intercept) 41.4019837 2.4266906 17.061 2.5e-16 ***
#wt -3.4179165 0.9545642 -3.581 0.001278 **
#disp -0.0823950 0.0182460 -4.516 0.000104 ***
#I(disp^2) 0.0001277 0.0000328 3.892 0.000561 ***
#---
```



```
#Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#Residual standard error: 2.391 on 28 degrees of freedom
#Multiple R-squared: 0.8578, Adjusted R-squared: 0.8426
```

3つのがあるので、てはめられたモデルは、

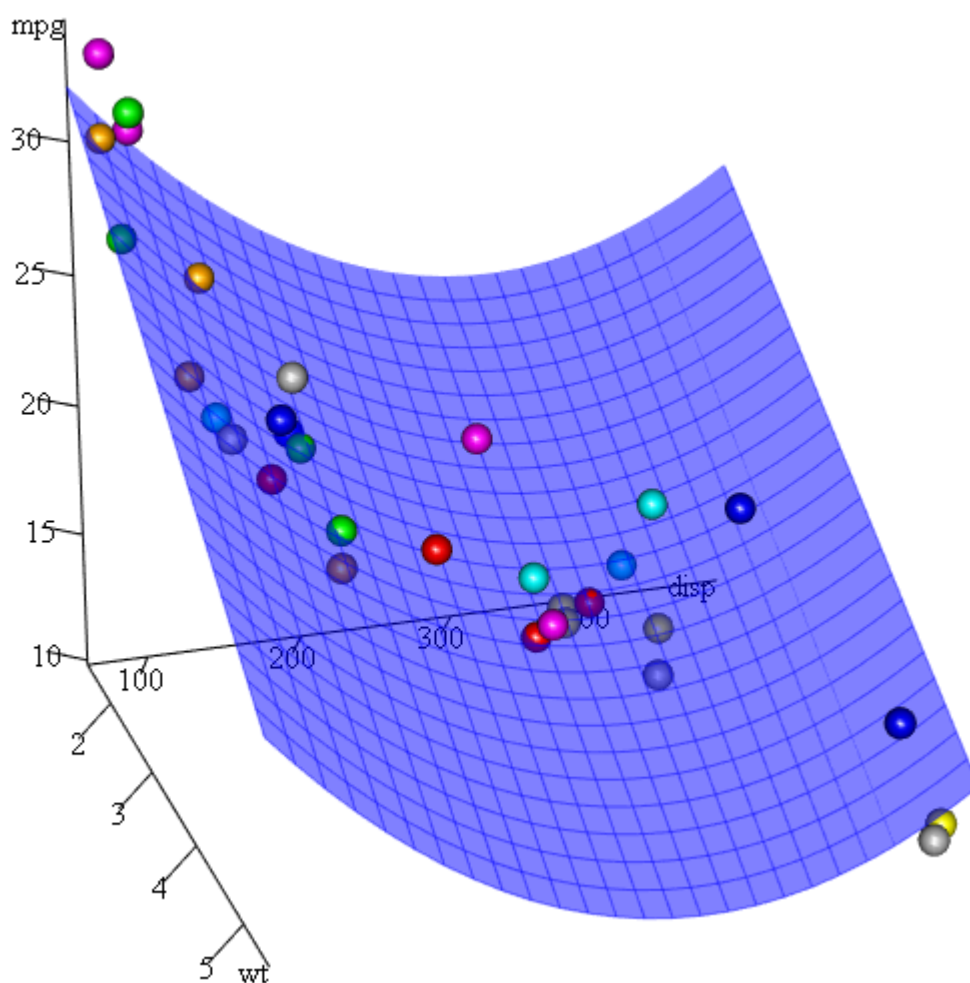
```
mpg = 41.4020-3.4179*wt-0.0824*disp+0.0001277*disp^2
```

をするもう1つのは、パラメータ `raw=TRUE` `poly` をするです。そうでないはをしますは `help(poly)` を。じがられます

```
summary(lm(mpg ~ wt+poly(disp, 2, raw=TRUE),mtcars))
```

に、されたのプロットをするがあるはどうなりますかさて、`R 3D` プロットをするためのくのオプションがあります。ここでは、`p3d` パッケージの `Fit3d` をします。

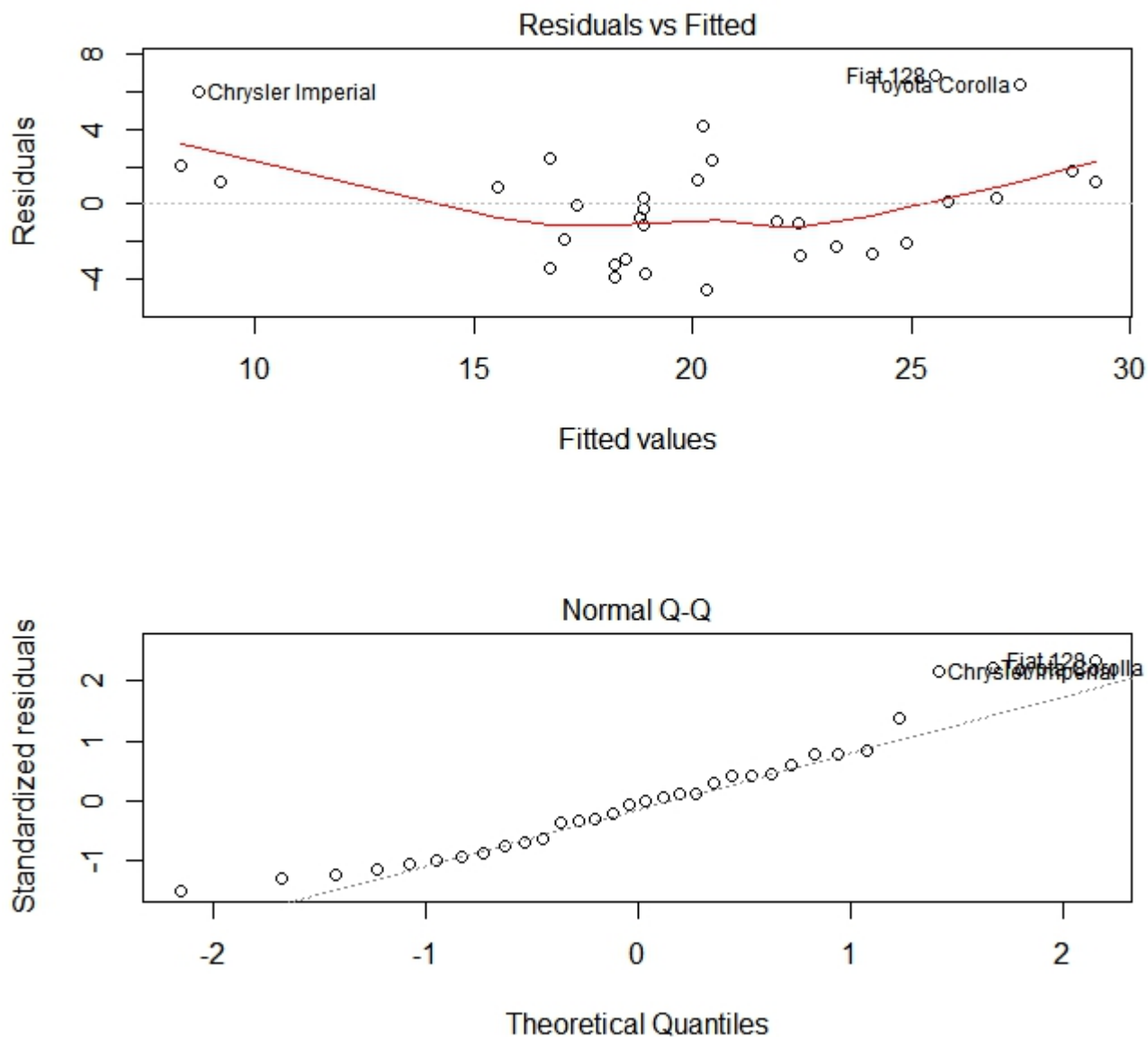
```
library(p3d)
Init3d(family="serif", cex = 1)
Plot3d(mpg ~ disp+wt, mtcars)
Axes3d()
Fit3d(fit1)
```



モデルをした、をし、モデルがであり、のデータとうまくしているかどうかをすることがです。

これは、プロットおよびのプロットをべることによってうことができます。

```
fit the model
fit <- lm(mpg ~ wt, data = mtcars)
#
par(mfrow=c(2,1))
plot model object
plot(fit, which =1:2)
```



これらのプロットは、モデルをするにわた2つのをチェックします。

1. このは`mpg` のは、のこのは`wt` によってえられます。このはっているとしています。したがって、は、プレディクタのすべてのにをしてをにしなければなりません。この、はでであり、であるがあり、にのがあることがわかります。
2. のされたは、そののりにしていること。したがって、はするがあります。データの、のQQプロットのはにあるか、またはにくなければなりません。ここにはいくらかのスキューがあ

ります。

## 'predict'をする

モデルがされると、しいデータでテストするながpredictされます。このでは、mtcarsビルトインデータセットをして、にしてガロンあたりのマイルをさせます。

```
my_md1 <- lm(mpg ~ disp, data=mtcars)
my_md1

Call:
lm(formula = mpg ~ disp, data = mtcars)

Coefficients:
(Intercept) disp
 29.59985 -0.04122
```

がディスプレイメントきのしいデータソースをとっていれば、ガロンあたりのマイルをることができました。

```
set.seed(1234)
newdata <- sample(mtcars$disp, 5)
newdata
[1] 258.0 71.1 75.7 145.0 400.0

newdf <- data.frame(disp=newdata)
predict(my_md1, newdf)
 1 2 3 4 5
18.96635 26.66946 26.47987 23.62366 13.11381
```

このプロセスのものは、のデータとじをつしいデータフレームをすることです。この、のデータにはdispというラベルのがあり、じのしいデータをずびすことになりました。

あぶない

いくつかのなとしをてみましょう

1. しいオブジェクトにdata.frameをししないでください

```
predict(my_md1, newdata)
Error in eval(predvars, data, env) :
 numeric 'envir' arg not of length one
```

2. しいデータフレームでじをししない

```
newdf2 <- data.frame(newdata)
predict(my_md1, newdf2)
Error in eval(expr, envir, enclos) : object 'disp' not found
```

さ

のをするには、しいデータののyがです。このでは、 `newdf` は 'mpg' と 'disp' のがです。

```
newdf <- data.frame(mpg=mtcars$mpg[1:10], disp=mtcars$disp[1:10])
mpg disp
1 21.0 160.0
2 21.0 160.0
3 22.8 108.0
4 21.4 258.0
5 18.7 360.0
6 18.1 225.0
7 14.3 360.0
8 24.4 146.7
9 22.8 140.8
10 19.2 167.6

p <- predict(my_mdl, newdf)

#root mean square error
sqrt(mean((p - newdf$mpg)^2, na.rm=TRUE))
[1] 2.325148
```

オンラインでモデルをむ <https://riptutorial.com/ja/r/topic/801/モデル-->

き

NLPは、によってされたテキストからをりすことにをてたコンピュータサイエンスのです。

## Examples

をする

へのもなアプローチそしてこれまでににもにされているは、センテンスをトークンにすることです。例えば、はでなをち、けったりしたりするトークンにながありますけられたまたはバイト。がされると、トークンのはされます。のバグのモデルとしてられているへのこのアプローチ。

はであり、トークンにしてみがりてられる。1のでは、Rパッケージ`tm`いてコーパスコーパス からタームをする。

```
require(tm)
doc1 <- "drugs hospitals doctors"
doc2 <- "smog pollution environment"
doc3 <- "doctors hospitals healthcare"
doc4 <- "pollution environment water"
corpus <- c(doc1, doc2, doc3, doc4)
tm_corpus <- Corpus(VectorSource(corpus))
```

このでは、`Corpus`と`VectorSource` 2つのをつパッケージ`tm`によってされた`Corpus`クラスの`Corpus`をしました。これはベクトルから`VectorSource`オブジェクトをします。オブジェクト`tm_corpus`は、をすためののメタデータとオプションのメタデータのリストです。

```
str(tm_corpus)
List of 4
 $ 1:List of 2
 ..$ content: chr "drugs hospitals doctors"
 ..$ meta :List of 7
 $ author : chr(0)
 $ timestamp: POSIXlt[1:1], format: "2017-06-03 00:31:34"
 $ description : chr(0)
 $ heading : chr(0)
 $ id : chr "1"
 $ language : chr "en"
 $ origin : chr(0)
 - attr(*, "class")= chr "TextDocumentMeta"
 ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
 [truncated]
```

たちがっていたら`Corpus`、たちはにまれるトークンにむことができ`Corpus`なのをします。これをうには、`tmtm_map`をします。`tm_map`は、`applytm_map`とに、にをしてコーパスのをします。

