

 免費電子書

學習

# R Language

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#r

.....	1
<b>1: R</b> .....	<b>2</b>
.....	2
<b>Stack OverflowR Docs</b> .....	<b>2</b>
<b>R</b> .....	<b>2</b>
Examples.....	2
R.....	2
Windows.....	2
<b>Windows</b> .....	<b>2</b>
<b>OSX / macOS</b> .....	<b>2</b>
1.....	2
2.....	3
<b>DebianUbuntu</b> .....	<b>3</b>
<b>Red HatFedora</b> .....	<b>3</b>
<b>Archlinux</b> .....	<b>3</b>
.....	3
.....	3
R.....	4
.....	4
R.....	4
.....	5
<b>R</b> .....	<b>6</b>
<b>2: *</b> .....	<b>8</b>
.....	8
<b>*apply</b> .....	<b>8</b>
Examples.....	8
apply.....	8
.....	9
`data.frames` `lapply` `mapply`.....	10
.....	11

<b>lapply</b>	<b>11</b>
<b>lapply</b>	11
<b>sapply</b>	11
<b>mapply</b>	11
.....	12
.....	12
<b>3: .Rprofile</b>	<b>14</b>
.....	14
<b>Examples</b>	14
<b>.Rprofile -</b>	14
<b>R</b>	<b>14</b>
.....	14
.....	14
.....	14
.....	14
<b>CRAN</b>	<b>14</b>
.....	14
.....	15
.....	15
.....	15
.....	15
<b>.Rprofile</b>	15
.....	15
.....	15
.....	15
<b>4: data.table</b>	<b>17</b>
.....	17
.....	17
.....	18
.....	18
.....	18
<b>Examples</b>	18

data.table	18
	18
	19
<b>data.frame</b>	19
<b>data.table</b>	19
	20
	20
	20
	20
data.table	21
<b>.SD</b>	21
<b>.SDcols</b>	21
<b>.N</b>	22
data.framedata.table	22
	22
<b>data.framedata.table</b>	23
data.table	24
<b>5: Date</b>	26
	26
	26
	26
	26
Examples	26
	26
	26
	28
<b>6: dplyr</b>	29
	29
Examples	29
dplyr	29

..... 29

..... 30

..... 31

..... 32

..... 33

..... 33

..... 33

..... 34

..... 36

dplyr::filter() - ..... 36

dplyr::distinct() - ..... 37

  >..... 37

  dplyrNSE..... 38

**7: GGLOT2**..... **40**

..... 40

Examples..... 40

  ..... 40

  ..... 40

  ..... 44

  ..... 46

  ..... 46

  ..... 46

  ..... 46

  ..... 46

  ..... 46

  ..... 46

  ..... 46

  ..... 46

  qplot..... 46

**8: GPU**..... **47**

..... 47

Examples.....	47
gpuR gpuMatrix.....	47
gpuR vclMatrix.....	47
<b>9: JSON.....</b>	<b>48</b>
Examples.....	48
JSON/R.....	48
<b>10: lubridate.....</b>	<b>50</b>
.....	50
.....	50
Examples.....	50
lubridate.....	50
.....	<b>51</b>
.....	<b>51</b>
.....	51
.....	51
Lubridate.....	52
lubridate.....	52
.....	53
.....	53
.....	55
.....	55
.....	56
<b>11: Meta.....</b>	<b>57</b>
.....	57
Examples.....	57
.....	57
.....	57
.....	<b>57</b>
.....	<b>57</b>
.....	<b>57</b>
.....	<b>57</b>

<b>12: R MarkdownRStudio</b> .....	<b>58</b>
.....	58
Examples .....	58
.....	58
.....	58
.....	59
.....	59
.....	60
.....	60
.....	61
.....	62
<b>13: R memento</b> .....	<b>63</b>
.....	63
Examples .....	63
.....	63
.....	63
.....	63
<b>Dataframes</b> .....	<b>63</b>
.....	63
.....	63
.....	64
.....	64
<b>14: RCPP</b> .....	<b>66</b>
Examples .....	66
.....	66
Rcpp .....	66
Rcpp .....	67
.....	68
<b>15: RESTful R</b> .....	<b>69</b>
.....	69

Examples.....	69
opencpu.....	69
<b>16: RMarkdownknitr.....</b>	<b>70</b>
.....	70
.....	70
.....	70
.....	70
Examples.....	71
Rstudio.....	71
ioslides.....	72
<b>17: RMD.....</b>	<b>75</b>
.....	75
.....	75
Examples.....	75
.....	75
.....	77
.....	77
<b>18: RODBC.....</b>	<b>78</b>
Examples.....	78
RODBCExcel.....	78
SQL Server Management.....	78
.....	78
<b>19: roxygen2.....</b>	<b>79</b>
.....	79
Examples.....	79
roxygen2.....	79
<b>roxygen2.....</b>	<b>79</b>
.....	79
<b>20: R.....</b>	<b>81</b>
.....	81
Examples.....	81



`grep` .....	81
<b>21: R -</b> .....	<b>83</b>
Examples .....	83
.....	83
NA .....	83
.....	83
<b>22: R</b> .....	<b>85</b>
.....	85
Examples .....	85
S3 .....	85
<b>23: R</b> .....	<b>87</b>
Examples .....	87
.....	87
<b>24: RLaTeXknitr</b> .....	<b>90</b>
.....	90
.....	90
.....	90
Examples .....	90
KnitrLatexR .....	90
RLatexKnitr .....	91
RTLaTeXKnitr .....	91
<b>25: RI / O</b> .....	<b>93</b>
Examples .....	93
RdsRDataRda .....	93
Enviromments .....	93
<b>26: R</b> .....	<b>94</b>
Examples .....	94
RRPDFPMF .....	94
<b>27: Spark APISparkR</b> .....	<b>95</b>
.....	95
Examples .....	95

Spark.....	95
RSpark.....	95
Spark Cluster.....	95
.....	95
RDD.....	95
.....	95
csv.....	96
<b>28: sqldf.....</b>	<b>97</b>
Examples.....	97
.....	97
<b>29: strsplit.....</b>	<b>99</b>
.....	99
Examples.....	99
.....	99
<b>30: tidyverse.....</b>	<b>100</b>
Examples.....	100
tbl_df.....	100
tidyverse.....	100
<b>tidyverse.....</b>	<b>100</b>
.....	100
.....	101
<b>31: Web.....</b>	<b>102</b>
.....	102
.....	102
Examples.....	102
rvest.....	102
rvest.....	103
<b>32: xgboost.....</b>	<b>104</b>
Examples.....	104
xgboost.....	104
<b>33:.....</b>	<b>107</b>
.....	

Examples.....	107
foreach.....	107
.....	107
.....	109
mcparrallelDo.....	109
.....	<b>109</b>
.....	<b>109</b>
<b>34:</b> .....	<b>111</b>
Examples.....	111
.....	111
proc.time.....	111
.....	112
.....	113
microbenchmark.....	114
<b>35: CSVTSV</b> .....	<b>116</b>
.....	116
.....	116
.....	116
Examples.....	116
.csv.....	116
<b>R</b> .....	<b>116</b>
.....	117
.....	<b>117</b>
data.table.....	117
.....	118
.tsvR.....	118
.csv.....	119
<b>R</b> .....	<b>119</b>
.....	<b>119</b>
csv.....	119
.....	.....

<b>R</b> .....	<b>120</b>
<b>readr</b> .....	<b>120</b>
<b>36: data.table</b> .....	<b>121</b>
.....	121
.....	121
.....	121
Examples.....	121
data.table - I.....	121
data.table - II.....	122
<b>37: devtools</b> .....	<b>124</b>
.....	124
.....	124
Examples.....	124
.....	124
.....	<b>124</b>
.....	<b>125</b>
.....	<b>125</b>
1.....	125
2.....	125
.....	<b>125</b>
.....	<b>125</b>
Github.....	125
CRAN.....	126
.....	126
.....	<b>126</b>
.....	<b>126</b>
<b>38: hclust</b> .....	<b>127</b>
.....	127
.....	127
Examples.....	127

1 - hclust.....	127
2 - hclust.....	131
<b>39: igraph.....</b>	<b>134</b>
Examples.....	134
.....	134
<b>40: RMarkdown.....</b>	<b>136</b>
Examples.....	136
.....	136
LaTeX Preamble.....	138
.....	139
R-markdown.....	139
R-markdown.....	140
R-markdown.....	140
R-markdown.....	141
<b>41: S4.....</b>	<b>143</b>
.....	143
Examples.....	143
.....	143
<b>42: stringi.....</b>	<b>153</b>
.....	153
Examples.....	153
.....	153
.....	153
.....	154
.....	154
<b>43: texreg.....</b>	<b>155</b>
.....	155
.....	155
.....	155
Examples.....	155
.....	155

<b>44:</b>	<b>157</b>
.....	157
Examples.....	157
.....	157
<b>45:</b>	<b>164</b>
.....	164
Examples.....	164
GLCM.....	164
.....	166
<b>46: I / O.....</b>	<b>168</b>
.....	168
Examples.....	168
.....	168
<b>47:</b>	<b>170</b>
Examples.....	170
.....	170
<b>48:</b>	<b>172</b>
.....	172
.....	172
Examples.....	172
.....	172
.....	172
.....	172
.....	172
<b>49:</b>	<b>173</b>
.....	173
.....	173
Examples.....	173
.....	173

.....	173
<b>50:</b> .....	<b>178</b>
Examples.....	178
.....	178
split-apply-combinesplit.....	179
<b>51:</b> .....	<b>182</b>
Examples.....	182
.....	182
<b>52:</b> .....	<b>183</b>
Examples.....	183
.....	183
<b>53:</b> .....	<b>185</b>
Examples.....	185
.....	185
SEQ.....	185
.....	186
.....	187
rep.....	188
.....	189
<b>54:</b> .....	<b>191</b>
Examples.....	191
.....	191
<b>55:</b> .....	<b>192</b>
Examples.....	192
Rcpp.....	192
.....	192
<b>56: HashMap</b> .....	<b>194</b>
Examples.....	194
.....	194
.....	194
.....	194
.....	194

.....	195
.....	195
.....	196
.....	197
listenv .....	198
<b>57: R .....</b>	<b>199</b>
.....	199
.....	199
.....	<b>199</b>
Examples .....	199
.....	199
dput()dput() dget() .....	199
.....	199
<b>58: .....</b>	<b>200</b>
.....	200
Examples .....	200
.....	200
<b>59: .....</b>	<b>201</b>
.....	201
.....	201
.....	<b>201</b>
.....	<b>202</b>
Examples .....	202
.....	202
.....	203
factor factor_approach .....	204
ifelse ifelse_approach .....	204
list_approach .....	204
.....	204
.....	205
.....	206
.....	.....



.....	210
.....	210
<b>60:</b> .....	<b>212</b>
Examples .....	212
viridis - .....	212
RColorBrewer .....	213
.....	213
- .....	213
R .....	213
.....	213
<b>61: R</b> .....	<b>214</b>
Examples .....	214
.....	214
.....	214
RStudio .....	215
.....	216
<b>62:</b> .....	<b>217</b>
Examples .....	217
.zip .....	217
.zip .....	217
.tar .....	217
.tar .....	217
.zip .....	217
<b>63: &lt;&gt;</b> .....	<b>219</b>
.....	219
Examples .....	219
.....	219
<b>64:</b> .....	<b>220</b>
.....	220
.....	220
.....	220

Examples.....	220
.....	220
.....	220
.....	221
.....	221
<b>R.....</b>	<b>222</b>
<b>tidyr.....</b>	<b>222</b>
<b>data.table.....</b>	<b>222</b>
<b>65: .....</b>	<b>223</b>
.....	223
Examples.....	223
map.....	223
Google Viz50.....	227
.....	228
LeafletHTML.....	230
ShinyLeaflet.....	231
<b>66: I / Oshapefile.....</b>	<b>234</b>
.....	234
Examples.....	234
Shapefile.....	234
<b>67: .....</b>	<b>235</b>
Examples.....	235
.....	235
<b>68: .....</b>	<b>237</b>
.....	237
.....	237
Examples.....	237
.....	237
Matplot.....	240
.....	246
.....	248

par()	248
layout()	249
.....	250
.....	251
R_Plots	252
<b>69: I / OExcelSASSPSSStata</b>	<b>254</b>
Examples	254
rio	254
Excel	254
<b>xlsxexcel</b>	<b>254</b>
<b>XLconnectExcel</b>	<b>255</b>
<b>openxlsxexcel</b>	<b>255</b>
<b>readxlsxexcel</b>	<b>256</b>
<b>RODBCexcel</b>	<b>256</b>
<b>gdataexcel</b>	<b>257</b>
StataSPSSSAS	257
.....	258
<b>70:</b>	<b>260</b>
.....	260
.....	260
Examples	260
.....	260
.....	262
.....	263
.....	264
.....	264
.....	266
.....	266
.....	267
.....	267
<b>71:</b>	<b>269</b>
.....	

.....	269
.....	269
.....	<b>269</b>
Examples.....	269
.....	269
<b>CRAN</b> .....	<b>269</b>
<b>Bioconductor</b> .....	<b>270</b>
.....	270
GitHub.....	270
CLI - pacman.....	271
.....	272
<b>72: /</b> .....	<b>273</b>
.....	273
.....	273
tryCatch.....	273
.....	273
“”.....	273
Examples.....	273
tryCatch.....	273
tryCatchtryCatch.....	273
.....	274
.....	275
<b>73:</b> .....	<b>276</b>
Examples.....	276
.....	276
<b>74:</b> .....	<b>278</b>
Examples.....	278
.....	278
.....	279
<b>75:</b> .....	<b>281</b>
.....	.....

Examples.....	281
.....	281

**76: ..... 282**

.....	282
.....	<b>282</b>
.....	<b>282</b>

Examples.....	283
---------------	-----

.....	283
-------	-----

For.....	284
----------	-----

.....	284
-------	-----

.....	284
-------	-----

vapply.....	284
-------------	-----

colMeans.....	284
---------------	-----

.....	284
-------	-----

.....	284
-------	-----

while.....	285
------------	-----

repeat.....	285
-------------	-----

break.....	286
------------	-----

**77: ..... 288**

.....	288
-------	-----

Examples.....	288
---------------	-----

.....	288
-------	-----

**78: ..... 289**

Examples.....	289
---------------	-----

.....	289
-------	-----

**79: I / O..... 291**

.....	291
-------	-----

.....	<b>291</b>
-------	------------

Examples.....	291
---------------	-----

MySQL.....	291
------------	-----

.....	.....
-------	-------

291	
.....	<b>291</b>
MongoDB.....	291
<b>80:</b> .....	<b>292</b>
.....	292
Examples.....	292
.....	292
.....	<b>292</b>
.....	293
<b>Gapminder</b> .....	<b>293</b>
<b>2015 -</b> .....	<b>293</b>
.....	293
.....	<b>293</b>
.....	293
.....	<b>293</b>
<b>81:</b> .....	<b>294</b>
.....	294
Examples.....	294
data.frame.....	294
.....	295
<b>[ [ [ \$</b> .....	<b>295</b>
data[rows, columns].....	295
.....	295
.....	296
.....	296
.....	296
.....	296
.....	296
data[columns].....	296
data[[one_column]].....	297
\$.....	297
\$.....	297
.....	

.....298

.....298

data.frames.....298

.....298

.....298

.....298

.....299

do.call.....299

data.frame.....300

.....301

**82: .....303**

Examples.....303

  N-gram Word.....303

**83: .....307**

Examples.....307

  aov.....307

  Anova.....307

**84: .....309**

.....309

.....309

.....309

.....309

.....309

Examples.....310

.....310

.....310

.....310

.....311

**85: POSIXctPOSIXlt.....312**

.....312

.....

.....	312
.....	312
.....	312
Examples.....	312
.....	312
.....	312
.....	313
.....	313
.....	313
.....	313
<b>86:</b> .....	<b>315</b>
.....	315
Examples.....	315
.....	315
ts.....	316
<b>87: R</b> .....	<b>318</b>
Examples.....	318
Windows.....	318
<b>88: R</b> .....	<b>319</b>
.....	319
Examples.....	319
R.....	319
installr PackageR.....	319
.....	320
.....	323
R.....	324
<b>89:</b> .....	<b>325</b>
.....	325
Examples.....	325
barplot.....	325



<b>90:</b>	<b>332</b>
.....	332
.....	332
.....	332
.....	<b>332</b>
.....	<b>332</b>
Examples.....	332
.....	332
.....	332
.....	<b>332</b>
.....	<b>333</b>
.....	<b>333</b>
.....	<b>333</b>
.....	333
.....	334
<b>91:</b>	<b>336</b>
Examples.....	336
.....	336
<b>92:</b>	<b>337</b>
.....	337
.....	337
Examples.....	337
.....	337
.....	337
.....	337
.....	337
<b>93:</b>	<b>338</b>
.....	338
.....	338

.....	338
.....	338
.....	338
.....	338
.....	338
Examples .....	338
.....	338
.....	338
.....	339
.....	339
.....	339
"YYYYMMDD" .....	339
.....	340
.....	340
R .....	341
PerlPOSIX .....	341
/ .....	341
<b>94:</b> .....	<b>342</b>
Examples .....	342
.....	342
.....	343
.....	344
.....	345
.....	346
<b>95:</b> .....	<b>347</b>
.....	347
Examples .....	347
.....	347
.....	347
<b>96:</b> .....	<b>348</b>
Examples .....	348
.....	.....

<b>::</b> .....	<b>348</b>
1.....	348
2.....	349
3 <sup>'''</sup> .....	349
4.....	350
5.....	351
6.....	352
7.....	353
8cor.....	354
heatmap.....	356
<b>97:</b> .....	<b>362</b>
.....	362
Examples.....	362
R.....	362
<b>98:</b> .....	<b>363</b>
Examples.....	363
randomForestSRC.....	363
-.....	364
Kaplan Meier.....	365
<b>99: R</b> .....	<b>368</b>
.....	368
Examples.....	368
.....	368
<b>R</b> .....	<b>368</b>
.....	368
<b>100:</b> .....	<b>370</b>
.....	370
Examples.....	370
.....	370
<b>101:</b> .....	<b>372</b>

Examples.....	372
XY.....	372
.shp.....	373
<b>rgdal.....</b>	<b>373</b>
.....	373
<b>TMAP.....</b>	<b>373</b>
<b>102: .....</b>	<b>375</b>
.....	375
Examples.....	375
.....	375
.....	375
<b>103: &gt;.....</b>	<b>378</b>
.....	378
.....	378
.....	378
.....	378
.....	378
%>%.....	378
.....	378
.....	378
.....	378
Examples.....	378
.....	378
.....	379
<>.....	380
\$.....	380
dplyrggplot2.....	381
T>.....	381
<b>104: .....</b>	<b>383</b>
.....	383
.....	383
Examples.....	383
boxplot(graphics).....	383

<b>boxplotSepal.Length</b> .....	<b>383</b>
<b>Boxplot</b> .....	<b>384</b>
.....	384
.....	385
.....	386
.....	386
.....	387
<b>plot=FALSE</b> .....	<b>387</b>
boxplot .....	387
.....	387
.....	387
.....	388
.....	388
.....	388
.....	388
.....	388
<b>105:</b> .....	<b>390</b>
Examples .....	390
.....	390
.....	390
.....	390
.....	391
.....	391
.....	391
<b>106: R</b> .....	<b>392</b>
Examples .....	392
RCurl .....	392
<b>107:</b> .....	<b>393</b>
.....	393
.....	393
Examples .....	393

mtcars.....	393
.....	395
.....	396
.....	397
.....	399
".....	400
<b>108:</b> .....	<b>402</b>
.....	402
.....	402
Examples.....	402
.....	402
NA.....	402
NA.....	402
TRUE / FALSE/NA.....	403
.....	404
.....	<b>404</b>
.....	<b>404</b>
.....	<b>404</b>
<b>109:</b> .....	<b>405</b>
.....	405
Examples.....	405
R.....	405
dplyr.....	406
data.table.....	407
<b>110:</b> .....	<b>409</b>
.....	409
Examples.....	409
.....	409
<b>111:</b> .....	<b>411</b>
.....	411
Examples.....	411
ArimaAR1.....	411

<b>112:</b>	<b>420</b>
.....	420
.....	<b>420</b>
Examples	420
`rle`	420
R	420
data.table	421
.....	422
<b>113: + eval</b>	<b>423</b>
.....	423
Examples	423
.....	423
<b>114:</b>	<b>424</b>
.....	424
.....	424
.....	<b>424</b>
Examples	424
.....	424
<b>115: RODE</b>	<b>425</b>
.....	425
.....	425
.....	425
Examples	425
.....	425
Lotka-Volterra	427
ODE - R	428
ODE - C	429
ODE - fortran	430
ODE - .....	432
<b>116:</b>	<b>434</b>
.....	434

Examples.....	434
.....	434
.....	<b>434</b>
.....	<b>434</b>
.....	434
“”.....	435
.....	<b>435</b>
//.....	436
/.....	436
<b>117:</b> .....	<b>437</b>
Examples.....	437
.....	437
.....	438
<b>118:</b> .....	<b>439</b>
.....	439
Examples.....	439
.....	439
.....	440
.....	440
.....	<b>440</b>
.....	<b>441</b>
<b>119:</b> .....	<b>442</b>
Examples.....	442
.....	442
.....	442
.....	443
.....	<b>443</b>
.....	<b>443</b>
.....	<b>443</b>
<b>Matrix</b> .....	<b>443</b>
“”.....	444



<b>120:</b> .....	<b>445</b>
.....	445
Examples .....	445
.....	445
.....	445
.....	446
.....	446
.....	447
.....	447
<b>121:</b> .....	<b>448</b>
.....	448
Examples .....	448
.....	448
.....	<b>448</b>
.....	<b>448</b>
.....	<b>448</b>
<b>122:</b> .....	<b>450</b>
.....	450
Examples .....	450
.....	450
.....	450
<b>123: R</b> .....	<b>452</b>
.....	452
.....	452
Examples .....	452
R .....	452
<b>R</b> .....	<b>452</b>
<b>R</b> .....	<b>452</b>
Linux/ .....	452
.....	453
littler .....	453
.....	

453	
R	453
apt-getDebianUbuntu	453
.rittler	453
shebangedlittler	454
<b>124:</b>	<b>455</b>
.....	455
.....	455
.....	<b>455</b>
Examples	455
.....	455
.....	455
NAs	455
<b>125: tidy</b>	<b>457</b>
.....	457
Examples	457
spread	457
gather	457
h21	457
<b>126:</b>	<b>459</b>
Examples	459
.....	459
.....	<b>459</b>
.....	<b>459</b>
ui.R	459
server.R	459
.....	459
.....	460
.....	461
.....	462
1	462

2.....	462
.....	462
.....	464
.....	<b>464</b>
<b>Reactive Log Visualizer.....</b>	<b>464</b>
<b>127: .....</b>	<b>466</b>
.....	466
.....	466
Examples.....	466
.....	466
.....	<b>466</b>
.....	<b>466</b>
.....	<b>467</b>
.....	<b>467</b>
.....	468
<b>128: .....</b>	<b>469</b>
Examples.....	469
.....	469
.....	469
.....	470
010.....	470
01.....	470
100.5.....	470
0.2.....	470
3105.....	470
100.8.....	470
2.....	470
1.5.....	471
01.....	471
15.....	471
a = 1b = 0.5Beta.....	471

Gamma3= 0.5.....	471
Cauchy01.....	471
01.....	471
Weibull0.51.....	471
Wilcoxon1020.....	472
53.....	472
<b>129:</b> .....	<b>473</b>
.....	473
Examples.....	473
- .....	473
<b>130:</b> .....	<b>475</b>
.....	475
Examples.....	475
dplyr.....	475
<b>131:</b> .....	<b>477</b>
.....	477
.....	477
Examples.....	477
.....	477
.....	477
.....	477
.....	<b>479</b>

---

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](mailto:info@zzzprojects.com)

---

# 1: R

---

## Stack Overflow R Docs

◦

---

## R

- R◦
- ◦
- 10◦
- R arrays dimR""◦
- R◦ ◦ ◦
- 
- 

## Examples

R.

[RStudio](#)R. RStudioR◦

## Windows

[Visual Studio](#) 2015 Update 3 [RR Tools](#) IntelliSense◦ R.

---

## Windows

1. [CRAN](#)R for WindowsR.
2. ◦
3. ◦
4. ◦

---

## OSX / macOS

### 1

0. [XQuartz](#)

- 1.

CRAN.R.

2. ◦
3. ◦

RR-MacGUI ◦ GUIR.app/ Applications / FolderDoc ◦ R.appR ◦ R  
/Library/Frameworks/R.Framework/Versions/ ◦ RStudioRGUIR ◦

## 2

1. <https://brew.sh/homebrewmacOS>
2. `brew install R`

MacR-SIG-Mac ◦

---

## DebianUbuntu

apt-getR ◦ CRAN ◦ CRAN"" ◦

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

CRANCRAN ◦ CRAN ◦ `install.packages()` ◦ Linux

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

---

## Red HatFedora

```
sudo dnf install R
```

---

## Archlinux

RExtra package repo ◦

```
sudo pacman -S r
```

ArchlinuxRArchWiki R ◦

```
"Hello World!"
```

◦

`help()?R` ◦ `help.search()??` ◦

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

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

<https://www.r-project.org/help.html>

## R

---

R◦ R

## R

RWindowsRGuiRGuiLinuxR

```
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.

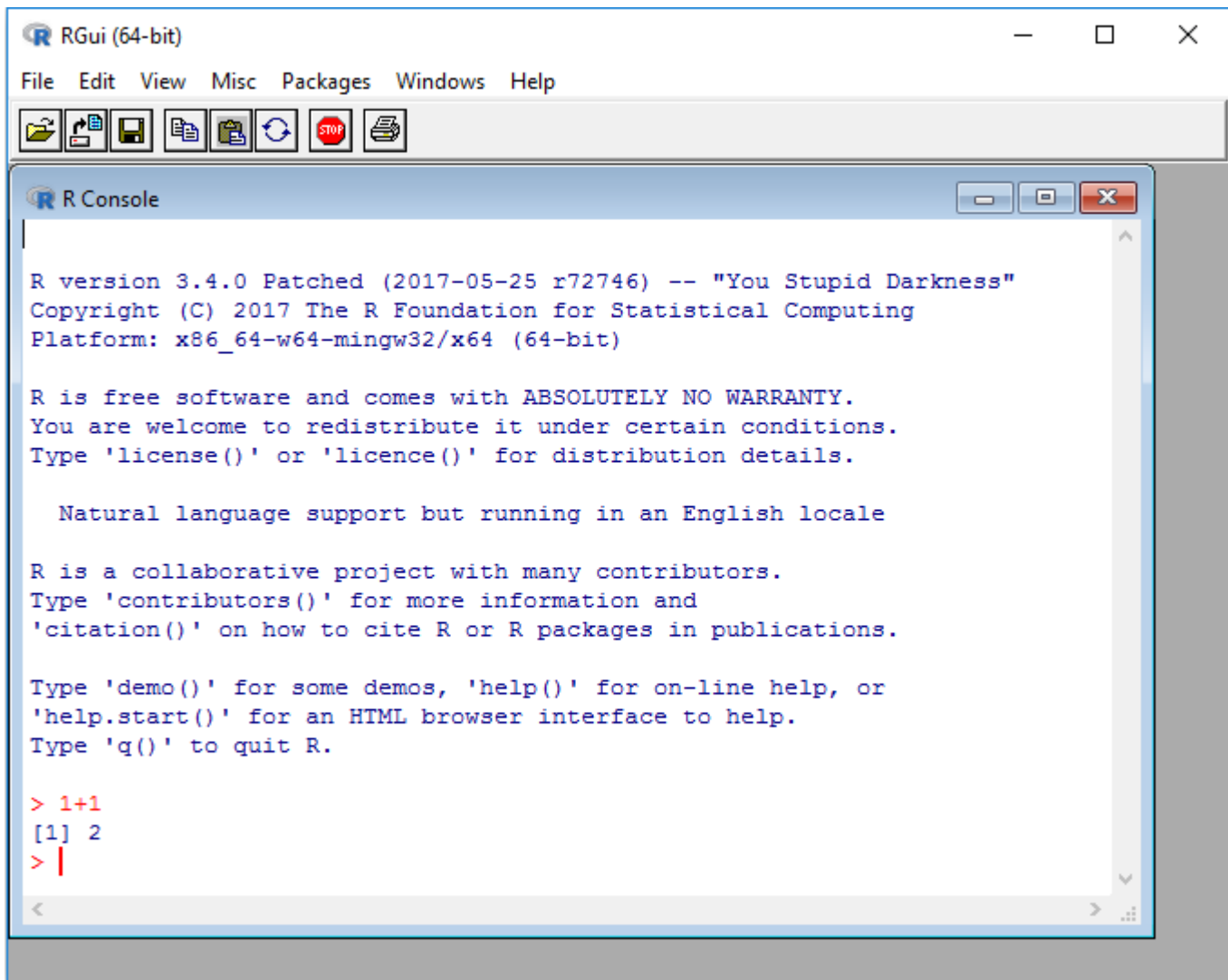
R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Information und 'citation()',
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.

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
> █
```

WindowsRGuiWindowsR





> R. R

```
1+1
```

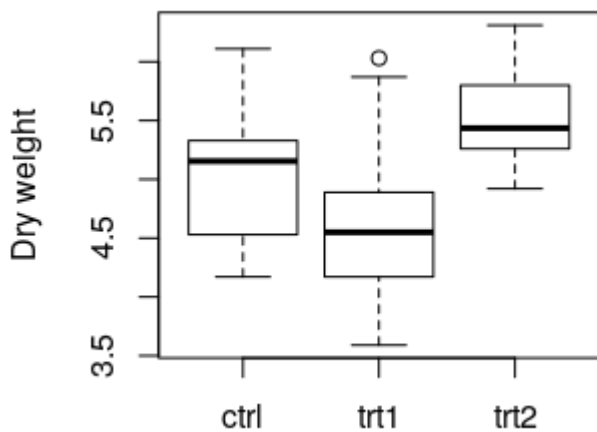
2. [1]R. 2.

R. PlantGrowth R

R##. ##R.

```
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")
```



```
data(PlantGrowth) PlantGrowth PlantGrowth PlantGrowth
```

- [CSVTSV](#)
- [I/O Excel SAS SPSS Stata](#)

```
str(PlantGrowth) PlantGrowth data.frame R data.frame 30 $weight num group Rfactor
```

```
anova(lm(...)) ANOVA weight ~ group "weight group group" R.data data = ...
```

```
Pr(>F) p = 0.01591 Tukey
```

```
boxplot(...) weight ~ group "group ylab = ...y"
```

```
q() Ctrl - D R
```

## R

```
• R RR
```

```
plants.R
```

```
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()
```

## R

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

```
plant_result.txt R
```

```
pngdev.offdev.off ◦ png("FILENAME", width = ..., height = ...)PNG ◦ dev.off() ◦ dev.off() ◦
```

[R https://riptutorial.com/zh-TW/r/topic/360/r](https://riptutorial.com/zh-TW/r/topic/360/r)

## 2: \*

\*applyfor° for\*apply

1. °
2. °
3. °

for\*apply°

for\*applyforfor°

### \*apply

\*apply°

apply	matrix data.framearray	
sapply	list	
lapply	list	list
vapply	`	
mapply	lists	list

“”。

## Examples

### apply

apply°

iris° iris3150°

```
> 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
```

° forRapply

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

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

- iris[1:4, 2, mean]
- 2 (rows); 1 (column)

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

## R colMeans/rowMeans

0.5 mean

```
> 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*

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

apply

apply \*apply

◦

- paste0()

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

- list.files() ◦

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

X

lapply

readRDS

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

for◦

◦ library()

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

## `data.frames` `lapply` `mapply`

◦

```
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)))
```

data.frame◦

```
#* 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)
```

magrittr

```
library(magrittr)
```

```

library(broom)
Combined <- lapply(1:4,
                  function(i) mtcars[sample(1:nrow(mtcars),
                                           size = nrow(mtcars),
                                           replace = TRUE), i] %>%
                    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", .))

```

## lapplysapplymapply

R.

- `lapply()` =
- `sapply()` = `lapply()`
  - `vapply()` = `sapply()`
- `mapply()` = `lapply()` ◦ `sapply()`
  - `Map()` `mapply()` `SIMPLIFY = FALSE`

### lapply

`lapply()`

- `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]]))

```

### sapply

`sapply()`

```

# 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() lapply() lapply() m

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

## ◦ Hadley Wickham **Functionals**

```
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
}
```

randomise f **Uniform** ◦ set.seed

```
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
```

base::lapply f x ◦ ...f meanna.rm

```
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]]  
# [1] 4
```

\* <https://riptutorial.com/zh-TW/r/topic/3567/--->

# 3: .Rprofile

Efficient R

## Examples

### .Rprofile -

```
.RprofileR.RprofileR.Rprofile ◦ RRprofile.site ◦ Rprofile.site.RprofileR◦  
  
RStudioRStudio.Rprofile ◦
```

---

### .Rprofile◦

## R

```
# 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.
```

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

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

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

## CRAN

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

R◦

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

---

R◦ .Last.value

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

**.RprofileRR◦**

---

◦

---

```
help(Startup)◦ Profile◦ Rprofile Profile.site◦ R${RHOME}/etc◦ RenviroRenviro.site ~/.Renviro
```

◦

**.Rprofile**

```
# 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")
  }
}
```

```
# 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)
```

```
# 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)
```

[.Rprofile](https://riptutorial.com/zh-TW/r/topic/4166/-rprofile) <https://riptutorial.com/zh-TW/r/topic/4166/-rprofile>

## 4: data.table

Data.tableR. [data.table](#).

- DT[i, j, by]  
DT [whereselect | update | doby]
- DT[...][...]  
#chaining
- ##### Shortcuts, special functions and special symbols inside DT[...]
- ◦  
list
- J  
#inlist
- =  
#in j
- .N  
#in  
#in j
- ◦  
#in ji
- .SD  
#in j  
.SDcols
- .GRP  
#in j
- ◦  
#in jby
- V1V2.....  
j
- ##### Joins inside DT[...]
- DT1 [DT2onj]  
#joed
- ◦ \*  
DT2#external
- = .EACHI
- DT1 [DT2onj]
- DT1 [DT2onrollj]  
#joedon =
- ##### Reshaping, stacking and splitting
- DTid.varsmeasure.vars  
measure.vars = patterns...
- dcastDT
- rbindDT1DT2...  
stackdata.tables
- rbindlistDT\_listidcol  
data.tables
-

## DTby

### data.table

- ##### Some other functions specialized for data.tables
- foverlaps
- overlap joins
- 
- 
- fintersectfsetdiffunionfsetequaluniqueduplicatedanyDuplicated
- #set-theory operation
- uniqueN
- rowidvDTcols
- #colsID1.N
- rleidvDTcols
- colsID1.GRP
- shiftDTntype = c("lag" "lead")
- setordersetcolordersetnamesetkeysetindexsetattr

---

## data.table

```
# 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")
```

◦ - StackOverflow - ◦

---

data.table◦ library(data.table)data.table::freadfread◦ help("fread")?fread◦ ?data.table::fread◦

## Examples

### data.table

data.tableRdata.frame◦ class() "data.table" "data.frame" data.frame data.table◦ data.table◦

---

data.table

```
library(data.table)
```

```
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 data.table

```
sapply(DT, class)
#           x           y           z
# "character" "integer" "logical"
```

```
dt <- fread("my_file.csv")
```

read.csv fread°

## data.frame

data.table data.frame list data.table

```
# example data.frame
DF <- data.frame(x = letters[1:5], y = 1:5, z = (1:5) > 3)
# modification
setDT(DF)
```

<-DF° data.frame

```
sapply(DF, class)
#           x           y           z
# "factor" "integer" "logical"
```

## data.table

list data.frame data.table setDT data.table as.data.table° °

R as.data.table data.table °

```
mat <- matrix(0, ncol = 10, nrow = 10)

DT <- as.data.table(mat)
# or
```

```
DT <- data.table(mat)
```

DT[where, select|update|do, by] **data.table**◦

- “where”<sub>i</sub>
- “select | update | do”<sub>j</sub>

◦

```
mtcars = data.table(mtcars, keep.rownames = TRUE)
```

**j**:=

```
mtcars[, mpg_sq := mpg^2]
```

NULL

```
mtcars[, mpg_sq := NULL]
```

**:=**

```
mtcars[, `:=`(mpg_sq = mpg^2, wt_sqrt = sqrt(wt))]  
# or  
mtcars[, c("mpg_sq", "wt_sqrt") := .(mpg^2, sqrt(wt))]
```

```
mtcars[, c("mpg_sq", "mpg2_hp") := .(temp1 <- mpg^2, temp1/hp)]
```

LHS := RHS.()◦

```
vn = "mpg_sq"  
mtcars[, (vn) := mpg^2]
```

set

```
set(mtcars, j = "hp_over_wt", v = mtcars$hp/mtcars$wt)
```

**i**“where”

```
mtcars[1:3, newvar := "Hello"]  
# or  
set(mtcars, j = "newvar", i = 1:3, v = "Hello")
```

**data.frame**◦ **i**“join”◦

levels<-names<- ◦ **data.table**◦



setnames data.table data.frame setattr

```
# 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")
```

◦ ◦

```
# 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

### .SD

.SD data.table by

.SDlapply data.table

mtcars

```
mtcars = data.table(mtcars) # Let's not include rownames to keep things simpler
```

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 vs am gear carb ◦ mean ◦ ◦ .SDcols

### .SDcols

.SDcols data.table .SD

o

gear cyl gearcyl

```
# 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
```

mean o byo

```
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
```

- cols\_chosen o .SDcols
- .SDcols **column numbers** o mtcars[ , lapply(.SD, mean), .SDcols = c(1,3:7)]

---

## .N

.No

```
iris[, .(count=.N), by=Species]

# Species count
#1: setosa 50
#2: versicolor 50
#3: virginica 50
```

---

## data.frame data.table

data.table R data.frame matrix 2D array o A[rows, cols]o

matrix data.frame 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!
```

◦

```
ma[2:3, ] # \
df[2:3, ] # }---> returns the 2nd and 3rd rows
dt[2:3, ] # /
```

◦ ◦

```
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")] # /
```

```
mycols <- 2:3
ma[, mycols] # \
df[, mycols] # }---> returns the 2nd and 3rd columns
dt[, mycols, with = FALSE] # /

dt[, mycols] # ---> Raises an error
```

mycols ◦ dtmycols ◦

data.table ◦ data.frame ◦ dt ◦ dt[, 2:3] ◦ dt[, mycols] ◦ 2:3 ◦ mycols ◦

---

## data.frame ◦ data.table

data.frame ◦ data.table ◦ data.frame ◦

1. ◦
2. ◦
3. data.frame ◦ print.data.frame ◦
4. list ◦
5. data.frame ◦
6. data.table ◦

◦ [, ]

```
A[1:10, ]
A[A$var > 17, ] # A[var > 17, ] just works for data.table
```

◦ \$[[ ]]

```
A$var
colname <- 'var'
A[[colname]]
A[[1]]
```

```
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'))
```

• `data.frame` `row.names` `data.table` `key` `row.names` `data.table`

```
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.