```
tm_corpus <- tm_map(tm_corpus, tolower)
tm_corpus <- tm_map(tm_corpus, removeWords, stopwords("english"))
tm_corpus <- tm_map(tm_corpus, removeNumbers)
tm_corpus <- tm_map(tm_corpus, PlainTextDocument)
tm_corpus <- tm_map(tm_corpus, stemDocument, language="english")
tm_corpus <- tm_map(tm_corpus, stripWhitespace)
tm_corpus <- tm_map(tm_corpus, PlainTextDocument)
```

これらのにいて、に

```
tdm <- TermDocumentMatrix(tm_corpus)
```

それは

```
<<TermDocumentMatrix (terms: 8, documents: 4)>>
Non-/sparse entries: 12/20
Sparsity : 62%
Maximal term length: 9
Weighting : term frequency (tf)
```

それをにすることできる

```
as.matrix(tdm)
```

|           | Docs         |              |              |              |
|-----------|--------------|--------------|--------------|--------------|
| Terms     | character(0) | character(0) | character(0) | character(0) |
| doctor    | 1            | 0            | 1            | 0            |
| drug      | 1            | 0            | 0            | 0            |
| environ   | 0            | 1            | 0            | 1            |
| healthcar | 0            | 0            | 1            | 0            |
| hospit    | 1            | 0            | 1            | 0            |
| pollut    | 0            | 1            | 0            | 1            |
| smog      | 0            | 1            | 0            | 0            |
| water     | 0            | 0            | 0            | 1            |

は、4つの、4つのコラムで、づいたように `environment` への `environ` トークンのをします。

のでは、ペアのトークン/ドキュメントになつまり、ドキュメントにされるトークンのインスタンスをみけしました。

オンラインでをむ <https://riptutorial.com/ja/r/topic/10119/>

## 124:

き

はデータをする

## Examples

の

フードのでは、は2のなのベクトルです。ベクトルとに、は1つのデータクラスしかてません。の  
ように、`matrix`をしてをすることができます。

```
matrix(data = 1:6, nrow = 2, ncol = 3)
[,1] [,2] [,3]
[1,] 1 3 5
[2,] 2 4 6
```

このとおり、これは、1から6までの23のすべてののをします。`data`パラメータは、のベクトルを  
とり、`nrow`マトリックスのをし、`ncol`のをします。により、マトリックスはカラムによってされ  
る。デフォルトは、にすように`byrow`パラメータでできます。

```
matrix(data = 1:6, nrow = 2, ncol = 3, byrow = TRUE)
[,1] [,2] [,3]
[1,] 1 2 3
[2,] 4 5 6
```

はであるはありません。どのベクトルもにできます。えば

```
matrix(data = c(TRUE, TRUE, TRUE, FALSE, FALSE, FALSE), nrow = 3, ncol = 2)
[,1] [,2]
[1,] TRUE FALSE
[2,] TRUE FALSE
[3,] TRUE FALSE
matrix(data = c("a", "b", "c", "d", "e", "f"), nrow = 3, ncol = 2)
[,1] [,2]
[1,] "a" "d"
[2,] "b" "e"
[3,] "c" "f"
```

ベクトルとに、はとしてされ、でびされます。のとははをけることができます。あなたは  
`rownames`と`colnames`をもってこれらをすることができます。にすように、とははをたず、`NULL`でされ  
`NULL`。ただし、をりてることができます。

```
mat1 <- matrix(data = 1:6, nrow = 2, ncol = 3, byrow = TRUE)
rownames(mat1)
NULL
```

```
colnames(mat1)
NULL
rownames(mat1) <- c("Row 1", "Row 2")
colnames(mat1) <- c("Col 1", "Col 2", "Col 3")
mat1
Col 1 Col 2 Col 3
Row 1 1 2 3
Row 2 4 5 6
```

ベクトルとに、には1つのデータしかないことにすることができます。のデータをつをしようとする  
と、データはのデータクラスににされます。

`class`、`is`、および`as is`、クラス1のベクトルでされたのとして、データをチェックしするため  
にすることができます。

```
class(mat1)
[1] "matrix"
is.matrix(mat1)
[1] TRUE
as.vector(mat1)
[1] 1 4 2 5 3 6
```

オンラインでをむ <https://riptutorial.com/ja/r/topic/9019/>



## 125: のデータをプレーンテキストファイルCSV、TSVなどでみきする

- ファイル、ヘッダー= TRUE、sep = "、"、 = "" "、dec = "。"、りつぶし=、comment.char = "、 ...
- ファイル、ヘッダー= TRUE、sep = ";"、 もり= "" "、dec = "、"、りつぶし=、comment.char = " "、 ...
- read\_csvfile、col\_names = TRUE、col\_types = NULL、ロケール= default\_locale、na = c ""、NA "、comment = " "、trim\_ws = TRUE、skip = 0、n\_max = -1、progress = interactive
- データ= fread、sep = "auto"、sep2 = "auto"、nrows = -1L、header = "auto"、na.strings = "NA"、stringsAsFactors = FALSE、verbose = getOption "datatable デフォルト "integer64" dec = ifsep= "0";デフォルト "integer64" dec = ifsep= " "" "" ""、"、"、"、col.names、check.names = FALSE、エンコーディング=" "、strip.white = TRUE、showProgress = getOption "datatable.showProgress "、デフォルトTRUEデータ。テーブル= getOption "datatable.fread.datatable"デフォルトTRUE

### パラメーター

| パラメータ        |                                          |
|--------------|------------------------------------------|
| ファイル         | みむCSVファイルの                               |
| ヘッダ          | .csvファイルにのヘッダーがまれていますか                   |
| セップ          | のセルをる                                    |
| もり           | characterをするための                          |
| dec          | としてされる                                   |
| める           | logicalTRUEの、のはのフィールドでめられます。             |
| comment.char | charactercsvファイルのコメントとしてされる。こののにあるはされます。 |
| ...          | read.tableにされるな                          |

プレーンテキストにエクスポートすると、いのためにクラスのようなデータにエンコードされたのがになることにしてください。そのようなをとしないケースでは、[.RData](#)や[Feather](#)のようなフォーマットがよりかもしれません。

のタイプのファイルのは、とからリンクされたのいくつかのトピックでりげられています。

## Examples

.csv ファイルのインポート

### ベースRをしたインポート

コンマリファイルCSVは、`read.table`をラップするが、`sep = ","`をしてカンマにりをする  
`read.csv`をしてインポートできます。

```
get the file path of a CSV included in R's utils package
csv_path <- system.file("misc", "exDIF.csv", package = "utils")

path will vary based on installation location
csv_path
[1] "/Library/Frameworks/R.framework/Resources/library/utils/misc/exDIF.csv"

df <- read.csv(csv_path)

df
Var1 Var2
1 2.70 A
2 3.14 B
3 10.00 A
4 -7.00 A
```

ユーザーフレンドリーなオプション`file.choose`は、ディレクトリをブラウズすることができます

```
df <- read.csv(file.choose())
```

### ノート

- `read.table`とはなり、`read.csv`デフォルトは`header = TRUE`で、のがとしてされます。
- これらのすべてののは、`as.is = TRUE`または`stringsAsFactors = FALSE`、を`factor`クラスにデフォルトでし`stringsAsFactors = FALSE`。
- `read.csv2`バリエーションのデフォルトは`sep = ";"`コンマがとしてされ、セミコロンがフィールドりとしてされるのデータにする`dec = ","`

### パッケージをしたインポート

`readr`パッケージの`read_csv`は、よりなパフォーマンス、ファイルのバー、およびの`read.csv`よりもなデフォルトオプション `stringsAsFactors = FALSE`などを `stringsAsFactors = FALSE`。

```
library(readr)

df <- read_csv(csv_path)

df
A tibble: 4 x 2
Var1 Var2
<dbl> <chr>
1 2.70 A
2 3.14 B
3 10.00 A
4 -7.00 A
```

## data.table をったインポート

`data.table` パッケージは `fread` しています。 `read.table` とていますが、 `fread` は、ファイルのりをにしながら、よりでがあります。

```
get the file path of a CSV included in R's utils package
csv_path <- system.file("misc", "exDIF.csv", package = "utils")

path will vary based on R installation location
csv_path
[1] "/Library/Frameworks/R.framework/Resources/library/utils/misc/exDIF.csv"

dt <- fread(csv_path)

dt
Var1 Var2
1: 2.70 A
2: 3.14 B
3: 10.00 A
4: -7.00 A
```

`input` はをすです。

- ファイル "filename.csv" 、
- ファイルにするシェルコマンド えば "grep 'word' filename" 、または
- そのもの えば、 "input1, input2\n A, B \n C, D" 。

`fread` は `data.table` の `[]` のにした `data.frame` クラスをする `data.table` クラスのオブジェクトをします。の `data.frame` をすには、 `data.table` パラメータを `FALSE` し `FALSE` 。

```
df <- fread(csv_path, data.table = FALSE)

class(df)
[1] "data.frame"

df
Var1 Var2
1 2.70 A
2 3.14 B
3 10.00 A
4 -7.00 A
```

## ノート

- `fread`は`read.table`と同じオプションがありません。しているの1つは`na.comment`であり、ソースファイルに`#`がまれているとなにつながるがあります。
- `fread`のみを`"`のために`quote`パラメータ。
- `fread`は、のをするためにわずかな5しかしません。

### .tsv ファイルをとしてインポートするなR

くのは、ファイルへのパスをとときに`file.path`をしません。しかし、Windows、Mac、Linuxマシンでしているは、は`paste`わりにパスをすることをおめします。

```
FilePath <- file.path(AVariableWithFullProjectPath, "SomeSubfolder", "SomeFileName.txt.gz")

Data <- as.matrix(read.table(FilePath, header=FALSE, sep = "\t"))
```

に、これはほとんどのにとってです。

のきさがにきいので、をみむにメモリりてのをするがあります。これはをでみむことをします。

のをえてみましょう。この、`FilePath`は8970 8970ファイルがまれ、79のセルに0のがまれています。

```
system.time(expr=Data<-as.matrix(read.table(file=FilePath,header=FALSE,sep=" ")))
```

`system.time`はファイルをむために267かかるとべています。

```
 user system elapsed
265.563 1.949 267.563
```

に、このファイルは1ずつみることができますが、

```
FilePath <- "SomeFile"
connection<- gzfile(FilePath,open="r")
TableList <- list()
Counter <- 1
system.time(expr= while (length(Vector<-as.matrix(scan(file=connection, sep=" ", nlines=1,
quiet=TRUE))) > 0) {
 TableList[[Counter]]<-Vector
 Counter<-Counter+1
})

 user system elapsed
165.976 0.060 165.941
close(connection)
system.time(expr=(Data <- do.call(rbind,TableList)))

 user system elapsed
0.477 0.088 0.565
```

また、`read.matrix`メソッドをした`futile.matrix`パッケージもあります。コードは、1でしたのと

じことがらかになります。

**.csv** ファイルのエクスポート

---

## ベースRをしたエクスポート

`write.csv()` をしてデータをCSVファイルにきむことができます。

```
write.csv(mtcars, "mtcars.csv")
```

にされたパラメータには、 `row.names = FALSE` および `na = ""` されます。

---

## パッケージをしたエクスポート

`readr::write_csv` は `readr::write_csv` よりもはるかにで、 `write.csv` れません。

```
library(readr)

write_csv(mtcars, "mtcars.csv")
```

の **csv** ファイルをインポートする

```
files = list.files(pattern="*.csv")
data_list = lapply(files, read.table, header = TRUE)
```

これにより、すべてのファイルがみまれ、リストにされます。その、すべての `data.frame` がしをつ、それらを1つのきな `data.frame` にすることができます

```
df <- do.call(rbind, data_list)
```

ファイルのインポート

のファイルは、のような、のりでられていないされているテキストファイルです,か,むしろ をつ。データは、でめられます。

| Column1 | Column2 | Column3           | Column4  | Column5 |
|---------|---------|-------------------|----------|---------|
| 1647    | pi      | 'important'       | 3.141596 | 28318   |
| 1731    | euler   | 'quite important' | 2.718285 | 43656   |
| 1979    | answer  | 'The Answer.'     | 42       | 42      |

このデータテーブルがディレクトリのローカルファイル `constants.txt` にするとしましょう。

## ベースRでのインポート

```
df <- read.fwf('constants.txt', widths = c(8,10,18,7,8), header = FALSE, skip = 1)
```

```
df
#> V1 V2 V3 V4 V5
#> 1 1647 pi 'important' 3.14159 6.28318
#> 2 1731 euler 'quite important' 2.71828 5.43656
#> 3 1979 answer 'The Answer.' 42 42.0000
```

- タイトルは与えられていません Column4Column5
- `widths` パラメータは、のを使います。
- されていないヘッダーは、`read.fwf()` みることはできません。

---

## readrでのインポート

```
library(readr)

df <- read_fwf('constants.txt',
 fwf_cols(Year = 8, Name = 10, Importance = 18, Value = 7, Doubled = 8),
 skip = 1)

df
#> # A tibble: 3 x 5
#> Year Name Importance Value Doubled
#> <int> <chr> <chr> <dbl> <dbl>
#> 1 1647 pi 'important' 3.14159 6.28318
#> 2 1731 euler 'quite important' 2.71828 5.43656
#> 3 1979 answer 'The Answer.' 42.00000 42.00000
```

- `readr`の`fwf_*`ヘルパーは、のを使います。 `fwf_empty`
- `readr`はRよりい
- タイトルをデータファイルからインポートすることはできません

オンラインでのデータをプレーンテキストファイルCSV、TSVなどでみきするをむ

<https://riptutorial.com/ja/r/topic/481/>のデータをプレーンテキストファイル-csv-tsvなど-でみきする

## 126:

1. `x = character`、レベル、ラベル=レベル、`= NA`、`is.ordered`、`nmax = NA`
2. `?factor`する?`factor`オンラインで[ドキュメントをしてください](#)。

クラス `factor` つオブジェクトは、のセットをつベクトルです。

1. に `integer` ベクトルとしてされます。
2. これは、のを `levels` をします。
3. そのクラスは `factor` としてされます

するために、のから1000のベクトルをしましょう。

```
set.seed(1)
Color <- sample(x = c("Red", "Blue", "Green", "Yellow"),
 size = 1000,
 replace = TRUE)
Color <- factor(Color)
```

の `Color` をることができます

```
1. It is stored internally as an `integer` vector
typeof(Color)
```

```
[1] "integer"
```

```
2. It maintains a `levels` attribute the shows the character representation of the values.
3. Its class is stored as `factor`
attributes(Color)
```

```
$levels
[1] "Blue" "Green" "Red" "Yellow"

$class
[1] "factor"
```

ファクタオブジェクトのなは、データストレージのです。はよりもになメモリがなくてみます。このようなは、くのコンピュータがのマシンよりもはるかにられたリソースをとっていたににましものでしたののにあるのなについては、[stringsAsFactors](#) [Unauthorized Biography](#)をしてください。メモリのいは、`Color`オブジェクトでもることができます。このように、`Color`をととしてするには、`Factor`オブジェクトの1.7のメモリがです。

```
Amount of memory required to store Color as a factor.
object.size(Color)
```

```
4624 bytes
```

```
##* Amount of memory required to store Color as a character
object.size(as.character(Color))
```

```
8232 bytes
```

## をレベルにマッピングする

ファクタのなはオブジェクトをとしますが、がするはレベルです。えば、

```
head(Color)
```

```
[1] Blue Blue Green Yellow Red Yellow
Levels: Blue Green Red Yellow
```

のがです

```
head(as.numeric(Color))
```

```
[1] 1 1 2 4 3 4
```

Rがをにマッチさせるのはのとおりです。

```
head(levels(Color)[as.numeric(Color)])
```

```
[1] "Blue" "Blue" "Green" "Yellow" "Red" "Yellow"
```

これらのを

```
head(Color)
```

```
[1] Blue Blue Green Yellow Red Yellow
Levels: Blue Green Red Yellow
```

## なの

2007に、Rは、ベクトルのメモリをしたのハッシングをしました [stringsAsFactors Unauthorized Biography](#)。がより1.7くのスペースをとるとした、それはRののバージョンでされたことにしてください。これは、ベクトルのメモリが2007よりにされたことをします。

なRのハッシュやのコンピュータのはるかにきなメモリリソースのおかげで、のをするのメモリのはにさなになりました。Rコミュニティにおけるなは、ほとんどのでのにするベクトルのです。からのシフトのなはのとおりです。



1. および/またはやかにされたデータの
2. ユーザーがキャラクターではなくをとっていることを忘れてしまったときに、ファクターがみどおりにしない

のケースでは、フリーテキストやオープンレスポンスフィールドをとしてするのはありません。レベルごとにのをするパターンはしありません。あるいは、データがくされていない、じカテゴリえば、「」、「」、および「」にするのレベルをすることがある。そのような、くのは、にするに、これらのをとしてすることをむがまったくわれない。

2のケースでは、ユーザーがベクトルをとっていると、のメソッドがしたとおりにしないことがあります。このなは、スクリプトとコードをデバッグするにとをくことがあります。にえば、これはユーザーのちとみなすことができますが、ほとんどのユーザーはをしないでこれらのをにすることができます。

## Examples

のな

は、Rのカテゴリをす1つのです。は、にのベクトルとしてされます。えられたベクトルののは、のレベルとしてられています。デフォルトでは、レベルがユーザーによってされない、Rはベクトルのののセットをし、これらのをでソートし、それらをレベルとしてします。

```
charvar <- rep(c("n", "c"), each = 3)
f <- factor(charvar)
f
levels(f)

> f
[1] n n n c c c
Levels: c n
> levels(f)
[1] "c" "n"
```

レベルのをするは、レベルをですするための1つのオプションがあります。

```
levels(factor(charvar, levels = c("n", "c")))

> levels(factor(charvar, levels = c("n", "c")))
[1] "n" "c"
```

にはくがあります。たとえば、レベルにラベルをけることができます。

```
> f <- factor(charvar, levels=c("n", "c"), labels=c("Newt", "Capybara"))
> f
[1] Newt Newt Newt Capybara Capybara Capybara
Levels: Newt Capybara
```

りてることができるもう1つのプロパティは、がけられているかどうかです。

```
> Weekdays <- factor(c("Monday", "Wednesday", "Thursday", "Tuesday", "Friday", "Sunday",
"Saturday"))
> Weekdays
[1] Monday Wednesday Thursday Tuesday Friday Sunday Saturday
Levels: Friday Monday Saturday Sunday Thursday Tuesday Wednesday
> Weekdays <- factor(Weekdays, levels=c("Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday", "Sunday"), ordered=TRUE)
> Weekdays
[1] Monday Wednesday Thursday Tuesday Friday Sunday Saturday
Levels: Monday < Tuesday < Wednesday < Thursday < Friday < Saturday < Sunday
```

のレベルがされなくなったら、`droplevels()` をして `droplevels()` することができます

```
> Weekend <- subset(Weekdays, Weekdays == "Saturday" | Weekdays == "Sunday")
> Weekend
[1] Sunday Saturday
Levels: Monday < Tuesday < Wednesday < Thursday < Friday < Saturday < Sunday
> Weekend <- droplevels(Weekend)
> Weekend
[1] Sunday Saturday
Levels: Saturday < Sunday
```

## レベルをリストにする

おそらく、カテゴリの1つにおけるなデータのために、レベルをよりないグループにすることがま  
しいがあります。カテゴリのりやがなるにもするがあります。としてする

```
set.seed(1)
colorful <- sample(c("red", "Red", "RED", "blue", "Blue", "BLUE", "green", "gren"),
size = 20,
replace = TRUE)
colorful <- factor(colorful)
```

Rはとをするため、このベクトルのテーブルはのようになります。

```
table(colorful)
```

```
colorful
blue Blue BLUE green gren red Red RED
 3 1 4 2 4 1 3 2
```

ただし、これはデータののをすものではなく、カテゴリは、の3つのタイプににできます。3つ  
のがされています。はなのようにえますが、にははありません。2のはなをしています、でコ  
ストがかかります。3はらかなではありませんが、コンパクトでがいです。

## factor factor\_approach をしたレベルの

```
factor(as.character(colorful),
levels = c("blue", "Blue", "BLUE", "green", "gren", "red", "Red", "RED"),
labels = c("Blue", "Blue", "Blue", "Green", "Green", "Red", "Red", "Red"))
```

```
[1] Green Blue Red Red Blue Red Red Red Blue Red Green Green Green
Blue Red Green
[17] Red Green Green Red
Levels: Blue Blue Blue Green Green Red Red Red
Warning message:
In `levels<`(`*tmp*`, value = if (nl == nL) as.character(labels) else
paste0(labels, :
duplicated levels in factors are deprecated
```

レベルがしていることにしてください。たちはまだ""の3つのカテゴリーをっていますが、それはレベルをまとめるといふをしません。さらに、されたレベルはされなくなりました。これは、このコードがエラーをするがあることをします。

## ifelse ifelse\_approach をしてレベルをする

```
factor(ifelse(colorful %in% c("blue", "Blue", "BLUE"),
 "Blue",
 ifelse(colorful %in% c("green", "gren"),
 "Green",
 "Red")))
```

```
[1] Green Blue Red Red Blue Red Red Red Blue Red Green Green Green
Blue Red Green
[17] Red Green Green Red
Levels: Blue Green Red
```

このコードはのをしますが、ネストされたifelseステートメントをするがあります。このアプローチにはもはありませんが、ネストされたifelseステートメントのはなであり、にわなければなりません。

## ファクターレベルをリストにする list\_approach

あまりでないレベルのは、のがのカテゴリであるリストをすることです。は、のカテゴリにマップするがあるのレベルのベクトルです。これには、しいオブジェクトをりてることなく、ファクタのlevelsにするというがあります。

```
levels(colorful) <-
 list("Blue" = c("blue", "Blue", "BLUE"),
 "Green" = c("green", "gren"),
 "Red" = c("red", "Red", "RED"))
```

```
[1] Green Blue Red Red Blue Red Red Red Blue Red Green Green Green
Blue Red Green
[17] Red Green Green Red
Levels: Blue Green Red
```

## アプローチのベンチマーキング

これらのアプローチのそれぞれをするのにするをにする。スペースのために、このをするコードはされません

```
Unit: microseconds
 expr min lq mean median uq max neval cld
factor 78.725 83.256 93.26023 87.5030 97.131 218.899 100 b
ifelse 104.494 107.609 123.53793 113.4145 128.281 254.580 100 c
list_approach 49.557 52.955 60.50756 54.9370 65.132 138.193 100 a
```

リストアプローチは `ifelse` アプローチの2のさでされます。ただし、にのデータがあるをいて、のは、マイクロまたはミリでされるがあります。そのようなわずかなで、はどのアプローチをするかのをくはありません。わりに、いれた、なアプローチをして、あなたとあなたののがのレビューでするをしてください。

は、1つのベクトルをえるRでカテゴリをするである`x`をしてにすることができる`as.character()`、デフォルト `factor()` と `as.factor()` のにをりてますベクトル、レベル、ラベルなどがあります。レベルは`x`がなるのであるであり、ラベルはされたであるか、ユーザーによってされます。

ファクタがどのようにするかをとして、デフォルト、カスタムレベル、カスタムレベル、およびラベルをむをします。

```
standard
factor(c(1,1,2,2,3,3))
[1] 1 1 2 2 3 3
Levels: 1 2 3
```

ファクタがすることができるなのがベクトルののよりもきいことをユーザがっている、インスタンスがするがあります。このために、たちは `factor()` レベルをりてます。

```
factor(c(1,1,2,2,3,3),
 levels = c(1,2,3,4,5))
[1] 1 1 2 2 3 3
Levels: 1 2 3 4 5
```

スタイルのために、ユーザはレベルにラベルをりてることができる。デフォルトでは、ラベルはレベルのです。ここでは、のなレベルのそれぞれにラベルをりてます。

```
factor(c(1,1,2,2,3,3),
 levels = c(1,2,3,4,5),
 labels = c("Fox", "Dog", "Cow", "Brick", "Dolphin"))
[1] Fox Fox Dog Dog Cow Cow
Levels: Fox Dog Cow Brick Dolphin
```

、は、`==`と`!=`とがじレベルであるにのみすることができます。ののは、がなるレベルをするためにしくえてもする。

```
factor(c(1,1,2,2,3,3), levels = c(1,2,3)) == factor(c(1,1,2,2,3,3), levels = c(1,2,3,4,5))
Error in Ops.factor(factor(c(1, 1, 2, 2, 3, 3), levels = c(1, 2, 3)), :
 level sets of factors are different
```

これは、RHSのなレベルは、Rがなでそれらをするためにについてのなをっていないことをするのであります。

<、<=、>および>=は、けられたにしてのみできます。これらは、としてのをするカテゴリをすることができる。orderedファクタは、ordered = TRUEをfactorにするか、orderedをすることでできます。