```
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

### SETKEY pre 1.9.6

1.9.6 `data.table` • [201595 544] “setkey” “key =”

```
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` •

```
setkey(DT, y)
```

```
tables()
```

```
> tables()
      NAME NROW NCOL MB COLS  KEY
[1,] DT      5    3  1 x,y,z y
Total: 1MB
```

◦

```
#> DT
#   x y     z
#1: e 1  TRUE
#2: d 2  TRUE
#3: c 3 FALSE
#4: b 4 FALSE
#5: a 5 FALSE
```

### v1.9.6◦ data.table"on="◦ ◦

20171 “on”◦

keysetindex(DT, key.col)setindexv(DT, "key.col.string") DTdata.table◦ setindex(DT, NULL)◦

indices(DT)◦

“on”◦ ◦ ◦

yDT

```
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
```

[data.table](https://riptutorial.com/zh-TW/r/topic/849/data-table) <https://riptutorial.com/zh-TW/r/topic/849/data-table>

# 5: Date

- 

- `Date` 1970-01-01 `UNIX` ◦ ◦
- 
- ◦
- `POSIXct` `POSIXlt` ◦
- `sys.Date()` `Date`

- `lubridate` `ymd` `mdyas.Date` `Date`; `lubridate` ◦
- `data.table` `Date` `Date` `Datedouble` ◦

## Examples

```
Datesformat(date, format="%Y-%m-%d")POSIXct as.POSIXct()POSIXlt as.POSIXlt()
```

```
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"
```

?`strptime` ◦

as.Date()◦

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

as.Date()◦ %Y-%m-%d - - ◦

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

◦ "dm-yy" "dm-YYYY" "md-yy" "md-YYYY" "YYYY-md" "YYYY-dm" ◦ "-" "/" ◦ "1986116" "1986116" "1986116" "1986116" ◦ **as.Date**"YYYY-md"◦

"9-6-1962" "%d-%m-%Y" ◦

```
#
# It tries to interpret 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
>
```

◦ **as.Date**◦

%d	
%m	
%y	
%Y	4
%b	3
%B	

**format**

```
> as.Date("9-6-1962", format="%d-%m-%Y")
```

```
[1] "1962-06-09"  
>
```

◦

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

◦ %b ◦

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

'-'/' ◦ ◦

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

◦ NA ◦ %B

```
> 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 ◦ **origintz** ◦

**RDate** `as.Date()` ISO 8601 YYYY-MM-DD `strptime` style ◦

```
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"
```

**Date** <https://riptutorial.com/zh-TW/r/topic/9015/date>



---

# 6: dplyr

dplyr is an R package available on CRAN

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

dplyr version 0.5.0

- 
- 
- SQLite
- PostgreSQL/
- MySQL/ MariaDB
- BIGQUERY
- MonetDB
- 

## Examples

### dplyr

dplyr works with [data.frame](#), [data.table](#), [database](#), and [dplyrRcpp](#).

dplyr verbs: `filter`, `arrange`, `select`, `mutate`, `summarise`, `group_by`

- 
- - \$
  - . . .

`mtcars` is a `dplyr` `tbl_df` with rownames, converted to a `tibble`.

```
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
```

```
filter◦ data.frameTRUEFALSE
```

## 4 - 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
```

## ◦ 46- cyl5 - 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
```

```
filterslice◦ slice2data.frame◦
```

## 69

```
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
```

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

```
slice(mtcars_tbl, 6:9)slice(mtcars_tbl, 6:9)
```

```
n()
```

```
arrange◦ dplyrdata.frame◦◦
```

```
- 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
```

/arrange - mpg - 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
```

select° mpg disp wt qsec vsmtcars\_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
```

∴ dispvs carbcars

```
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
```

```
select(mtcars_tbl, ~(hp:qsec))
```

◦ starts\_with() ends\_with() contains() matches() num\_range() one\_of() everything() select ◦  
?select\_helpers?select ◦

```
select()◦
```

```
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
```

◦

```
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◦ dplyrdplyr **mutate**◦ data.frame ◦

```
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
```

weight\_poundsweight\_ton ◦ R mutate◦

transmutemutate

```
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
```

summarise◦ ◦

mpgdisp

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

# 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◦ ◦

cylmpgmeansd

```
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
```

```
carshpgear cylmpg gear mpg > 20hp > 75
```

```
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
```

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

◦ %>%dplyr◦

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

dplyrdplyr summarise\_all()◦

```
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

```
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> <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( ~ cyl)
```

```
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
```

```
helper ?select_helpers
```

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

```
funs
```

```
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
```

```
o
```

```
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
```

summarise\_if

meannumericcyl

```
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>
```

mean°

cylmean

```
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
```

**dplyr::filter()** 

```
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()` ▀

```
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
```

>

> `dplyr::mtcars` help("mtcars") **summarize**

```
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.

`summarize` ◦ `n()` ◦

```
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.

```

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

```

## dplyrNSE

dplyrNSE. Shiny. selectselect\_

```

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

```

interlazyeval

```

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
```

[dplyr https://riptutorial.com/zh-TW/r/topic/4250/dplyr](https://riptutorial.com/zh-TW/r/topic/4250/dplyr)

# 7: GGPLOT2

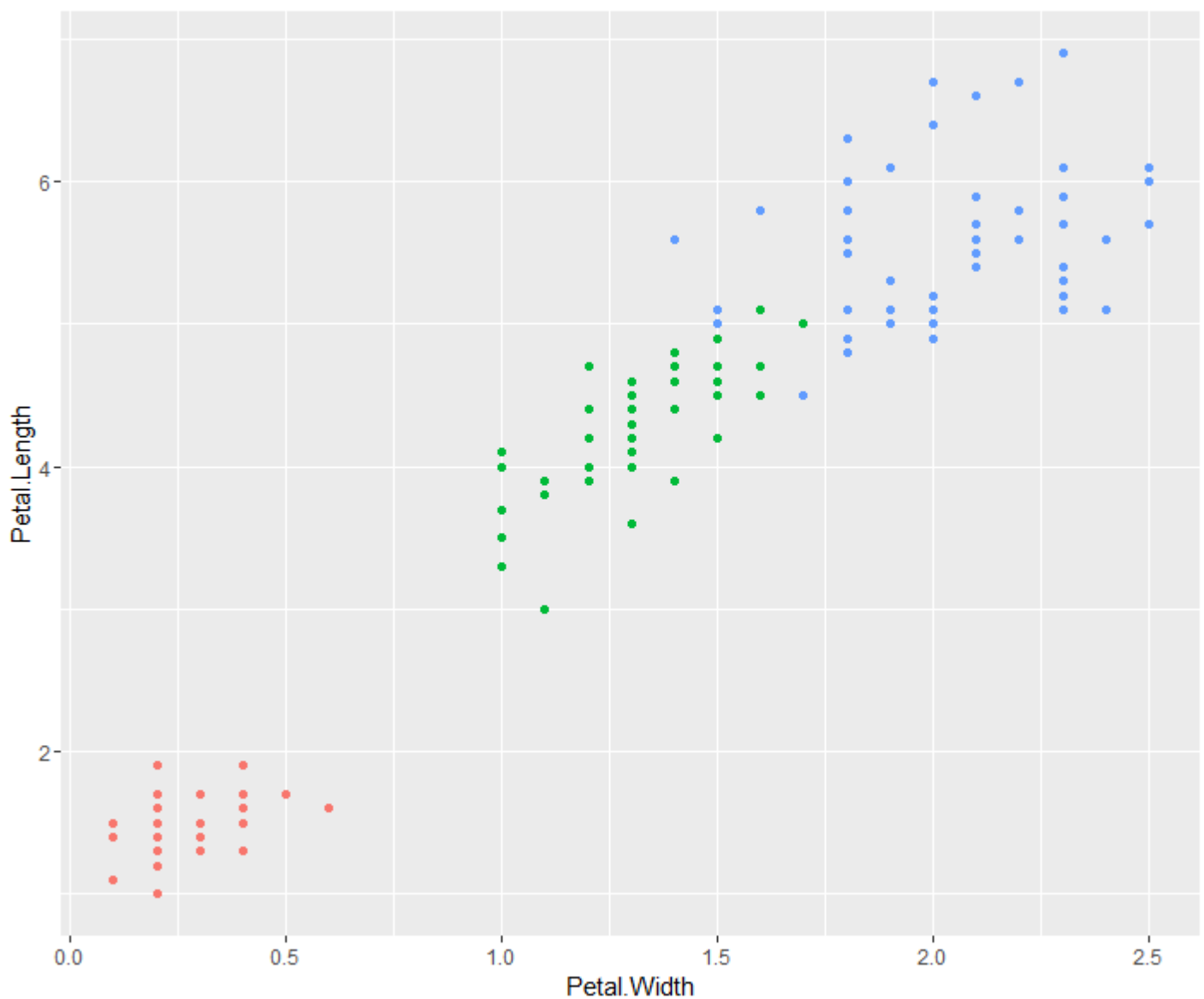
ggplot2 <http://ggplot2.tidyverse.org/> ◦

data.frame ◦

RStudio“ggplot2”◦

## Examples

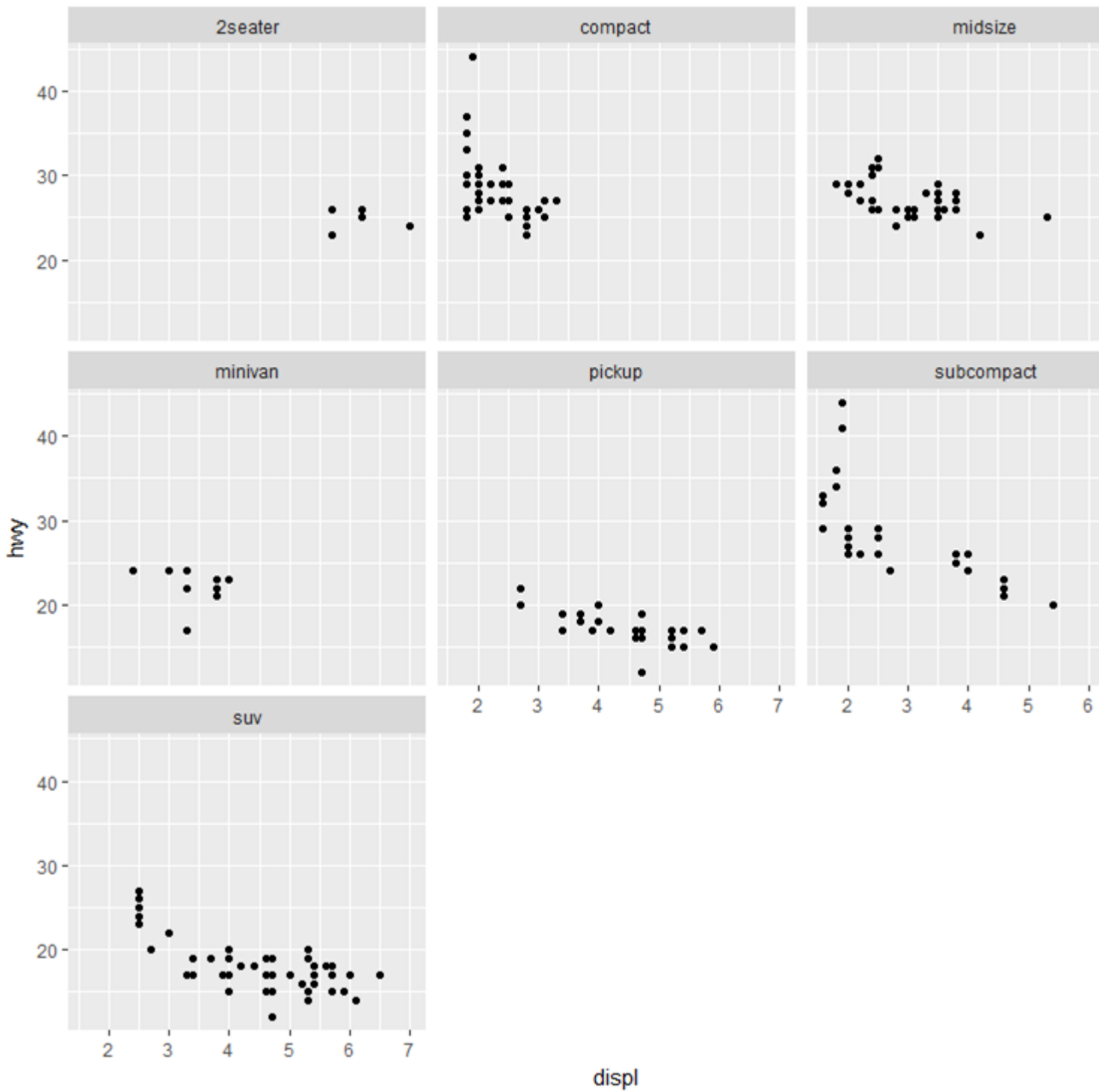
```
library(ggplot2)
ggplot(iris, aes(x = Petal.Width, y = Petal.Length, color = Species)) +
  geom_point()
```



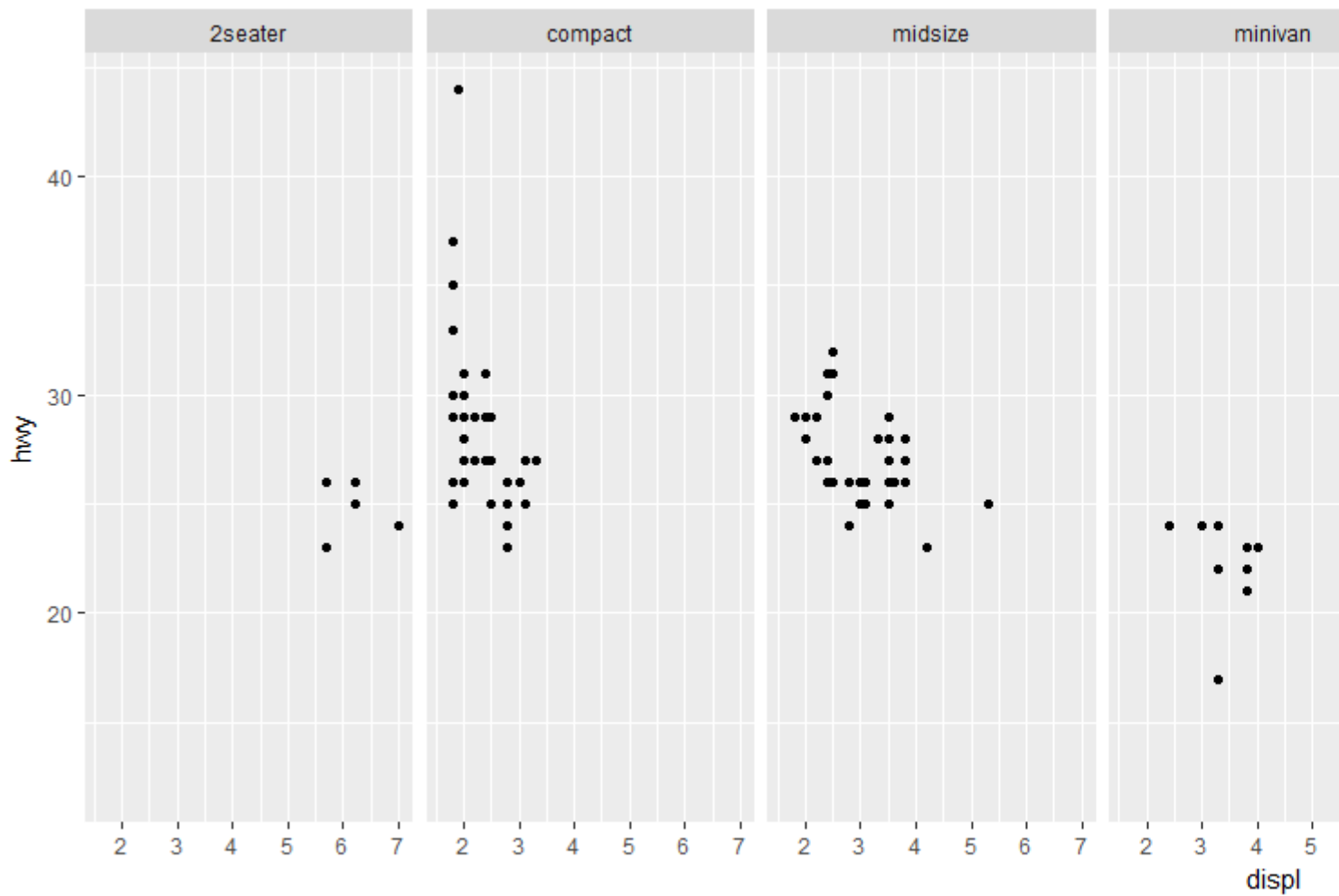
facet ◦ ◦ ggplot2mpg ◦

```
ggplot(mpg, aes(x = displ, y = hwy)) +
```

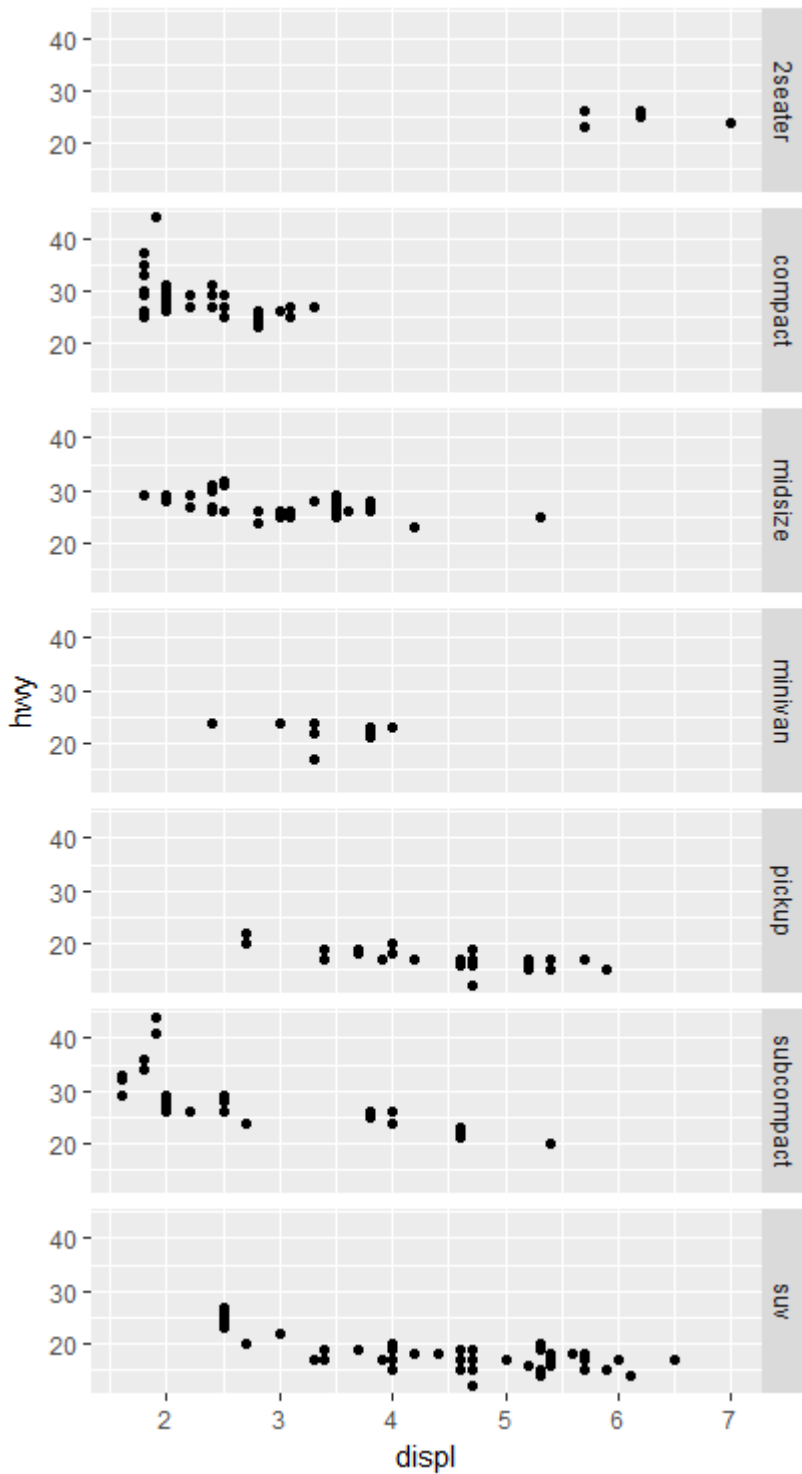
```
geom_point() +  
facet_wrap(~class)
```



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

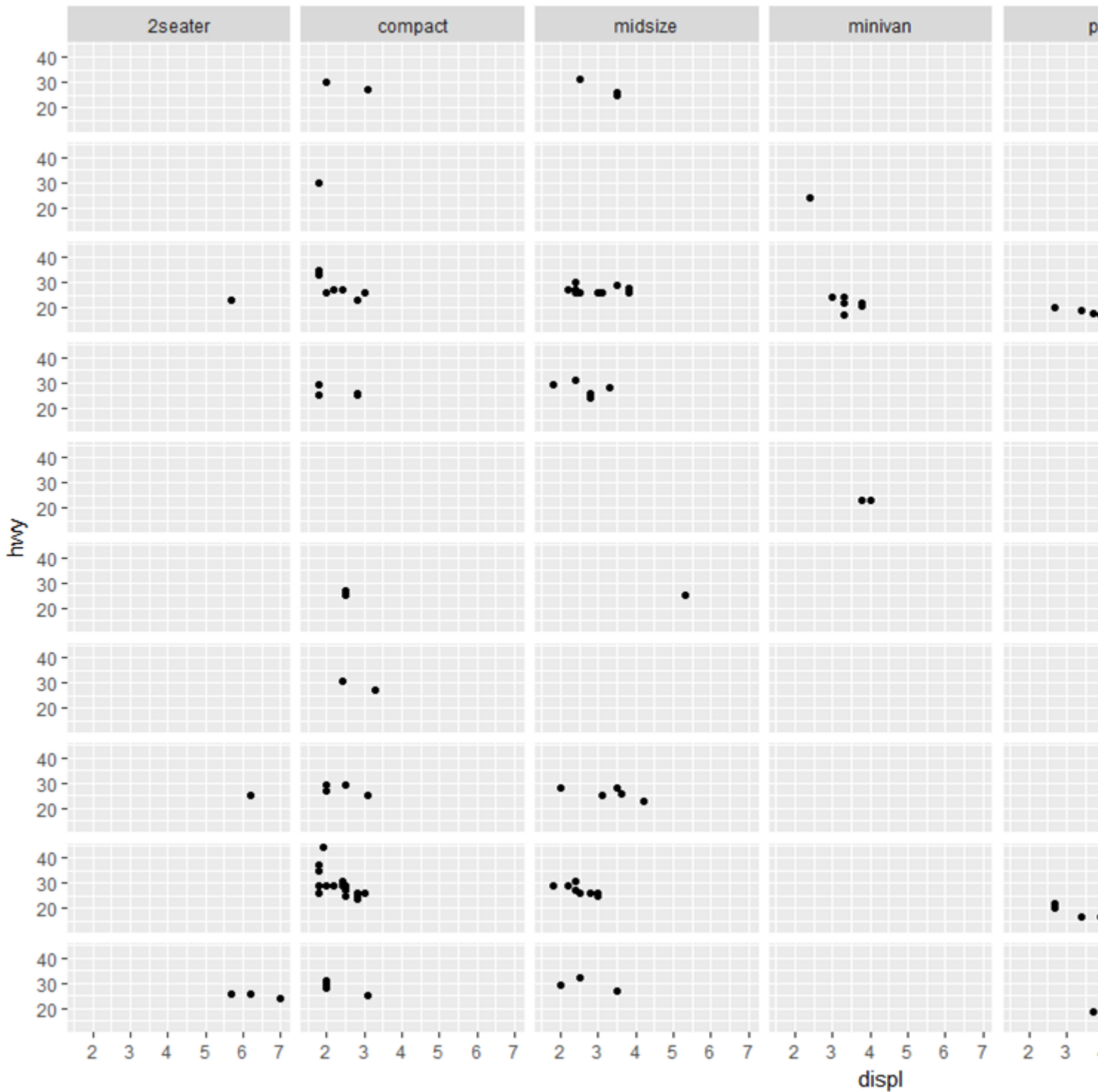


```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point() +  
  facet_grid(class~.)
```



2

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



ggplot2 20

```
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
```

sweetsWideggplot2 **R** reshape2 data.table tidyr

```
# 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')

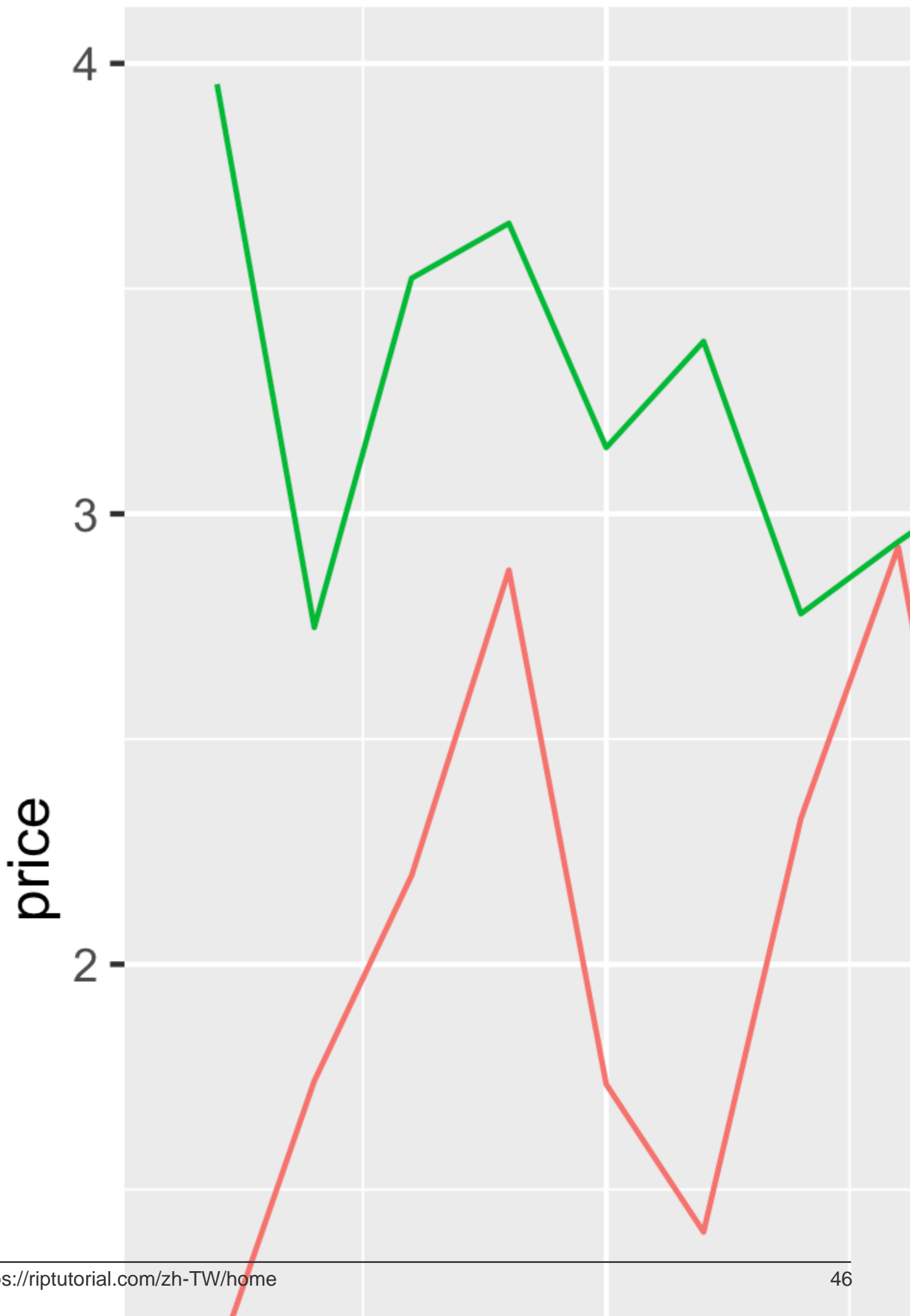
# gather from 'tidyr'
library(tidyr)
sweetsLong <- gather(sweetsWide, sweet, price, chocolate:candy)
```

```
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
```

o

sweetsLongo

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



---

## 8: GPU

GPU。CUDAOpenCL。NVIDIA CUDANVIDIA GPU。NVIDIAAMDIntelICPUGPUSDK。RGPU。

CUDA ToolkitOpenCL SDKR。R GPUCUDANVIDIA GPU。

1. [gputools](#)
2. [cudaBayesreg](#)
3. [HiPLARM](#)
4. [gmatrix](#)

OpenCL

1. [OpenCL - ROpenCL](#)
2. [gpuR -](#)

- GPU。

## Examples

### gpuR gpuMatrix

```
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

```
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
```

GPU <https://riptutorial.com/zh-TW/r/topic/4680/gpu>

# 9: JSON

## Examples

### JSON/R

`jsonlite` `WebJSON` `JSON` `fromJSON()` `JSON` `fromJSON()` `toJSON()` `Web` `vectors` `matrices` `data.frames`  
`JSON`.

### JSON

```
library(jsonlite)

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

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

### JSON

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

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

```
## 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 JSON

```
## 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

```

## 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"

```

JSON <https://riptutorial.com/zh-TW/r/topic/2460/json>

# 10: lubridate

- `ymd_hms...quiet = FALSEtz = "UTC"locale = Sys.getlocale("LC_TIME")`
- `tzzone = ""`
- `intervalstartendtzzone = attrstart"tzzone"`
- `num = NULLunits = "seconds"...`
- `num = NULLunits = "second"...`

## CRAN

```
install.packages("lubridate")
```

## Github

```
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

```
vignette("lubridate")
```

foo

```
help(foo)      # help about function foo
?foo           # same thing

# Example
# help("is.period")
# ?is.period
```

foo

```
example("foo")

# Example
# example("interval")
```

## Examples

### lubridate

lubridate◦

	R
ÿ	%y %Y
myd	%m %b %h %B
d	%d %e
H	%H %I%p
mhs	%M
	%S

```
ymd() "2016-07-22" ymd_hms() "2016-07-22 13:04:47" ◦
```

```
/ -◦ ◦
```

## date Date◦

```
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"
```

```
ymd_hmsymd_hmymd_h ◦ as.POSIXctstrptime tz="UTC"◦
```

## datetime POSIXct◦

```
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"
```

```
lubridateas.POSIXctstrptimestrptime
```

parse_date_time	POSIXct	◦ strptime style with %lubridate datetime"ymd hms"◦◦
parse_date_time2	POSIXct;lt = TRUE	◦ strptime%◦

	POSIXlt	
fast_strptime	POSIXlt;lt = FALSE POSIXct	◦ % -delimited strptime - / :◦

```
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\_time2fast\_strptimeC◦

?parse\_date\_time ◦

## Lubridate

Lubridateymd()◦ ymd◦

```
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

```
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"

```

◦ ◦ 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

```

lubridate ◦ ◦

```

# 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

```

◦

```

duration(60, "seconds")
## [1] "60s"

```

```
duration(2, "minutes")
## [1] "120s (~2 minutes)"
```

◦

```
dseconds dminutes◦
?quick_durations◦
```

```
dseconds(60)
## [1] "60s"

dhours(2)
## [1] "7200s (~2 hours)"

dyears(1)
## [1] "31536000s (~365 days)"
```

◦

```
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"
```

◦

```
as.duration(span)
[1] "43199s (~12 hours)"
```

◦

```
periodseconds hoursperiod◦ ?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◦
```

```
is.period(years(1))
## [1] TRUE
```

```
is.period(dyears(1))
## [1] FALSE
```

```
now_dt <- ymd_hms(now(), tz="IST")
now_dt
## [1] "2016-07-22 13:53:09 IST"
```

round\_date()◦

```
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() floor\_date()◦

```
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()◦

```
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"
```

DST◦

```
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°

```
nyc_time <- now("America/New_York")
nyc_time
## [1] "2016-07-22 05:49:08 EDT"

# corresponding Europe/Moscow time
with_tz(nyc_time, tzone = "Europe/Moscow")
## [1] "2016-07-22 12:49:08 MSK"
```

force\_tzX°

```
nyc_time <- now("America/New_York")
nyc_time
## [1] "2016-07-22 05:49:08 EDT"

force_tz(nyc_time, tzone = "Europe/Moscow") # only timezone changes
## [1] "2016-07-22 05:49:08 MSK"
```

lubridate <https://riptutorial.com/zh-TW/r/topic/2496/lubridate>

---

# 11: Meta

RR ◦

## Examples

QA ◦

- ◦ ◦
- ◦ ◦
- ◦

```
library(help = "datasets")
```

## Docs

- ?data.frame ◦ SO Docs ◦ R ◦
- "" ◦

---

R> >+◦ Docs◦

- 
- ◦
  - ###◦
  - [1]◦
  - ◦

---

=<-R◦ x<-1 x <- 1x < -1

---

◦ ◦ ◦ ◦

---

H1◦

Meta <https://riptutorial.com/zh-TW/r/topic/5410/meta->

# 12: R MarkdownRStudio

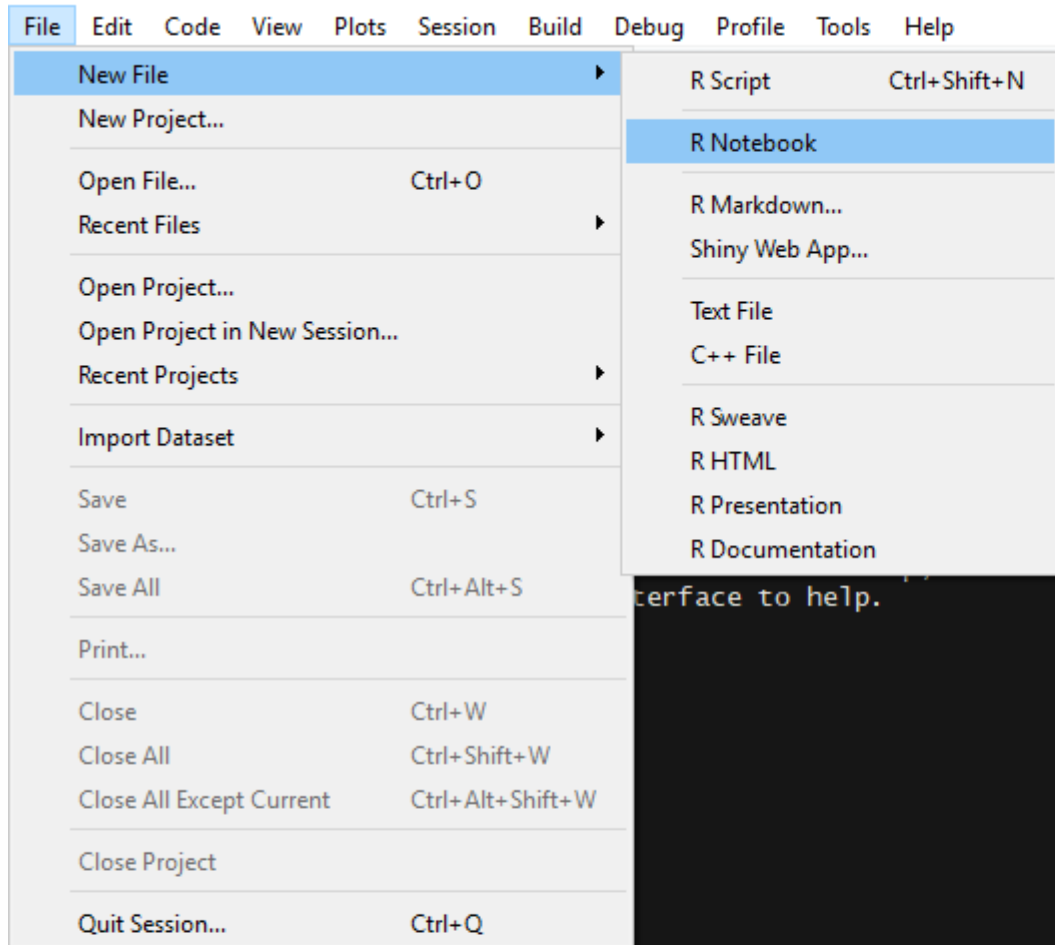
RR Markdown。 R MarkdownR Notebook/。 RRStudio1.0RStudio。