```
x <- factor(1:3, labels = c('low', 'medium', 'high'), ordered = TRUE)
print(x)
[1] low medium high
Levels: low < medium < high

y <- ordered(3:1, labels = c('low', 'medium', 'high'))
print(y)
[1] high medium low
Levels: low < medium < high

x < y
[1] TRUE FALSE FALSE
```

については、[Factorのドキュメント](#)をしてください。

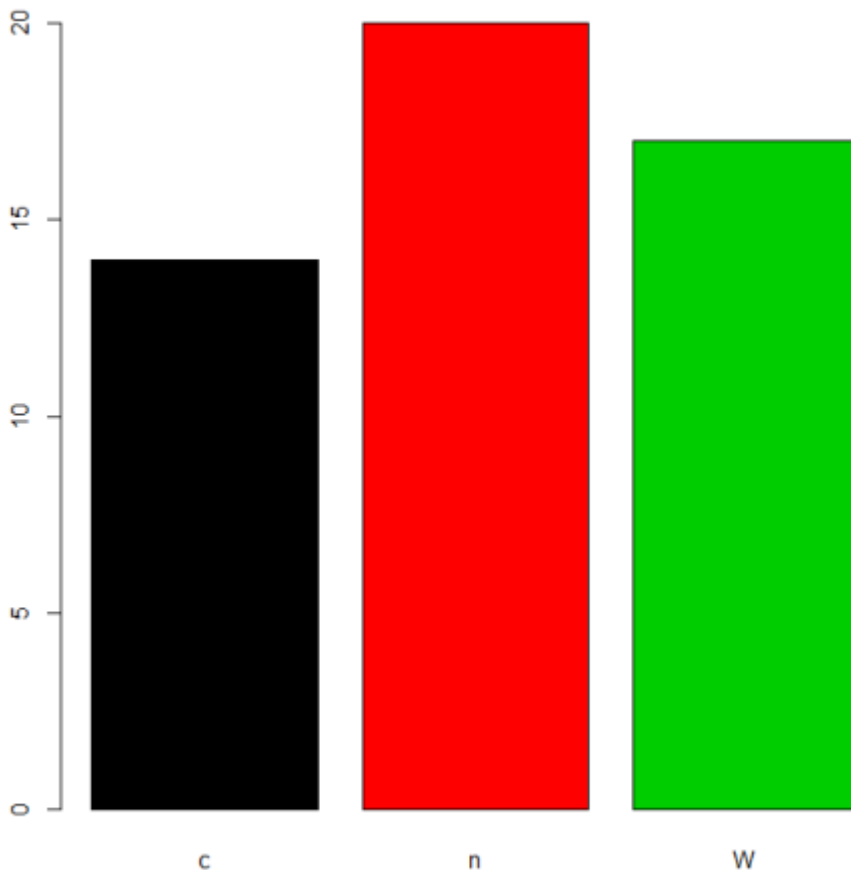
およびべえの

ファクタがデフォルトでされるとき、levelsはにされるas.characterによってas.characterされ、アルファベットにべられます。

```
charvar <- rep(c("W", "n", "c"), times=c(17,20,14))
f <- factor(charvar)
levels(f)
[1] "c" "n" "W"
```

によっては、levelsのデフォルトのアルファベット/のいがけられます。たとえば、をplotするだけの、のようになります。

```
plot(f,col=1:length(levels(f)))
```



しかし、`levels`なるがなは、`levels`または`labels`パラメータでこれをするがありますここでの「」のはけられたとはなります。にじてそのタスクをするためのくがあります。

## 1.をする

であれば、`levels`パラメータをして、なでをすることができます。

```
ff <- factor(charvar, levels = c("n", "W", "c"))
levels(ff)
[1] "n" "W" "c"

gg <- factor(charvar, levels = c("W", "c", "n"))
levels(gg)
[1] "W" "c" "n"
```

レベルがのレベルとなる、`levels`パラメータをなの「フィルタ」にする`labels`パラメータをしますが、として`factor`ベクトルの`"levels"`のをそのままします`labels`

```
fm <- factor(as.numeric(f), levels = c(2,3,1),
 labels = c("nn", "WW", "cc"))
levels(fm)
[1] "nn" "WW" "cc"

fm <- factor(LETTERS[1:6], levels = LETTERS[1:4], # only 'A'-'D' as input
```

```

labels = letters[1:4]) # but assigned to 'a'-'d'
fm
[1] a b c d <NA> <NA>
#Levels: a b c d

```

## 2. relevelをする

1つのな level があるがある、たちは relevel をすることができます。これは、えは、をするために base カテゴリがなののコンテキストでします。

```

g<-relevel(f, "n") # moves n to be the first level
levels(g)
[1] "n" "c" "W"

```

できるように、f と g はじです

```

all.equal(f, g)
[1] "Attributes: < Component 'levels': 2 string mismatches >"
all.equal(f, g, check.attributes = F)
[1] TRUE

```

## 3. べえの

、な、された、またはのについて levels を reorder があるがあります。 levels についてべえましよう

```

table(g)
g
n c W
20 14 17

```

reorder はです help(reorder) してくださいが、このでは、のものがです x、このは factor; x は、x とじさのです。x にされ、x レベルによってされるで、levels をする FUN は、デフォルトでします。そのは、じであり、そのレベルはべえられています。

```

g.ord <- reorder(g, rep(1, length(g)), FUN=sum) #increasing
levels(g.ord)
[1] "c" "W" "n"

```

するをるために、の -1

```

g.ord.d <- reorder(g, rep(-1, length(g)), FUN=sum)
levels(g.ord.d)
[1] "n" "W" "c"

```

、はのとじです。

```

data.frame(f, g, g.ord, g.ord.d)[seq(1, length(g), by=5),] #just same lines
f g g.ord g.ord.d
1 W W W W

```

```
6 W W W W
11 W W W W
16 W W W W
21 n n n n
26 n n n n
31 n n n n
36 n n n n
41 c c c c
46 c c c c
51 c c c c
```

にするがある、のをして `levels` をべえることができ `levels`。 については、 `iris` データ `help("iris")` を `Sepal.Width`、 `Species` の `Sepal.Width` をしてべえます。

```
miris <- iris #help("iris") # copy the data
with(miris, tapply(Sepal.Width, Species, mean))
setosa versicolor virginica
3.428 2.770 2.974

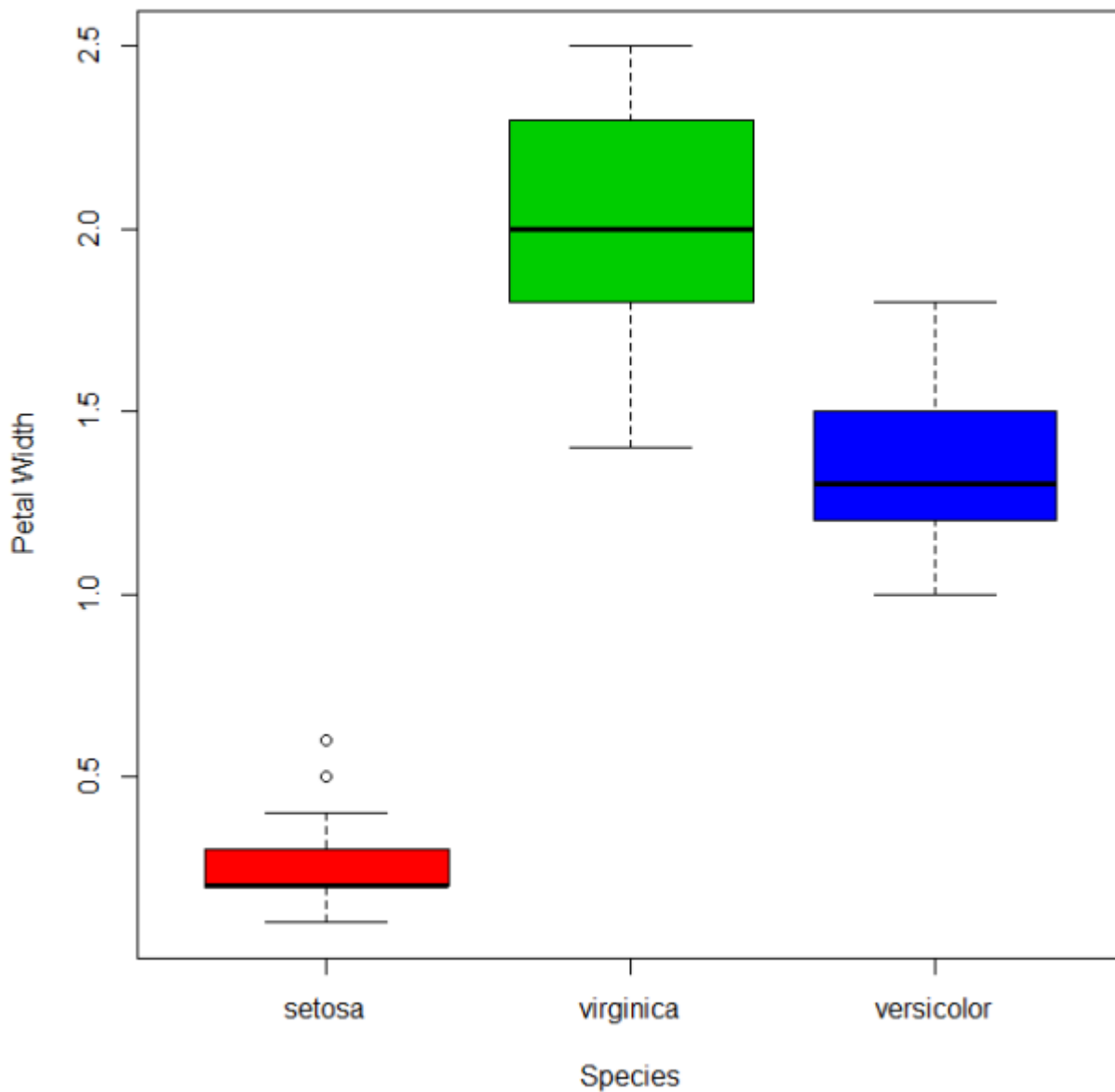
miris$Species.o <- with(miris, reorder(Species, -Sepal.Width))
levels(miris$Species.o)
[1] "setosa" "virginica" "versicolor"
```

の `boxplot` `with(miris, boxplot(Petal.Width~Species))` のにな `especies` がされます `setosa`、 `versicolor` の、 および `virginica` のしかし、 したをいて、 々はそのがのをる。 `Sepal.Width`

```
boxplot(Petal.Width~Species.o, data = miris,
 xlab = "Species", ylab = "Petal Width",
 main = "Iris Data, ordered by mean sepal width", varwidth = TRUE,
 col = 2:4)
```



Iris Data, ordered by mean sepal width



さらに、`levels`のをしたり、グループにまとめたり、しい`levels`することもでき`levels`。そのためは、じ`levels`をし`levels`。

```
f1<-f
levels(f1)
[1] "c" "n" "w"
levels(f1) <- c("upper","upper","CAP") #rename and grouping
levels(f1)
[1] "upper" "CAP"

f2<-f1
levels(f2) <- c("upper","CAP", "Number") #add Number level, which is empty
levels(f2)
[1] "upper" "CAP" "Number"
f2[length(f2):(length(f2)+5)]<-"Number" # add cases for the new level
table(f2)
f2
```

```
upper CAP Number
33 17 6

f3<-f1
levels(f3) <- list(G1 = "upper", G2 = "CAP", G3 = "Number") # The same using list
levels(f3)
[1] "G1" "G2" "G3"
f3[length(f3):(length(f3)+6)]<-"G3" ## add cases for the new level
table(f3)
f3
G1 G2 G3
33 17 7
```

## けられた

に、`ordered`とはなり、の`factors`はデータをすのにされ、2つはのデータをすることがわかります。は、けられたの`levels`のをするのはありませんが、`labels`をすることはできます。

```
ordvar<-rep(c("Low", "Medium", "High"), times=c(7,2,4))

of<-ordered(ordvar,levels=c("Low", "Medium", "High"))
levels(of)
[1] "Low" "Medium" "High"

of1<-of
levels(of1)<- c("LOW", "MEDIUM", "HIGH")
levels(of1)
[1] "LOW" "MEDIUM" "HIGH"
is.ordered(of1)
[1] TRUE
of1
[1] LOW LOW LOW LOW LOW LOW LOW MEDIUM MEDIUM HIGH HIGH HIGH HIGH

Levels: LOW < MEDIUM < HIGH
```

## ゼロからのの

は、Rのレベルとばれるのカテゴリからをるをするためにされます。たとえば、、、の4つのレベルで、バッテリーのエネルギーレベルによってけられるもあります。に、5つのなるサンプリングサイトについて、これらのレベルをのようになすことができます。

フル、フル、ノーマル、、

、データベースまたはのでは、これらのデータのりいは、カテゴリまたはレベルにけられたのインデックスによってわれます。えられたでは、1=、2=、3=、4=、のように5つのサンプルをコードするとします。

4、4、3、1、2

あなたの、えばデータベースからは、エンコードされたのリストと、をlevel-keywordとけるカタログだけがするがあります。そのからどのようにRのをすることができますか

サンプルをす20ののベクトルをシミュレートします。それぞれのサンプルは4つのなるの1つをち

ます。

```
set.seed(18)
ii <- sample(1:4, 20, replace=T)
ii
```

```
[1] 4 3 4 1 1 3 2 3 2 1 3 4 1 2 4 1 3 1 4 1
```

のステップは、レベルまたはカテゴリが1から4までのである、のシーケンスからのをすることです。

```
fii <- factor(ii, levels=1:4) # it is necessary to indicate the numeric levels
fii
```

```
[1] 4 3 4 1 1 3 2 3 2 1 3 4 1 2 4 1 3 1 4 1
レベル1 2 3 4
```

は、にindexタグでしたをドレスするがあります。

```
levels(fii) <- c("empty", "low", "normal", "full")
fii
```

```
[1] ノーマル ノーマル ノーマル
[11]
レベルのいのフル
```

オンラインでをむ <https://riptutorial.com/ja/r/topic/1104/>

## 127: クラス

き

は、ベクトルのモードおよびのクラスです。

`TRUE`、`FALSE`、`NA`はベクトルのものです。3つはです。`T`と`F`は、クリーンなRセッションでは`TRUE`と`FALSE`になり`FALSE`が、`T`と`F`どちらもされていないため、これらのにデフォルトのをすると、ユーザーはになるがあります。

### Examples

には、さのベクトルをけてすもの`!`、`|`、`&`、`xor()`と、`&&`、`||`のののみをするものの2があります。2のソートは、に`if`の`cond`としてされます。

|                         | ない       | バツ                                 |
|-------------------------|----------|------------------------------------|
| そして、                    | ベクトルおよび  | <code>x</code> と <code>y</code>    |
| <code>&amp;&amp;</code> | とのみ      | <code>x &amp;&amp; y</code>        |
| <code> </code>          | ベクトルまたは  | <code>x   y</code>                 |
| <code>  </code>         | またはのみ    | <code>x    y</code>                |
| <code>xor</code>        | のベクトルされた | <code>xorx</code> 、 <code>y</code> |

`||`はのをし、のが`TRUE`の、はされません。のがなのである、これによりをできます。`&&`は、ののが`FALSE`のときに、2のをしないで`FALSE`をします。

```
> x <- 5
> x > 6 || stop("X is too small")
Error: X is too small
> x > 3 || stop("X is too small")
[1] TRUE
```

がかどうかをチェックするには、`is.logical()`をします。

をにするには、`as.logical()`をします。

```
> x <- 2
> z <- x > 4
```

```
> z
[1] FALSE
> class(x)
[1] "numeric"
> as.logical(2)
[1] TRUE
```

に `as.numeric()` をすると、`double` がされます。 `NA` であり、との `NA` されます `NA` があいまいである。

## NA の

については、 をしてください。

```
> TRUE & NA
[1] NA
> FALSE & NA
[1] FALSE
> TRUE || NA
[1] TRUE
> FALSE || NA
[1] NA
```

オンラインでクラスをむ <https://riptutorial.com/ja/r/topic/9016>/クラス

## 128: いといでデータをする

き

Rでは、のデータがデータフレームにされます。このトピックでは、のをするさまざまなについてします。

## なパッケージ

- データの、スタッキング、
- をってする
- `splitstackshape`

## Examples

リサーブ

データを`reshape`ためのものなRは、`reshape`です。については`?reshape`をしてください。

```
create unbalanced longitudinal (panel) data set
set.seed(1234)
df <- data.frame(identifier=rep(1:5, each=3),
 location=rep(c("up", "down", "left", "up", "center"), each=3),
 period=rep(1:3, 5), counts=sample(35, 15, replace=TRUE),
 values=runif(15, 5, 10))[-c(4,8,11),]
```

```
df
```

|    | identifier | location | period | counts | values   |
|----|------------|----------|--------|--------|----------|
| 1  | 1          | up       | 1      | 4      | 9.186478 |
| 2  | 1          | up       | 2      | 22     | 6.431116 |
| 3  | 1          | up       | 3      | 22     | 6.334104 |
| 5  | 2          | down     | 2      | 31     | 6.161130 |
| 6  | 2          | down     | 3      | 23     | 6.583062 |
| 7  | 3          | left     | 1      | 1      | 6.513467 |
| 9  | 3          | left     | 3      | 24     | 5.199980 |
| 10 | 4          | up       | 1      | 18     | 6.093998 |
| 12 | 4          | up       | 3      | 20     | 7.628488 |
| 13 | 5          | center   | 1      | 10     | 9.573291 |
| 14 | 5          | center   | 2      | 33     | 9.156725 |
| 15 | 5          | center   | 3      | 11     | 5.228851 |

`data.frame`はアンバランスであることにしてください。つまり、ユニット2は1のにをしていますが、ユニット3と4には2のにがしています。また、ピリオドには2つのがありますカウントと、およびとがわからない2つのがあります。

ロングトウワイド

data.frameをワイドフォーマットにするには、

```
reshape wide on time variable
df.wide <- reshape(df, idvar="identifier", timevar="period",
 v.names=c("values", "counts"), direction="wide")
df.wide
 identifier location values.1 counts.1 values.2 counts.2 values.3 counts.3
1 1 up 9.186478 4 6.431116 22 6.334104 22
5 2 down NA NA 6.161130 31 6.583062 23
7 3 left 6.513467 1 NA NA 5.199980 24
10 4 up 6.093998 18 NA NA 7.628488 20
13 5 center 9.573291 10 9.156725 33 5.228851 11
```

しているは、NAsでめられていることにしてください。

くする、「v.names」はとともにするをします。がでないは、"drop"をしてするにすることができます。data.frameからnon-varying / non-idのみをする、v.namesはになります。

```
reshape(df, idvar="identifier", timevar="period", direction="wide",
 drop="location")
```

## ワイドからロングまで

のdf.wideでくするには、のがです。

```
reshape(df.wide, direction="long")
```

しかし、これはにはトリッキーです。

```
remove "." separator in df.wide names for counts and values
names(df.wide)[grep("\\.", names(df.wide))] <-
 gsub("\\.", "", names(df.wide)[grep("\\.", names(df.wide))])
```

これでなでは、のについてエラーがします。

reshapeがにreshapeがなでは、するためにreshapeにワイドののをグループreshapeようにreshape "varying"をするがあります。このは、またはインデックスのベクトルのリストをとります。

```
reshape(df.wide, idvar="identifier",
 varying=list(c(3,5,7), c(4,6,8)), direction="long")
```

long reshapingでは、"v.names"をしてののをすることができます。

ときどき "varying"のは、のどのがvalueをし、timeをreshapeかをreshapeにらせる "sep"のによってけることができます。

データの

この、データはテーブルにされます。に、このデータをワイドおよびロングフォーマットですることができます。ワイドフォーマットでは、にはのがあります。

|      | さ[cm] | [] |
|------|-------|----|
| アリソン | 178   | 20 |
| ボブ   | 174   | 45 |
| カール  | 182   | 31 |

ただし、すべてのが1つのにあり、が2つのにあるようないのがあります。

| アリソン | さ[cm] | 178 |
|------|-------|-----|
| ボブ   | さ[cm] | 174 |
| カール  | さ[cm] | 182 |
| アリソン | []    | 20  |
| ボブ   | []    | 45  |
| カール  | []    | 31  |

このプロセスをするために、Base Rとパッケージをすることができます。オプションについて、mtcarsデータセットがされます。デフォルトでは、このデータセットはいです。パッケージをさせるために、をのとしてします。

```
mtcars # shows the dataset
data <- data.frame(observation=row.names(mtcars),mtcars)
```

## ベースR

Rには、ワイドフォーマットとロングフォーマットのにできる2つの、stack()とunstack()ます。

```
long <- stack(data)
long # this shows the long format
wide <- unstack(long)
wide # this shows the wide format
```

しかし、これらはなユースケースではにになります。いにも、サードパーティのパッケージをするのオプションがあります。



## tidyrパッケージ

このパッケージは `gather()` をしてワイドからロングにし、 `spread()` をしてロングワイドからワイドにします。

```
library(tidyr)
long <- gather(data, variable, value, 2:12) # where variable is the name of the
variable column, value indicates the name of the value column and 2:12 refers to
the columns to be converted.
long # shows the long result
wide <- spread(long, variable, value)
wide # shows the wide result (~data)
```

---

## data.tableパッケージ

`data.table`パッケージは `reshape2` をし、 `melt()` をしてワイドからロングに、 `dcast()` からワイドにします。

```
library(data.table)
long <- melt(data, 'observation', 2:12, 'variable', 'value')
long # shows the long result
wide <- dcast(long, observation ~ variable)
wide # shows the wide result (~data)
```

オンラインでいといのでデータをするをむ <https://riptutorial.com/ja/r/topic/2904/いといのでデータをする>

## 129: モデリング

### Examples

なモデルフィッティング

しありません は、のについてのやフィードバックをするためのチャンネルがわからないので、  
はここでします。このよりいをしてください @DataTxは、これが「にで、で、またはフォーマットのが」とべています。きなの:-)はされないなので、ここでさやをするためにがされているのか、  
そしてここではができないのかについてのガイダンスがちます。

Rのあるいは "mixed"または " `lme4` "モデルをフィッティングするためのなパッケージは、 `nlme` い  
と `lme4` しいです。これらのパッケージはくのマイナーなでなりますが、にによくとモデルになり  
ます。

```
library(nlme)
library(lme4)
m1.nlme <- lme(Reaction~Days, random=~Days|Subject, data=sleepstudy, method="REML")
m1.lme4 <- lmer(Reaction~Days+(Days|Subject), data=sleepstudy, REML=TRUE)
all.equal(fixef(m1.nlme), fixef(m1.lme4))
[1] TRUE
```

すべき

- はしなります
- `nlme` まだいされているえばピニエイロとベイツは、S-PLUS 2000モデルとしたが、ベイツ  
ら 2015のソフトウェアの `/vignette("lmer", package="lme4")` のために `lme4`
- `lme4` はで、したランダムなをにフィットできます
- `nlme` は、からりしたモデルのpをします `lme4` は、 `lmerTest` や `afex` などのアドオンパッケージが  
`afex`
- `nlme` は、//におけるまたはのモデルをにする。

のGLMM FAQは、モデルGLMMにをてていますが、よりくのをします。

オンラインでモデリングをむ <https://riptutorial.com/ja/r/topic/3460/モデリング>

## 130: と

き

DplyrとRのくのは、プログラミングのためのNSEとプログラミングのためのSEをしています<sup>1</sup>。

たとえば、`summarise()`はをしますが、をする`summarise_()`します。

`lazyeval`ライブラリをすると、をNSEににりえることができます。

## Examples

なdplyrの

NSEは、プログラミングでするがあります。ただし、しいパッケージでしいをするは、SEバージョンをするがよいでしょう。

dplyrとlazyevalをロードする

```
library(dplyr)
library(lazyeval)
```

フィルタリング

NSEバージョン

```
filter(mtcars, cyl == 8)
filter(mtcars, cyl < 6)
filter(mtcars, cyl < 6 & vs == 1)
```

SEバージョンしいパッケージにをプログラミングするときにする

```
filter_(mtcars, .dots = list(~ cyl == 8))
filter_(mtcars, .dots = list(~ cyl < 6))
filter_(mtcars, .dots = list(~ cyl < 6, ~ vs == 1))
```

NSEバージョン

```
summarise(mtcars, mean_disp)
summarise(mtcars, mean_disp = mean_disp)
```

SEバージョン

```
summarise_(mtcars, .dots = lazyeval::interp(~ mean(x), x = quote(dispatch)))
summarise_(mtcars, .dots = setNames(list(lazyeval::interp(~ mean(x), x = quote(dispatch)),
"mean_disp")))
```

```
summarise_(mtcars, .dots = list("mean_disp" = lazyeval::interp(~ mean(x), x = quote(displ))))
```

## NSEバージョン

```
mutate(mtcars, displ_l = displ / 61.0237)
```

## SEバージョン

```
mutate_(
 .data = mtcars,
 .dots = list(
 "displ_l" = lazyeval::interp(
 ~ x / 61.0237, x = quote(displ)
)
)
)
```

オンラインでとをむ <https://riptutorial.com/ja/r/topic/9365/>と

## 131: テストの

### Examples

かなりな

ビルトインのデータセットをします。モルモットにオレンジジュースとビタミンCをえた、のにながあるかどうかにかがあります。

なはのとおりで。