## Examples

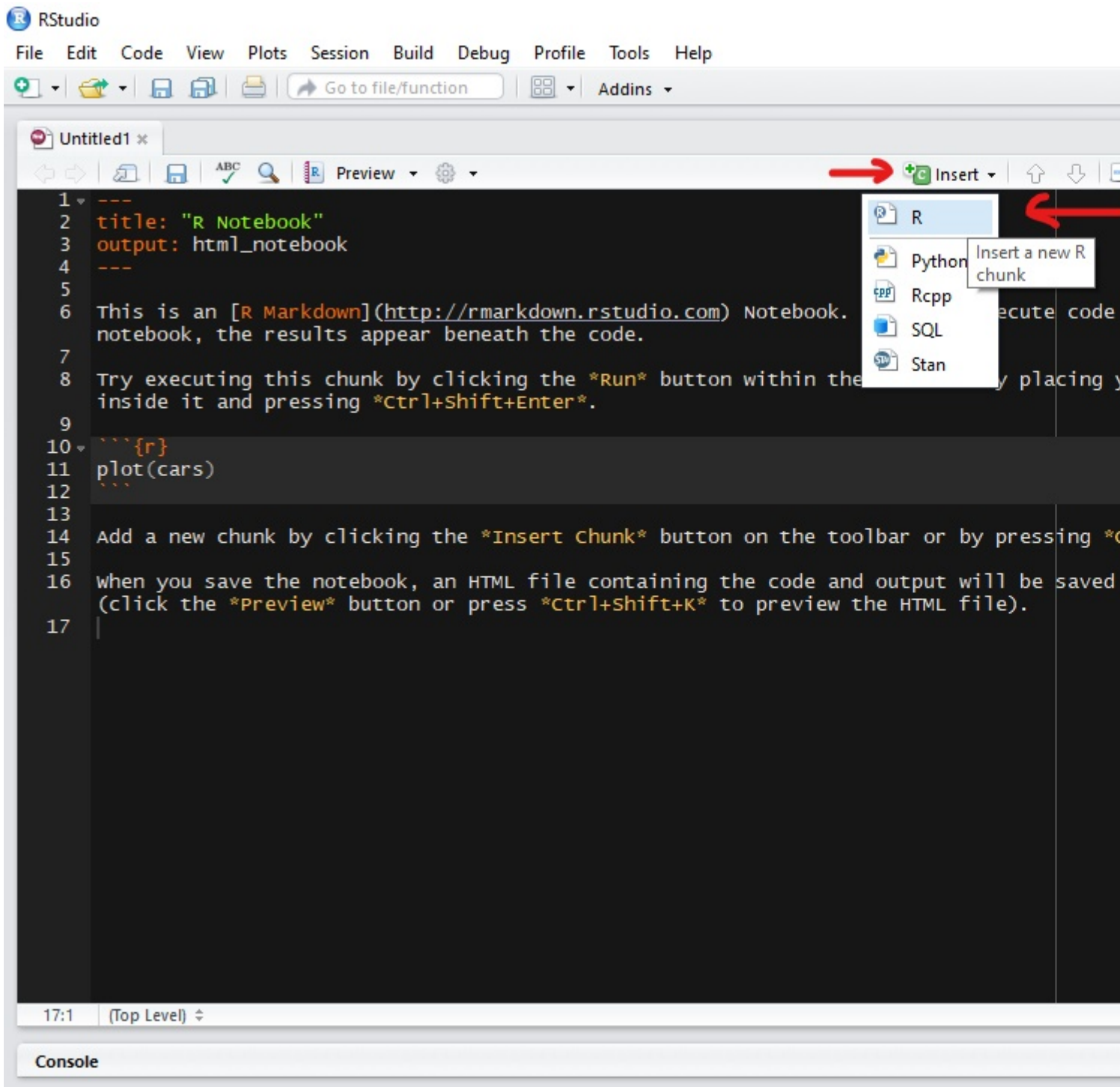
- > - > RRStudio

R NotebookRStudio。 RStudio

RStudio



。 RR。 Ctrl + Alt + IOS XCmd + Option + I

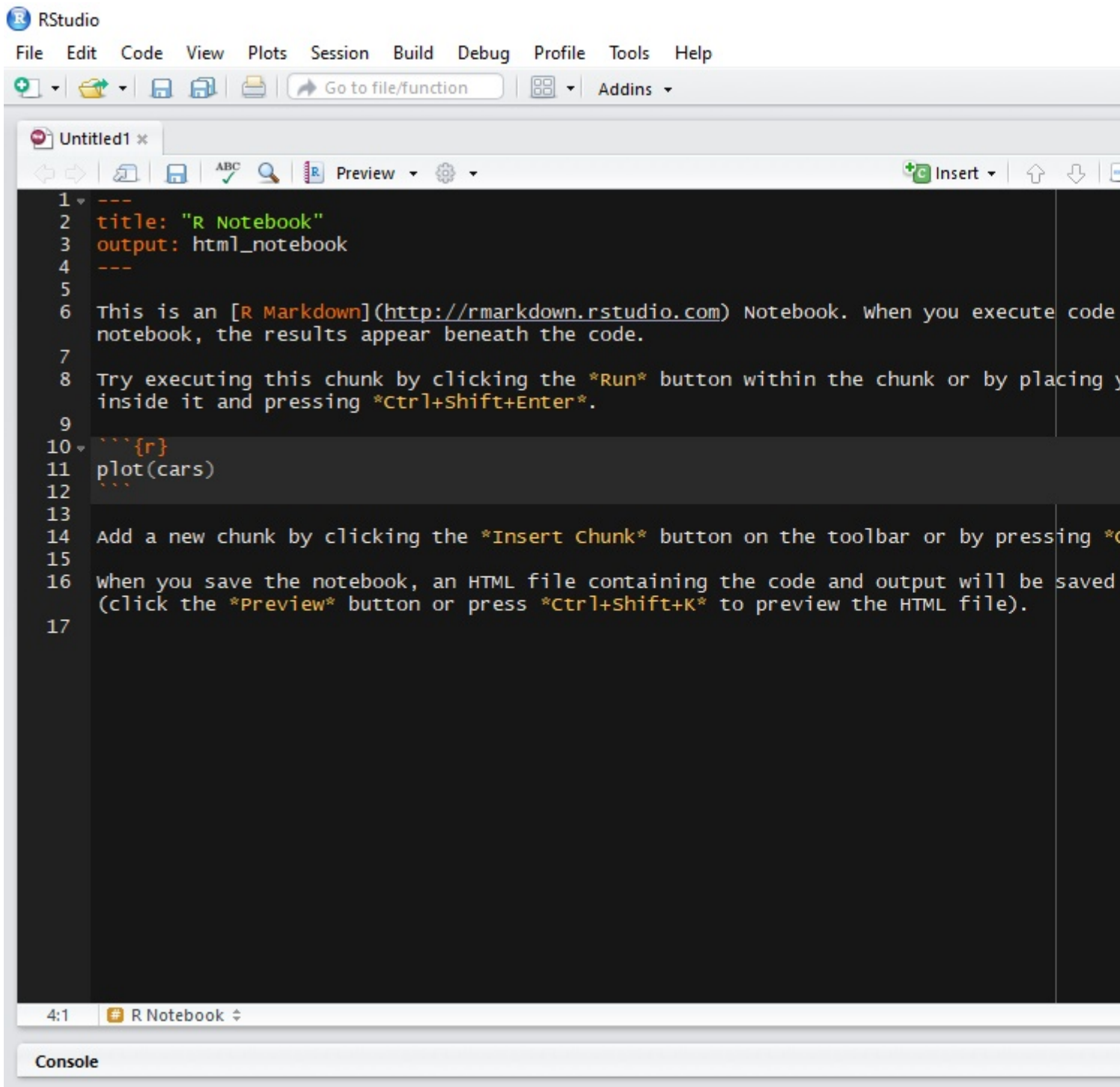


“”。 Ctrl + Shift + Enter OS X Cmd + Shift + Enter

。

。

Ctrl + Alt + IOS XCmd + Option + I



◦ R;R◦

Run◦ ◦ **Ctrl + Alt + ROS XCmd + Option + R**

**Restart RRun All ChunksRunR◦**

**Run All Chunks AboveRun All Chunks Below◦**



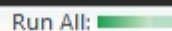
Untitled1\* x

```
14 data("iris")
15 head(iris,5)
16
```

	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

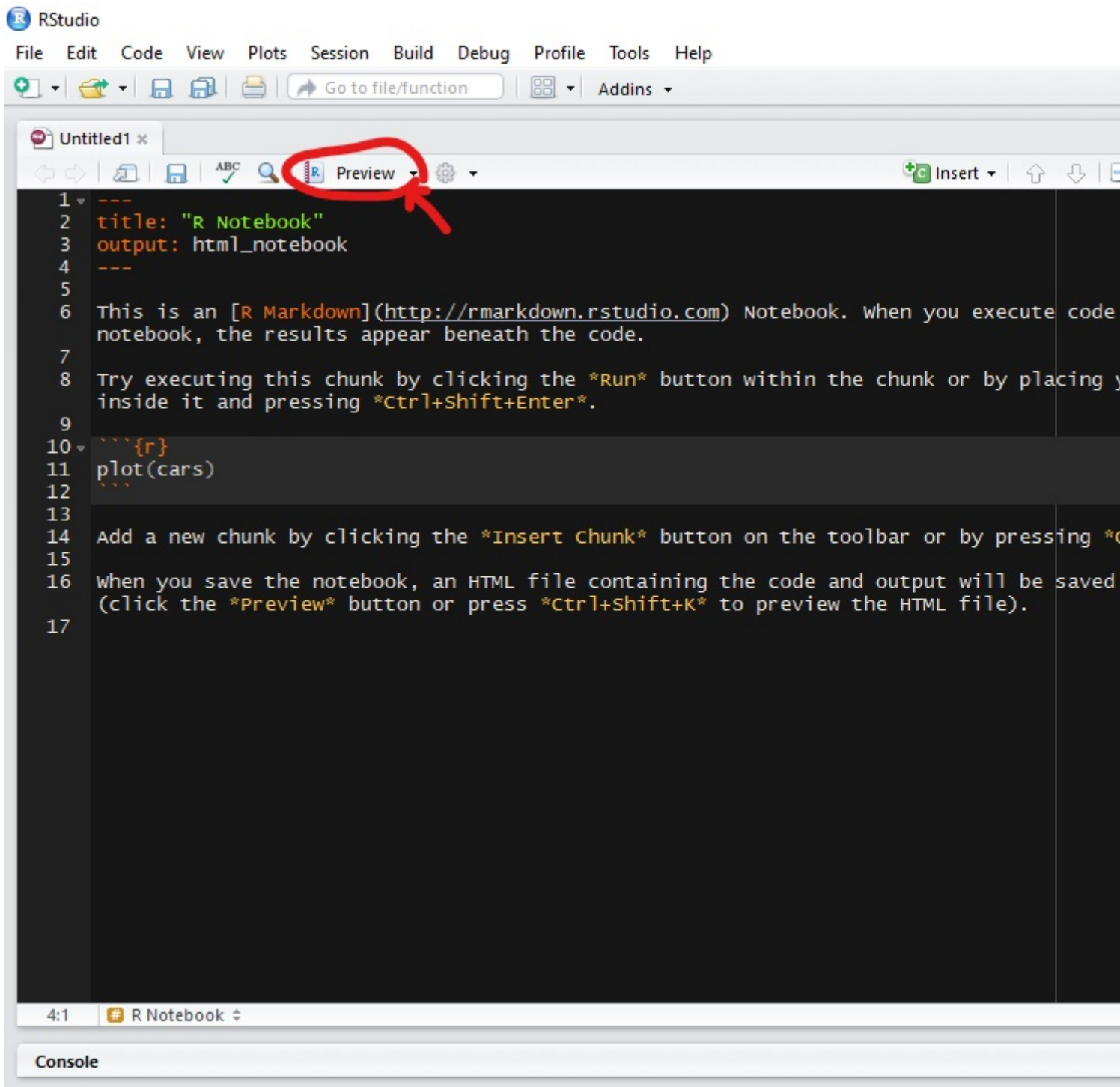
5 rows

```
17
18 Divide Iris data to x (contain the all features) and y (only the classes)
19 {r}
20 x <- subset(iris, select=-species)
21 y <- iris$species
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
```

Run All: 

。 “ ”。

“pdf\_document”“html\_notebook”



.Rmd .nb.html HTML.Rmd

[RStudio docs](#)

[R Markdown RStudio https://riptutorial.com/zh-TW/r/topic/10728/r-markdown-rstudio-](https://riptutorial.com/zh-TW/r/topic/10728/r-markdown-rstudio-)

---

# 13: R memento

R.

succint.

## Examples

---

```
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
```

---

## Dataframes

```
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"
```

---

```
l <- list(a = 500, "aaa", 98.2)
length(l) == 3
class(l[1]) == "list"
class(l[[1]]) == "numeric"
class(l$a) == "numeric"
```

---

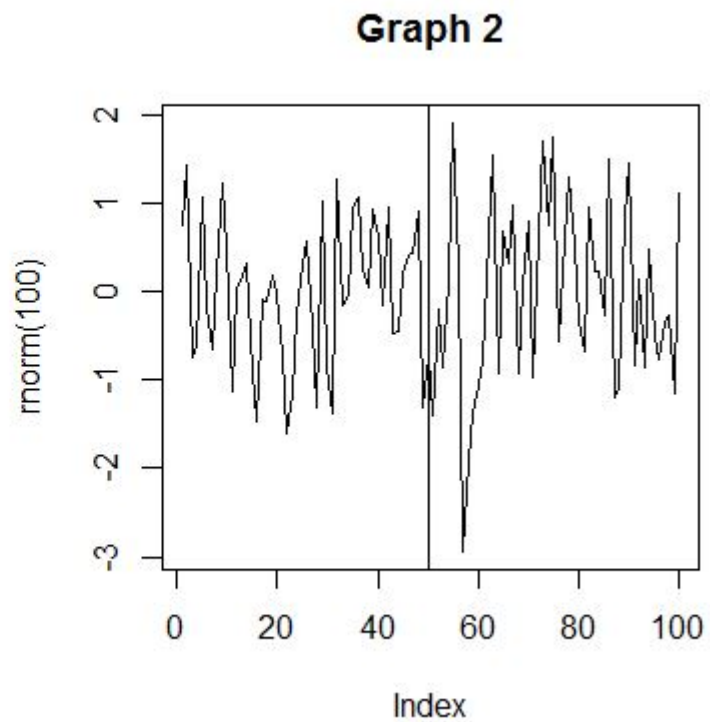
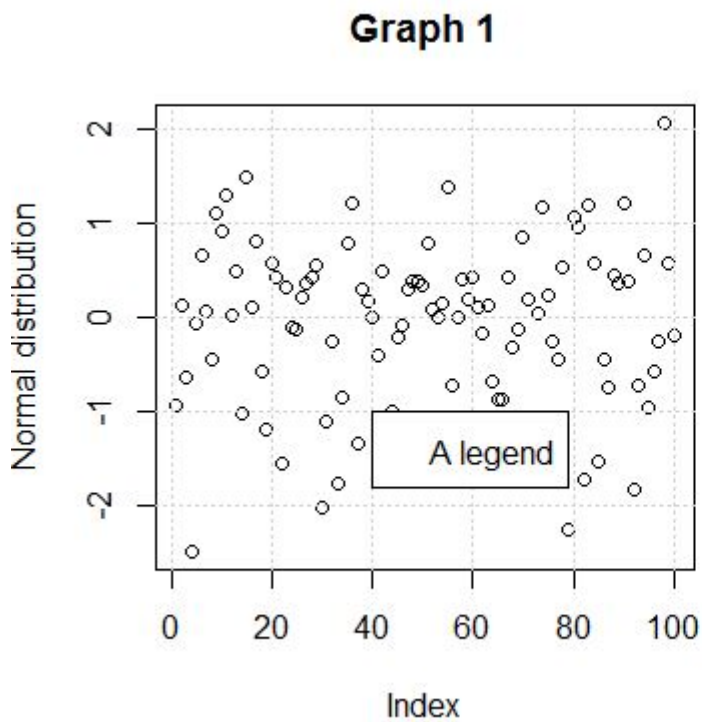
```
env <- new.env()
env[["foo"]] = "bar"
env2 <- env
env2[["foo"]] = "BAR"

env[["foo"]] == "BAR"
get("foo", envir = env) == "BAR"
rm("foo", envir = env)
env[["foo"]] == NULL
```

```
# 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)
```



```
# 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")
```

R memento <https://riptutorial.com/zh-TW/r/topic/10827/r-memento>

# 14: RCPP

## Examples

`Rcpp::cppFunction()` `R` `cppFunction()` `evalCpp()` `sourceCpp()` `C++` `cppFunction()`

`RC++`

```
# 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++`

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

## Rcpp

`Rcpp` `Attributes` `RC++`

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

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

`sourceCpp()` `C++`

`C++`

```
// 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
}

```

## RC++

```

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

```

## Rcpp

### C++

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

```
// 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)]]
```

RcppRcpp.hRcpp<PACKAGE>.h [RcppArmadillo](#) ◦

```
// [[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)]]
```

**RCPP** <https://riptutorial.com/zh-TW/r/topic/1404/rcpp>



---

# 15: RESTful R

OpenCPUWeb。

## Examples

### opencpu

<https://www.opencpu.org/apps.html>

R

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

URLR。XMLhtmlJSON。

### cURLR

```
#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"
```

R。

/ocpu/tmp/

```
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。

/ocpu/tmp/x009f9e7630/R/.valrnorm(5) /ocpu/tmp/x009f9e7630/R/consolelernorm(5)。

RESTful R <https://riptutorial.com/zh-TW/r/topic/8323/restful-r>

# 16: RMarkdownknitr

- ○ YAML

```

“ r format(Sys.time(), '%d %B, %Y') ”

10. html html_output . PDF pdf_document ..

```

		HTML	PDF		ODT	RTF	MD	github	ioslides	slidy	
citation_package	natbib biblatex noneLaTeX		X				X				X
code_folding	R“”“”	X									
colortheme											X
CSS	CSS	X							X	X	
	“png”	X	X				X	X	X	X	X
										X	
fig_caption		X	X	X	X				X	X	X
fig_height fig_width		X	X	X	X	X	X	X	X	X	X
	“” “pygments” “zenburn” “textmate”	X	X	X						X	X
	in_header before_body after_body	X	X		X		X	X	X	X	X
									X	X	X
keep_md	knitr.md	X		X	X	X			X	X	
keep_tex	knitr.tex		X								X

		HTML	PDF		ODT	RTF	MD	github	ioslides	slidy	
latex_engine	“”pdflatex“ ”xelatex“ lualatex”		X								X
LIB_DIR	Bootstrap MathJax	X							X	X	
mathjax	URL MathJax/ URL	X							X	X	
md_extensions	MarkdownR Markdown	X	X	X	X	X	X	X	X	X	X
number_sections		X	X								
pandoc_args	Pandoc	X	X	X	X	X	X	X	X	X	X
preserve_yaml	YAML						X				
reference_docx	docxdocx			X							
self_contained	doc	X							X	X	
slide_level											X
									X		
	em-dashes ... <sup>o</sup>	X							X	X	
	Pandoc	X	X	X						X	X
	Bootswatch Beamer	X									X
TOC		X	X	X		X	X	X			X
toc_depth		X	X	X		X	X	X			
toc_float		X									

## Examples

### Rstudio

## .Rmd.R

render Rstudio

```
---  
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

jQueryCSSknitrioslides jQuery

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

```
jQueryDOM HTML $(document).ready(function() { ... }) .title-slide .backdrop.segue"  
<footer></footer> </slide> label
```

## CSS

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

- label
- 12
- 2060

o

```

---
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;
    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.

```

# Slide 1

This is slide 1.

My amazing footer

RMarkdownknitr <https://riptutorial.com/zh-TW/r/topic/2999/rmarkdownknitr>

# 17: RMD

| YAML            |  |
|-----------------|--|
| toc             |  |
| number_sections |  |
| bibliography    |  |
| csl             |  |

- RMD。
- `install.packages("rmarkdown")` Rrmarkdown 。
- Rmarkdown。YAML `link-citations: true`
- 

|               |          |
|---------------|----------|
| MODS          | .mods    |
| BibLaTeX      | 。        |
|               | .bibtex  |
| RIS           | .ris     |
| EndNote       | .enl     |
| EndNote XML   | .XML     |
| ISI           | .wos     |
| MEDLINE       | .medline |
| COPAC         | .copac   |
| JSON citeproc | .json    |

## Examples

RMDYAML。PDFtoc。

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

```
output:
  pdf_document:
    number_sections: yes
toc: yes
bibliography: bibliography.bib
---
```

bibliography.bib

```
@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@bibkey**Meyer2000 ◦

```
# 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**`rmarkdown::render("<path-to-your-RMD-file">)`



# 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.

# 18: RODBC

## Examples

### RODBCExcel

RODBCRRDMSWindowsExcelSQL

```
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 Management

RODBCSQL Server: 'Driver'SQL Server" "Atilla"sqlQuery

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

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

SQL Server: [connectionstrings.com](https://connectionstrings.com)

datasourceName.schema.objectname

RODBC <https://riptutorial.com/zh-TW/r/topic/2471/rodbc>

---

# 19: roxygen2

|       |   |
|-------|---|
|       |   |
|       |   |
|       |   |
|       | - |
|       | S |
|       |   |
| PARAM |   |

## Examples

roxygen2

---

# roxygen2

[roxygen2](#) Hadley Wickham.

R#'. @

```
#' @author The Author
```

```
mean<-function(x) sum(x)/length(x)
```

```
#' 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)
```

- #' Mean
- @param @export
- @keyword
- @importFrom @import
- @example
  - ;
  - \dontrun -

devtools::document() devtools::document() ◦ devtools::check() ◦

roxygen2 <https://riptutorial.com/zh-TW/r/topic/5171/roxygen2>

# 20: R

R ◦ R?regex ◦ ICU ◦

## Examples

### `grep`

```
# 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 “x”“y”

```
grep('5|8', mystring)
# [1] 1 2
grep('com|org', mystring)
# [1] 5 6
```

### . ◦ “”

```
grep('The number .', mystring)
# [1] 1 2
```

```
tricky <- c('www.personal.org', 'My friend is a cyborg')
grep('.org', tricky)
# [1] 1 2
```

### \ ◦ R ◦

```
grep('\.org', tricky)
# Error: '\.' is an unrecognized escape in character string starting "\"
grep('\\.org', tricky)
# [1] 1
```

[]

```
grep('[13]', mystring)
# [1] 3 4
grep('[@/]', mystring)
# [1] 5 7
```

◦ [0-4]**0,1,2,34** [AZ] [Az] [A-z0-9]

```
grep('[0-4]', mystring)
# [1] 3 4
grep('[A-Z]', mystring)
# [1] 1 2 4 5 6
```

**R**◦ [:lower:]az [:upper:]AZ [:alpha:]Az [:digit:]0-9 [:alnum:]A-z0-9 ◦ ;[[:digit:]] ◦  
[@[:digit:]]/@ /0-9 ◦

```
grep('[[:digit:]]', mystring)
# [1] 1 2 3 4
grep('[@[:digit:]]/', mystring)
# [1] 1 2 3 4 5 7
```

^ ◦ [^5]"5"◦

```
grep('The number [^5]', mystring)
# [1] 2
```

**R** <https://riptutorial.com/zh-TW/r/topic/9743/r>

# 21: R -

## Examples

◦

```
data("GermanCredit")
variances<-apply(GermanCredit, 2, var)
variances[which(variances<=0.0025)]
```

◦

“nearZeroVar.....“

```
library(caret)
names(GermanCredit)[nearZeroVar(GermanCredit)]
```

## NA

```
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 |            |            |            |            |

## NonDDream20

◦ ◦

```
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
```

MPG。 cyldisp。 ◦

R - <https://riptutorial.com/zh-TW/r/topic/7561/r--->



## 22: R

R。

S3S4S6。

## Examples

### S3

S3ROO。

S3。 class。

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

class

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

attr

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

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

R。

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

summary.bicycle

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

this is a bicycle

[R https://riptutorial.com/zh-TW/r/topic/9723/r](https://riptutorial.com/zh-TW/r/topic/9723/r)

# 23: R

## Examples

R”R。

“”

- 
- R
- 
- applyR

applyR。

/

```
apply(mtcars, 1, mean)
      Mazda RX4      Mazda RX4 Wag      Datsun 710      Hornet 4 Drive      Hornet
Sportabout      29.90727      Valiant      Duster 360      23.59818      38.73955
53.66455      35.04909      29.98136      59.72000      29.98136
      Merc 240D      Merc 230      Merc 280      Merc 280C      Merc
450SE      Merc 450SL      Merc 450SLC      31.86000      31.78727
46.43091      46.50000      46.35000
Cadillac Fleetwood Lincoln Continental Chrysler Imperial Fiat 128      Honda
Civic      Toyota Corolla      Toyota Corona      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
```

1. data.frame(matrix(apply(iris, 2, class)str(iris)sapply(iris, class))
2. Rnrow(mtcars) meanR
3. R

```
rowMeans(mtcars)
      Mazda RX4      Mazda RX4 Wag      Datsun 710      Hornet 4 Drive      Hornet
Sportabout      29.90727      Valiant      Duster 360      23.59818      38.73955
53.66455      35.04909      29.98136      59.72000      29.98136
      Merc 240D      Merc 230      Merc 280      Merc 280C      Merc
450SE      Merc 450SL      Merc 450SLC      31.86000      31.78727
46.43091      46.50000      46.35000
Cadillac Fleetwood Lincoln Continental Chrysler Imperial Fiat 128      Honda
Civic      Toyota Corolla      Toyota Corona      65.97227      19.44091
      66.23273      66.05855      65.97227
```

|                          |               |               |              |            |                          |
|--------------------------|---------------|---------------|--------------|------------|--------------------------|
| 17.74227                 | 18.81409      | 24.88864      |              |            |                          |
| Dodge Challenger<br>X1-9 | Porsche 914-2 | AMC Javelin   | Lotus Europa | Camaro Z28 | Pontiac Firebird<br>Fiat |
|                          | 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   |                          |

R◦ data.frame matrix ◦ rowMeans◦

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

mtcarsdata.frame list vector ◦

```
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
 [2] 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
 [18] 24.77909 24.88027 60.97182 34.50818 63.15545 26.26273
```

NA◦

R.

```
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
```

RCCR◦

R<sub>rowsum</sub>

```
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
```

◦

◦ data.frame matrix

◦ ◦

m

```
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
```

```
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
```

```
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
```

[R https://riptutorial.com/zh-TW/r/topic/3327/r](https://riptutorial.com/zh-TW/r/topic/3327/r)

# 24: RLaTeXknitr

1. `<< internal-code-chunk-nameoptions ... >> =`  
#R  
@
2. `\ Sexpr {#R Code Here}`
3. `<< read-external-R-file >> =`  
read\_chunk'R-file.R'  
@  
`<< external-code-chunk-nameoptions ... >> =`  
@

|            |                  |
|------------|------------------|
|            | TRUE / FALSE - R |
|            | TRUE / FALSE - R |
|            | TRUE / FALSE - R |
|            | TRUE / FALSE - R |
|            | TRUE / FALSE - R |
| fig.width  | - R              |
| fig.height | - R              |

KnitrLaTeXR。 。 knitrLaTeX.texR.RR noweb.Rnw。 .RnwLaTeXR。

KnitrPDF。

.RnwPDF。 RLaTeX'knitting'。 knitr。 [knitr](#)

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

.texr-noweb.texPDF

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

## Examples

KnitrLatexR.

KnitrRRLaTeX。 。 R/R。 。 。

```
# 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}
```

## R.

```
## 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)
```

## R<sub>LaTeX</sub>Knitr

### KnitrR<sub>LaTeX</sub>. approach.

```
# 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  $\text{\Sexpr{length(x)}}$ .

\end{document}
```

## R<sub>LaTeX</sub>Knitr

### KnitrR<sub>LaTeX</sub>. approach.

```
# 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}
```

**R**LaTeXknitr <https://riptutorial.com/zh-TW/r/topic/4334/rlatexknitr>



# 25: RI / O.

## Examples

### RdsRDataRda

`.rds.Rdata .rda RR` `write.table`

- R
- R.

`saveRDS / readRDSR` .

`.rdsiris`

```
saveRDS(object = iris, file = "my_data_frame.rds")
```

```
iris2 <- readRDS(file = "my_data_frame.rds")
```

`save().Rdata` .

2

```
save(iris, cars, file = "myIrisAndCarsData.Rdata")
```

```
load("myIrisAndCarsData.Rdata")
```

## Enviromments

`saveload`

```
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
```

RI / O. <https://riptutorial.com/zh-TW/r/topic/5540/ri---o->

# 26: R

## Examples

### RRPDFPMF

#### PMF

10.

dbinom

```
> dbinom(2, 10, 1/6)
[1] 0.29071
```

### POESONPMF

20.

dpois

```
> dpois(18, 20)
[1] 0.08439355
```

### PDF

52x = 2.5pdf

```
> dnorm(2.5, mean=5, sd=2)
[1] 0.09132454
```

[R https://riptutorial.com/zh-TW/r/topic/4333/r](https://riptutorial.com/zh-TW/r/topic/4333/r)

# 27: Spark API SparkR

SparkR [Spark](#) ◦ ◦ [SparkR](#) [SparkR](#)

## Examples

### Spark

### RSpark

SparksRSpark Cluster◦

```
library(SparkR)
sc <- sparkR.init() # connection to Spark context
sqlContext <- sparkRSQL.init(sc) # connection to SQL context
```

IDESpark◦

## Spark Cluster

[Apache Spark](#) ◦ [javaSpark Cluster](#) [Microsoft Azure \[topic site\]](#) [IBM](#) ◦

Spark◦ ◦ [SparkRDD](#)◦

- ◦ [RDD](#)◦

### 3 GB big csv

```
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

```
mtrdd <- createDataFrame(sqlContext, mtcars)
```

## CSV

### csvSparkCSV

```
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)
```

## CSV

```
train <- read.df(sqlContext, "/train.csv", header= "true", source =
"com.databricks.spark.csv", inferSchema = "true")
```

```
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)
```

Spark API SparkR <https://riptutorial.com/zh-TW/r/topic/5349/spark-api-sparkr->

# 28: sqldf

## Examples

```
sqldf() sqldf(SQLiteR, SQL)
```

```
ggplot2 "10"
```

```
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"10

```
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
```

“”SQL。

1

```
sqldf("select count(*) from diamonds where carat > 1 and color = 'E'")
```

```
count(*)  
1      1892
```

```
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
```

price cut

```
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
```

sqldf <https://riptutorial.com/zh-TW/r/topic/2100/sqldf>

## 29: strsplit

- strsplit
- X
- 
- =
- perl = FALSE
- useBytes = FALSE

### Examples

strsplit ◦ Rdata.frame ◦

strsplit

```
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"
```

split ◦ temp2temp ◦ split ◦

```
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. ◦
2. ◦ RR ◦

strsplit <https://riptutorial.com/zh-TW/r/topic/2762/strsplit>

# 30: tidyverse

## Examples

### tbl\_df

tbl\_df [tibble](#) [diff tidyverse](#)

as\_data\_frame [tbl\\_df](#)

```
library(tibble)
mtcars_tbl <- as_data_frame(mtcars)
```

### data.frame [tbl\\_dfs](#)

```
# 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
```

- 32 x 11
- dbl
- `options(tibble.print_max = [number])`

dplyr [tbl\\_dfs](#) [group\\_by\(\)](#)

## tidyverse

[tidyverse](#)

[tidyverse](#) [RHadley / Rstudio](#) [tidyverse](#)

[tidyverse](#) [API](#) [tidyverse](#) [tidyverse](#)

[TidyverseR for Data Science](#) [tidyverse](#) [tidyverse](#)



R°

```
install.package("tidyverse")  
library("tidyverse")
```

/° /°

- 
- [ggplot2 SO\\_doc](#)
  - [dplyr Rcpp SO\\_doc](#)
  - [tidyr SO\\_doc](#)
  - [readr](#) °
  - [purrr JSunderscore.jslodashlazy.jsR](#)°
  - [tibble](#) °
  - [magrittr SO\\_doc](#)
  
  - [hms](#)
  - [stringr](#)
  - [lubridate /SO\\_doc](#)
  - [forcats](#) °
  
  - [DBI RDBMS](#)
  - [SPSS SAS Stata SO\\_doc](#)
  - [httr httrcurl Web API](#)
  - [jsonlite WebJSON](#)
  - [readxl read.xls.xlsx SO\\_doc](#)
  - [rvest rvest SO\\_doc](#)
  - [xml2 XML](#)
  
  - [modelr](#)
  -

tidyverse

- [knitr API](#)° [SO\\_docs](#)
- [rmarkdown Rstudio](#)° [SO\\_docs](#)

[tidyverse](https://riptutorial.com/zh-TW/r/topic/1395/tidyverse) <https://riptutorial.com/zh-TW/r/topic/1395/tidyverse>

# 31: Web

◦ R◦

Base R◦ [RSelenium](#) [httr](#) [curl](#) [RCurl](#) [XML](#) [xml2](#) [rvest](#)◦

Web API◦ ◦

◦ ◦

## Examples

### rvest

[rvest](#)◦ [Hadley Wickham](#)◦ Python's [Beautiful Soup](#)◦ [Hadley](#)◦ [xml2](#)◦ [libxml2](#)◦ [HTML](#)◦

[tidyverse](#) [rvest](#)◦

- `xml2::read_html`/`xml2::read_html`◦ [HTML](#)
- [CSSXPath](#)◦ `html_node`/`html_nodes`
- `html_text`/`html_table`◦ [R](#)◦

### R

```
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
```

data.frame<sup>NA</sup>

- jQuery read\_html<sup>R</sup>Selenium<sup>◦</sup>

## rvest

ID.

- ◦

```
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)
```

5. \*\*\*\*\*.

Web <https://riptutorial.com/zh-TW/r/topic/2890/web>

# 32: xgboost

## Examples

### xgboost

```
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))
```

xgboost <https://riptutorial.com/zh-TW/r/topic/3239/xgboost>

# 33:

◦ package::function() ◦ caret plsplyr ◦

Microsoft R Open Revolution RBLAS / LAPACK.

## Examples

### foreach

foreachR.CPU ◦ doSNOW ◦

foreach1100000 ◦

```
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.combine ◦ ◦ c ◦ "+" ◦

foreach ◦ k ◦

```
◦ ◦ c cbind rbind "+" "*" .....
◦ TRUE i ◦ FALSE FALSE ◦ ◦
.packages basemass randomForestc("mass", "randomForest")
```

parallel ◦

### localhost

```
parallel::detectCores(all.tests = FALSE, logical = TRUE)
```

### localhost

```
parallelCluster <- parallel::makeCluster(parallel::detectCores())
```

◦ mtcars ◦ cylmpg ◦

```

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 zlevels ° °

```

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))
}

```

```

worker <- function(zlevel) {
  fitmodel(zlevel, datax, datay, dataz)
}

```

parallel° parallel° ° °

```

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)
}

```

°

```

parallelcluster <- parallel::makeCluster(parallel::detectCores())
models <- parallel::parLapply(parallelcluster, zlevels,
  wrapper(datax, datay, dataz))

```



◦

```
parallel::stopCluster(parallelcluster)
```

parallelapply()par ◦

**RNG** ◦ set.seed() ◦ ◦

◦ parallel snow ◦

```
s <- seed
for (i in 1:numofcores) {
  s <- nextRNGStream(s)
  # send s to worker i as .Random.seed
}
```

◦

```
clusterSetRNGStream(cl = parallelcluster, iseed)
```

## mcparallelDo

mcparallelDo **UnixLinuxMacOSXR** ◦ ◦ [future](#) ◦

```
data(ToothGrowth)
```

## mcparallelDofork

```
mcparallelDo({glm(len ~ supp * dose, data=ToothGrowth)}, "interactionPredictorModel")
```

```
binaryPredictorModel <- glm(len ~ supp, data=ToothGrowth)
gaussianPredictorModel <- glm(len ~ dose, data=ToothGrowth)
```

## mcparallelDotargetEnvironment.GlobalEnv

```
summary(interactionPredictorModel)
```

```
# 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) {  
    mcpipelineDo({2+2}, targetValue = "output")  
  }  
  mcpipelineDoCheck()  
  if (exists("output")) print(i)  
}
```

<https://riptutorial.com/zh-TW/r/topic/1677/>

# 34:

## Examples

### RCPU

```
system.time(print("hello world"))

# [1] "hello world"
#      user  system elapsed
#      0      0      0
```

```
system.time({
  library(numbers)
  Primes(1,10^5)
})
```

```
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

proc.time() CPU。

```
proc.time()

#      user      system    elapsed
# 284.507    120.397 515029.305
```

。

```
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)
```

```
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()/proc.time()/
```

```
print(t1 <- system.time(replicate(1000,12^2)))
## user  system elapsed
## 0.000  0.000  0.002
```

```
proc.time
```

```
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" ...
```

[lineprof](#) Hadley Wickham<sup>o</sup> `auto.arima`

```
library(lineprof)
library(forecast)

l <- lineprof(auto.arima(AirPassengers))
shine(l)
```

o R<sup>o</sup>

# 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               |          |    |    |    |

## Microbenchmark. hello world.

```
system.time(print("hello world"))  
  
# [1] "hello world"  
#   user  system elapsed  
#    0      0      0
```

## system.timeproc.timeproc.time. "hello world". microbenchmark

```
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
```

## print("hello world") 10044. "hello world"100.

```
cat("hello world\n") print("hello world")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
```

cat()print()°

microbenchmark

```
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
```

## microbenchmark

microbenchmark“”°

data.table°

**3**data.table id timestatus° id -> 7false

```
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

```

expression\_3°

[data.table -](#)

[data.table - data.table](#)

<https://riptutorial.com/zh-TW/r/topic/2149/>

# 35: CSVTSV

- `read.csvfileheader = TRUEsep = ""quote = ""dec = "fill = TRUEcomment.char = "...`
- `read.csv2fileheader = TRUEsep = ";"quote = ""dec = "fill = TRUEcomment.char = "...`
- `readr :: read_csvfilecol_names = TRUEcol_types = NULLlocale = default_localena = c"""NA"comment = "trim_ws = TRUEskip = 0n_max = -1 progress = interactive`
- `data.table :: freadsep = "auto"sep2 = "auto"nrows = -1Lheader = "auto"na.strings = "NA"stringsAsFactors = FALSEverbose = getOption"datatable" .verbose"autostart = 1Lskip = 0Lselect = NULLdrop = NULLcolClasses = NULLinteger64 = getOption"datatable.integer64"default"integer64"dec = ifsep=" " " "else""col.namescheck.names = FALSEencoding = "unknown"strip.white = TRUEshowProgress = getOption"datatable.showProgress"defaultTRUE data. table = getOption"datatable.fread.datatable"defaultTRUE`

|              |                 |
|--------------|-----------------|
|              | CSV             |
|              | logical.csv     |
|              | character       |
|              | character       |
|              | character       |
|              | logicalTRUE.    |
| comment.char | charactercsv. . |
| ...          | read.table      |

◦ [.RDataFeather](#)◦

/◦

## Examples

`.csv`

# R

CSV`read.csvread.table sep = ", "◦`



```
# 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
```

file.choose

```
df <- read.csv(file.choose())
```

- `read.table read.csvheader = TRUE` ◦
- `as.is = TRUE stringsAsFactors = FALSE factor` ◦
- `read.csv2 sep = ";" dec = ","` ◦

---

`readr read_csv 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

`data.table fread` ◦ `read.table fread` ◦

```
# 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

- "filename.csv"
- **shell** "grep 'word' filename"
- "input1, input2 \n A, B \n C, D" ◦

freaddata.tabledata.frame **data.table** [] ◦ **data.frame**data.tableFALSE

```
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
```

- freadread.table ◦ na.comment # ◦
- fread" for quote ◦
- fread5 ◦

## .tsvR

file.path ◦ **WindowsMacLinux**paste ◦

```
FilePath <- file.path(AVariableWithFullProjectPath, "SomeSubfolder", "SomeFileName.txt.gz")

Data <- as.matrix(read.table(FilePath, header=FALSE, sep = "\t"))
```

◦

◦

FilePath8970 8970**79** ◦

```
system.time(expr=Data<-as.matrix(read.table(file=FilePath,header=FALSE,sep=" ") ))
```

system.time**267** ◦

```
user system elapsed
265.563  1.949 267.563
```

```

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

```

futile.matrixread.matrix1◦

## .CSV

# R

write.csv() CSV

```
write.csv(mtcars, "mtcars.csv")
```

row.names = FALSEna = "" ◦

readr::write\_csvwrite.csv ◦

```
library(readr)
write_csv(mtcars, "mtcars.csv")
```

## CSV

```
files = list.files(pattern="*.csv")
data_list = lapply(files, read.table, header = TRUE)
```

◦ data.frame data.frame

```
df <- do.call(rbind, data_list)
```

,; ◦ ◦

```
Column1 Column2 Column3 Column4Column5
```

```
1647 pi 'important' 3.141596.28318
1731 euler 'quite important' 2.718285.43656
1979 answer 'The Answer.' 42 42
```

constants.txt

---

## R

```
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
```

- Column4Column5
- widths
- read.fwf()

---

## readr

```
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<sub>fwf\_\*</sub> fwf\_empty
- readrbase R
- 

CSVTSV <https://riptutorial.com/zh-TW/r/topic/481/-csv-tsv->

# 36: data.table

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

|                      |             |
|----------------------|-------------|
| <b>id.vars</b>       | melt        |
|                      | melt        |
| <b>value.name</b>    | melt        |
| <b>value.var</b>     | dcast       |
|                      | dcastLHSRHS |
| <b>fun.aggregate</b> |             |

`data.table` ◦ `data.table` ◦

## Examples

### data.table - I.

`datasets` 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 ◦ `data.table` ◦ `data.frame` ◦

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

◦ ◦ ◦ **long** ◦ ◦ USArrests ◦

```
      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

```

melt◦

```

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

```

melt◦ USArrests UrbanPop◦ Murder AssaultRape 10◦ UrbanPop◦ id.vars◦

```

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

```

variable.name Murder Assault value.name◦ ◦

```

      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◦

```

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

```

```

      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 - II

dcastdcast ◦

```

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

```

◦

```

      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

```

**LHSRHS** ◦ `dcast` ◦ `value.var` ◦

`dcastfun.aggregate` ◦ `Decile` ◦

```

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

```

`Decile ~ Crime` ◦ `fun.aggregate` ◦

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

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

|    | 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    |

◦ `fun.aggregate` ◦

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

◦

|    | 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 |

[data.table](https://riptutorial.com/zh-TW/r/topic/6934/data-table) <https://riptutorial.com/zh-TW/r/topic/6934/data-table>

---

# 37: devtools

devtoolsR.