```
teethVC = ToothGrowth[ToothGrowth$supp == 'VC',]
teethOJ = ToothGrowth[ToothGrowth$supp == 'OJ',]

permutationTest = function(vectorA, vectorB, testStat){
 N = 10^5
 fullSet = c(vectorA, vectorB)
 lengthA = length(vectorA)
 lengthB = length(vectorB)
 trials <- replicate(N,
 {index <- sample(lengthB + lengthA, size = lengthA, replace = FALSE)
 testStat((fullSet[index]), fullSet[-index]) })

 trials
}
vec1 =teethVC$len;
vec2 =teethOJ$len;
subtractMeans = function(a, b){ return (mean(a) - mean(b))}
result = permutationTest(vec1, vec2, subtractMeans)
observedMeanDifference = subtractMeans(vec1, vec2)
result = c(result, observedMeanDifference)
hist(result)
abline(v=observedMeanDifference, col = "blue")
pValue = 2*mean(result <= (observedMeanDifference))
pValue
```

CSVをみんだ、をします

```
permutationTest = function(vectorA, vectorB, testStat){
 N = 10^5
 fullSet = c(vectorA, vectorB)
 lengthA = length(vectorA)
 lengthB = length(vectorB)
 trials <- replicate(N,
 {index <- sample(lengthB + lengthA, size = lengthA, replace = FALSE)
 testStat((fullSet[index]), fullSet[-index]) })

 trials
}
```

これは2つのベクトルをり、それらのをにシャッフルし、シャッフルされたベクトルにしてtestStatをします。teststatのがりであるtrialsにされます。

これは $N = 10^5$ う。 $N$ はのパラメータであるがにいいことにしてください。

これにより、新しいデータセット、 `trials`、には2つのにないにじるのあるのがされます。

に、テストをします。

```
subtractMeans = function(a, b){ return (mean(a) - mean(b)) }
```

テストをします。

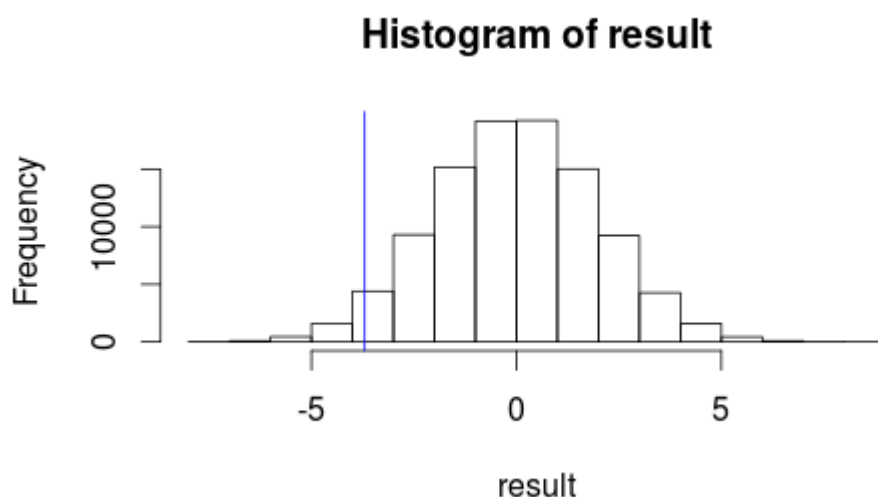
```
result = permutationTest(vec1, vec2, subtractMeans)
```

々ののされたをする

```
observedMeanDifference = subtractMeans(vec1, vec2)
```

々のテストのヒストグラムでの々ののをてみましょう。

```
hist(result)
abline(v=observedMeanDifference, col = "blue")
```



たちのされたがランダムなチャンスによってにこりそうなるようにはえませんが...

2つのにない、なのであるpをします。

```
pValue = 2*mean(result >= (observedMeanDifference))
```

それをちょっとちいてみましょう

```
result >= (observedMeanDifference)
```

のようなブールベクトルをします。

```
FALSE TRUE FALSE FALSE TRUE FALSE ...
```

resultがobservedMeanになるたびにTRUEます。

ファンクションmean、このベクトルをする1についてTRUEと0のためFALSE、そしてたちのえる1ミックスのを、たちのシャッフルベクトルがいをする。すなわちか、々がするものしかったです。

に、テストのがにであるために、2をけます。にされたよりも"な"をりたいのです。

っているのは、0.06093939としたpをすること0.06093939です。こののはですが、はビタミンCがオレンジジュースよりもかなりくのをするようにえるといいます。

オンラインでテストのをむ <https://riptutorial.com/ja/r/topic/3216/テストの>

## クレジット

| S. No |                        | Contributors                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1     | Rのい                    | <a href="#">42-</a> , <a href="#">akraf</a> , <a href="#">Ale</a> , <a href="#">Andrea Cirillo</a> , <a href="#">Andrew Brēza</a> , <a href="#">Axeman</a> , <a href="#">Community</a> , <a href="#">Craig Vermeer</a> , <a href="#">d.b</a> , <a href="#">dotancohen</a> , <a href="#">Francesco Dondi</a> , <a href="#">Frank</a> , <a href="#">G5W</a> , <a href="#">George Bonebright</a> , <a href="#">GForce</a> , <a href="#">Giorgos K</a> , <a href="#">Gregor</a> , <a href="#">H. Pauwelyn</a> , <a href="#">kartoffelsalat</a> , <a href="#">kdopen</a> , <a href="#">Konrad Rudolph</a> , <a href="#">L.V.Rao</a> , <a href="#">Imckeogh</a> , <a href="#">Lovy</a> , <a href="#">Matt</a> , <a href="#">mnoronha</a> , <a href="#">pitosalas</a> , <a href="#">polka</a> , <a href="#">Rahul Saini</a> , <a href="#">RetractedAndRetired</a> , <a href="#">russellpierce</a> , <a href="#">Steve_Corrin</a> , <a href="#">theArun</a> , <a href="#">Thomas</a> , <a href="#">torina</a> , <a href="#">user2100721</a> , <a href="#">while</a> |
| 2     | *ファミリーをする              | <a href="#">Benjamin</a> , <a href="#">FisherDisinformation</a> , <a href="#">Gavin Simpson</a> , <a href="#">jcb</a> , <a href="#">Karolis Koncevičius</a> , <a href="#">kneijenhuijs</a> , <a href="#">Maximilian Kohl</a> , <a href="#">nrussell</a> , <a href="#">omar</a> , <a href="#">Robert</a> , <a href="#">seasmith</a> , <a href="#">zacdav</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 3     | .Rprofile              | <a href="#">42-</a> , <a href="#">Dirk Eddebuettel</a> , <a href="#">ikashnitsky</a> , <a href="#">Karolis Koncevičius</a> , <a href="#">Nikos Alexandris</a> , <a href="#">Stedy</a> , <a href="#">Thomas</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 4     | ANOVA                  | <a href="#">Ben Bolker</a> , <a href="#">DataTx</a> , <a href="#">kneijenhuijs</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 5     | data.tableでピボット/アンピボット | <a href="#">Sun Bee</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 6     | Dateクラス                | <a href="#">alistaire</a> , <a href="#">coatless</a> , <a href="#">Frank</a> , <a href="#">L.V.Rao</a> , <a href="#">MichaelChirico</a> , <a href="#">Steve_Corrin</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 7     | devtoolsを使ってパッケージを作る   | <a href="#">Frank</a> , <a href="#">Lovy</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 8     | dplyr                  | <a href="#">4444</a> , <a href="#">Alihan Zihna</a> , <a href="#">ikashnitsky</a> , <a href="#">Robert</a> , <a href="#">skoh</a> , <a href="#">Sumedh</a> , <a href="#">theArun</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 9     | ggplot2                | <a href="#">akraf</a> , <a href="#">Alex</a> , <a href="#">alistaire</a> , <a href="#">Andrea Cirillo</a> , <a href="#">Artem Klevtsov</a> , <a href="#">Axeman</a> , <a href="#">baptiste</a> , <a href="#">blmoore</a> , <a href="#">Boern</a> , <a href="#">gitblame</a> , <a href="#">ikashnitsky</a> , <a href="#">Jaap</a> , <a href="#">jmax</a> , <a href="#">loki</a> , <a href="#">Matt</a> , <a href="#">Mine Cetinkaya-Rundel</a> , <a href="#">Paolo</a> , <a href="#">smci</a> , <a href="#">Steve_Corrin</a> , <a href="#">Sumedh</a> , <a href="#">Taylor Ostberg</a> , <a href="#">theArun</a> , <a href="#">void</a> , <a href="#">YCR</a> , <a href="#">Yun Ching</a>                                                                                                                                                                                                                                                                                                                                                                    |
| 10    | GPUコンピューティング           | <a href="#">cdeterman</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 11    | hclustによるクラスタリング       | <a href="#">Frank</a> , <a href="#">G5W</a> , <a href="#">Tal Galili</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 12    | igraphパッケージに           | <a href="#">Boysenb3rry</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |



| よるネットワーク |                          |                                                                                                                                                               |
|----------|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 13       | JSON                     | <a href="#">SymbolixAU</a>                                                                                                                                    |
| 14       | LaTeXのR with knitr       | <a href="#">JHowIX</a>                                                                                                                                        |
| 15       | R MarkdownノートブックRStudio  | <a href="#">dmail</a>                                                                                                                                         |
| 16       | Rcpp                     | <a href="#">Artem Klevtsov</a> , <a href="#">coatless</a> , <a href="#">Dirk Eddelbuettel</a>                                                                 |
| 17       | RESTfulなRサービス            | <a href="#">YCR</a>                                                                                                                                           |
| 18       | RMarkdownとknitrプレゼンテーション | <a href="#">Martin Schmelzer</a> , <a href="#">YCR</a>                                                                                                        |
| 19       | RMarkdownをしたレポートの        | <a href="#">ikashnitsky</a> , <a href="#">Karolis Koncevičius</a> , <a href="#">Martin Schmelzer</a>                                                          |
| 20       | RMDの                     | <a href="#">J_F</a> , <a href="#">RamenChef</a>                                                                                                               |
| 21       | RODBC                    | <a href="#">akrun</a> , <a href="#">Hack-R</a> , <a href="#">Parfait</a> , <a href="#">Tim Coker</a>                                                          |
| 22       | Rコードベクトルのベストプラクティス       | <a href="#">Axeman</a> , <a href="#">David Arenburg</a> , <a href="#">snaut</a>                                                                               |
| 23       | RでODEをく                  | <a href="#">J_F</a>                                                                                                                                           |
| 24       | Rでツイートをする                | <a href="#">Umberto</a>                                                                                                                                       |
| 25       | Rとパッケージライブラリの            | <a href="#">Eric Lecoutre</a>                                                                                                                                 |
| 26       | Rにおけるオブジェクトプログラミング       | <a href="#">Jon Ericson</a> , <a href="#">rcorty</a>                                                                                                          |
| 27       | Rにをく                     | <a href="#">Aksela</a> , <a href="#">ikashnitsky</a> , <a href="#">kaksat</a>                                                                                 |
| 28       | RのWebクロール                | <a href="#">Pankaj Sharma</a>                                                                                                                                 |
| 29       | RのバイナリのI/O               | <a href="#">Frank</a> , <a href="#">ikashnitsky</a> , <a href="#">Mario</a> , <a href="#">russellpierce</a> , <a href="#">zacdav</a> , <a href="#">zx8754</a> |
| 30       | Rの - の                   | <a href="#">Joy</a>                                                                                                                                           |
| 31       | Rの                       | <a href="#">Alexey Shiklomanov</a>                                                                                                                            |
| 32       | Rバージョンの                  | <a href="#">dmail</a>                                                                                                                                         |

|    |                           |                                                                                                                                                                                                                                                                                                                                                                                                          |
|----|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 33 | Rをいた                      | <a href="#">Pankaj Sharma</a>                                                                                                                                                                                                                                                                                                                                                                            |
| 34 | S4クラスをしたステートマシンパターンの      | <a href="#">David Leal</a>                                                                                                                                                                                                                                                                                                                                                                               |
| 35 | sqldf                     | <a href="#">Hack-R</a> , <a href="#">Miha</a>                                                                                                                                                                                                                                                                                                                                                            |
| 36 | stringiパッケージによる           | <a href="#">bartektartanus</a> , <a href="#">FisherDisinformation</a>                                                                                                                                                                                                                                                                                                                                    |
| 37 | strsplit                  | <a href="#">Imo</a>                                                                                                                                                                                                                                                                                                                                                                                      |
| 38 | texregをしてにしたでモデルをエクスポートする | <a href="#">Frank</a> , <a href="#">ikashnitsky</a>                                                                                                                                                                                                                                                                                                                                                      |
| 39 | Webスクレープと                 | <a href="#">alistaire</a> , <a href="#">Dave2e</a>                                                                                                                                                                                                                                                                                                                                                       |
| 40 | xgboost                   | <a href="#">Hack-R</a>                                                                                                                                                                                                                                                                                                                                                                                   |
| 41 | イントロスペクション                | <a href="#">Jason</a>                                                                                                                                                                                                                                                                                                                                                                                    |
| 42 | キャレット                     | <a href="#">highBandWidth</a> , <a href="#">Steve_Corrin</a>                                                                                                                                                                                                                                                                                                                                             |
| 43 | クラス                       | <a href="#">42-</a> , <a href="#">AkselA</a> , <a href="#">David Heckmann</a> , <a href="#">dayne</a> , <a href="#">Frank</a> , <a href="#">Gregor</a> , <a href="#">Jaap</a> , <a href="#">kneijenhuijs</a> , <a href="#">L.V.Rao</a> , <a href="#">Nathan Werth</a> , <a href="#">Steve_Corrin</a>                                                                                                     |
| 44 | グラフィックスのクラススキーム           | <a href="#">ikashnitsky</a> , <a href="#">munirbe</a>                                                                                                                                                                                                                                                                                                                                                    |
| 45 | コードプロファイリング               | <a href="#">Ben Bolker</a> , <a href="#">Glen Moutrie</a> , <a href="#">Jav</a> , <a href="#">SymbolixAU</a> , <a href="#">USER_1</a>                                                                                                                                                                                                                                                                    |
| 46 | コンビナトリアル                  | <a href="#">Frank</a> , <a href="#">Karolis Koncevičius</a>                                                                                                                                                                                                                                                                                                                                              |
| 47 | サブセッティング                  | <a href="#">42-</a> , <a href="#">Agriculturist</a> , <a href="#">alexis_laz</a> , <a href="#">alistaire</a> , <a href="#">dayne</a> , <a href="#">Frank</a> , <a href="#">Gavin Simpson</a> , <a href="#">Gregor</a> , <a href="#">L.V.Rao</a> , <a href="#">Mario</a> , <a href="#">mrip</a> , <a href="#">RamenChef</a> , <a href="#">smci</a> , <a href="#">user2100721</a> , <a href="#">zx8754</a> |
| 48 | シャイニー                     | <a href="#">alistaire</a> , <a href="#">CClaire</a> , <a href="#">Christophe D.</a> , <a href="#">JvH</a> , <a href="#">russellpierce</a> , <a href="#">SymbolixAU</a> , <a href="#">tuomastik</a> , <a href="#">zx8754</a>                                                                                                                                                                              |
| 49 | スタンドアロンのRスクリプトをしてをする      | <a href="#">akraf</a> , <a href="#">herbaman</a>                                                                                                                                                                                                                                                                                                                                                         |
| 50 | スパーク API SparkR           | <a href="#">Maximilian Kohl</a>                                                                                                                                                                                                                                                                                                                                                                          |

|    |                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 51 | データのクリーニング     | <a href="#">Derek Corcoran</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 52 | データフレーム        | <a href="#">Alex</a> , <a href="#">Andrea Ianni</a> , <a href="#">Batanichek</a> , <a href="#">Carlos Cinelli</a> , <a href="#">Christophe D.</a> , <a href="#">DataTx</a> , <a href="#">David Arenburg</a> , <a href="#">David Robinson</a> , <a href="#">dayne</a> , <a href="#">Frank</a> , <a href="#">Gregor</a> , <a href="#">Hack-R</a> , <a href="#">kaksat</a> , <a href="#">R. Schifini</a> , <a href="#">scoa</a> , <a href="#">Sumedh</a> , <a href="#">Thomas</a> , <a href="#">Tomás Barcellos</a> , <a href="#">user2100721</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 53 | データフレームの       | <a href="#">Florian</a> , <a href="#">Frank</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 54 | データベーステーブルのI/O | <a href="#">Frank</a> , <a href="#">JHowIX</a> , <a href="#">SommerEngineering</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 55 | データ            | <a href="#">ikashnitsky</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 56 | データ            | <a href="#">akrun</a> , <a href="#">Allen Wang</a> , <a href="#">bartektartanus</a> , <a href="#">cderv</a> , <a href="#">David</a> , <a href="#">David Arenburg</a> , <a href="#">Dean MacGregor</a> , <a href="#">Eric Lecoutre</a> , <a href="#">Frank</a> , <a href="#">Jaap</a> , <a href="#">jogo</a> , <a href="#">L Co</a> , <a href="#">leogama</a> , <a href="#">Mallick Hossain</a> , <a href="#">micstr</a> , <a href="#">Nathan Werth</a> , <a href="#">oshun</a> , <a href="#">Peter Humburg</a> , <a href="#">Sowmya S. Manian</a> , <a href="#">stanekam</a> , <a href="#">Steve_Corrin</a> , <a href="#">Sumedh</a> , <a href="#">Tensibai</a> , <a href="#">user2100721</a> , <a href="#">Uwe</a>                                                                                                                                                                                                                                                                                          |
| 57 | テキストマイニング      | <a href="#">Hack-R</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 58 | デバッグ           | <a href="#">James Elderfield</a> , <a href="#">russellpierce</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 59 | パイプ>など         | <a href="#">42-</a> , <a href="#">Alexandru Papiu</a> , <a href="#">Alihan Zihna</a> , <a href="#">alistaire</a> , <a href="#">AndreyAkinshin</a> , <a href="#">Artem Klevtsov</a> , <a href="#">Atish</a> , <a href="#">Axeman</a> , <a href="#">Benjamin</a> , <a href="#">Carlos Cinelli</a> , <a href="#">CMichael</a> , <a href="#">DrPositron</a> , <a href="#">Franck Dernoncourt</a> , <a href="#">Frank</a> , <a href="#">Gal Dreiman</a> , <a href="#">Gavin Simpson</a> , <a href="#">Gregor</a> , <a href="#">ikashnitsky</a> , <a href="#">James McCalden</a> , <a href="#">Kay Brodersen</a> , <a href="#">Matt</a> , <a href="#">polka</a> , <a href="#">RamenChef</a> , <a href="#">Ryan Hilbert</a> , <a href="#">Sam Firke</a> , <a href="#">seasmith</a> , <a href="#">Shawn Mehan</a> , <a href="#">Simplans</a> , <a href="#">Spacedman</a> , <a href="#">SymbolixAU</a> , <a href="#">thelatemail</a> , <a href="#">tomw</a> , <a href="#">TriskalJM</a> , <a href="#">user2100721</a> |
| 60 | パターンマッチングと     | <a href="#">Abdou</a> , <a href="#">Alex</a> , <a href="#">Artem Klevtsov</a> , <a href="#">David Arenburg</a> , <a href="#">David Leal</a> , <a href="#">Frank</a> , <a href="#">Gavin Simpson</a> , <a href="#">Jaap</a> , <a href="#">NWaters</a> , <a href="#">R. Schifini</a> , <a href="#">SommerEngineering</a> , <a href="#">Steve_Corrin</a> , <a href="#">Tensibai</a> , <a href="#">thelatemail</a> , <a href="#">user2100721</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 61 | パッケージのインストール   | <a href="#">Aaghaz Hussain</a> , <a href="#">akraf</a> , <a href="#">alko989</a> , <a href="#">Andrew Brêza</a> , <a href="#">Artem Klevtsov</a> , <a href="#">Arun Balakrishnan</a> , <a href="#">Christophe D.</a> , <a href="#">CL.</a> , <a href="#">Frank</a> , <a href="#">gitblame</a> , <a href="#">Hack-R</a> , <a href="#">hongsy</a> , <a href="#">Jaap</a> , <a href="#">kaksat</a> , <a href="#">kneijenhuijs</a> , <a href="#">Imckeogh</a> , <a href="#">loki</a> , <a href="#">Marc Brinkmann</a> , <a href="#">Miha</a> , <a href="#">Peter Humburg</a> , <a href="#">Pragyaditya Das</a> , <a href="#">Raj Padmanabhan</a> , <a href="#">seasmith</a> , <a href="#">SymbolixAU</a> , <a href="#">theArun</a> , <a href="#">user890739</a> , <a href="#">xamgore</a> , <a href="#">zx8754</a>                                                                                                                                                                                               |
| 62 | パッケージの         | <a href="#">Frank</a> , <a href="#">Sowmya S. Manian</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 63 | ハッシュマップ        | <a href="#">nrussell</a> , <a href="#">russellpierce</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

|    |                     |                                                                                                                                                                                                                                                                                                                                                                   |
|----|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 64 | ヒートマップとヒートマップ.2     | <a href="#">AndreyAkinshin</a> , <a href="#">Nanami</a>                                                                                                                                                                                                                                                                                                           |
| 65 | フーリエと               | <a href="#">Hack-R</a>                                                                                                                                                                                                                                                                                                                                            |
| 66 | フォールトトレラント/コード      | <a href="#">Rappster</a>                                                                                                                                                                                                                                                                                                                                          |
| 67 | ベースプロット             | <a href="#">42-</a> , <a href="#">Alexey Shiklomanov</a> , <a href="#">catastrophic-failure</a> , <a href="#">FisherDisinformation</a> , <a href="#">Frank</a> , <a href="#">Giorgos K</a> , <a href="#">K.Daisey</a> , <a href="#">maRtin</a> , <a href="#">MichaelChirico</a> , <a href="#">RamenChef</a> , <a href="#">Robert</a> , <a href="#">symbolrush</a> |
| 68 | ベクトルの               | <a href="#">alistaire</a> , <a href="#">bartektartanus</a> , <a href="#">Jaap</a> , <a href="#">Karsten W.</a> , <a href="#">Imo</a> , <a href="#">Rich Scriven</a> , <a href="#">Robert</a> , <a href="#">Robin Gertenbach</a> , <a href="#">smci</a> , <a href="#">takje</a>                                                                                    |