1. R
2. roxygen2
3. devtools

## Examples

R . . .

./ R.

---

---

LaTeX.

roxygen

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

roxygen . *doxygen*.

*roxygen*

```
## 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)
}
```



◦

◦

---

`./script1.R./script2.R`

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

`./MyPackage/man/◦`

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

`./R CMD Rd2pdf MyPackage◦`

---

## 1.

`./MyPackage/DESCRIPTION◦ Package Version License Description Title AuthorMaintainer◦`

`Depends R version <3.2.0Imports R version> 3.2.0◦`

## 2.

◦ `R/ ./MyPackage/ R/R/man/◦`

- `data/◦ .RData``data()load()`
  - `tests/◦◦`
  - `src/ C / C ++ / FortranRcpp ...◦`
  - `exec/◦`
  - `misc/◦`
- 

`./MyPackage/Read-and-delete-me◦`

◦

`devtools::install("MyPackage")◦`

**tarball.**`./ R CMD build MyPackage◦`

---

# Github

*MyPackage* MyPackage/◦ ◦

## devtoolsgithub

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

# CRAN

CRAN ◦ R CMD check◦

◦ tarball◦

◦ ◦ ◦

## Vignettes◦

- 
- Rmarkdown `install.packages("rmarkdown")`
  - Pandoc
- 

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

```
./vignettes/MyVignette.Rmd◦
```

Markdown ◦

## MarkdownRMarkdown

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

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

```
./DESCRIPTION./DESCRIPTION./DESCRIPTION◦
```

devtools <https://riptutorial.com/zh-TW/r/topic/10884/devtools>

---

# 38: hclust

statshclust ◦

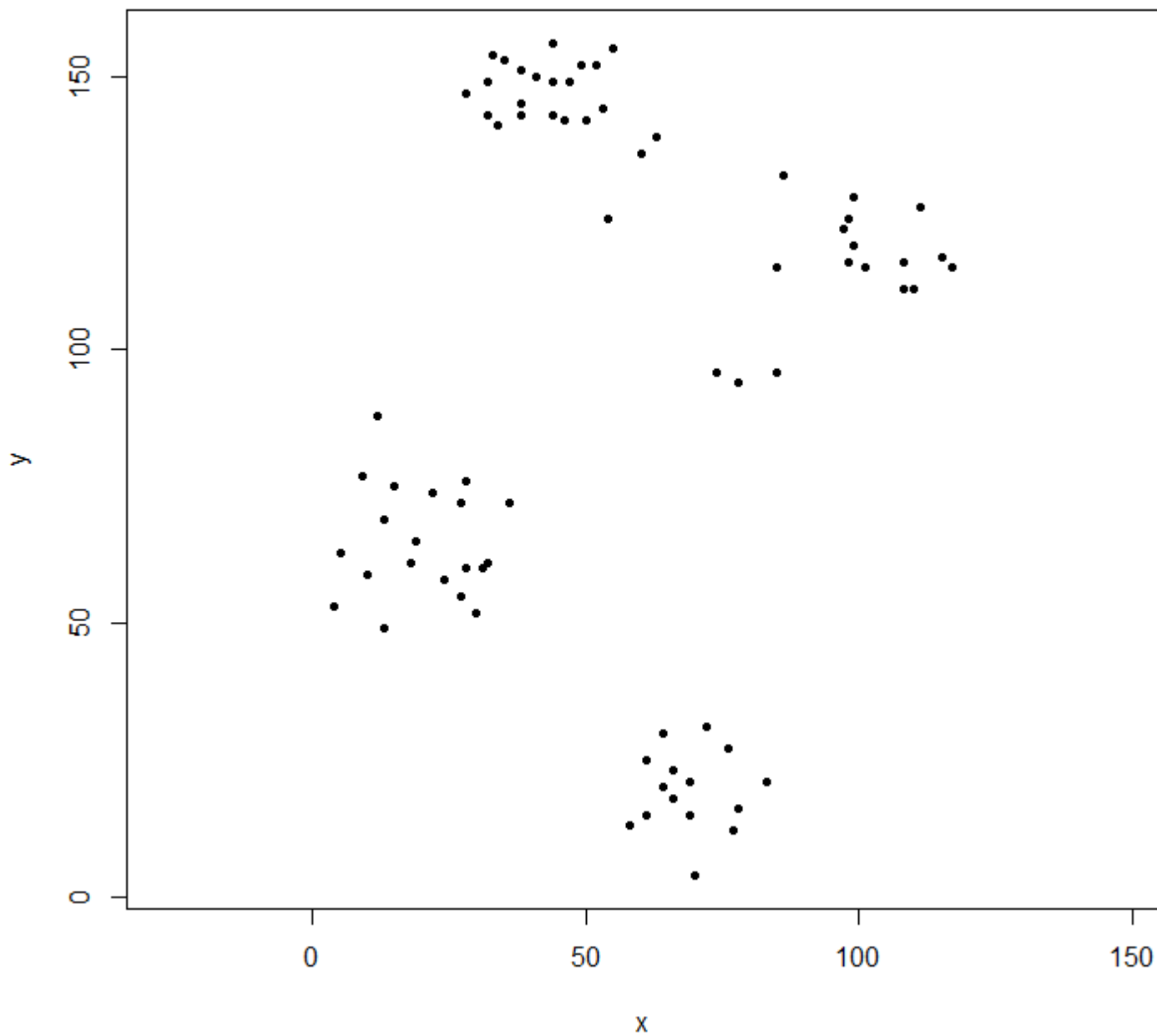
hclust [CRAN](#) ◦

## Examples

### 1 - hclust

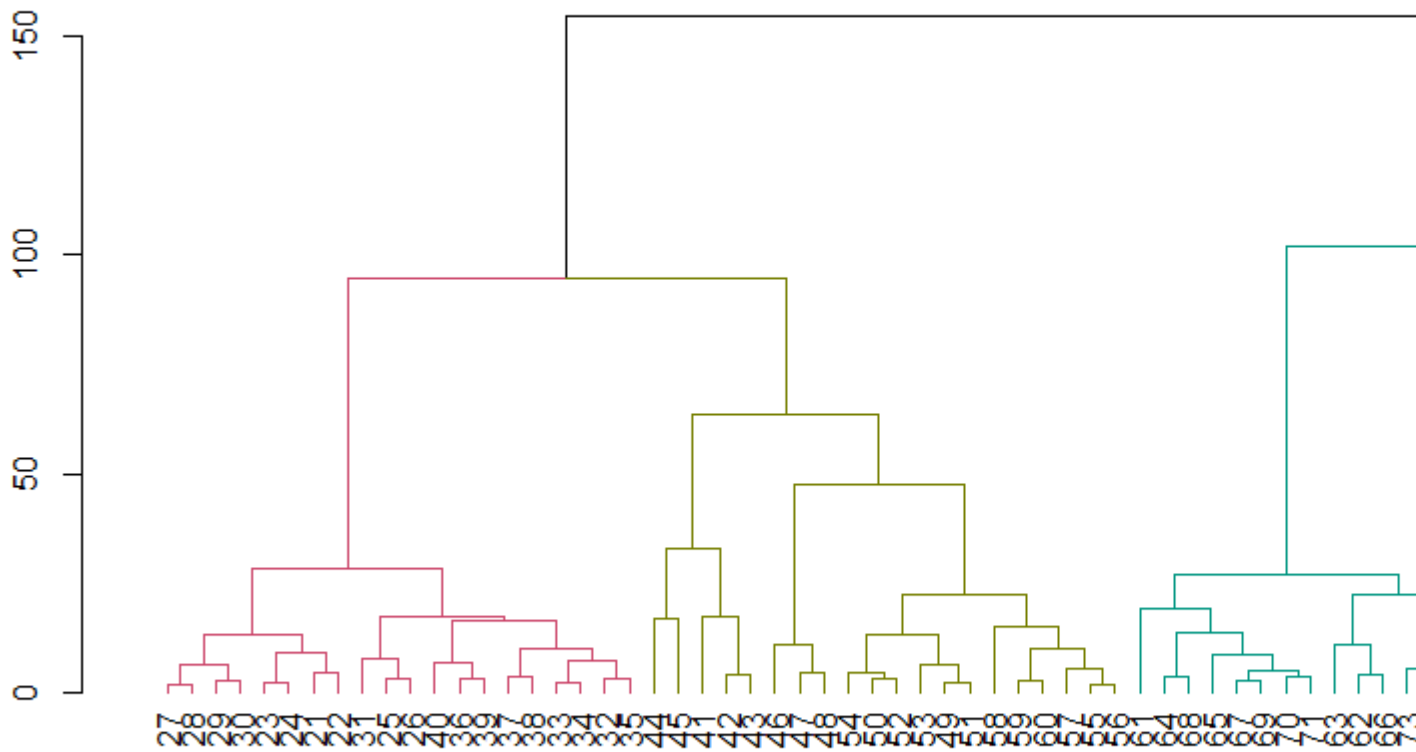
ruspini - ◦

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



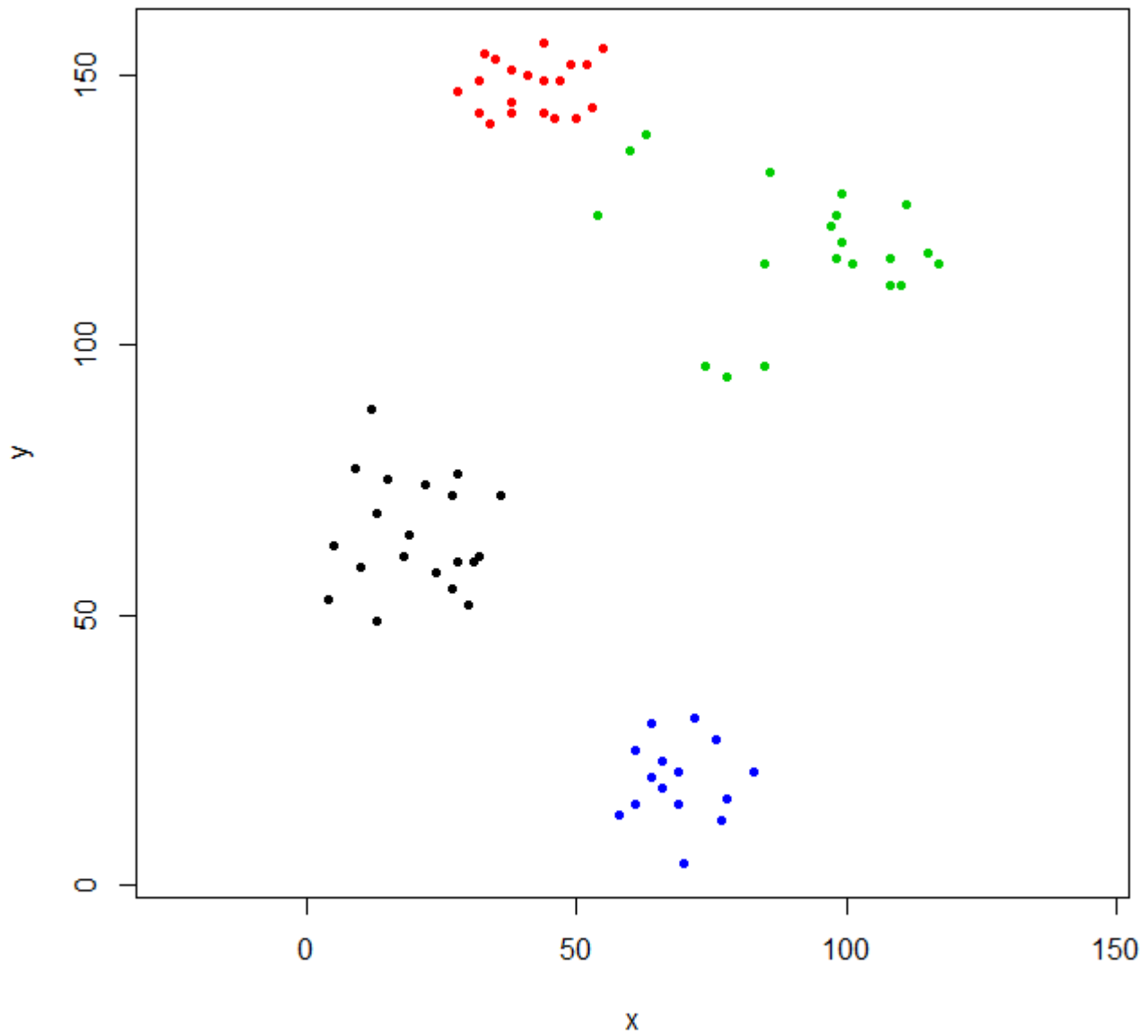
hclust ◦ ◦ hang ◦

```
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)
```



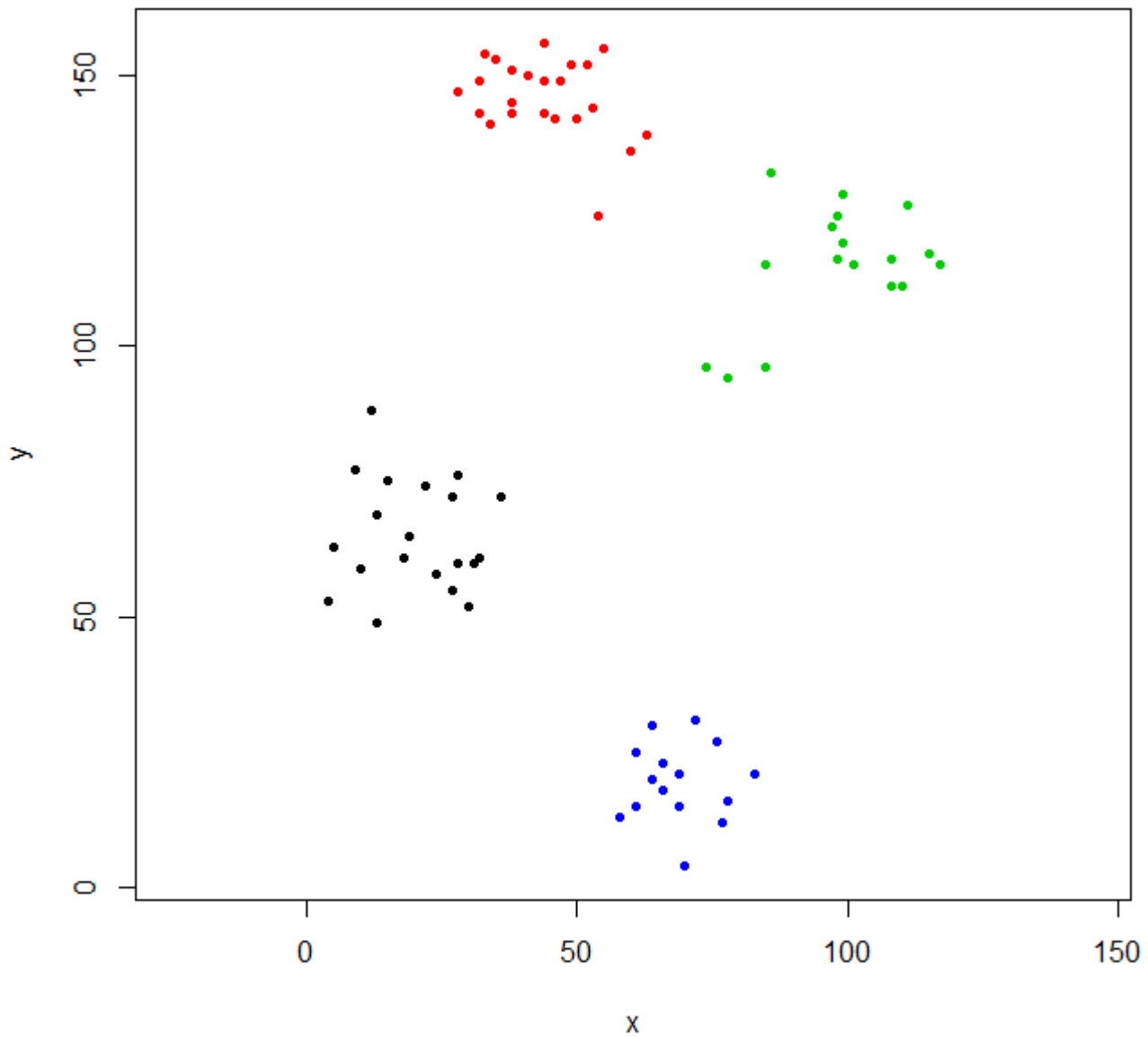
- k◦

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



◦ ◦

```
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)
```



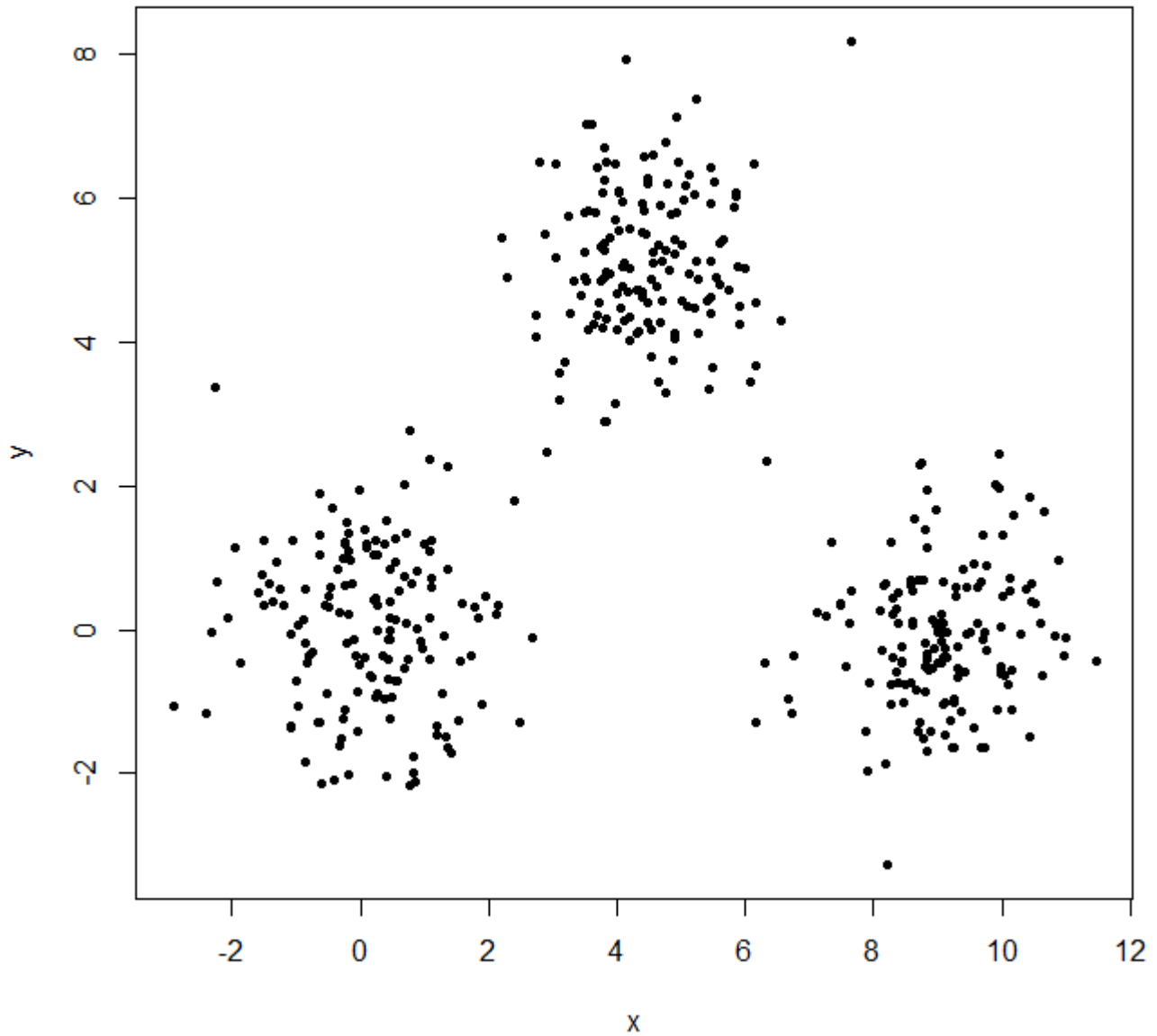
“” method.

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

## 2 - hclust

- 
- 

```
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)
```



```

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

```

hclust. "k".

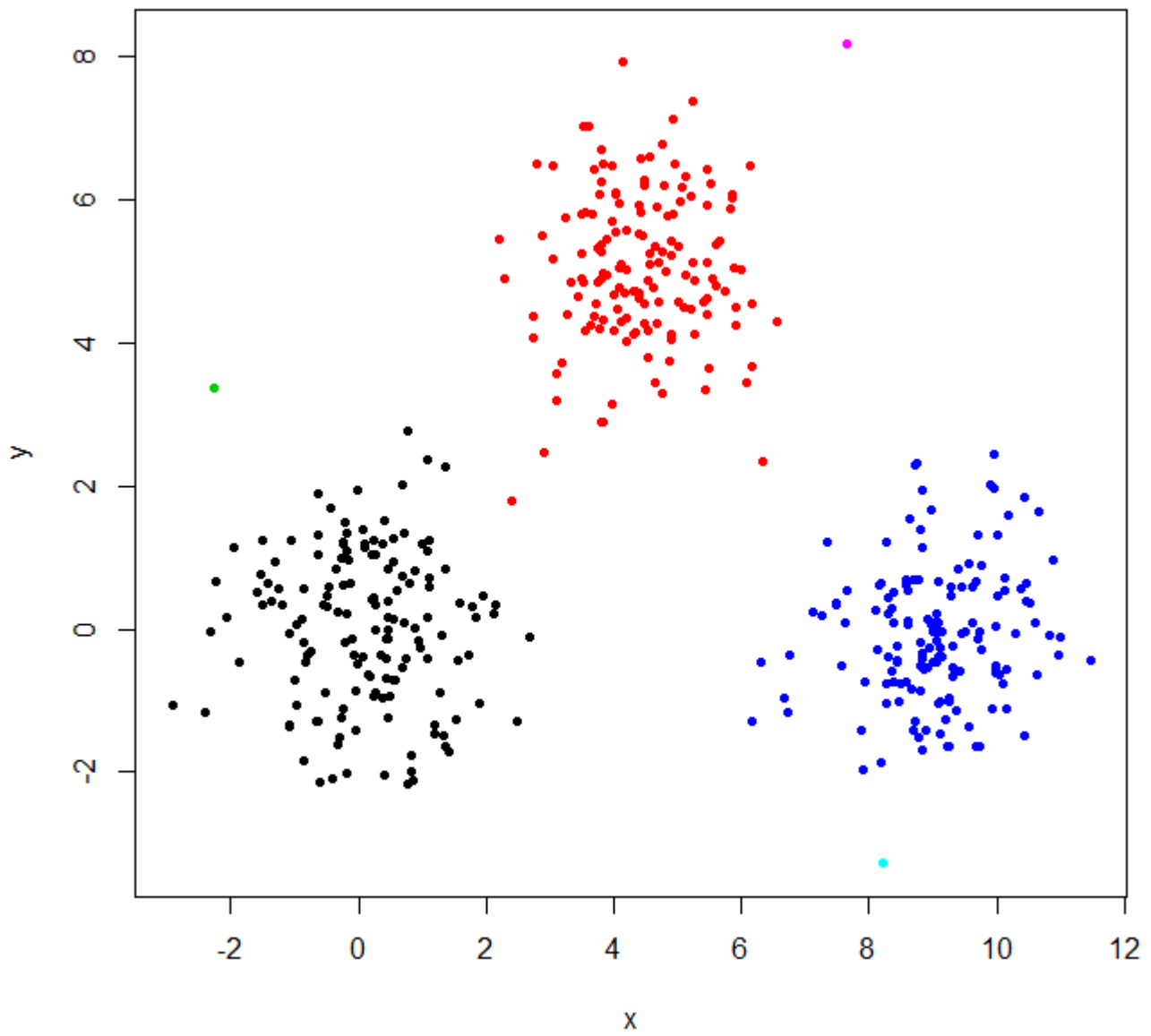
```

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)
```



[StackOverflow](#).

[hclust https://riptutorial.com/zh-TW/r/topic/8084/hclust](https://riptutorial.com/zh-TW/r/topic/8084/hclust)

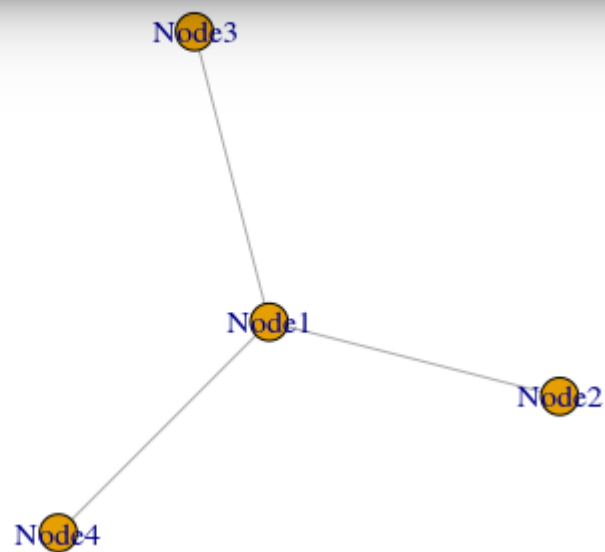
# 39: igraph

## Examples

Rigraph. R v.3.2.3igraph.

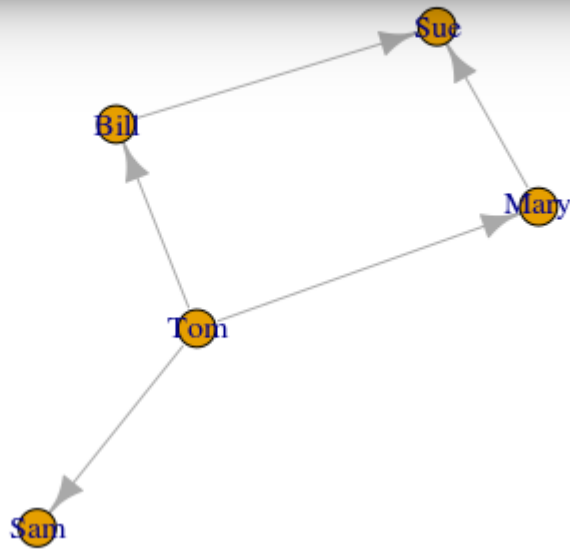
```
g<-graph.formula(Node1-Node2, Node1-Node3, Node4-Node1)
plot(g)
```

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



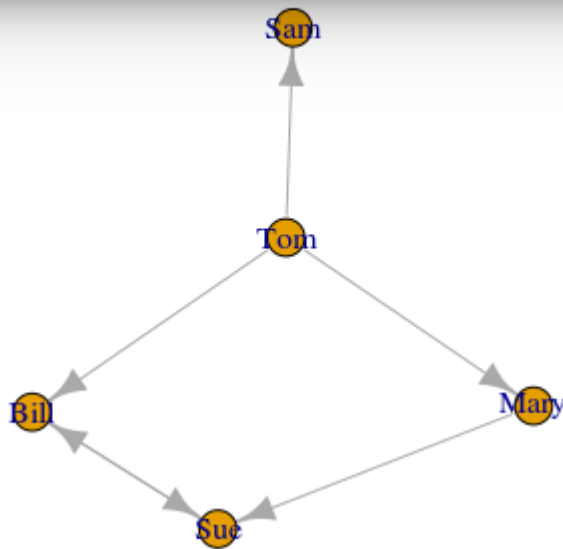
```
dg<-graph.formula(Tom->Mary, Tom->Bill, Tom->Sam, Sue->Mary, Bill->Sue)
plot(dg)
```

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



```
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)
> 
```



igraph <https://riptutorial.com/zh-TW/r/topic/4851/igraph>

---

# 40: RMarkdown

## Examples

HTMLLaTeX. ◦

- knitr
- xtable
- 

## HTML

```
---
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	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(df), type="html")
```

	mpg	cyl	disp	hp
Mazda RX4	21.000000	6	160	110
Mazda RX4 Wag	21.000000	6	160	110
Datsun 710	22.800000	4	108	93
Hornet 4 Drive	21.400000	6	258	110

### Print tables using `pander`

```
pander(df)
```

	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

## PDF

```
---  
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 Schmelzer  
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, caption = "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

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

## LaTeX Preamble

```
RMarkdownLaTeX\usepackage{}
```

## 1.YAMLheader-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)  
```\br/>  
# Section 1  
  
As you can see, this text uses the Computer Modern Font!
```

### Including LaTeX Preamble Commands in RMarkdown

#### Section 1

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

## 2.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!
```

**includes.tex**header-includes°

**LaTeX**template ° °

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

**bibtexcnaYAML**bibliography: ° biblio-style: ° °

```
---
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/>.

## R-markdown

# R-markdown

R-markdown markdown R . R .

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

o

```
```{r name, echo=TRUE, include=TRUE, ...}  
2*2  
````
```

o

- **echo** boolean
- **include**
- **fig.width** numeric
- **fig.height**
- **fig.cap**

tag=value o

# R-markdown

R-markdown Rr-markdown o

```
# 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-markdown

R `knitr` R-markdown Rmarkdown。

R-markdownpdf / html

1. `knitr` R-markdown markdown。
2. *pandoc* markdownpdf / html。

`knitr` `knit2html()` `knit2pdf()` **markdown**

income.Rmd **R**pdf

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

。

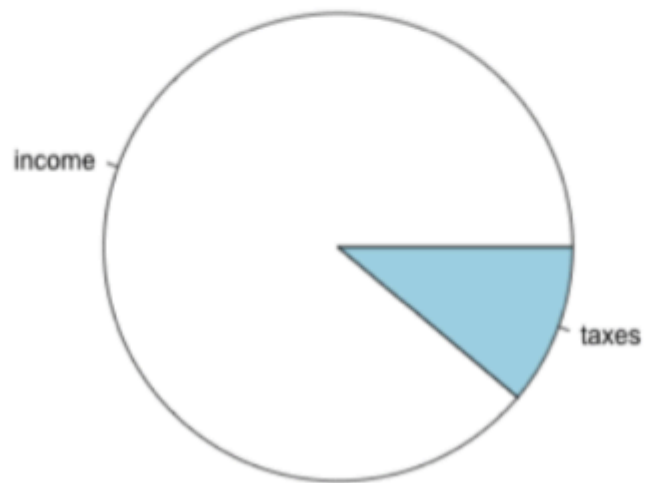
## Title

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
```



RMarkdown <https://riptutorial.com/zh-TW/r/topic/4572/rmarkdown>

# 41: S4

OOP Java GOF ""。

ROOS4 Object System。

## Examples

RS4。

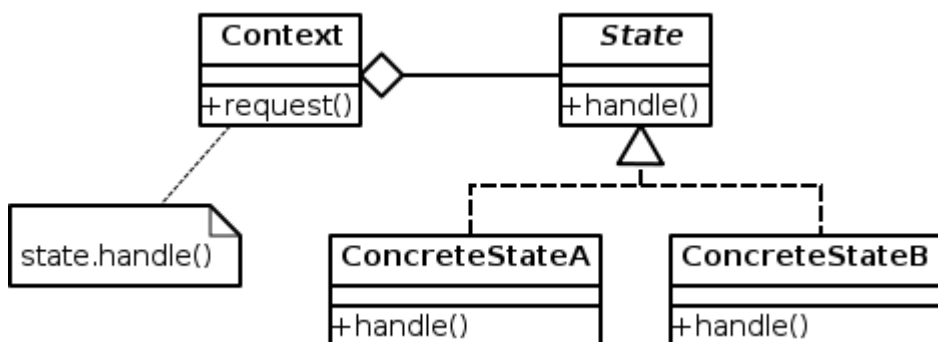
";"。 Name; [Address;]Phone。

```
GREGORY BROWN; 25 NE 25TH; +1-786-987-6543
DAVID SMITH; 786-123-4567
ALAN PEREZ; 25 SE 50TH; +1-786-987-5553
```

。 。 。 。

- `"^([AZ]'?\s+)* *[AZ]+(\s+[AZ]{1,2}\.\.? +)*[AZ]+((-|\s+)[AZ]+)*$"。` RAFAEL REAL, DAVID R. SMITH, ERNESTO PEREZ GONZALEZ, 0' CONNOR BROWN, LUIS PEREZ-MENA
- `"^\s[0-9]{1,4}(\s+[AZ]{1,2}[0-9]{1,2}[AZ]{1,2}|[AZ]\s[0-9]+)$"。` 11020 LE JEUNE ROAD 87 SW 27TH。
- `"^\s*(\s+1(-|\s+))*[0-9]{3}(-|\s+)[0-9]{3}(-|\s+)[0-9]{4}$"。` 305-123-4567, 305 123 4567, +1-786-123-4567。
- 。
- R"\"。
- [regex101.com](http://regex101.com)。

。



- Context **State Machine Process**。 handle() handle() Statehandle()。 。 Context
  - state
  - handle() ...
- State **State Machine**。 。
  - name, pattern
  - doAction() isState pattern...

- Concrete States **state** StateContext ◦ InitState NameState AddressState PhoneState ◦ ◦ ◦

handle() doAction() goNext() ◦ Context handle() StateContext doAction() ◦

## S4Person

```
setClass(Class = "Person",
  slots = c(name = "character", address = "character", phone = "character")
)
```

- setClass "initialize" prototype, representation ◦

```
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 methods methods ◦ R

```
> initialize
```

```
function (.Object, ...) {...}
```

setMethod **exactly** .Object ◦

show **Java** toString()

```
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)
  }
)
```

toString() **Java** ◦

Person **R** ◦

```
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)
}
)

```

ROO。 1setGeneric 2setMethod。。

S4State。

```

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
  }
)

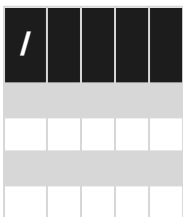
setMethod("show", signature = "State",
  definition = function(object) {
    info <- sprintf("%s@[name='%s', pattern='%s']", class(object),
      object@name, object@pattern)
    cat(info)
    invisible(NULL)
  }
)

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"))

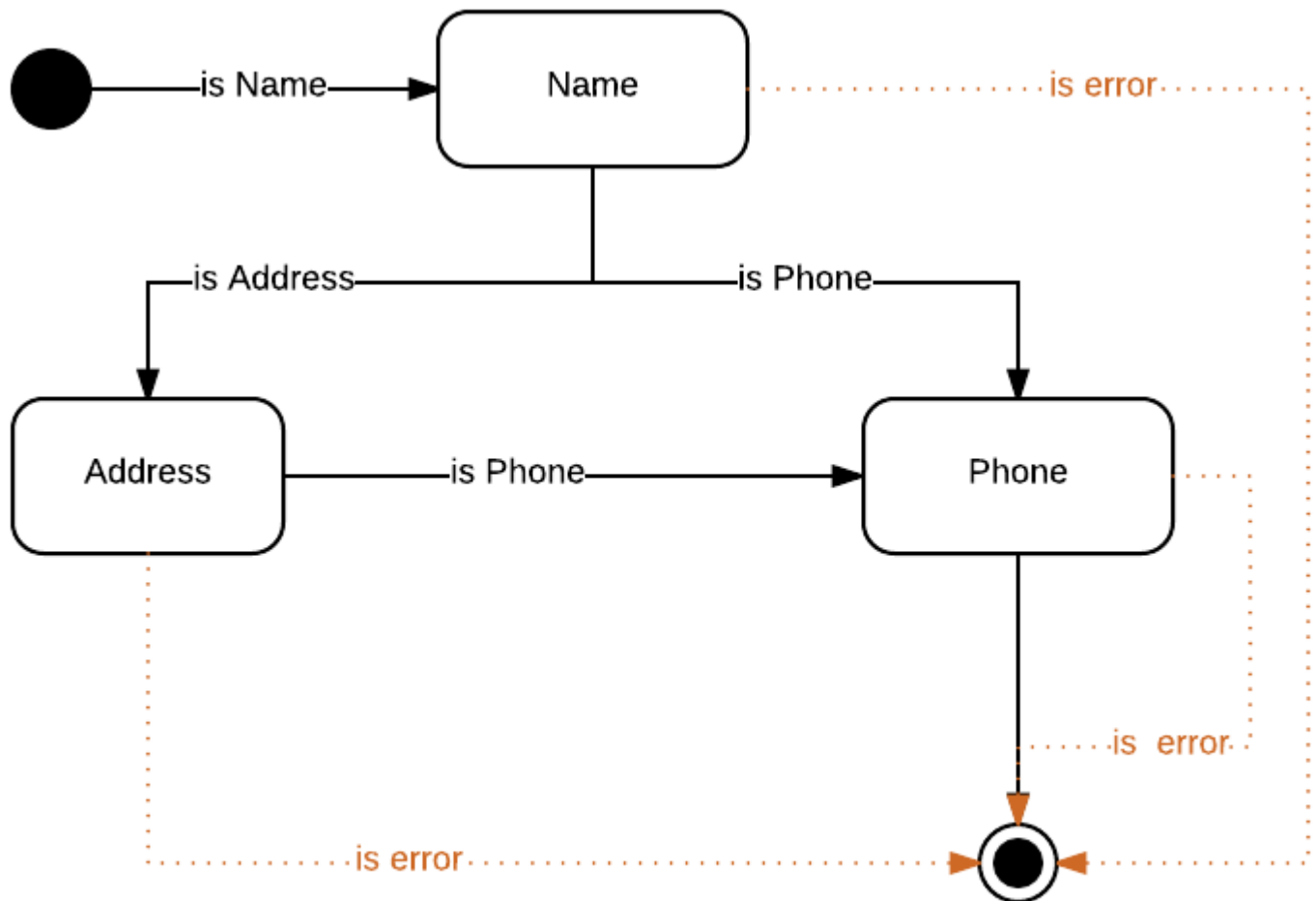
```

State name pattern isState() doAction()。



[row, col]=[i,j]ji。

。 UML



**is error:** when the input argument has an invalid pattern

State

```

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()
  }
)

```

Rcontains◦

show callNextMethod()

NA

◦

State

```
setMethod(f = "isState", signature = "InitState",
  definition = function(obj, input) {
    nameState <- new("NameState")
    result <- isState(nameState, input)
    return(result)
  }
)
```

pattern name ◦

```
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◦ @-operator◦ get/set OO get-set◦

doAction◦

```
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()
  }
)
```

grepl◦

```
setMethod(f="isState", signature="NameState",
  definition=function(obj, input) {
```

```

    result <- grepl(obj@pattern, input, perl=TRUE)
    return(result)
  }
)

```

```

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)
  }
)

```

◦

- personaddressphone ◦
- state

isState()◦ addressState, phoneState ◦

◦

```

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)
  }
)

```

```

setClass("PhoneState", contains = "State")

setMethod("initialize", "PhoneState",
  definition = function(.Object, name = "phone",
    pattern = "^\\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)
  }
)

```

context persons°

```

setMethod(f = "doAction", "PhoneState",
  definition = function(obj, input, context) {
    context <- addPerson(context, context@person)
    context@state <- new("InitState")
    return(context)
  }
)

```

Context°

```

setClass(Class = "Context",
  slots = c(state = "State", persons = "list", person = "Person")
)

```

- state
- person ◦
- persons ◦

name◦

```

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"))

```

- handle() statedoAction()◦
- addPerson personpersons◦
- parseLine()
- parseLines()
- as.df() persons◦

handle() contextstatedoAction()

```

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)
}
)

```

Rstrsplit()◦ handle() state person personscontext ◦

```

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)
  }
)

```

Becuae R obj

```

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)
  }
)

```

personsS4 Person◦ R◦ as.listPerson◦ lapply()persons◦ lappy()data.framepersons.list◦ rbind()

```

# 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)
  }
)

```

◦ ◦

```

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"  
)
```

context

```
context <- new("Context")  
context <- parseLines(context, s)
```

```
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

```
> show(context@persons[[1]])  
Person@[name='GREGORY BROWN', address='25 NE 25TH', phone='+1-786-987-6543']
```

```
> show(new("PhoneState"))  
PhoneState@[name='phone', pattern='^\s*(\+1(-|\s+))*[0-9]{3}(-|\s+)[0-9]{3}(-|\s+)[0-9]{4}$']
```

as.list()

```
> as.list(context@persons[[1]])  
$name  
[1] "GREGORY BROWN"  
  
$address  
[1] "25 NE 25TH"  
  
$phone  
[1] "+1-786-987-6543"  
  
>
```

ROO。 R OOOOP。 ◦ Java / C object.setID("A1") R setID(object, "A1") ◦ ◦ this"."◦ "Person"  
"isState"◦

S4Java / C◦ ◦ ◦ Stateif-elseif-else◦

◦

◦

S4 <https://riptutorial.com/zh-TW/r/topic/9126/s4>

## 42: stringi

```
install.packages("stringi")
```

```
require("stringi")
```

### Examples

```
stri_count_fixed("babab", "b")  
# [1] 3  
stri_count_fixed("babab", "ba")  
# [1] 2  
stri_count_fixed("babab", "bab")  
# [1] 1
```

```
length(gregexpr("b", "babab")[[1]])  
# [1] 3  
length(gregexpr("ba", "babab")[[1]])  
# [1] 2  
length(gregexpr("bab", "babab")[[1]])  
# [1] 1
```

```
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
```

### R

```
sapply(c("b", "ba"), function(x) length(gregexpr(x, "babab")[[1]]))  
# b ba  
# 3 2
```

- a

- a

```
stri_count_regex("a1 b2 a3 b4 aa", "a.")  
# [1] 3  
stri_count_regex("a1 b2 a3 b4 aa", "a\\d")  
# [1] 2
```

```
stri_dup("abc", 3)  
# [1] "abcabcabc"
```

### R

```
paste0(rep("abc", 3), collapse = "")
```

```
# [1] "abcabcabc"
```

```
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"
```

## R

```
> 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"
```

```
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."
```

```
stri_split_fixed("Apples, oranges and pineaplles.", c(" ", ",", "s"))
```

```
# [[1]]
# [1] "Apples," "oranges" "and" "pineaplles."
#
# [[2]]
# [1] "Apples" " oranges and pineaplles."
#
# [[3]]
# [1] "Apple" ", orange" " and pineapple" "."
```

[stringi https://riptutorial.com/zh-TW/r/topic/1670/stringi](https://riptutorial.com/zh-TW/r/topic/1670/stringi)

# 43: texreg

texreg: HTML.docMS Office Word.

- [CRAN](#)

## Examples

```
# 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')
```

|             | <b>Model 1</b> | <b>Model 2</b> | <b>Model 3</b> |
|-------------|----------------|----------------|----------------|
| (Intercept) | 37.29***       | 37.23***       | 38.75***       |
|             | (1.88)         | (1.60)         | (1.79)         |
| wt          | -5.34***       | -3.88***       | -3.17***       |
|             | (0.56)         | (0.63)         | (0.74)         |
| hp          |                | -0.03**        | -0.02          |
|             |                | (0.01)         | (0.01)         |
| cyl         |                |                | -0.94          |
|             |                |                | (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() ◦ ◦

```
# 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)

|             | <b>A</b>          | <b>B</b>          | <b>C</b>          |
|-------------|-------------------|-------------------|-------------------|
| (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

texreg <https://riptutorial.com/zh-TW/r/topic/9037/texreg>



# 44:

- ◦
- ◦ ◦ ◦
- ◦ ◦ ◦ ◦
- “”

- ◦ Joseph Fourier
- 

Dirac delta 4

- ◦  $\mathbb{R}$  original  $\mathbb{N}$  DFT group  $= \mathbb{Z} \text{ mod } N = S_1 \approx$  ◦ FFT DFT

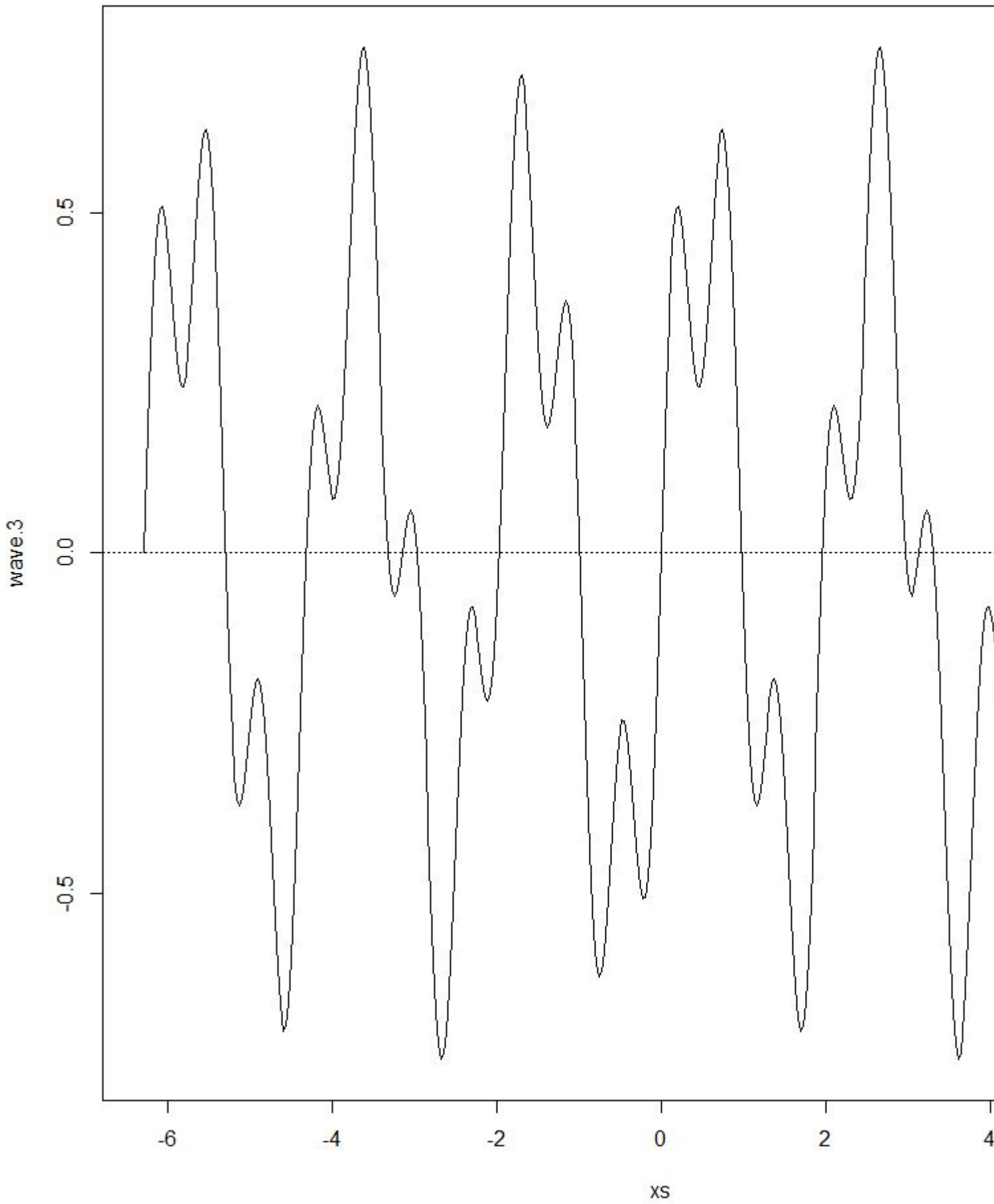
## Examples

Joseph Fourier ◦ ◦ ◦

```
# 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)

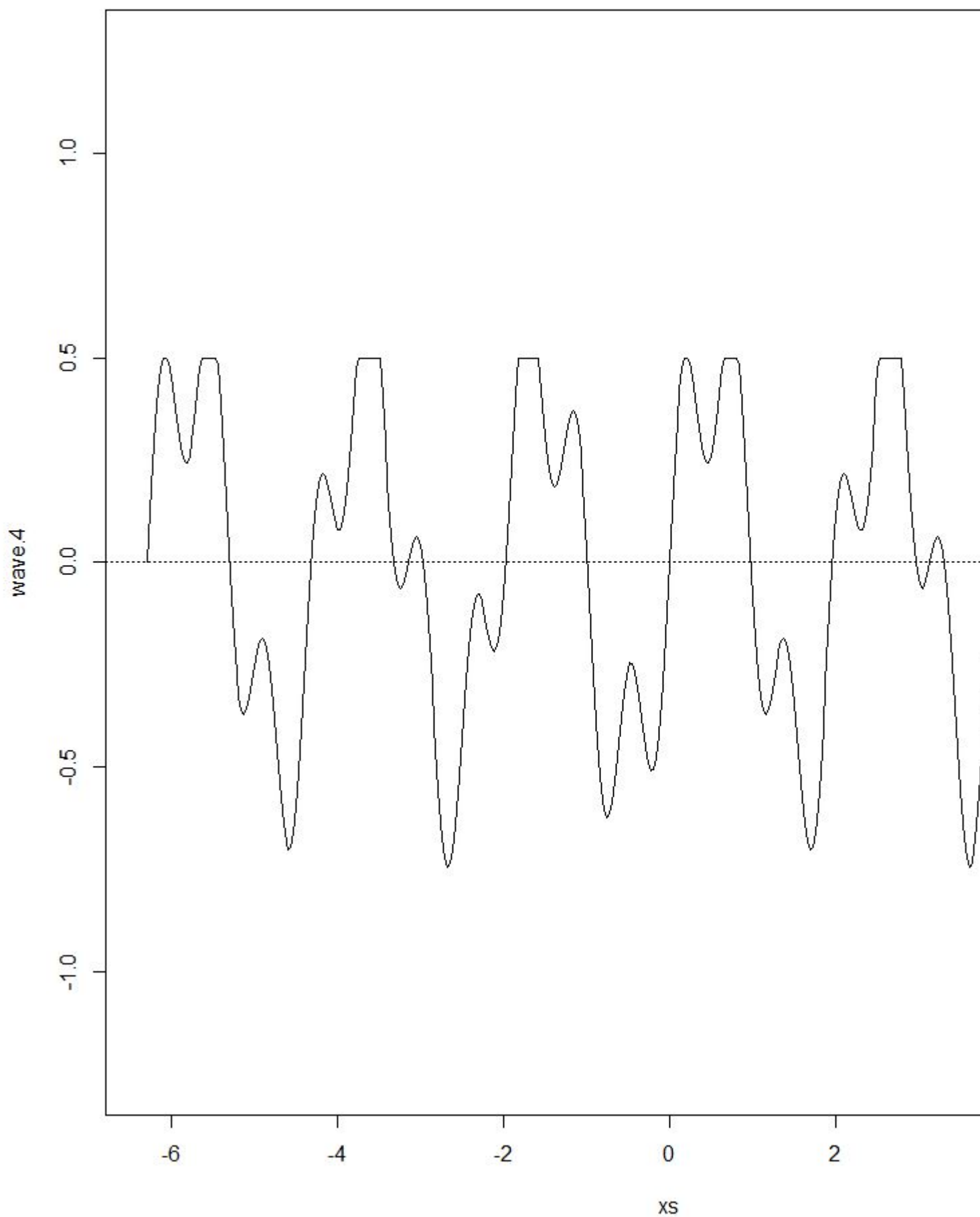
# Complex Wave
wave.3 <- 0.5 * wave.1 + 0.25 * wave.2
plot(xs, wave.3, type="l"); title("Eg complex wave"); abline(h=0, lty=3)
```

### 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



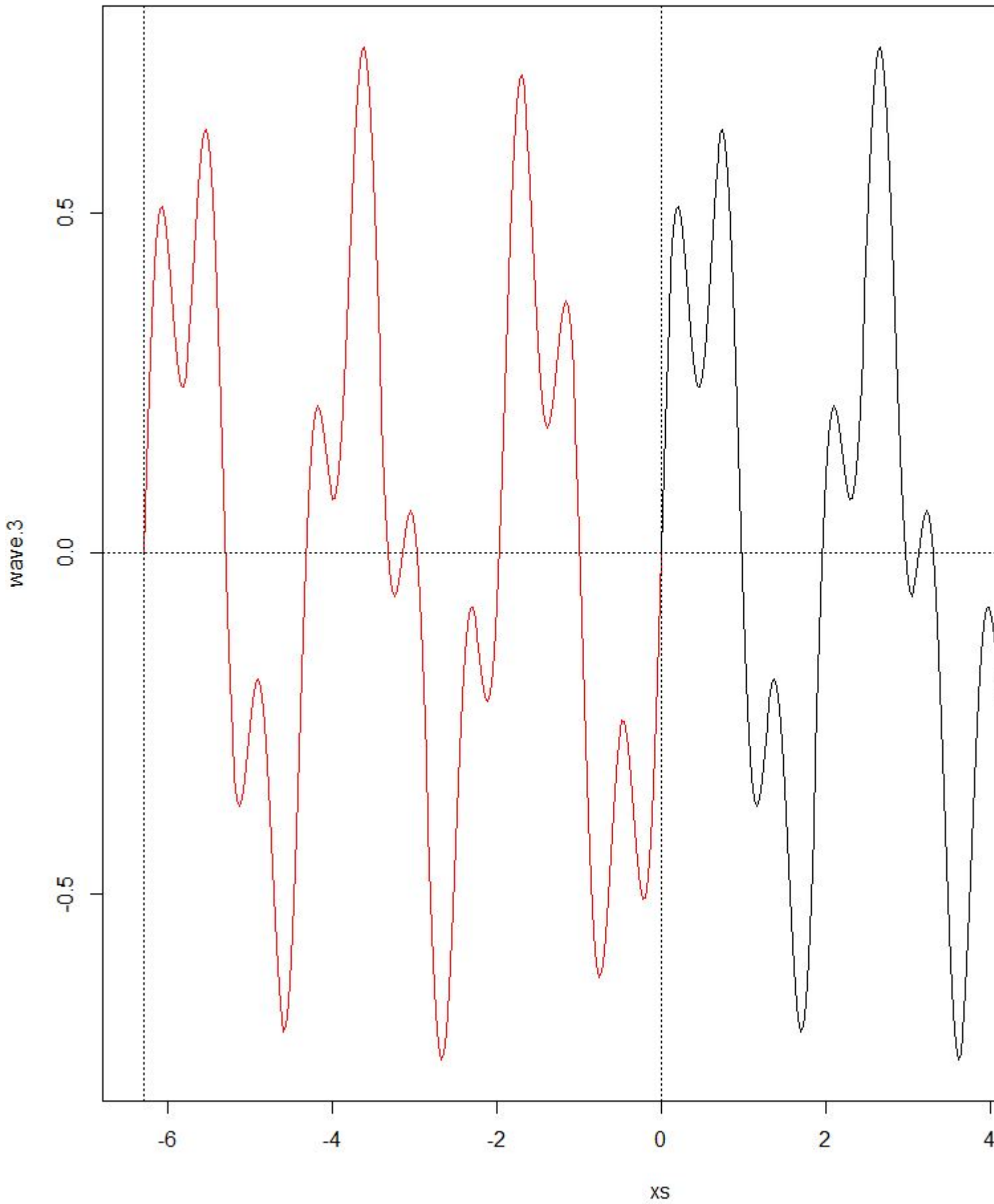
◦  $f\lambda = v / f_0 v$  ◦

- T
- $s_r$  ◦  $s_i N \cdot s_i = TN$
- $f_0 T$  ◦  $12\pi$  ◦  $f_0 T * f_0 T * f_0$

```
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



## R

```
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)}
```

<https://riptutorial.com/zh-TW/r/topic/4139/>

# 45:

I/O.

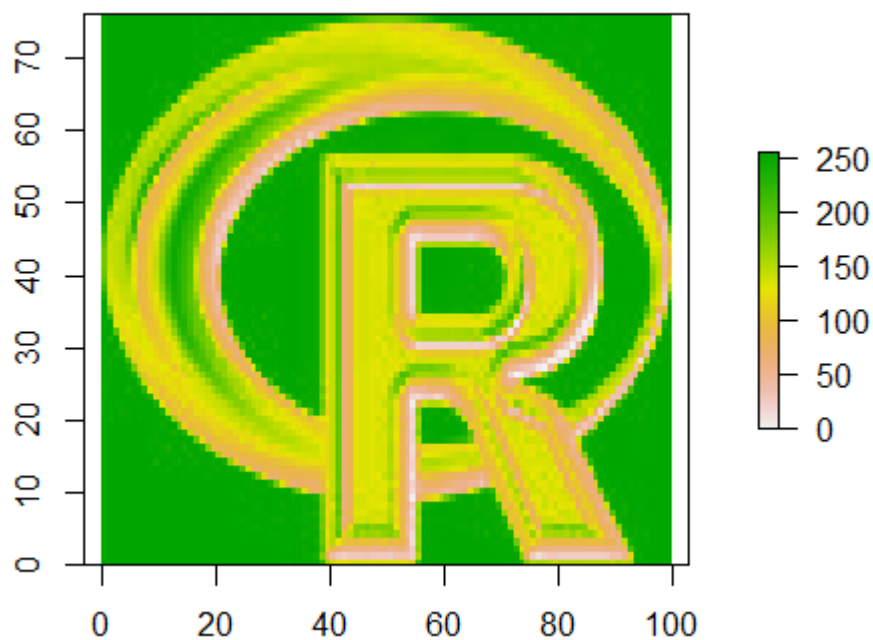
## Examples

### GLCM

Haralick1973. `glcmRasterLayer`

```
library(glcm)
library(raster)

r <- raster("C:/Program Files/R/R-3.2.3/doc/html/logo.jpg")
plot(r)
```

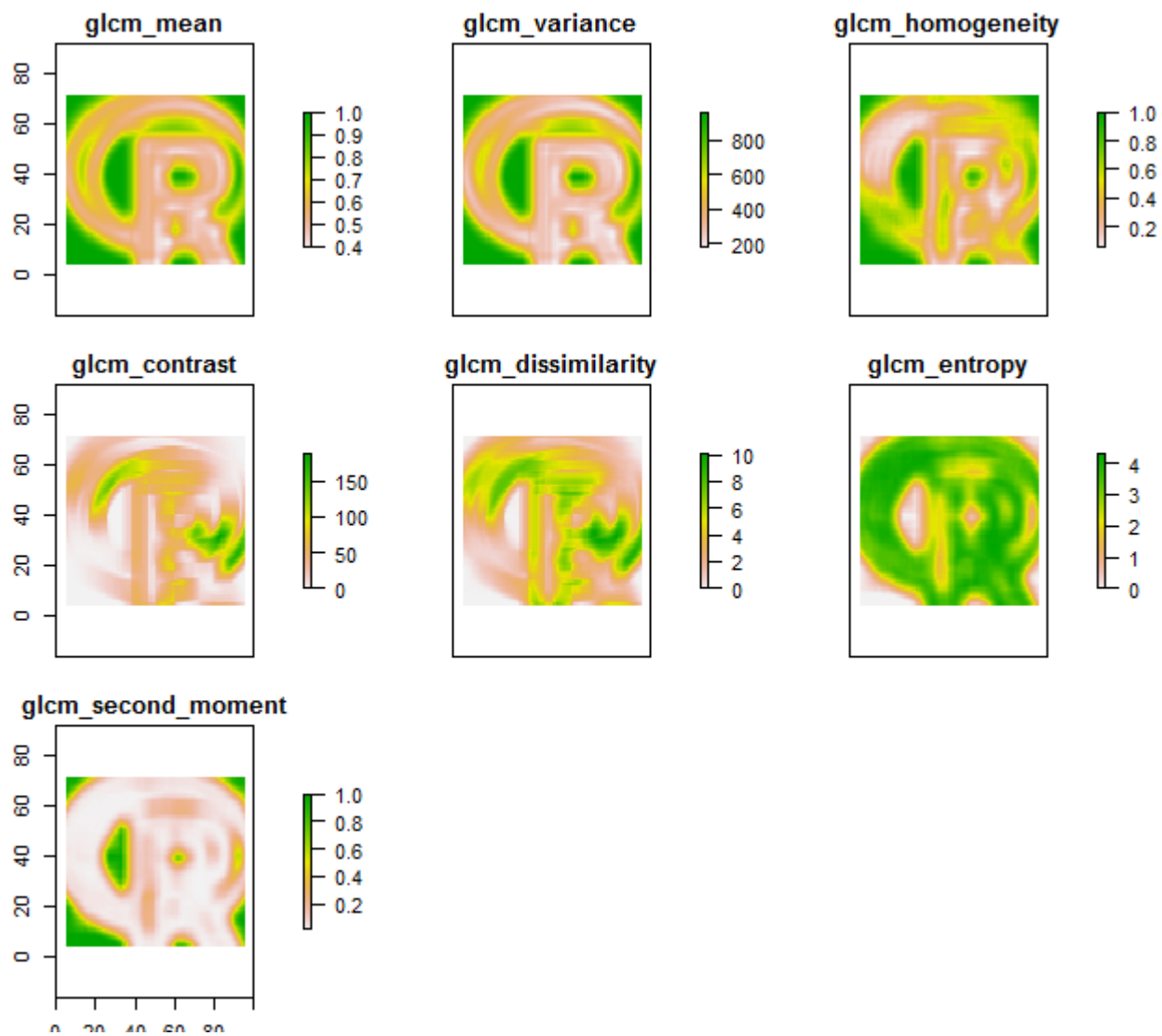


### GLCM

```
rglcm <- glcm(r,
  window = c(9,9),
  shift = c(1,1),
  statistics = c("mean", "variance", "homogeneity", "contrast",
    "dissimilarity", "entropy", "second_moment")
)

plot(rglcm)
```

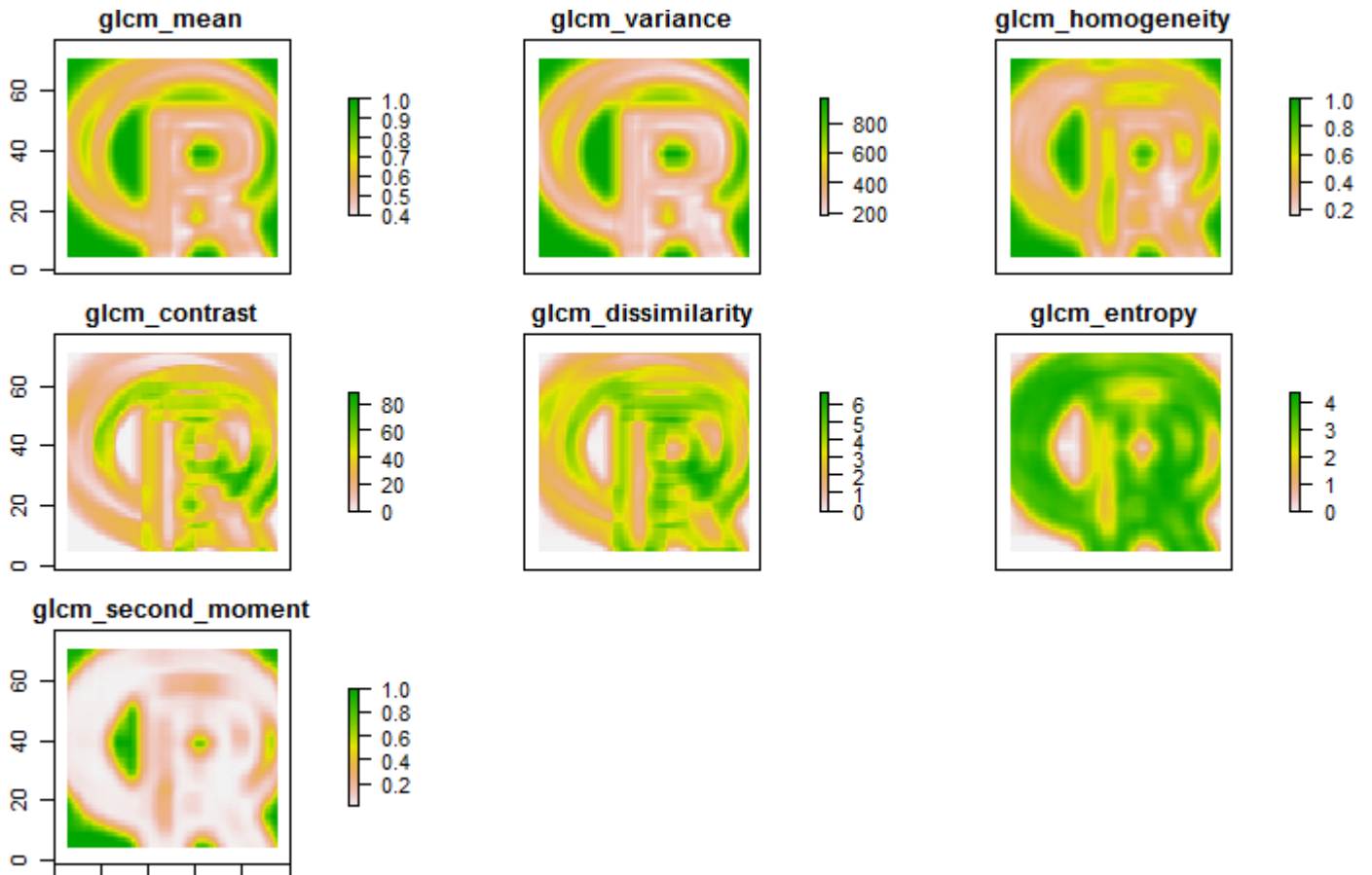




40°45°90°135° shift

```
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")
)

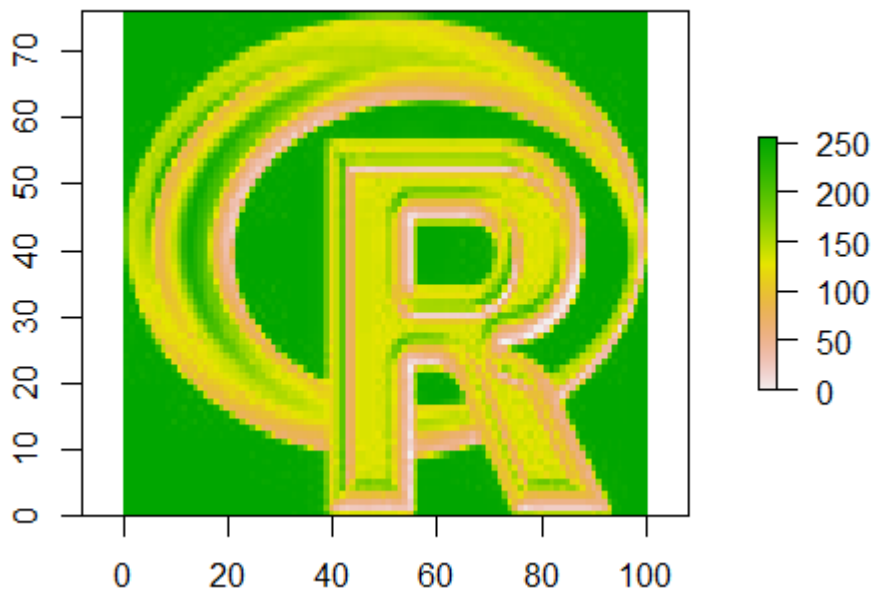
plot(rglcm1)
```



mmandn

```
library(raster)
library(mmand)

r <- raster("C:/Program Files/R/R-3.2.3/doc/html/logo.jpg")
plot(r)
```



9x9 disc boxdiamond

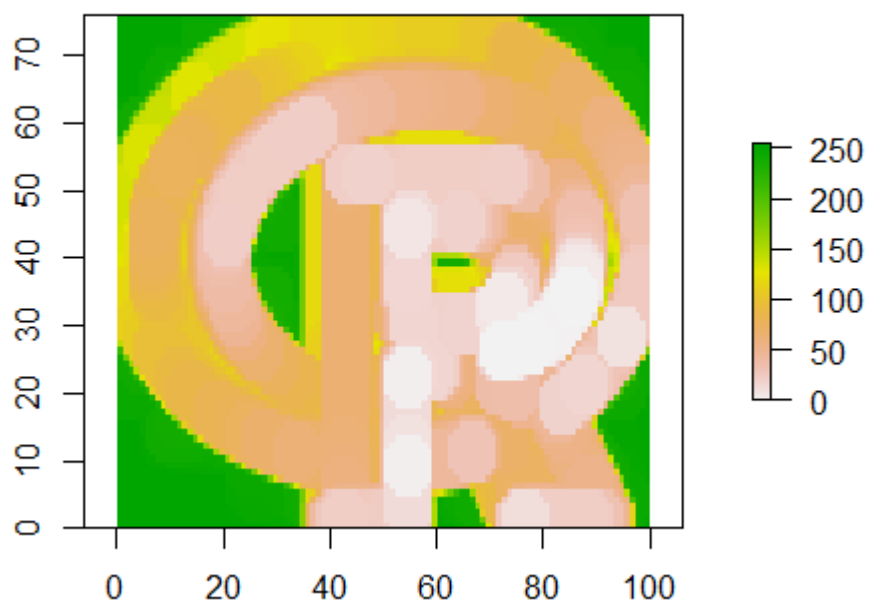
```
sk <- shapeKernel(c(9,9), type="disc")
```

erode() ◦

```
rArr <- as.array(r, transpose = TRUE)
rErode <- erode(rArr, sk)
rErode <- setValues(r, as.vector(aperm(rErode)))
```

erode() dilate() opening() closing() ◦

```
plot(rErode)
```



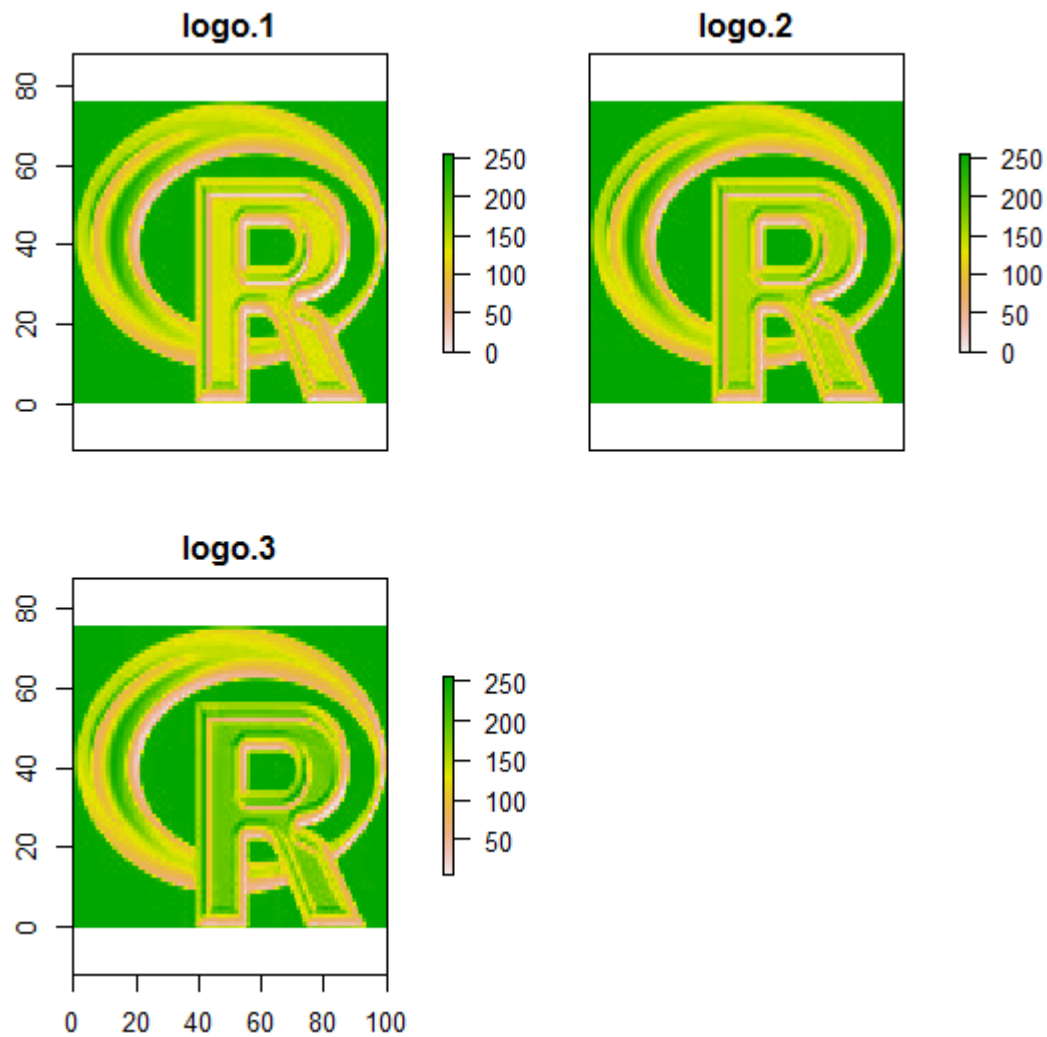
<https://riptutorial.com/zh-TW/r/topic/3726/>

# 46: I / O.

## Examples

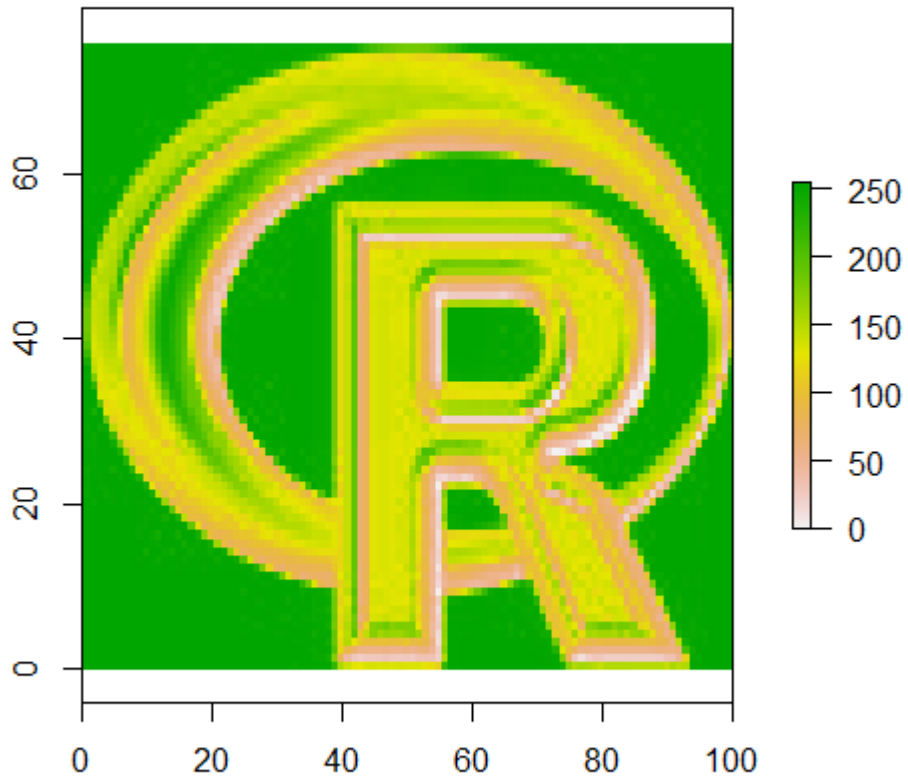
### R-Logo

```
library(raster)
r <- stack("C:/Program Files/R/R-3.2.3/doc/html/logo.jpg")
plot(r)
```



```
RasterStack[[1]]
```

```
plot(r[[1]])
```



I / O. <https://riptutorial.com/zh-TW/r/topic/5539/i---o->

# 47:

## Examples

R° ◦

a

```
a <- matrix(1:9, 3, 3)
```

```
> class(a)
[1] "matrix"
```

```
> a %% t(a)
      [,1] [,2] [,3]
[1,]  66   78   90
[2,]  78   93  108
[3,]  90  108  126
```

a

```
> dim(a)
[1] 3 3
> nrow(a)
[1] 3
> ncol(a)
[2] 3
```

head tailstr

```
> 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
```

◦ str◦ a

```
a <- c(a)
```

```
> class(a)
[1] "integer"
```

a°

```
> dim(a)
NULL
```

```
> length(a)
[1] 9
```

```
> class(a * 1.0)
[1] "numeric"
```

data.frames

```
a <- as.data.frame(a)
names(a) <- c("var1", "var2", "var3")
```

```
> names(a)
[1] "var1" "var2" "var3"
```

R°

<https://riptutorial.com/zh-TW/r/topic/3565/>

---

# 48:

R。

R。 [R.](#)

## Examples

“”。

```
data.frame table
```

---

◦ `getwd()`?`setwd` ◦

```
..w = options()$width
options(width = 500) # reduce text wrapping
sink(file = "mytab.txt")
  summary(mtcars)
sink()
options(width = ..w)
rm(..w)
```

---

## CSV

◦ `getwd()`?`setwd` ◦

```
write.csv(mtcars, file="mytab.csv")
```

- 
- `knitr::kable`
  - 
  - `tables::tabular`
  - [texreg](#)
  - `xtable`

`utilsSweave` **LaTeX**。

- 
- `KnitrRMarkdown`

<https://riptutorial.com/zh-TW/r/topic/9039/>



# 49:

R?Distributions◦

- **d-**
- **p-**
- **q-**
- **r-**

R?Distributions ◦

## Examples

\*norm◦

```
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)
```

0

```
dnorm(0)
```

0.3989423 ◦

```
pnorm(0) .5 ◦ 0◦
```

```
qnormpnorm◦ qnorm(.5)0 ◦
```

rnorm

```
rnorm(10)
```

10◦

```
rnorm(10, mean=4, sd= 3)
```

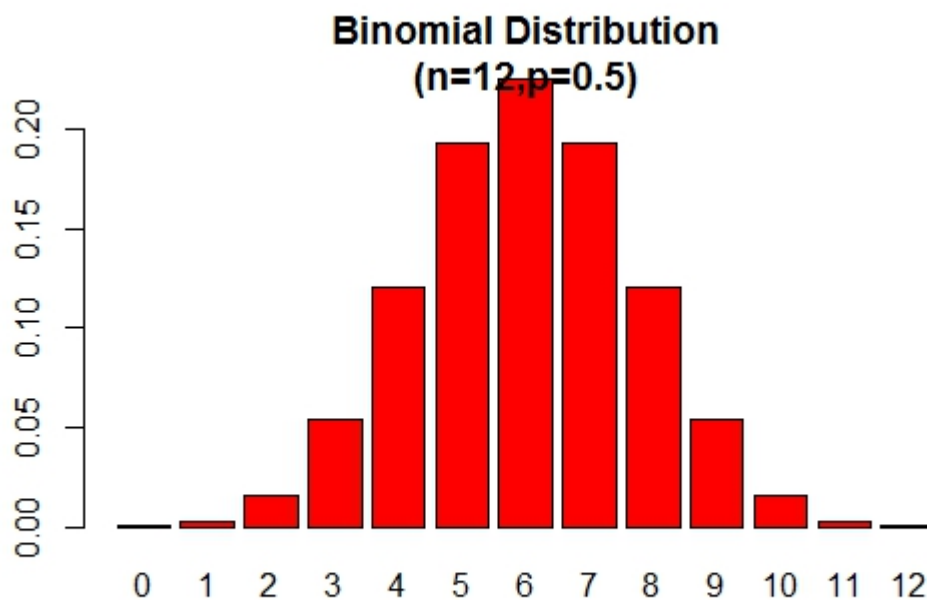
dbinom pbinom qbinomrbinom ◦

dbinom()◦ ◦ X◦ defining parameters n p ◦ n = 5 p = 0.5X0,1,2,3,4,5 ◦ dbinom(x,n,p)x = 0, 1, 2, 3, 4, 5P( X = x )◦

```
#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
>
```

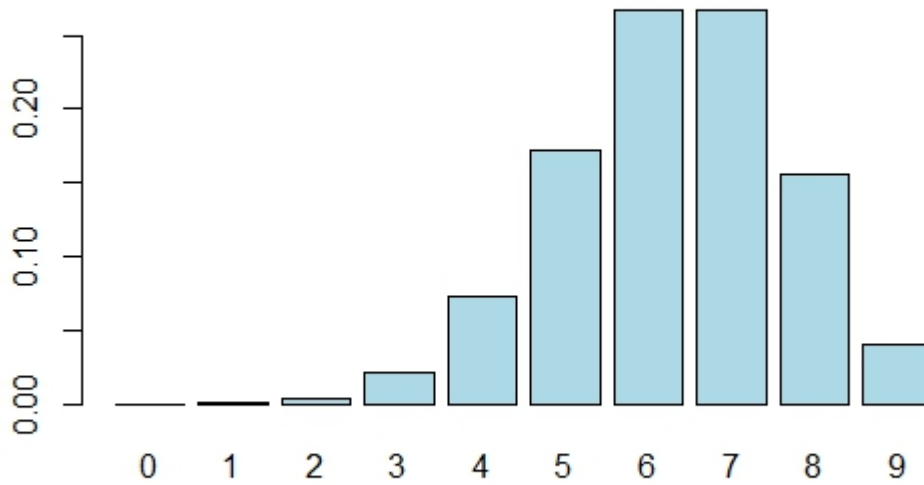
```
> x <- 0:12
> prob <- dbinom(x,12,.5)
> barplot(prob,col = "red",ylim = c(0,.2),names.arg=x,
           main="Binomial Distribution\n(n=12,p=0.5)")
```



p = 0.5◦ p0.5

```
> 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")
```

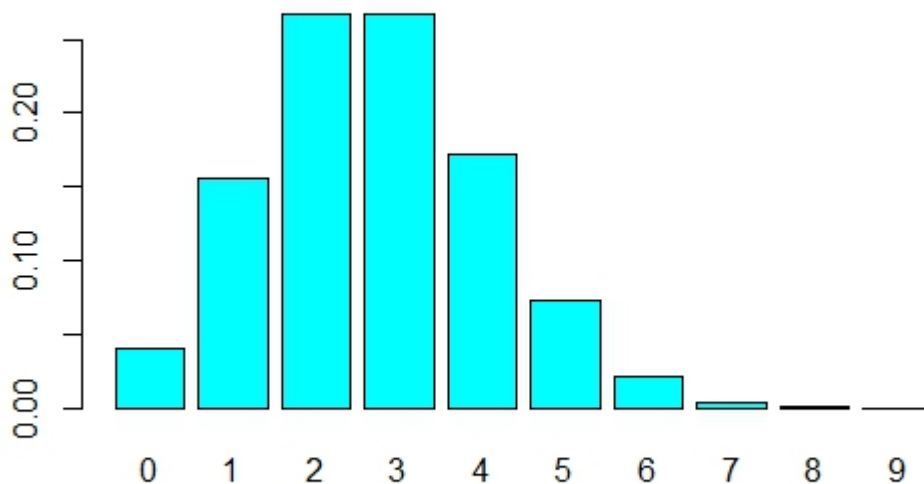
### Binomial Distribution (n=9, p=0.7)



p0.5 ◦

```
> 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")
```

### Binomial Distribution (n=9, p=0.3)



pbinom() ◦ P ( X <= x ) ◦ X ◦

```
# Calculating Probabilities  
# P(X <= 2) in a Bin(n=5,p=0.5) distribution  
> pbinom(2,5,0.5)
```

```
[1] 0.5
```

```
# P(X <= 2) = P(X=0) + P(X=1) + P(X=2)
> sum(dbinom(0:2,5,0.5))
[1] 0.5
```

$P(a \leq X \leq b)$

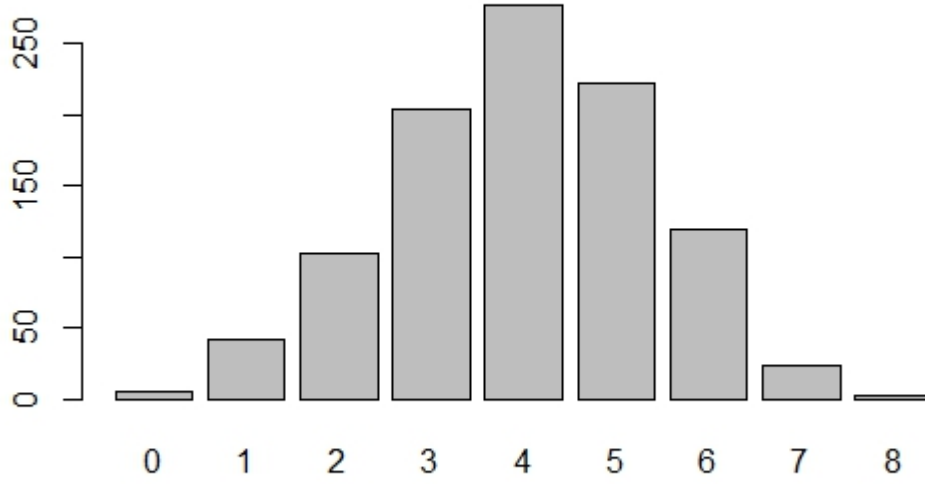
```
# P(3 <= X <= 5) = P(X=3) + P(X=4) + P(X=5) in a Bin(n=9,p=0.6) dist
> sum(dbinom(c(3,4,5),9,0.6))
[1] 0.4923556
>
```

```
> 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()` ◦

```
# 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)")
```

### Simulated Binomial Distribution ( $n=8, p=0.5$ )



<https://riptutorial.com/zh-TW/r/topic/1885/>

# 50:

## Examples

`split` `vector` `data.frame` `factor` / `group` `bucket` ◦ `for` `lapply` / `sapply` ◦

`split`

```
testdata <- c("e", "o", "r", "g", "a", "y", "w", "q", "i", "s", "b", "v", "x", "h", "u")
```

`vowels` `consonants` `vowels` ◦

```
vowels <- c('a','e','i','o','u','y')
letter_type <- ifelse(testdata %in% vowels, "vowels", "consonants")
```

`letter_type` `vector` `testdata` ◦ `split` `vowels` `consonants`

```
split(testdata, letter_type)
#$consonants
#[1] "r" "g" "w" "q" "s" "b" "v" "x" "h"

#$vowels
#[1] "e" "o" "a" "y" "i" "u"
```

/`letter_type` ◦

`split` `data.frames` ◦

`iris`

```
data(iris)
```

`split` `iris` `specie` `Species` `data.frame`

```
> 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
```

`setosa` ◦

`lapply`

```
> (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   |

:(/

```
> library(reshape)
> (topcor <- lapply(lcor, FUN=function(cormat){
  correlations <- melt(cormat,variable_name="correlatio");
  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   |

```
> (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-combinesplit

[split-apply-combine](#) ◦

[mtcars\\$mpg](#) ◦ [mtcars](#)

```
(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
# ...
```

◦ spl\$`4` spl\$`6` spl\$`8` spl\$`4"spl[["4"]]spl[["4"]]◦

lapply2 mpg

```
(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
```

rbind◦ rbind(best2[["4"]], best2[["6"]], best2[["8"]])◦

```
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
```

rbind 1best2 2◦



## split-apply-combine

```
do.call(rbind, lapply(split(mtcars, mtcars$cyl), function(x) tail(x[order(x$mpg),], 2)))
```

`lapply(split(x,f), FUN)?by`

```
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)))
```

<https://riptutorial.com/zh-TW/r/topic/1073/>

# 51:

## Examples

/o @DataTx"" :-).

R"" nlme lme4 o o

```
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
```

- 
- nlme *S-PLUS* Pinheiro Bates 2000 ; Bates 2015 / vignette("lmer", package="lme4") for lme4
- lme4
- nlme **p** lme4 lmerTest afex
- nlme//

[GLMM FAQ](#) GLMM.

<https://riptutorial.com/zh-TW/r/topic/3460/>

# 52:

## Examples

sum

```
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])
```

◦ base R colSums

```
colSums(df1[-1], na.rm = TRUE)
```

sum na.rm = TRUE **NA**

matrix

```
colSums(m1, na.rm = TRUE)
```

lapply/sapply/vapply

```
lapply(df1[-1], sum, na.rm = TRUE)
```

list ◦ vector

```
sapply(df1[-1], sum, na.rm = TRUE)
```

```
vapply(df1[-1], sum, na.rm = TRUE, numeric(1))
```

apply **with** MARGIN = 1

```
apply(m1, 2, FUN = sum, na.rm = TRUE)
```

dplyr data.table

```
library(dplyr)
df1 %>%
  summarise_at(vars(matches("^V\\d+")), sum, na.rm = TRUE)
```

summarise\_at sum ◦ V \\d+ ◦

data.table

```
library(data.table)
setDT(df1)[, lapply(.SD, sum, na.rm = TRUE), .SDcols = 2:ncol(df1)]
```

'data.frame' 'data.table' setDT(df1) .SDcols Data.table .SD sum ◦

---

/

```
df1 %>%  
  group_by(ID) %>%  
  summarise_at(vars(matches("^V\\d+")), sum, na.rm = TRUE)
```

sum summarise\_each summarise\_at

```
df1 %>%  
  group_by(ID) %>%  
  summarise_each(funs(sum(., na.rm = TRUE)))
```

data.table

```
setDT(df1)[, lapply(.SD, sum, na.rm = TRUE), by = ID]
```

<https://riptutorial.com/zh-TW/r/topic/2212/>

# 53:

## Examples

:

```
x <- 1:5
x
## [1] 1 2 3 4 5
```

```
10:4
# [1] 10 9 8 7 6 5 4
```

```
1.25:5
# [1] 1.25 2.25 3.25 4.25
```

```
-4:4
#[1] -4 -3 -2 -1 0 1 2 3 4
```

## SEQ

seq:1°

start 1°

end to

```
seq(5)
# [1] 1 2 3 4 5
```

```
seq(2, 5) # or seq(from=2, to=5)
# [1] 2 3 4 5
```

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
```

length.out seq

```
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
```

along.withlength.out = length(x)

```
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
```

seq\_along seq\_len seq.int ◦ seq\_along seq\_len **1NN** seq\_along seq\_len ◦

```
seq_along(x)
# [1] 1 2 3 4 5 6 7 8
```

seq\_along◦

```
# 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 seq◦

sequence ◦

```
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
```

**R**◦ vector() ◦

```
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.
```

**R**◦

```
integer(2) # is the same as vector('integer',2) and creates an integer vector with two
elements
character(2) # is the same as vector('integer',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
```

◦ c()◦ **C**◦

```
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.
```

**R11**◦ 1.1TF'a'◦ ◦

◦ R◦

```
c(1,1.1,'a',T) # all types (integer, numeric, character and logical) are converted to the
'lowest' type which is character.
```

[]◦

```
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'
```

```
vec_int[2] <- 5 # change the second value from 2 to 5
vec_int # returns [1] 1 5 3
```

:seq()◦

```
vec_int <- 1:10
vec_int # returns [1] 1 2 3 4 5 6 7 8 9 10
```

```
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"
```

◦ c

```
xc <- c('a' = 5, 'b' = 6, 'c' = 7, 'd' = 8)
```

```
> xc
a b c d
5 6 7 8
```

list

```
x1 <- list('a' = 5, 'b' = 6, 'c' = 7, 'd' = 8)
```

```
> x1
$a
[1] 5

$b
[1] 6

$c
[1] 7

$d
[1] 8
```

setNames

```
x <- 5:8
y <- letters[1:4]

xy <- setNames(x, y)
```

```
> xy
a b c d
5 6 7 8
```

c°

names

```
xy <- 5:8
names(xy) <- letters[1:4]
```

```
> xy["c"]
c
7
```

/°

```
mydf <- data.frame(let = c('c','a','b','d'))

> mydf
  let
1  c
2  a
3  b
4  d
```

nummydfmydfxy° matchxy

```
mydf$num <- xy[match(mydf$let, names(xy))]
```

```
> mydf
  let num
1  c   7
2  a   5
3  b   6
4  d   8
```

## rep

rep°

```
# 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
```

## /data.frame◦

```
# 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
```

---

rep

```
# 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
```

---

rep◦

---

seq reprep\_lenrep.int ◦ rep◦

```
# 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
```

---

R◦

- LETTERS **26**
- letters **26**
- month.abb
- month.name
- pi

◦

## 1

```
> 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

```
> 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"
```

<https://riptutorial.com/zh-TW/r/topic/1088/>

# 54:

## Examples

R Map Reduce Filter Find Position Negate ◦

Map

```
words <- list("this", "is", "an", "example")
Map(toupper, words)
```

Reduce◦

```
Reduce(`*`, 1:10)
```

Filter◦

```
Filter(is.character, list(1, "a", 2, "b", 3, "c"))
```

Find**TRUE**◦

```
Find(is.character, list(1, "a", 2, "b", 3, "c"))
```

Position**TRUE**◦

```
Position(is.character, list(1, "a", 2, "b", 3, "c"))
```

Negate**FALSETRUE**◦

```
is.noncharacter <- Negate(is.character)
is.noncharacter("a")
is.noncharacter(mean)
```

<https://riptutorial.com/zh-TW/r/topic/5050/>

# 55:

## Examples

### Rcpp

```
for(len=1; first x; icos(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)  
}
```

for  $\cos(x[i-1]+1)$ 。  $\mathbb{R}^n$ “ $x + 1$ ”。

### RcppC++

```
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;  
}")
```

```
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
```

RcppR0.031001.31。

```
Rcpplenfirst first x; icos(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)  
}
```

## R

```
library(compiler)
repeatedCosPlusOneCompiled <- cmpfun(repeatedCosPlusOne)
```

```
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
```

### 11.200.34。

repeatedCosPlusOne Reduce

```
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<sup>”</sup> repeatedCosPlusOne ◦ 2

```
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
```

<https://riptutorial.com/zh-TW/r/topic/1203/>

# 56: HashMap

## Examples

◦

R new.environment hash TRUE

```
H <- new.env(hash = TRUE)
H <- new.env()
```

size size 29.R environments size ◦ size

```
object.size(new.env())
# 56 bytes

object.size(new.env(size = 10e4))
# 56 bytes
```

[<-<-environment "" [<-

```
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
```

R object[[key]] <- value object\$key <- value key2 ◦

environment ◦

```
H[["key3"]] <- "original value"

H[["key3"]] <- "new value"

H[["key3"]]
#[1] "new value"
```

[[\$ [ ]

```
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

```

---

environment

```

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"

```

rm

```

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 *environment*

```

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 **R/**

```

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
```

vapplylist2env

```
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),
  envir = NULL,
  hash = TRUE
)

all.equal(sort(names(E2)), letters)
#[1] TRUE
```

for°

R° . . . R°

```
# 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')
  )
})

```

## listenv

[package:listenv](#) ◦ ◦ [package:future](#)[package:future](#) ◦ ◦ [package:hash](#) for write timing ◦

```

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')
  )
})

```

[HashMap](#) <https://riptutorial.com/zh-TW/r/topic/5179/hashmap>

---

# 57: R

“” ◦ ◦

◦ ◦ ◦

---

- PengRD2011. ◦ Science33460601226-1227. <http://doi.org/10.1126/science.1213847>
- PengRoger D.R. Leanpub2015.[https //leanpub.com/reportwriting](https://leanpub.com/reportwriting)◦

## Examples

`dput ()` `dput ()` `dget ()`

`dput ()` ◦ R◦

◦ `getwd ()?setwd` ◦

```
dput(mtcars, file = 'df.txt')
```

`dget ()` `RGlobalEnvironment`◦

```
df <- dget('df.txt')
```

R◦ ◦

R◦ ◦ R◦ `checkpoint`◦

2014-09-17CRAN - Microsoft R Archived Network◦ R

1. R◦
2. `checkpoint::checkpoint('YYYY-MM-DD')`◦

`checkpointR_home`.`checkpoint "~/"` ◦ ◦ `checkpoint.Rlibrary()require()` `CRAN`◦

**PRO**◦

**CONTRA**◦ ◦

R <https://riptutorial.com/zh-TW/r/topic/4087/r>

# 58:

## R

◦

```
c(1,2,3) + c(1,2,3,4,5,6)
[1] 2 4 6 5 7 9
```

```
c(1,2,3,1,2,3) + c(1,2,3,4,5,6)
```

```
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
```

```
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

◦

```
my_vec <- c(1,2,3,4,5,6,7,8,9,10)
my_vec[c(TRUE, FALSE)]

[1] 1 3 5 7 9
```

◦

```
my_vec <- c("foo", "bar", "soap", "mix")
my_vec == "bar"

[1] FALSE  TRUE FALSE FALSE
```

“”◦

<https://riptutorial.com/zh-TW/r/topic/5649/>

# 59:

1. `factor` = `character` = `levels` = `exclude` = `NA` = `ordered` = `is.ordered` = `x` = `max` = `NA`
2. `?factor`

`factor`

1. `integer`
2. `levels`
3. `factor`

1,000

```
set.seed(1)
Color <- sample(x = c("Red", "Blue", "Green", "Yellow"),
               size = 1000,
               replace = TRUE)
Color <- factor(Color)
```

`Color`

```
##* 1. It is stored internally as an `integer` vector
typeof(Color)
```

```
[1] "integer"
```

```
##* 2. It maintains a `levels` attribute that 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"
```

◦ ◦ [stringsAsFactors](#) ◦ `Color` ◦ `Color` 1.7 ◦

```
##* 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
```

o

```
head(Color)
```

```
[1] Blue   Blue   Green  Yellow Red    Yellow  
Levels: Blue Green Red Yellow
```

```
head(as.numeric(Color))
```

```
[1] 1 1 2 4 3 4
```

## R

```
head(levels(Color)[as.numeric(Color)])
```

```
[1] "Blue"  "Blue"  "Green" "Yellow" "Red"   "Yellow"
```

```
head(Color)
```

```
[1] Blue   Blue   Green  Yellow Red    Yellow  
Levels: Blue Green Red Yellow
```

---

2007R [stringsAsFactors](#) o 1.7R2007o

Ro Ro

1./

2.

o "''''''''''". o

o o o

## Examples

Ro . . Ro

```
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"
```

```
levels(factor(charvar, levels = c("n","c")))
> levels(factor(charvar, levels = c("n","c")))
[1] "n" "c"
```

o

```
> f <- factor(charvar, levels=c("n", "c"), labels=c("Newt", "Capybara"))
> f
[1] Newt      Newt      Newt      Capybara  Capybara  Capybara
Levels: Newt Capybara
```

```
> 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
```

```
droplevels()droplevels()
```

```
> 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
```

o o

```
set.seed(1)
colorful <- sample(c("red", "Red", "RED", "blue", "Blue", "BLUE", "green", "gren"),
                  size = 20,
                  replace = TRUE)
colorful <- factor(colorful)
```

Ro

```
table(colorful)
```

```
colorful
blue  Blue  BLUE green  gren   red   Red   RED
     3     1     4     2     4     1     3     2
```

◦ ◦ ◦ ◦ ◦

### factor factor\_approach

```
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
```

◦ “” ◦ ◦

### ifelse ifelse\_approach

```
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
```

ifelse◦ ifelse◦

### list\_approach

◦ levels◦

```
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
```

◦



```
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
```

ifelse° ° ° °

R° xas.character() factor()as.factor()° ° x°

°

```
# standard
factor(c(1,1,2,2,3,3))
[1] 1 1 2 2 3 3
Levels: 1 2 3
```

° factor()°

```
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
```

° ° °

```
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
```

==!° °

```
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
```

RHSR°

< <= >>=° ° factorordered = TRUEordered°

```
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
```

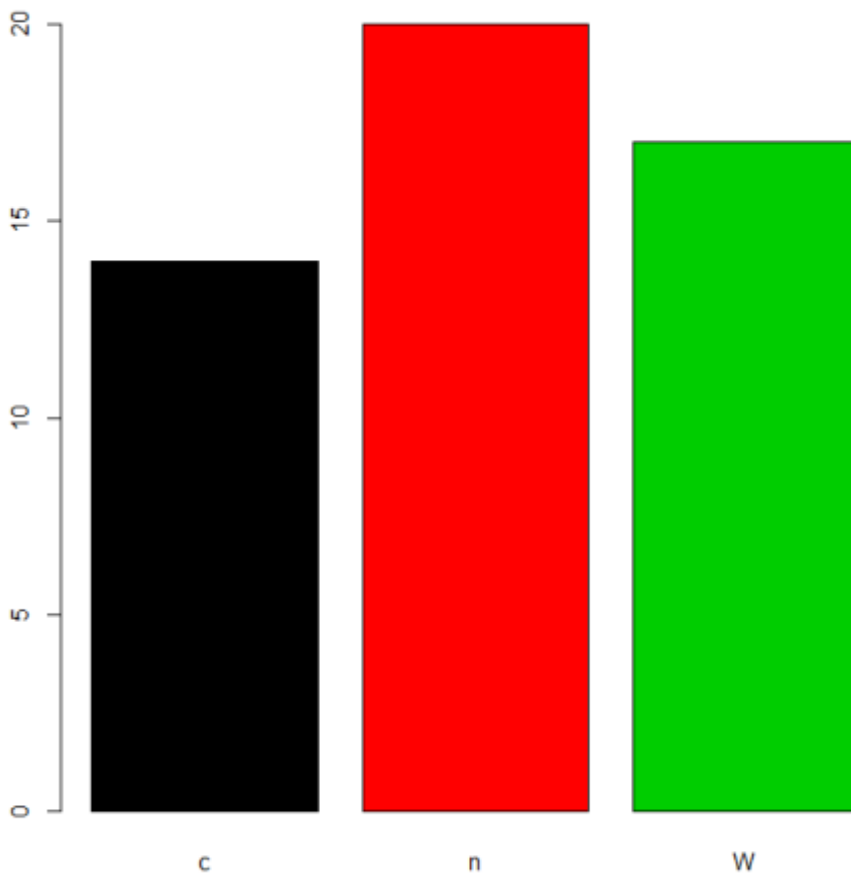
## Factor

levels as character as character

```
charvar <- rep(c("W", "n", "c"), times=c(17,20,14))
f <- factor(charvar)
levels(f)
# [1] "c" "n" "W"
```

levels / plot

```
plot(f, col=1:length(levels(f)))
```



levels levels labels "order"

## 1.

levels

```
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"
```

labelslevels "level" 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

levelrelevel ◦ base◦

```
g<-relevel(f, "n") # moves n to be the first level
levels(g)
# [1] "n" "c" "W"
```

fg

```
all.equal(f, g)
# [1] "Attributes: < Component "levels": 2 string mismatches >"
all.equal(f, g, check.attributes = F)
# [1] TRUE
```

## 3.

levelsreorder ◦ levels

```
table(g)
# g
#  n  c  W
# 20 14 17
```

reorderhelp(reorder) x ; X x;FUN Xxlevels◦ ◦

```
g.ord <- reorder(g, rep(1, length(g)), FUN=sum) #increasing
levels(g.ord)
# [1] "c" "W" "n"
```

-1

```
g.ord.d <- reorder(g,rep(-1,length(g)), FUN=sum)
levels(g.ord.d)
# [1] "n" "W" "c"
```

◦

```
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
```

levels ◦ iris help("iris") Sepal.WidthSpecies ◦

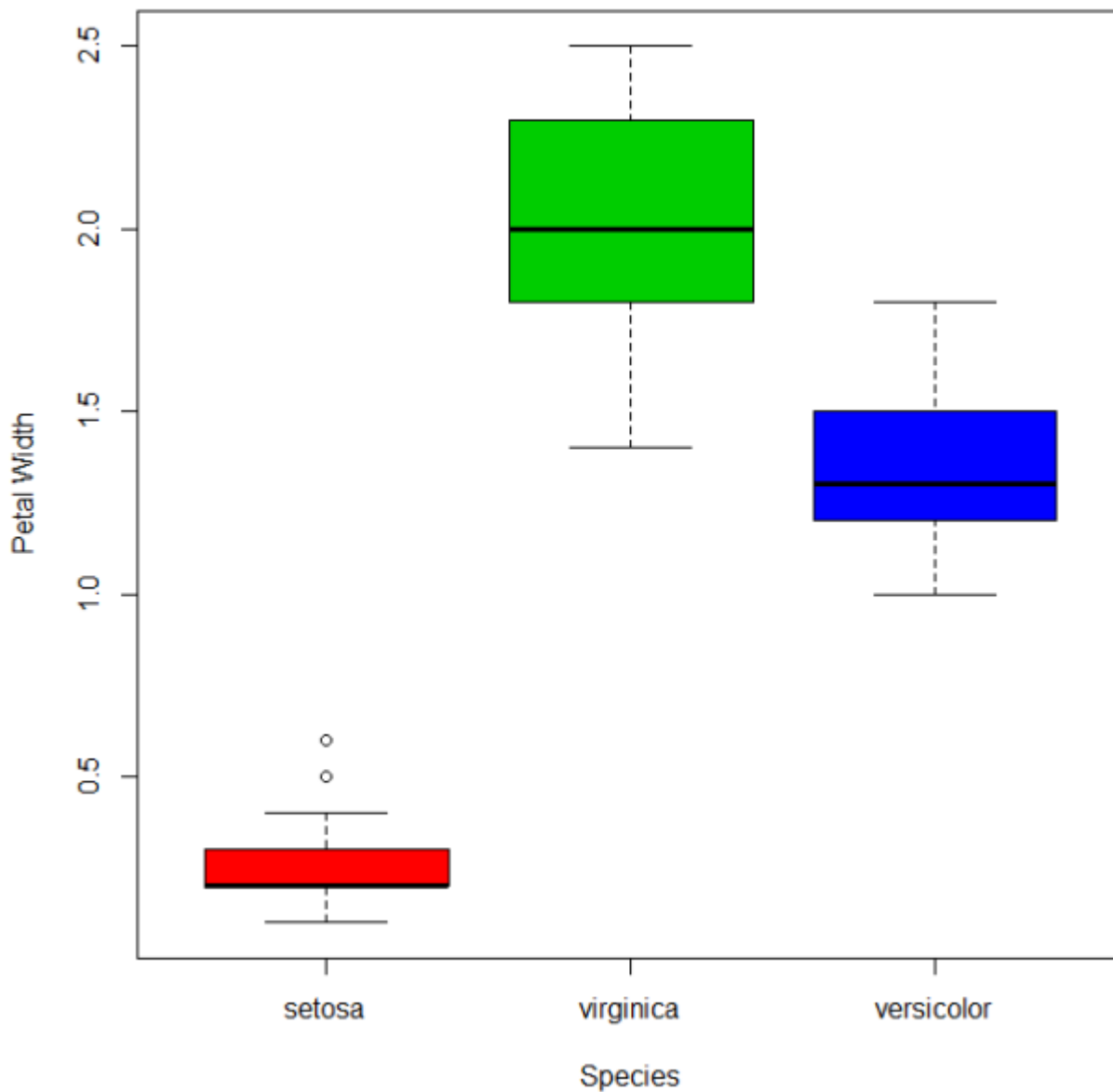
```
miris <- iris #help("iris") # copy the data
with(miris, tapply(Sepal.Width,Species,mean))
#   setosa versicolor virginica
#   3.428     2.770     2.974

miris$Species.o<-with(miris,reorder(Species,-Sepal.Width))
levels(miris$Species.o)
# [1] "setosa"     "virginica"  "versicolor"
```

boxplot with(miris, boxplot(Petal.Width~Species) **especies setosa versicolorvirginica** ◦  
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



levelslevels ◦ 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
```

-

orderedfactors ◦ levelslabels ◦

```
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
```

R◦ ◦ 5

◦ 1 =2 =3 =4 =5

**4431,2**

◦ R

20

```
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

14◦

```
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

1 2 3 4

```
levels(fii) <- c("empty", "low", "normal", "full")  
fii
```

[1]

[11]

<https://riptutorial.com/zh-TW/r/topic/1104/>

---

60:

## Examples

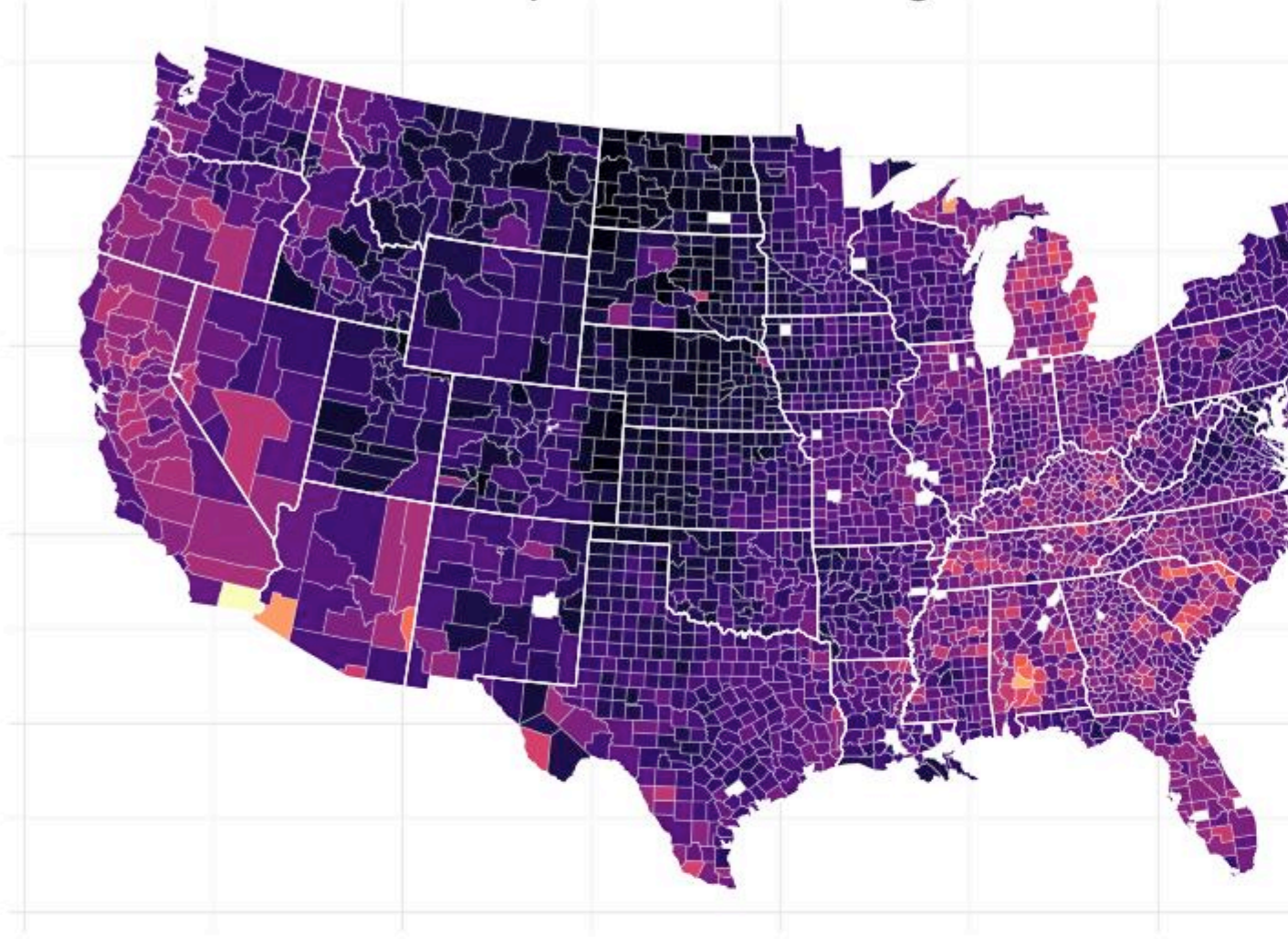
viridis -

Viridis [chromis viridis](#) Python [matplotlib](#) · R

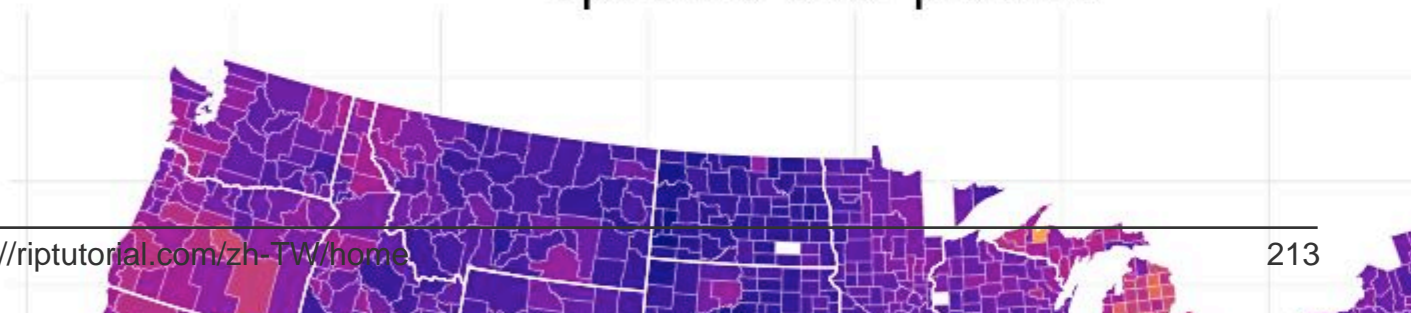
4 magma plasma infernoviridis · optionA B CD4



option A aka 'magma'



option C aka 'plasma'



# 61: R

## Examples

R Base ◦ ◦

◦

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

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

{ } ◦ ◦

◦ vec **6** ◦

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

length() **Base** ◦

```
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 multiplier 4 ◦

one two subtract.length ◦

◦ ◦ \*apply ◦

data.frame data.frame

```
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
```

1.

```
x <- sample(1:6, 12, replace=TRUE)
mat <- matrix(x, nrow=3)

apply(mat, 1, function(x) { seq(min(x), max(x)) })
```

```
(function() { 1 })()
[1] 1
```

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

## RStudio

◦  
“fun”RStudio IDE TAB.

The screenshot shows the RStudio IDE interface. On the left, a code editor displays the following code:

```
1
2 fun
3
4
5
6
7
8
```

A dropdown menu is open over the code, listing the following options:

- fun {snippet}
- function {base}
- functionBody {methods}
- functionBody<- {methods}

To the right of the dropdown, a yellow tooltip contains a template for a function definition:

```

${1:name} <- fun
  ${3:code}
}

```

◦

```
name <- function(variables) {
}

```

```
name <- function(df, x, y) {
  require(tidyverse)
  out <-
  return(out)
}
```

```
}
```

Global Options -> CodeEdit Snippets ◦

◦ `[]` ◦ **R**

```
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"))
```

**R** ◦ `dset[[vars[i]]]` ◦ `vars` ◦ `dset` ◦ `iris[["Sepal.Length"]]` ◦ `iris$Sepal.Length` ◦

**R** <https://riptutorial.com/zh-TW/r/topic/7937/r>

## 62:

### Examples

#### .zip

ziputilsunzipR

```
unzip(zipfile = "bar.zip", exdir = "./foo")
```

"bar.zip" "bar.zip" "foo" Tilde zipfile

#### .zip

ziputilsunzipR

```
unzip(zipfile = "bar.zip", list = TRUE)
```

"bar.zip" Tilde zipfile

#### .tar

tarutilsuntarR

```
untar(zipfile = "bar.tar", list = TRUE)
```

"bar.tar" Tilde tarfile

#### .tar

tarutilsuntarR

```
untar(tarfile = "bar.tar", exdir = "./foo")
```

"bar.tar" "bar.tar" "foo" Tilde tarfile

#### .zip

forzip

```
for (i in dir(pattern=".zip$"))  
  unzip(i)
```

dirpattern iunzipzip

<https://riptutorial.com/zh-TW/r/topic/4323/>

## 63: <>

NAMESPACE.

### Examples

◦ `“” zzz.Rutils.Rutils.R◦`

```
#' Pipe operator
#'  
#' @name %>%  
#' @rdname pipe  
#' @keywords internal  
#' @export  
#' @importFrom magrittr %>%  
#' @usage lhs \%>\% rhs  
NULL
```

<> <https://riptutorial.com/zh-TW/r/topic/10547/--lt--gt---->

# 64:

R ◦ ◦

- data.table
- tidyr
- splitstackshape

## Examples

Rreshape ◦ ?reshape◦

```
# 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.frame234◦ ◦

data.frame

```
# 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       |

NA◦



“v.names”。 “drop”。 data.frame/idv.names。

```
reshape(df, idvar="identifier", timevar="period", direction="wide",  
        drop="location")
```

df.wide

```
reshape(df.wide, direction="long")
```

```
# 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))])
```

。

reshape “” reshape。

```
reshape(df.wide, idvar="identifier",  
        varying=list(c(3,5,7), c(4,6,8)), direction="long")
```

“v.names”。

“sep” reshape。

。 。 。

|  | [cm] | [] |
|--|------|----|
|  | 178  | 20 |
|  | 174  | 45 |
|  | 182  | 31 |

。

|  | [cm] |     |
|--|------|-----|
|  | [cm] | 178 |
|  | [cm] | 174 |
|  | [cm] | 182 |
|  | []   | 20  |
|  | []   | 45  |
|  | []   | 31  |

Base R ◦ mtcars ◦ ◦ ◦

```
mtcars # shows the dataset
data <- data.frame(observation=row.names(mtcars),mtcars)
```

---

## R.

R stack() unstack() ◦

```
long <- stack(data)
long # this shows the long format
wide <- unstack(long)
wide # this shows the wide format
```

◦ ◦

---

## tidyr

gather() widelong spread() longwide ◦

```
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

data.table reshape2melt() dcast() ◦

```
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)
```

<https://riptutorial.com/zh-TW/r/topic/2904/>

# 65:

I/O

## Examples

### map

mapsmap() R

```
require(maps)  
map()
```



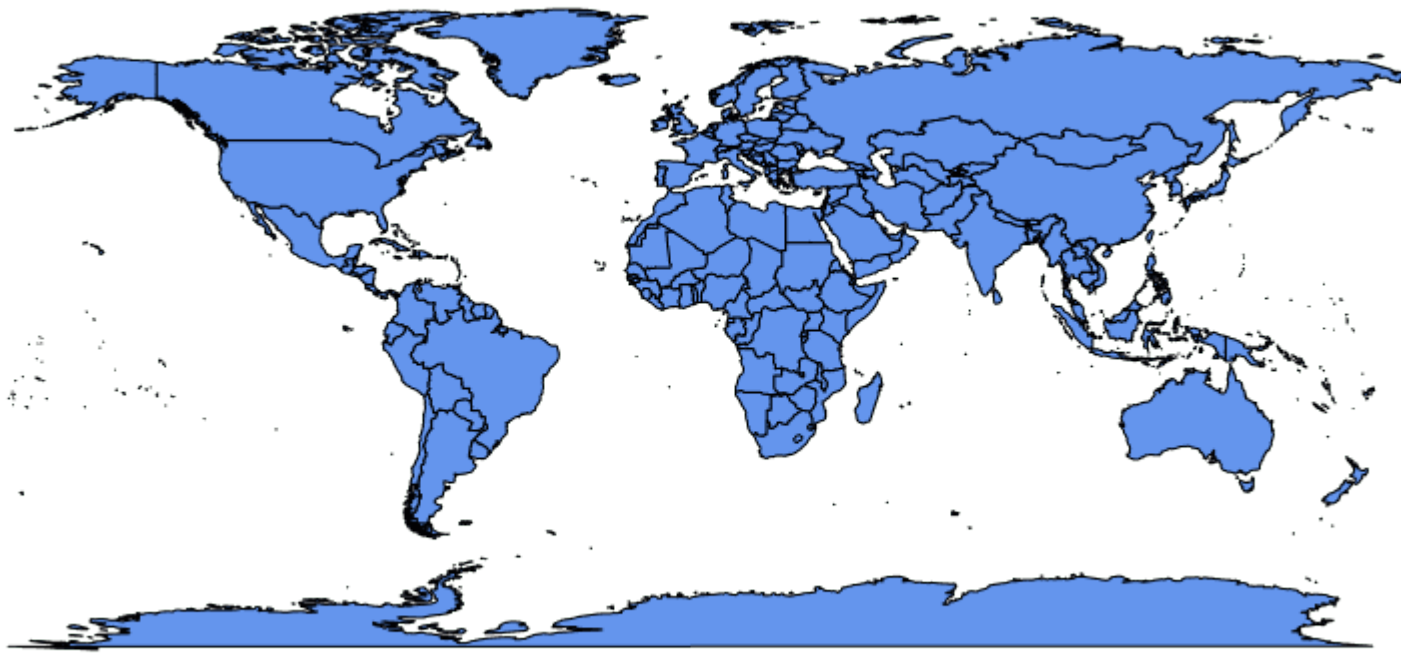
col

```
require(maps)  
map(col = "cornflowerblue")
```



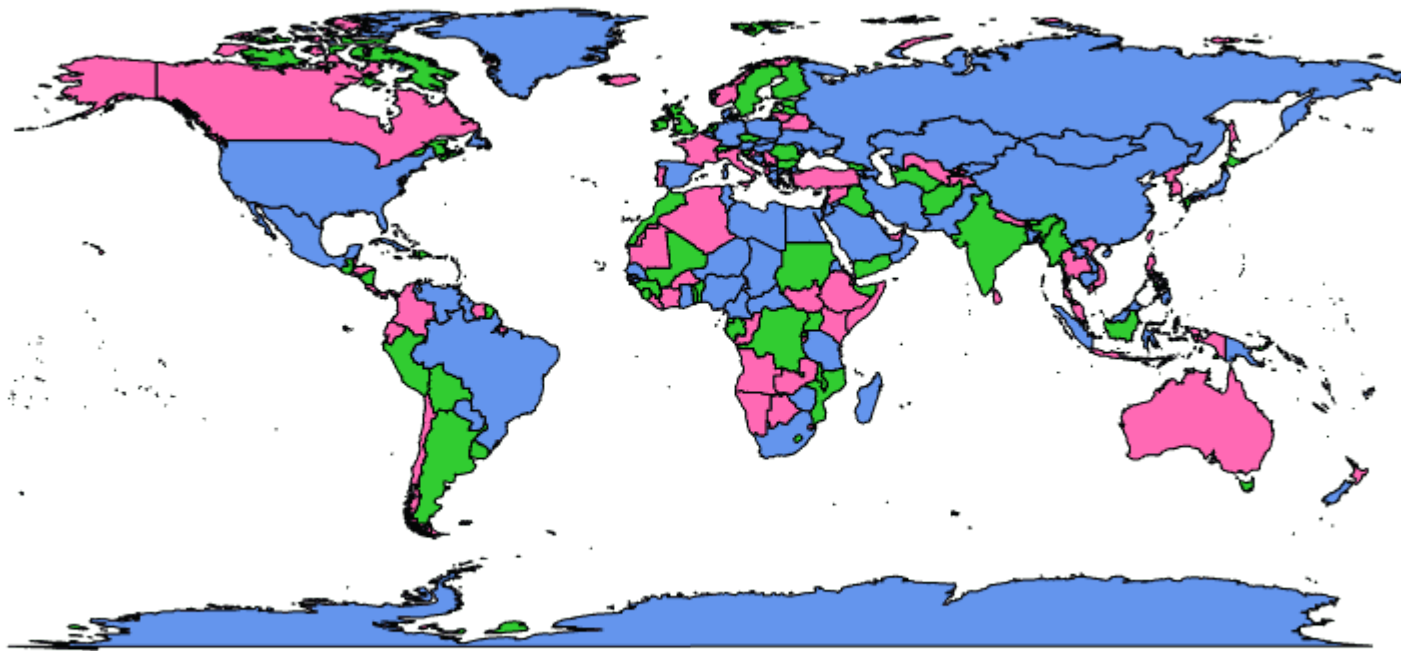
```
colfill = TRUE
```

```
require(maps)  
map(fill = TRUE, col = c("cornflowerblue"))
```



```
fill = TRUEcol
```

```
require(maps)  
map(fill = TRUE, col = c("cornflowerblue", "limegreen", "hotpink"))
```



col°

。 “”。

map() database "county" "state" unempcounty.fips

```
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
```

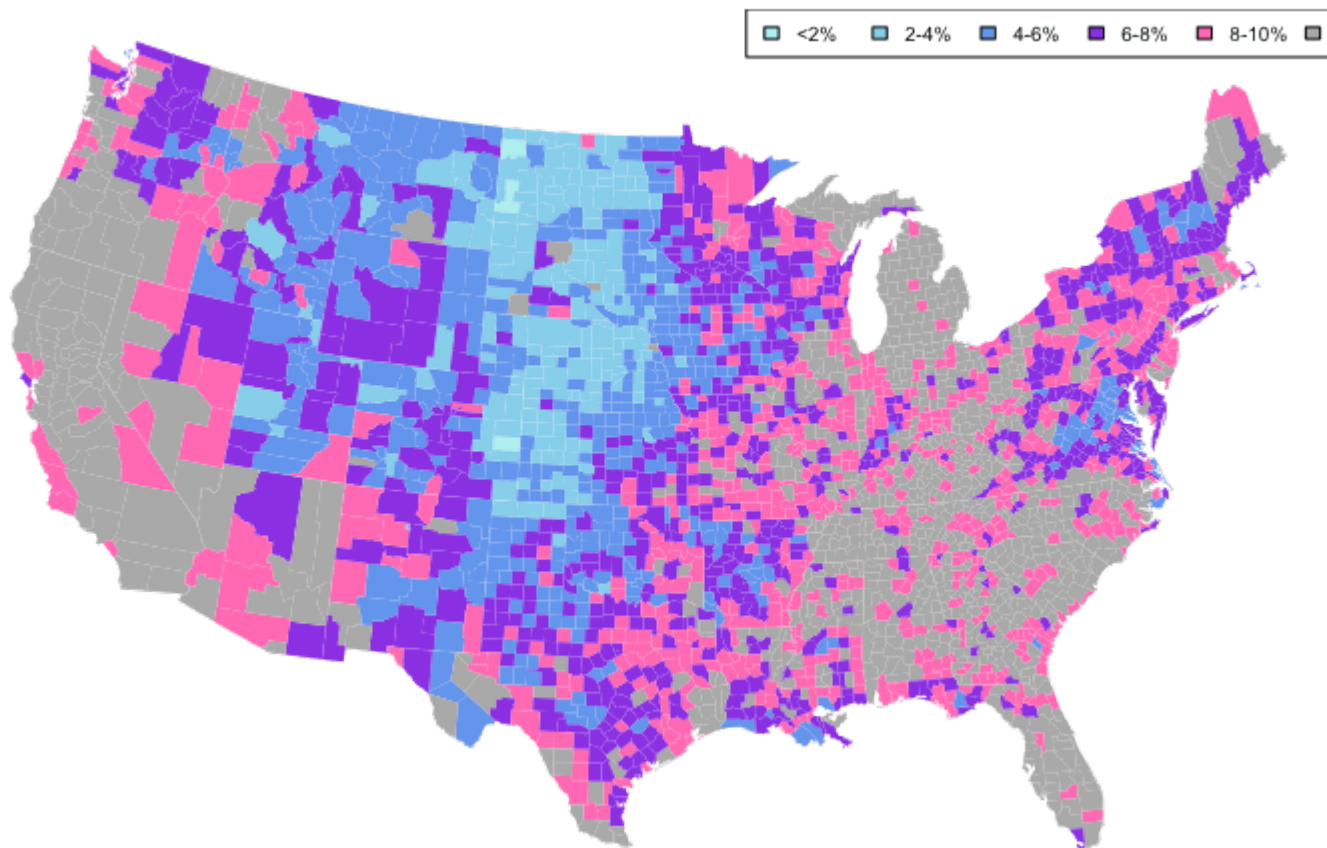
```

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 Viz50

50.

50. Google API `googleVis` `ggmapR` `googleMapsR` `googleMaps` .

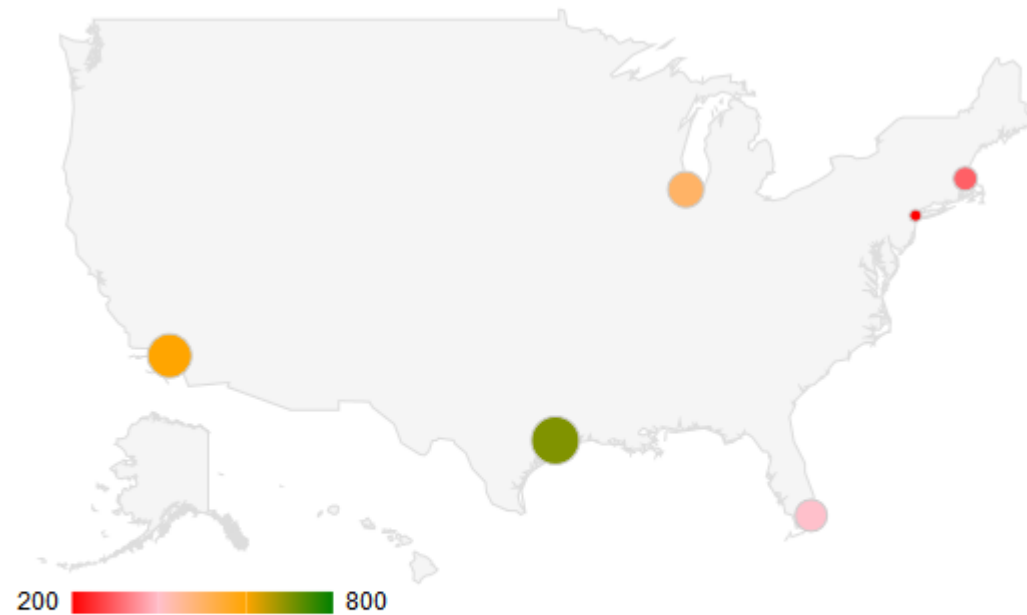
```

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)
```



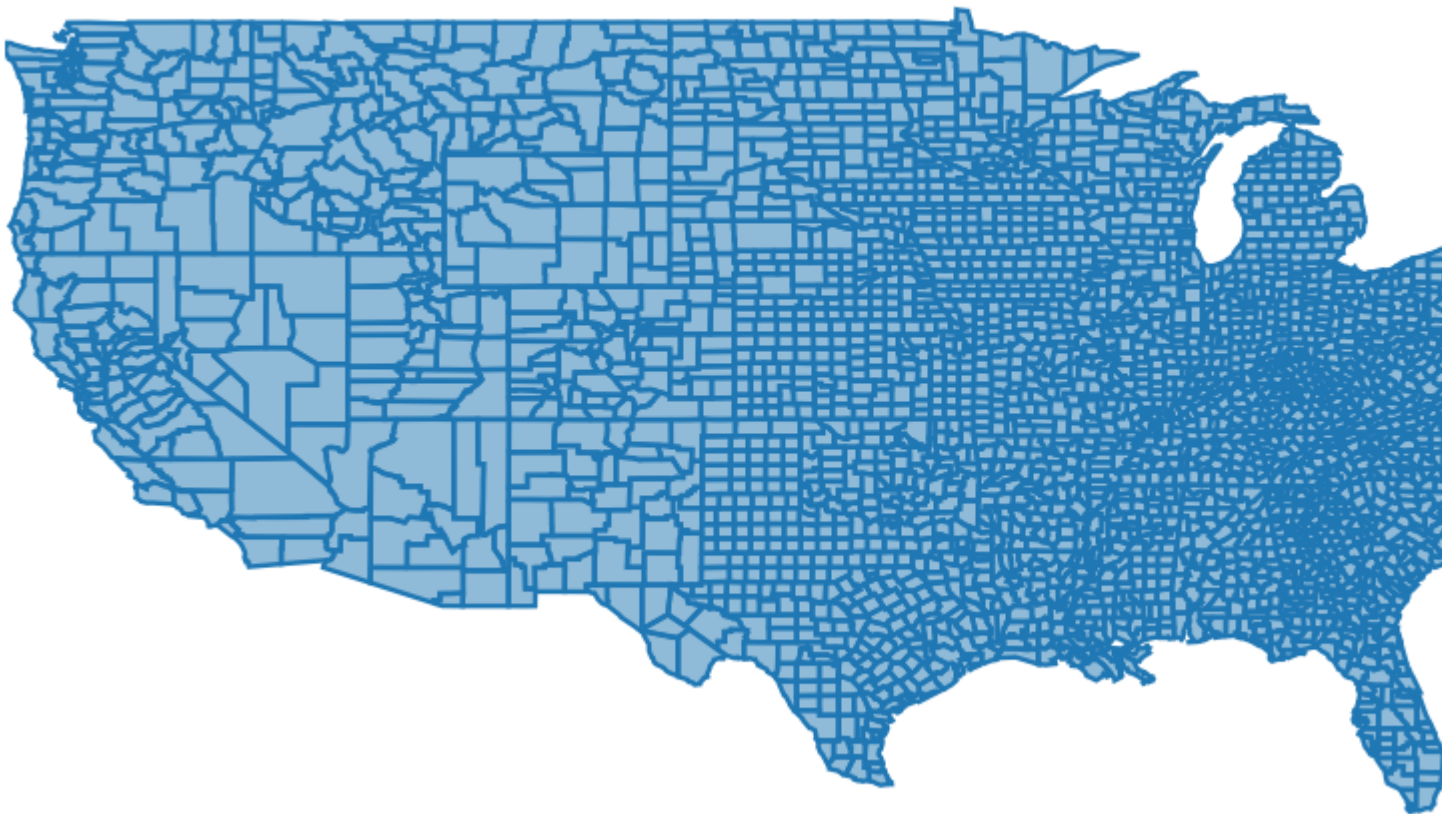
Data: CityPopularity • Chart ID: GeoChartID28504adb439a • googleVis-0.5.2  
 R version 3.1.0 (2014-04-10) • [Google Terms of Use](#) • [Documentation and Data Policy](#)

maps map() gvisGeoChart() ◦ colorvarlocationvar ◦ options height markers colorAxiscolors ◦

plotly ◦ plotly ◦ plot\_ly() ggplotly() **plotly** plot\_geo() plot\_mapbox() ◦

```
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)
  )
```





plot\_geo()plot\_mapbox()plot\_ly() ◦ ◦

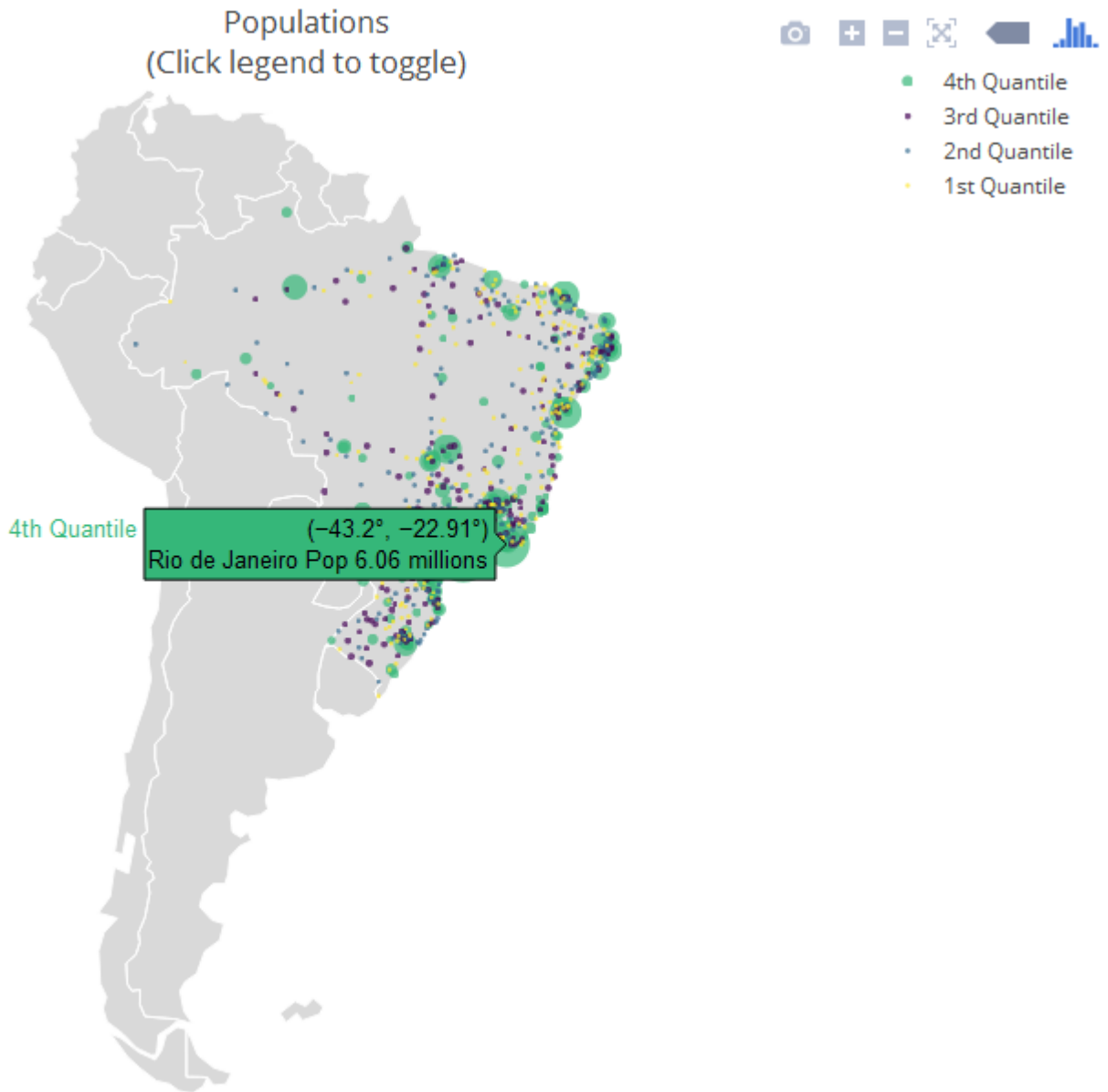
“”layout.geo◦ mapsworld.cities“”◦

poph; q◦ ge◦ ◦

```
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<br>(Click legend to toggle)')
```



## LeafletHTML

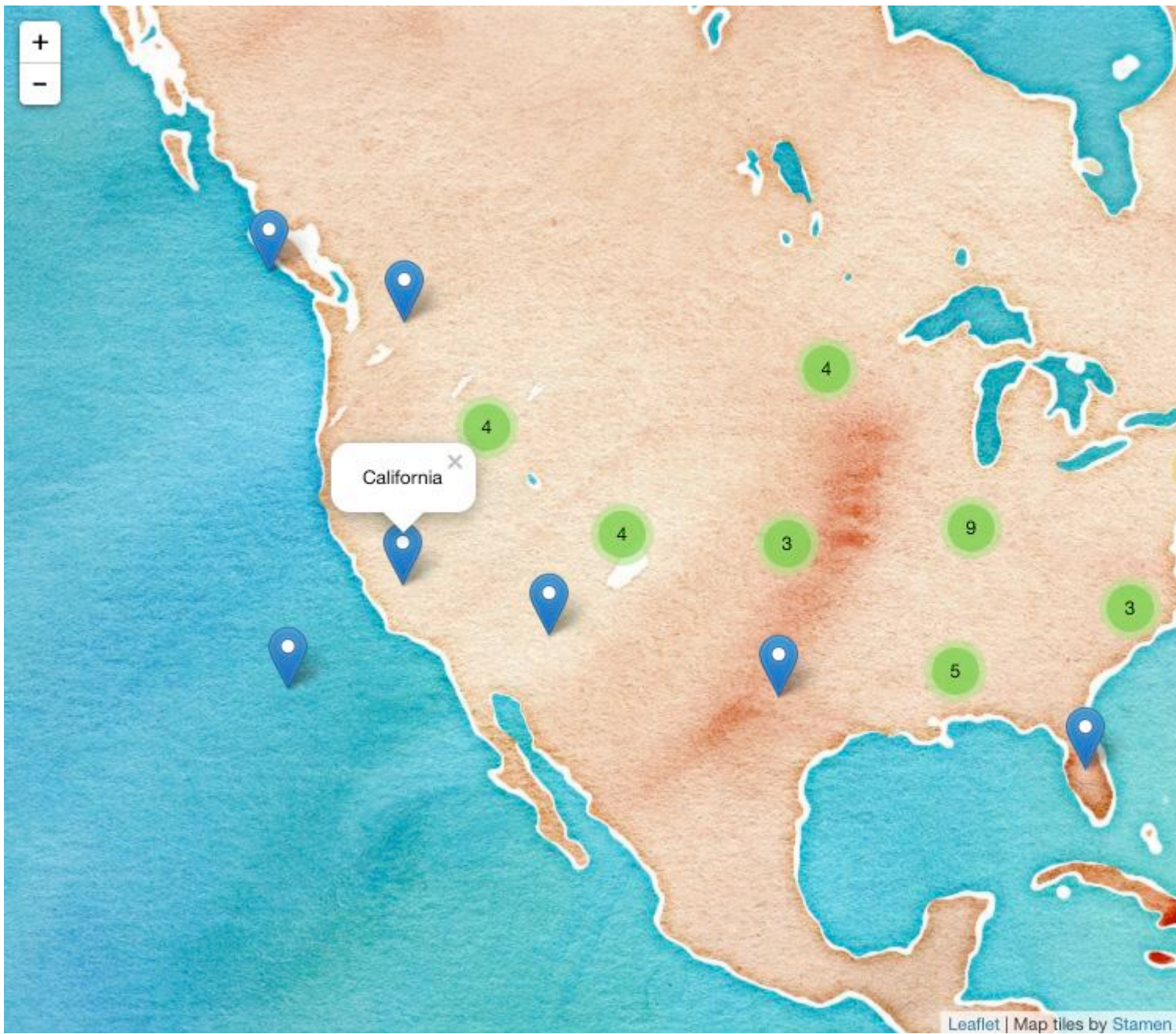
[LeafletJavaScriptWeb](#) ◦ [RStudioLeafletR](#) [leaflethtmlwidgets](#) ◦ [RMarkdownShiny](#) ◦

`leaflet()` ◦ ◦ `leaflet()` `data.frame` `function-style` ~ quotation ◦

`state.name` `state.center`

```
library(leaflet)

data.frame(state.name, state.center) %>%
  leaflet() %>%
  addProviderTiles('Stamen.Watercolor') %>%
  addMarkers(lng = ~x, lat = ~y,
            popup = ~state.name,
            clusterOptions = markerClusterOptions())
```



;

## ShinyLeaflet

### LeafletShiny

**ui** leafletOutput () 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)

```

renderLeaflet **reactive**.

leafletProxy().

leafletleafletProxy

```

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()

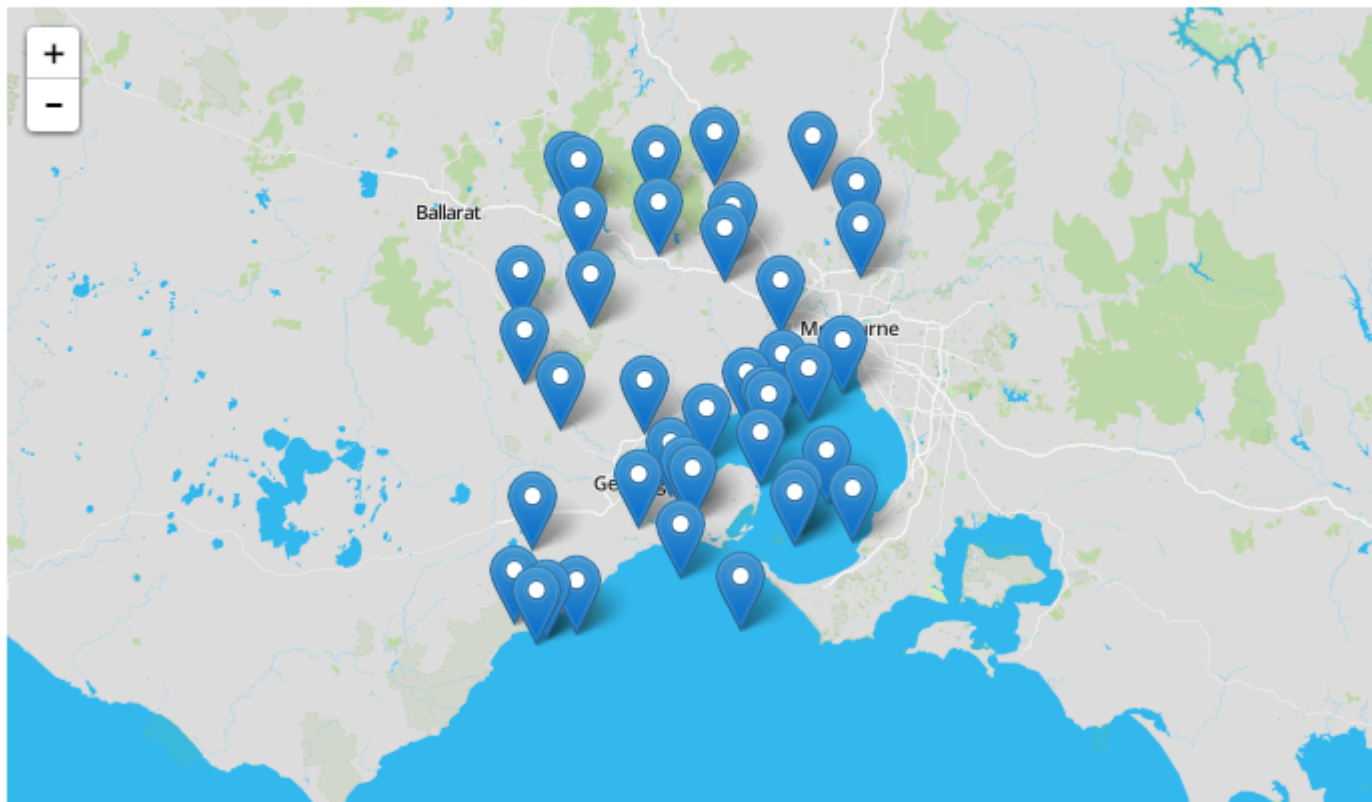
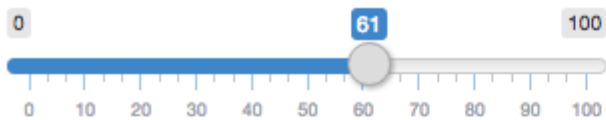
  })

}

```

```
shinyApp(ui, server)
```

values



<https://riptutorial.com/zh-TW/r/topic/1372/>

# 66: I / Oshapefile

## Examples

### Shapefile

`rgdal::readOGR(dsn="path/to/the/folder/containing/the/shapefile", layer="mapmap.shp")`

```
library(rgdal)
readOGR(dsn = "path\\to\\the\\folder\\containing\\the\\shapefile", layer = "map")
```

`writeOGR(Rmap, dsn="path/to/the/folder/containing/the/shapefile", layer="map", driver="ESRI Shapefile")`

```
writeOGR(Rmap, dsn = "path\\to\\the\\folder\\containing\\the\\shapefile", layer = "map",
         driver = "ESRI Shapefile" )
```

`read_shapefile("path/to/the/folder/containing/the/shapefile")`

I / Oshapefile <https://riptutorial.com/zh-TW/r/topic/5538/i---o-shapefile->

# 67:

## Examples

◦ C ◦

```
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

```
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
}
```

testStat ◦ teststattrials ◦

N = 10^5◦ N◦

trials ◦

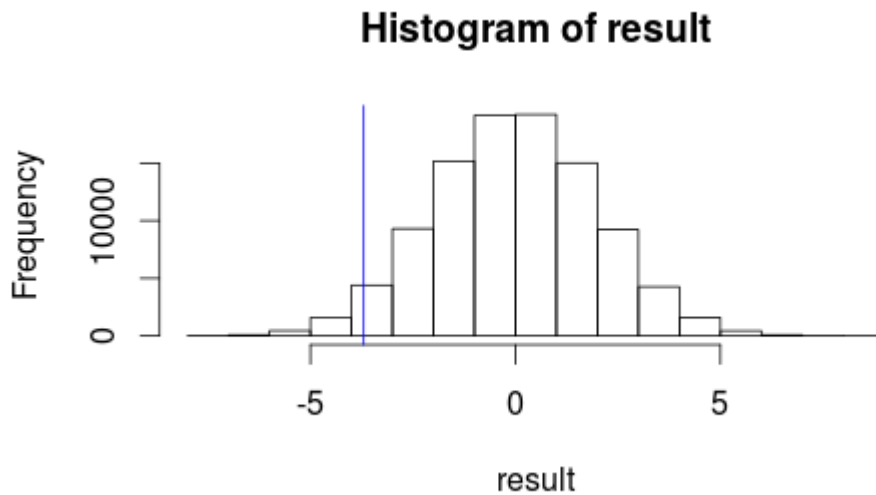
```
subtractMeans = function(a, b){ return (mean(a) - mean(b))}
```

```
result = permutationTest(vec1, vec2, subtractMeans)
```

```
observedMeanDifference = subtractMeans(vec1, vec2)
```

◦

```
hist(result)
abline(v=observedMeanDifference, col = "blue")
```



.....

p◦

```
pValue = 2*mean(result >= (observedMeanDifference))
```

```
result >= (observedMeanDifference)
```

```
FALSE TRUE FALSE FALSE TRUE FALSE ...
```

resultobservedMeanTRUE ◦

mean1TRUE 0FALSE 1◦

2<sup>4</sup>◦

p0.06093939 ◦ C◦

<https://riptutorial.com/zh-TW/r/topic/3216/>



## 68:

|      |   |
|------|---|
| x    | X° data\$variablexdata[,x]                        |
| y    | y° data\$variableydata[,y]                        |
| main |   |
| sub  |   |
| xlab | X   |
| ylab | y   |
| pch  |   |
| col  |   |
| type | ° "p" "l" "b" "c" "b" "o" "°" "h" "°" "s" "S" "n" |

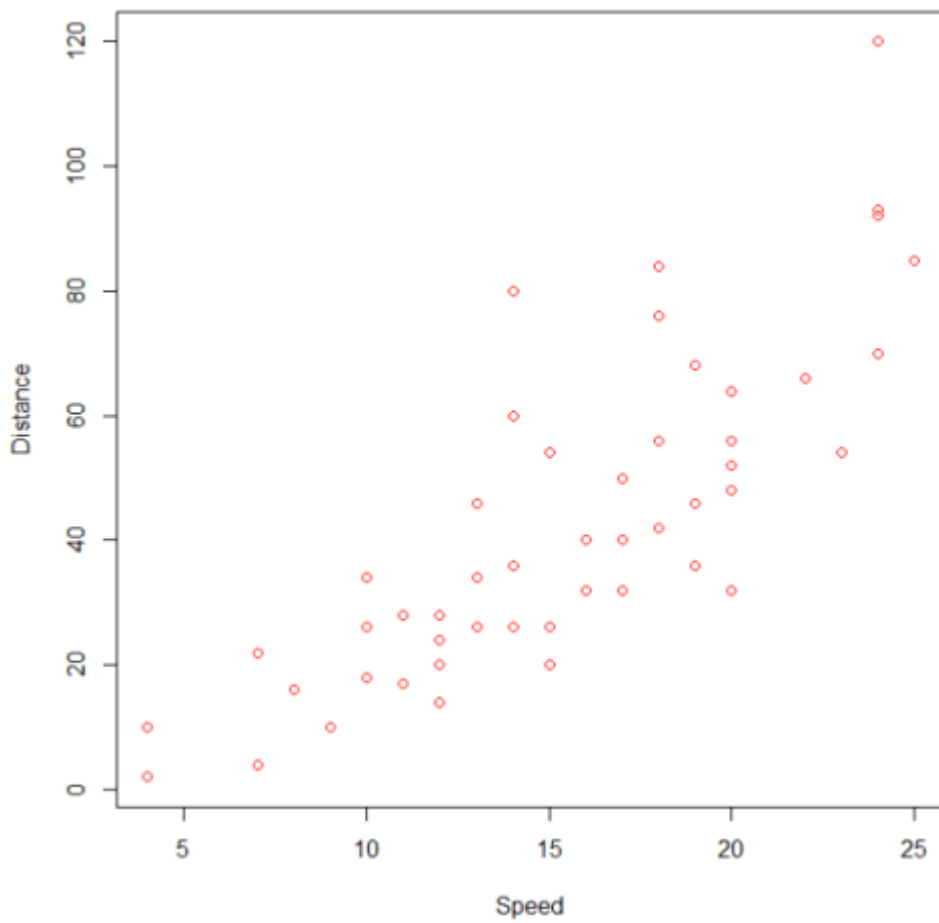
“”par° par ° °

## Examples

plot()° cars2020° helpcars°

```
plot(x = cars$speed, y = cars$dist, pch = 1, col = 1,
     main = "Distance vs Speed of Cars",
     xlab = "Speed", ylab = "Distance")
```

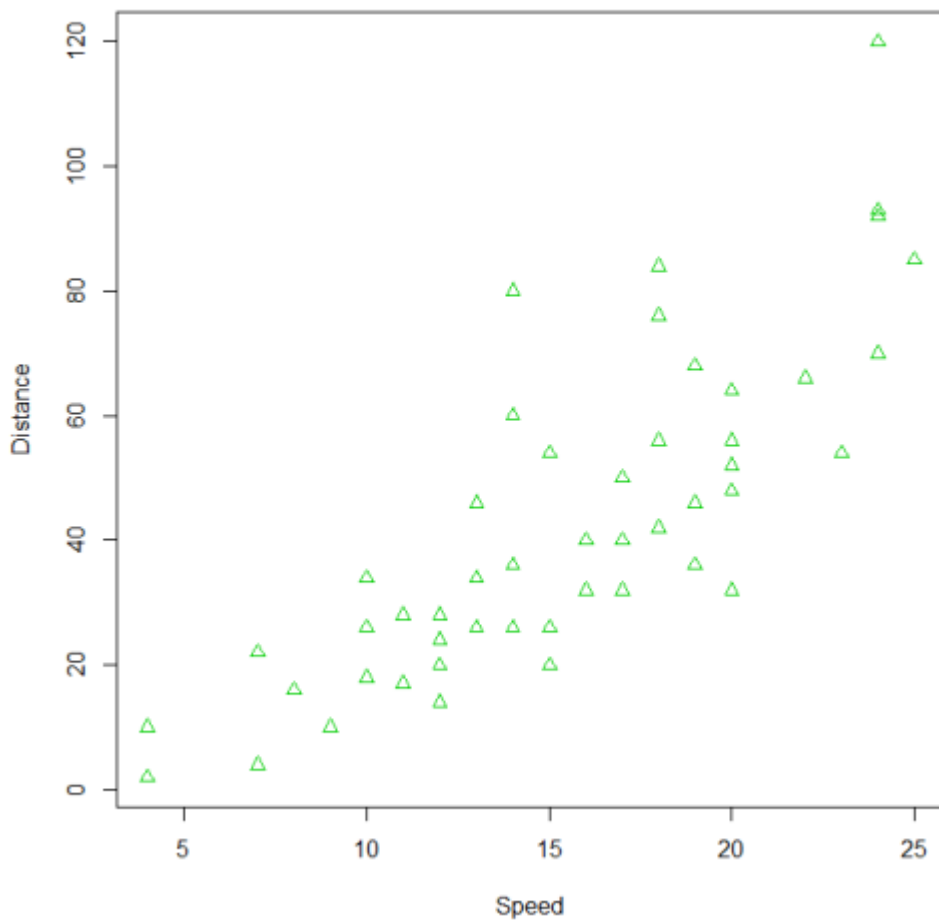
Distance to stop vs Speed of Cars



◦ ◦

```
with(cars, plot(dist~speed, pch = 2, col = 3,  
  main = "Distance to stop vs Speed of Cars",  
  xlab = "Speed", ylab = "Distance"))
```

Distance to stop vs Speed of Cars

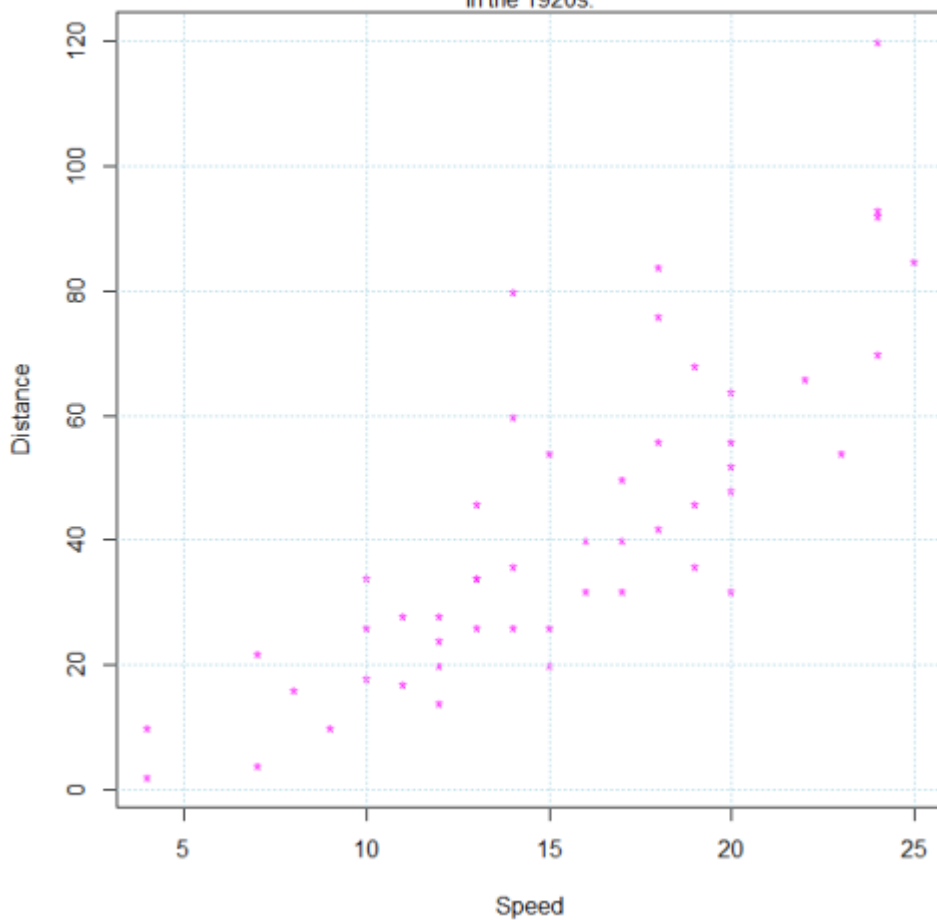


```
points() text() mtext() lines() grid()°
```

```
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")
```

## Distance to stop vs Speed of Cars

In the 1920s.



## Matplot

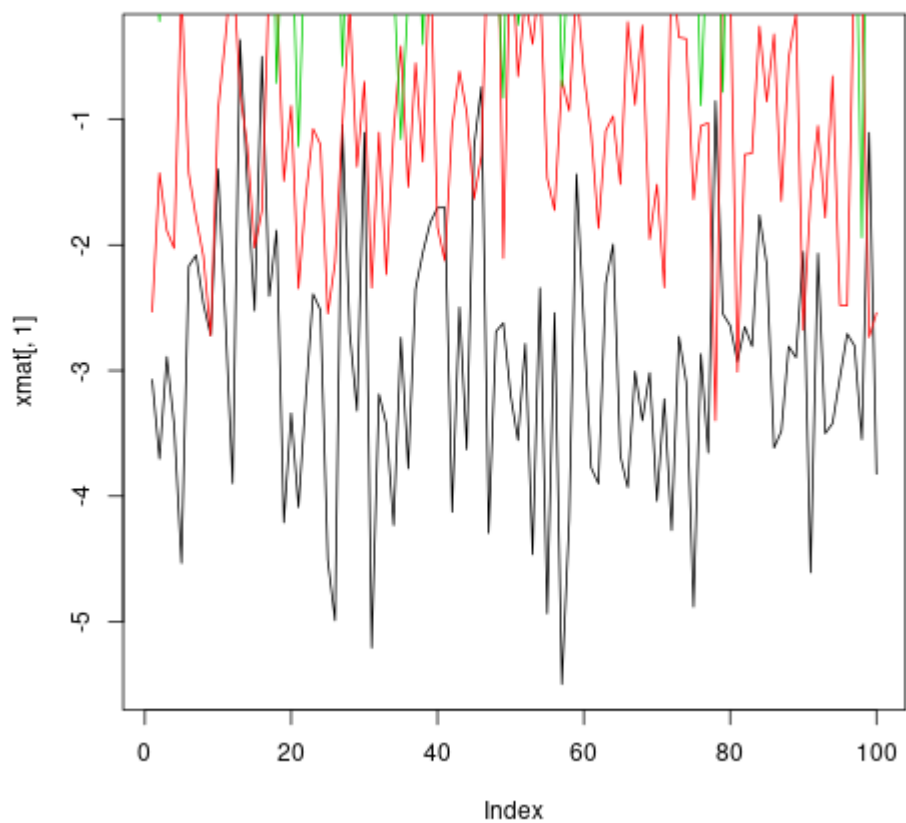
matplot◦

◦

```
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
```

plotpointslines◦

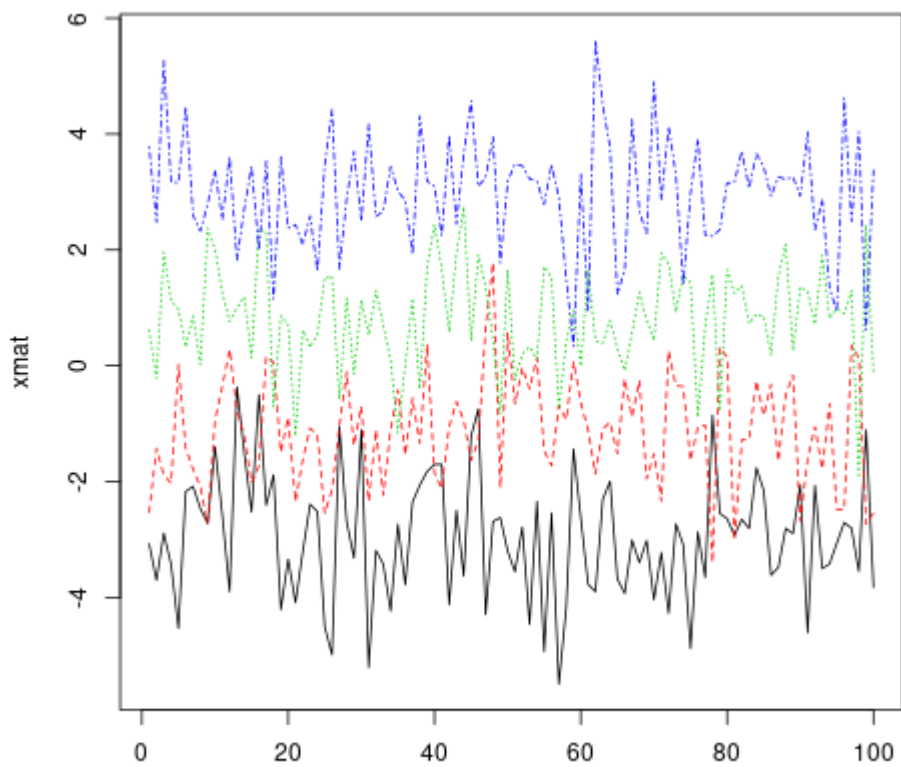
```
plot(xmat[,1], type = 'l')
lines(xmat[,2], col = 'red')
lines(xmat[,3], col = 'green')
lines(xmat[,4], col = 'blue')
```



plot◦

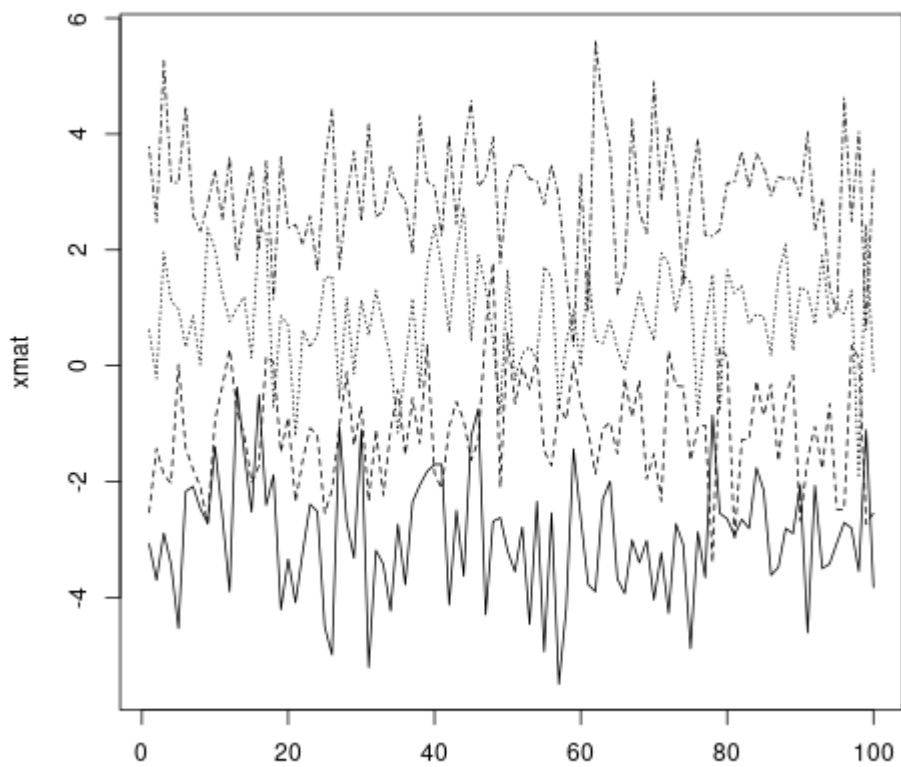
matplot◦

```
matplot(xmat, type = 'l')
```



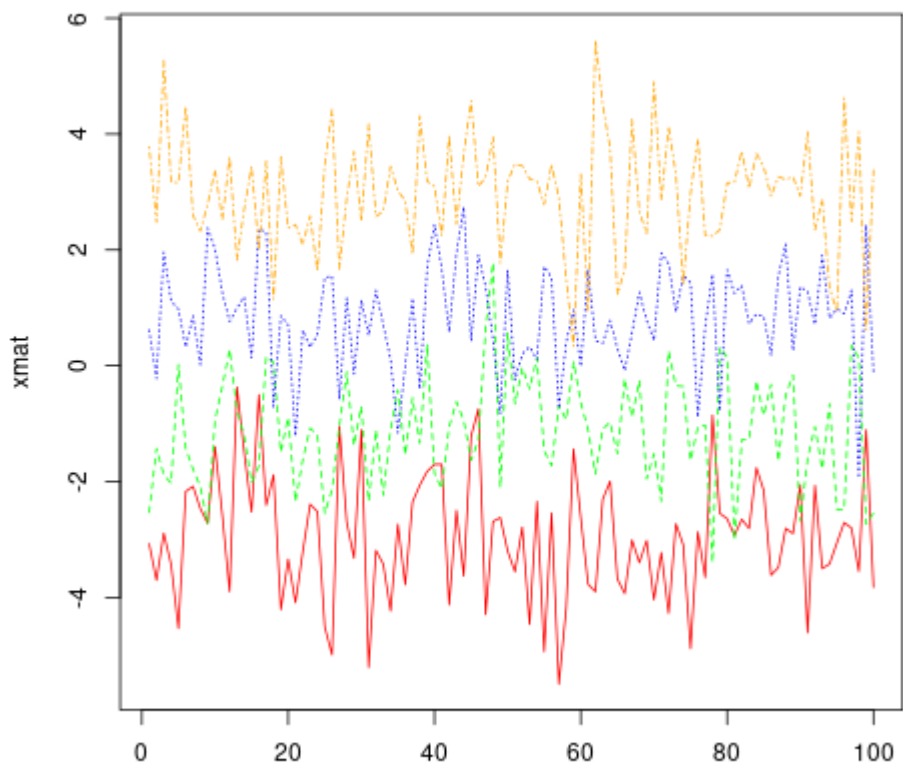
matplot col lty lty ° .....

```
matplot(xmat, type = 'l', col = 'black')
```



...R。

```
matplot(xmat, type = 'l', col = c('red', 'green', 'blue', 'orange'))
```

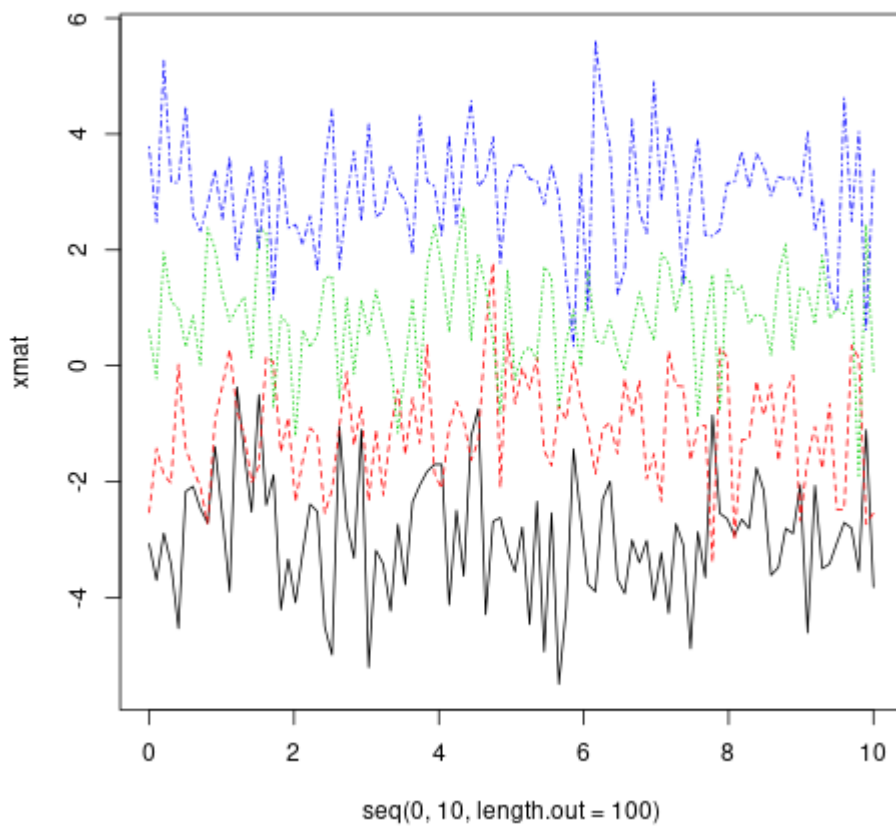


```
main xlab xmin plot ◦ ?par ◦
```

```
plotmatplotxy ◦ xy ◦
```

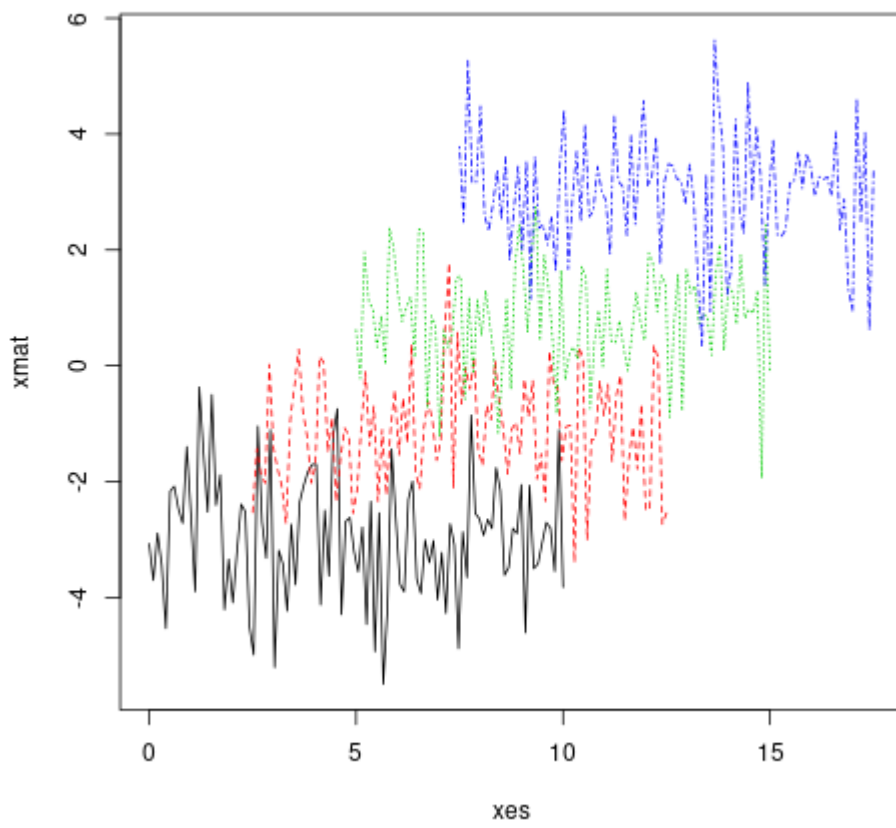
```
matplot(x = seq(0, 10, length.out = 100), y = xmat, type='l')
```





xy°

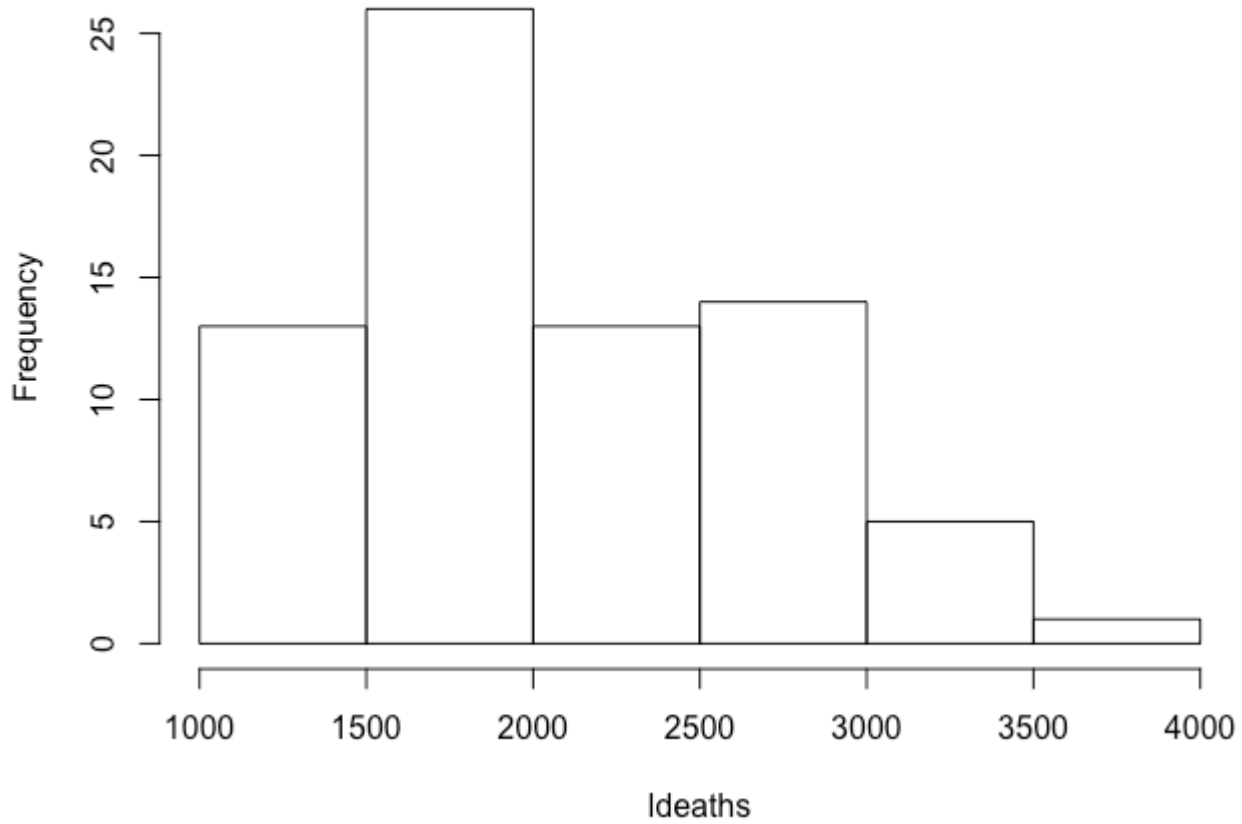
```
xes <- cbind(seq(0, 10, length.out = 100),  
             seq(2.5, 12.5, length.out = 100),  
             seq(5, 15, length.out = 100),  
             seq(7.5, 17.5, length.out = 100))  
matplot(x = xes, y = xmat, type = 'l')
```



o

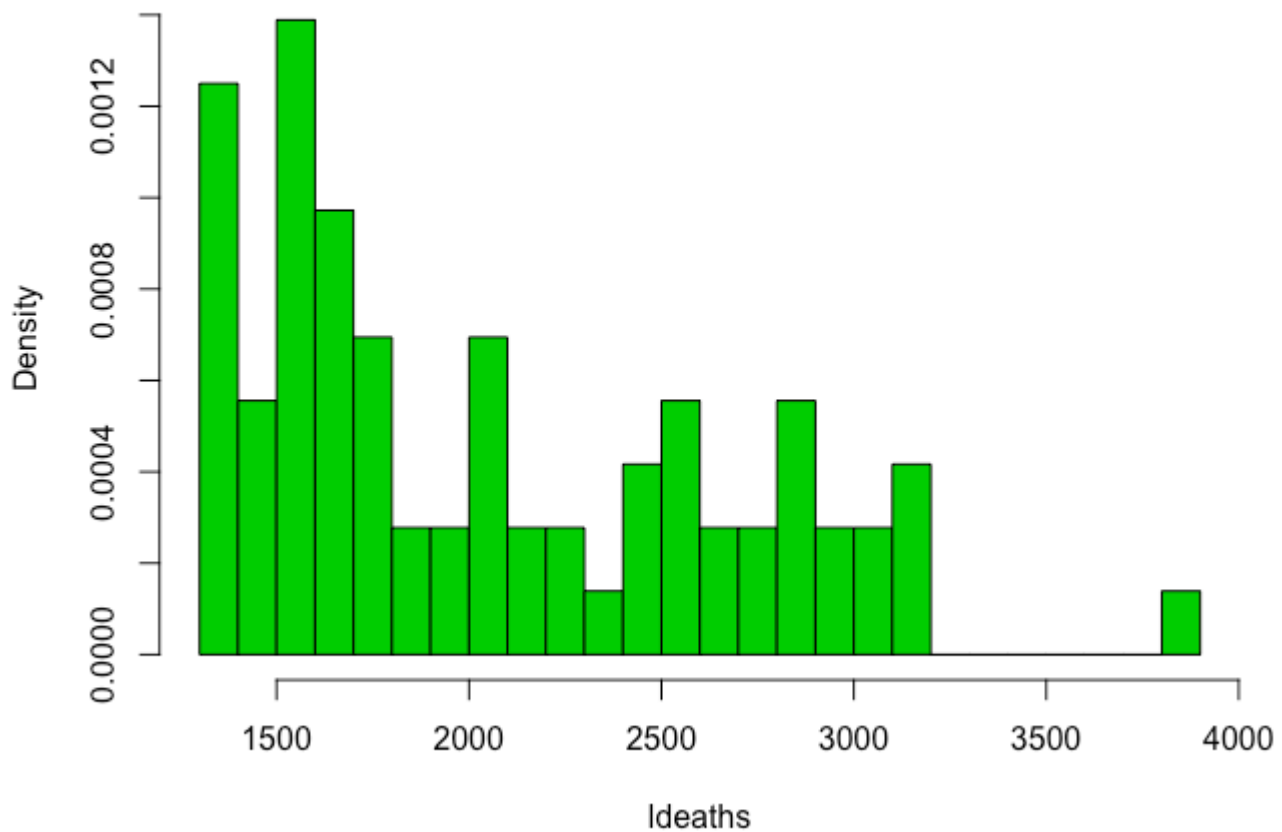
```
hist(ldeaths)
```

## Histogram of Ideaths



```
hist(ldeaths, breaks = 20, freq = F, col = 3)
```

## Histogram of Ideaths



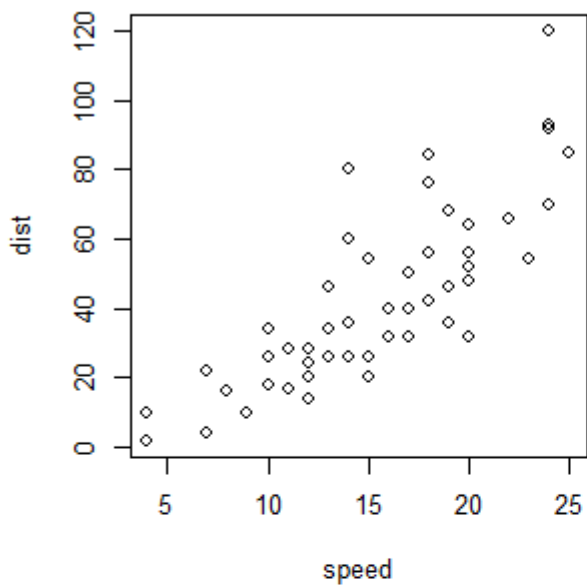
◦ `Rpar()` `layout()` ◦

`par()`

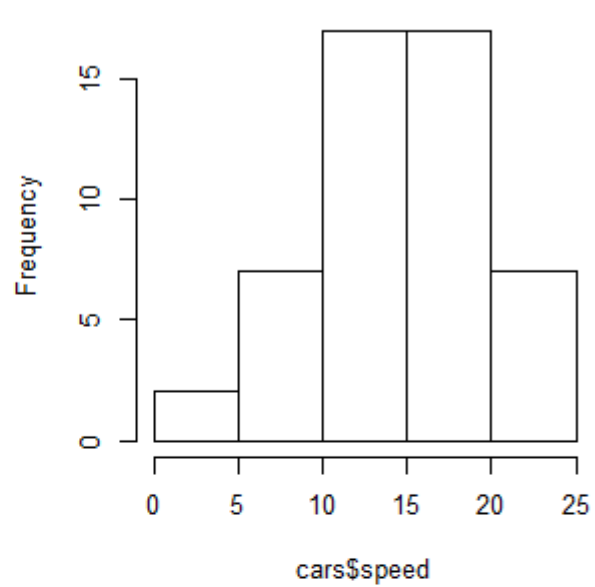
`par(mfrow=matrix(ncols=2, nrow=2))` ◦

```
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")
```

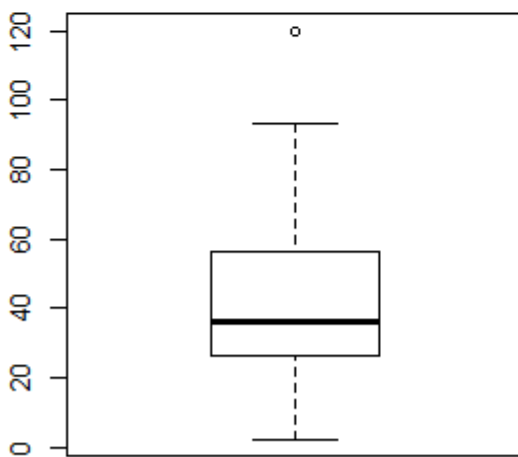
**Speed vs. Distance**



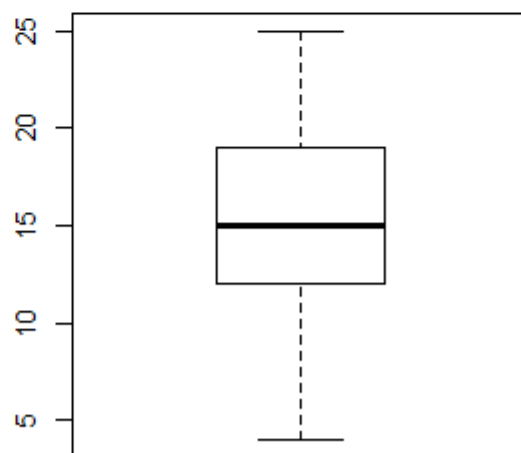
**Histogram of Speed**



**Boxplot of Distance**



**Boxplot of Speed**

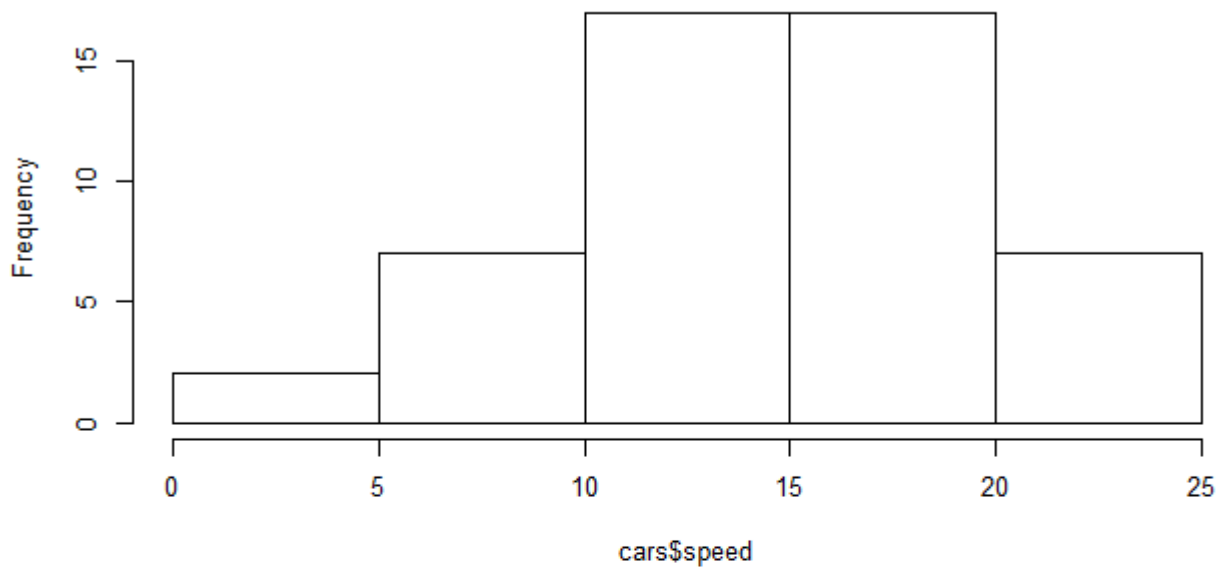


`layout ()`

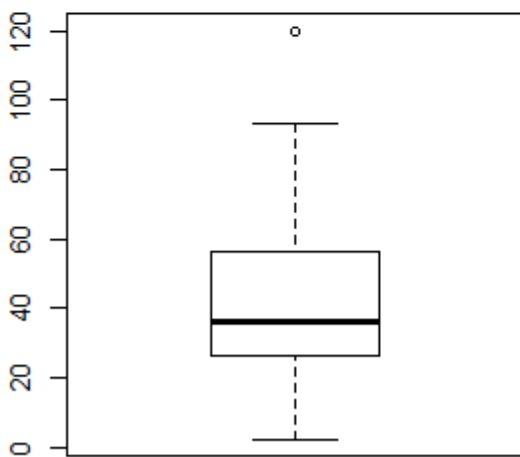
`layout ()` ◦

```
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")
```

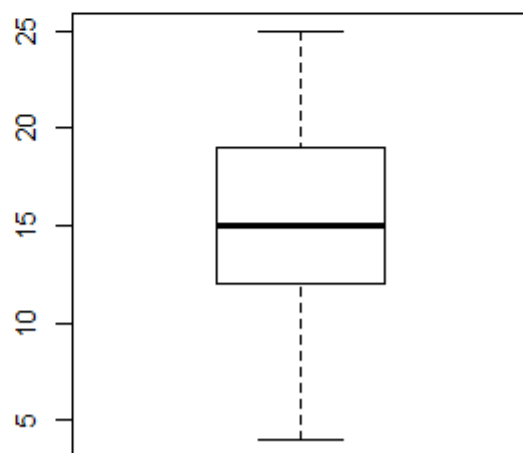
### Histogram of Speed



### Boxplot of Distance

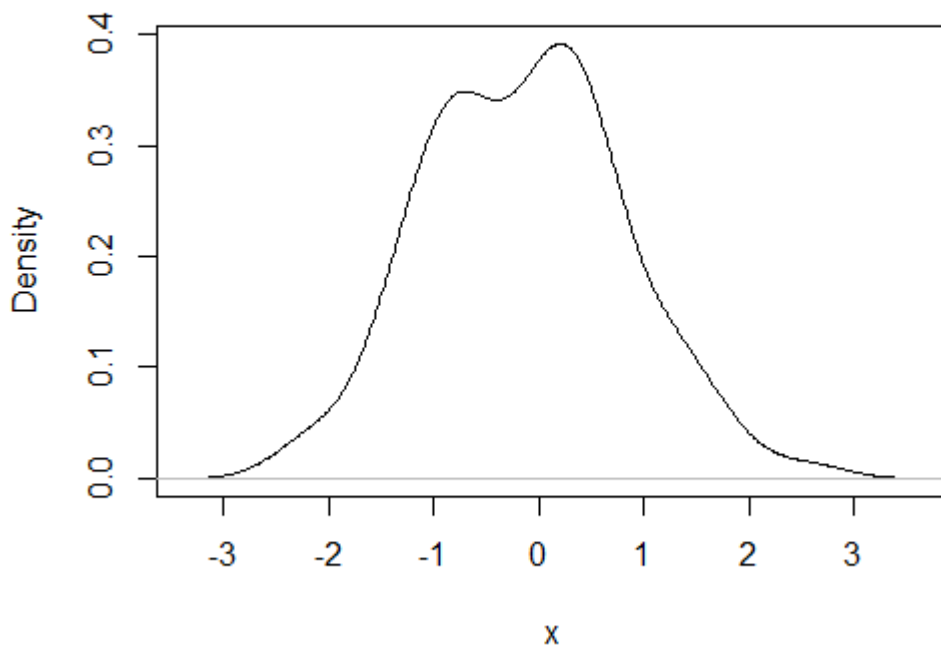


### Boxplot of Speed