| 69 | ボックスプロット            | <a href="#">Carlos Cinelli</a> , <a href="#">Christophe D.</a> , <a href="#">Karolis Koncevičius</a> , <a href="#">L.V.Rao</a>                                                                                                                                                                                                                                    |
| 70 | メタドキュメンテーションのガイドライン | <a href="#">Frank</a> , <a href="#">Gregor</a> , <a href="#">Stephen Leppik</a> , <a href="#">Steve_Corrin</a>                                                                                                                                                                                                                                                    |
| 71 | ユーザーをする             | <a href="#">Ashish</a> , <a href="#">DeveauP</a>                                                                                                                                                                                                                                                                                                                  |
| 72 | ラスターイメージのI/O        | <a href="#">Frank</a> , <a href="#">loki</a>                                                                                                                                                                                                                                                                                                                      |
| 73 | ラスタと                | <a href="#">Frank</a> , <a href="#">loki</a>                                                                                                                                                                                                                                                                                                                      |
| 74 | ランダムフォレストアルゴリズム     | <a href="#">G5W</a>                                                                                                                                                                                                                                                                                                                                               |
| 75 | ランダム                | <a href="#">TARehman</a>                                                                                                                                                                                                                                                                                                                                          |
| 76 | ランレングス              | <a href="#">Frank</a> , <a href="#">josliber</a> , <a href="#">Psidom</a>                                                                                                                                                                                                                                                                                         |
| 77 | リサイクル               | <a href="#">Frank</a> , <a href="#">USER_1</a>                                                                                                                                                                                                                                                                                                                    |
| 78 | リスト                 | <a href="#">Andrea Ianni</a> , <a href="#">BarkleyBG</a> , <a href="#">dayne</a> , <a href="#">Frank</a> , <a href="#">Hack-R</a> , <a href="#">Hairizuan Noorazman</a> , <a href="#">Peter Humburg</a> , <a href="#">RamenChef</a>                                                                                                                               |
| 79 | ロキシーゲン2             | <a href="#">DeveauP</a> , <a href="#">PAC</a>                                                                                                                                                                                                                                                                                                                     |
| 80 | モデル                 | <a href="#">Ben Bolker</a> , <a href="#">YCR</a>                                                                                                                                                                                                                                                                                                                  |
| 81 |                     | <a href="#">Artem Klevtsov</a> , <a href="#">jameselmore</a> , <a href="#">K.Daisey</a> , <a href="#">Imo</a> , <a href="#">loki</a> , <a href="#">russellpierce</a>                                                                                                                                                                                              |
| 82 |                     | <a href="#">bartektartanus</a> , <a href="#">FisherDisinformation</a> , <a href="#">Karolis Koncevičius</a> , <a href="#">Miha</a> , <a href="#">mnoronha</a>                                                                                                                                                                                                     |

|     |                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 83  | でをする                         | <a href="#">Alex</a> , <a href="#">David Leal</a> , <a href="#">Frank</a>                                                                                                                                                                                                                                                                                                                                                                                                             |
| 84  | によるRのメモ                      | <a href="#">Lovy</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 85  | がありません                       | <a href="#">Amit Kohli</a> , <a href="#">Artem Klevtsov</a> , <a href="#">Axeman</a> , <a href="#">Eric Lecoutre</a> , <a href="#">Frank</a> , <a href="#">Gregor</a> , <a href="#">Jaap</a> , <a href="#">kitman0804</a> , <a href="#">Imo</a> , <a href="#">seasmith</a> , <a href="#">Steve_Corrin</a> , <a href="#">theArun</a> , <a href="#">user2100721</a>                                                                                                                     |
| 86  |                              | <a href="#">Frank</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 87  | なR                           | <a href="#">Charmgoggles</a> , <a href="#">Frank</a> , <a href="#">ikashnitsky</a>                                                                                                                                                                                                                                                                                                                                                                                                    |
| 88  |                              | <a href="#">Frank</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 89  |                              | <a href="#">Eric Lecoutre</a> , <a href="#">etienne</a> , <a href="#">josliber</a> , <a href="#">Sathish</a> , <a href="#">Tensibai</a> , <a href="#">thelatemail</a> , <a href="#">user2100721</a>                                                                                                                                                                                                                                                                                   |
| 90  |                              | <a href="#">FisherDisinformation</a> , <a href="#">Frank</a> , <a href="#">L.V.Rao</a> , <a href="#">tenCupMaximum</a>                                                                                                                                                                                                                                                                                                                                                                |
| 91  | の                            | <a href="#">akrun</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 92  | フロー                          | <a href="#">Benjamin</a> , <a href="#">David Arenburg</a> , <a href="#">nrussell</a> , <a href="#">Robert</a> , <a href="#">Steve_Corrin</a>                                                                                                                                                                                                                                                                                                                                          |
| 93  | しいからベクトルするコードの               | <a href="#">egnha</a> , <a href="#">josliber</a>                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 94  | されたアーカイブのファイルのと              | <a href="#">catastrophic-failure</a> , <a href="#">Jeff</a>                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 95  | データファイルなどのI/O                | <a href="#">Alex</a> , <a href="#">Frank</a> , <a href="#">ikashnitsky</a>                                                                                                                                                                                                                                                                                                                                                                                                            |
| 96  | の                            | <a href="#">4444</a> , <a href="#">AkselA</a> , <a href="#">alistaire</a> , <a href="#">beetroot</a> , <a href="#">Carson</a> , <a href="#">Frank</a> , <a href="#">Hack-R</a> , <a href="#">HypnoGenX</a> , <a href="#">Robert</a> , <a href="#">russellpierce</a> , <a href="#">SymbolixAU</a> , <a href="#">symbolrush</a>                                                                                                                                                         |
| 97  |                              | <a href="#">42-</a> , <a href="#">Ale</a> , <a href="#">Axeman</a> , <a href="#">Craig Vermeer</a> , <a href="#">Frank</a> , <a href="#">L.V.Rao</a> , <a href="#">Imckeogh</a>                                                                                                                                                                                                                                                                                                       |
| 98  | の                            | <a href="#">Artem Klevtsov</a> , <a href="#">K.Daisey</a> , <a href="#">RamenChef</a>                                                                                                                                                                                                                                                                                                                                                                                                 |
| 99  | テーブルExcel、SAS、SPSS、StataのI/O | <a href="#">42-</a> , <a href="#">Alex</a> , <a href="#">alistaire</a> , <a href="#">Andrea Cirillo</a> , <a href="#">Carlos Cinelli</a> , <a href="#">Charmgoggles</a> , <a href="#">Crops</a> , <a href="#">Frank</a> , <a href="#">Jaap</a> , <a href="#">Jeromy Anglim</a> , <a href="#">kaksat</a> , <a href="#">Ken S.</a> , <a href="#">kitman0804</a> , <a href="#">Imo</a> , <a href="#">Miha</a> , <a href="#">Parfait</a> , <a href="#">polka</a> , <a href="#">Thomas</a> |
| 100 |                              | <a href="#">42-</a> , <a href="#">Axeman</a> , <a href="#">Qaswed</a> , <a href="#">Sathish</a>                                                                                                                                                                                                                                                                                                                                                                                       |
| 101 | +                            | <a href="#">YCR</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 102 |                              | <a href="#">d.b</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 103 | をする                          | <a href="#">DeveauP</a> , <a href="#">FisherDisinformation</a> , <a href="#">Frank</a>                                                                                                                                                                                                                                                                                                                                                                                                |

|     |                       |                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 104 | クラスとモード               | <a href="#">Frank</a> , <a href="#">Steve_Corrin</a>                                                                                                                                                                                                                                                                                                                                                                          |
| 105 |                       | <a href="#">David Robinson</a> , <a href="#">egnha</a> , <a href="#">Frank</a> , <a href="#">ikashnitsky</a> , <a href="#">RamenChef</a> , <a href="#">Sumedh</a>                                                                                                                                                                                                                                                             |
| 106 | をとってする                | <a href="#">Charmgoggles</a> , <a href="#">Frank</a> , <a href="#">Jeromy Anglim</a> , <a href="#">SymbolixAU</a> , <a href="#">user2100721</a>                                                                                                                                                                                                                                                                               |
| 107 | クラス                   | <a href="#">Frank</a> , <a href="#">Steve_Corrin</a>                                                                                                                                                                                                                                                                                                                                                                          |
| 108 | のみき                   | <a href="#">42-</a> , <a href="#">4444</a> , <a href="#">abhiieor</a> , <a href="#">cdrini</a> , <a href="#">dotancohen</a> , <a href="#">Frank</a> , <a href="#">Gregor</a> , <a href="#">kdopen</a> , <a href="#">Rich Scriven</a> , <a href="#">Thomas</a> , <a href="#">Uwe</a>                                                                                                                                           |
| 109 |                       | <a href="#">AkselA</a> , <a href="#">alistaire</a> , <a href="#">Angelo</a> , <a href="#">coatless</a> , <a href="#">David Leal</a> , <a href="#">Dean MacGregor</a> , <a href="#">Frank</a> , <a href="#">kneijenhuijs</a> , <a href="#">MichaelChirico</a> , <a href="#">scoa</a> , <a href="#">SymbolixAU</a> , <a href="#">takje</a> , <a href="#">theArun</a> , <a href="#">thelatemail</a>                              |
| 110 | クラスPOSIXctおよびPOSIXlt  | <a href="#">AkselA</a> , <a href="#">alistaire</a> , <a href="#">coatless</a> , <a href="#">Frank</a> , <a href="#">MichaelChirico</a> , <a href="#">SymbolixAU</a> , <a href="#">thelatemail</a>                                                                                                                                                                                                                             |
| 111 | と                     | <a href="#">Andras Deak</a> , <a href="#">Andrew Bryk</a> , <a href="#">coatless</a> , <a href="#">Hack-R</a> , <a href="#">JGreenwell</a> , <a href="#">Pankaj Sharma</a> , <a href="#">Steve_Corrin</a> , <a href="#">μ Muthupandian</a>                                                                                                                                                                                    |
| 112 | モデル                   | <a href="#">Andrew Bryk</a> , <a href="#">Steve_Corrin</a>                                                                                                                                                                                                                                                                                                                                                                    |
| 113 | グラフ                   | <a href="#">L.V.Rao</a>                                                                                                                                                                                                                                                                                                                                                                                                       |
| 114 |                       | <a href="#">loki</a>                                                                                                                                                                                                                                                                                                                                                                                                          |
| 115 | プログラミング               | <a href="#">Karolis Koncevičius</a>                                                                                                                                                                                                                                                                                                                                                                                           |
| 116 |                       | <a href="#">42-</a> , <a href="#">Benjamin</a> , <a href="#">David Leal</a> , <a href="#">etienne</a> , <a href="#">Frank</a> , <a href="#">MichaelChirico</a> , <a href="#">PAC</a>                                                                                                                                                                                                                                          |
| 117 |                       | <a href="#">alistaire</a> , <a href="#">Angelo</a> , <a href="#">Frank</a> , <a href="#">gitblame</a> , <a href="#">Hendrik</a> , <a href="#">scoa</a>                                                                                                                                                                                                                                                                        |
| 118 | のパッケージでパイプリーテを<>どのように | <a href="#">RobertMc</a>                                                                                                                                                                                                                                                                                                                                                                                                      |
| 119 |                       | <a href="#">42-</a> , <a href="#">Axeman</a> , <a href="#">Hack-R</a> , <a href="#">Marcin Kosiński</a>                                                                                                                                                                                                                                                                                                                       |
| 120 |                       | <a href="#">beetroot</a> , <a href="#">ikashnitsky</a> , <a href="#">loki</a> , <a href="#">maRtin</a>                                                                                                                                                                                                                                                                                                                        |
| 121 |                       | <a href="#">Batanichek</a> , <a href="#">FisherDisinformation</a> , <a href="#">Matt Sandgren</a> , <a href="#">Robert</a> , <a href="#">russellpierce</a> , <a href="#">Tensibai</a>                                                                                                                                                                                                                                         |
| 122 | モデル                   | <a href="#">Amstell</a> , <a href="#">Ben Bolker</a> , <a href="#">Carl</a> , <a href="#">Carlos Cinelli</a> , <a href="#">David Robinson</a> , <a href="#">fortune_p</a> , <a href="#">Frank</a> , <a href="#">highBandWidth</a> , <a href="#">ikashnitsky</a> , <a href="#">jaySf</a> , <a href="#">Robert</a> , <a href="#">russellpierce</a> , <a href="#">thelatemail</a> , <a href="#">USER_1</a> , <a href="#">WAF</a> |

|     |                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----|---------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 123 |                                             | <a href="#">CptNemo</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 124 |                                             | <a href="#">dayne</a> , <a href="#">Frank</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 125 | のデータをプレーン<br>テキストファイル<br>CSV、TSVなどでみ<br>きする | <a href="#">a.powell</a> , <a href="#">Aaghaz Hussain</a> , <a href="#">abhiieor</a> , <a href="#">Alex</a> , <a href="#">alistic</a> , <a href="#">Andrea Cirillo</a> , <a href="#">bartektartanus</a> , <a href="#">Carl Witthoft</a> , <a href="#">Carlos Cinelli</a> , <a href="#">catastrophic-failure</a> , <a href="#">cdrini</a> , <a href="#">Charmgoggles</a> , <a href="#">Crops</a> , <a href="#">DaveRGP</a> , <a href="#">David Arenburg</a> , <a href="#">Dawny33</a> , <a href="#">Derwin McGeary</a> , <a href="#">EDi</a> , <a href="#">Eric Lecoutre</a> , <a href="#">FoldedChromatin</a> , <a href="#">Frank</a> , <a href="#">Gavin Simpson</a> , <a href="#">gitblame</a> , <a href="#">Hairizuan Noorazman</a> , <a href="#">herbaman</a> , <a href="#">ikashnitsky</a> , <a href="#">Jaap</a> , <a href="#">Jeromy Anglim</a> , <a href="#">JHowIX</a> , <a href="#">joeyreid</a> , <a href="#">Jordan Kassof</a> , <a href="#">K.Daisey</a> , <a href="#">kitman0804</a> , <a href="#">kneijenhuijs</a> , <a href="#">Imo</a> , <a href="#">loki</a> , <a href="#">Miha</a> , <a href="#">PAC</a> , <a href="#">polka</a> , <a href="#">russellpierce</a> , <a href="#">Sam Firke</a> , <a href="#">stats-hb</a> , <a href="#">Thomas</a> , <a href="#">Uwe</a> , <a href="#">zacdav</a> , <a href="#">zelite</a> , <a href="#">zx8754</a> |
| 126 |                                             | <a href="#">42-</a> , <a href="#">Benjamin</a> , <a href="#">dash2</a> , <a href="#">Frank</a> , <a href="#">Gavin Simpson</a> , <a href="#">JulioSergio</a> , <a href="#">kneijenhuijs</a> , <a href="#">Nathan Werth</a> , <a href="#">omar</a> , <a href="#">Rich Scriven</a> , <a href="#">Robert</a> , <a href="#">Steve_Corrin</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 127 | クラス                                         | <a href="#">42-</a> , <a href="#">Frank</a> , <a href="#">Gregor</a> , <a href="#">L.V.Rao</a> , <a href="#">Steve_Corrin</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 128 | いといのでデータを<br>する                             | <a href="#">Charmgoggles</a> , <a href="#">David Arenburg</a> , <a href="#">demonplus</a> , <a href="#">Frank</a> , <a href="#">Jeromy Anglim</a> , <a href="#">kneijenhuijs</a> , <a href="#">Imo</a> , <a href="#">Steve_Corrin</a> , <a href="#">SymbolixAU</a> , <a href="#">takje</a> , <a href="#">user2100721</a> , <a href="#">zx8754</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 129 | モデリング                                       | <a href="#">Ben Bolker</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 130 | と                                           | <a href="#">PAC</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 131 | テストの                                        | <a href="#">Stephen Leppik</a> , <a href="#">tenCupMaximum</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |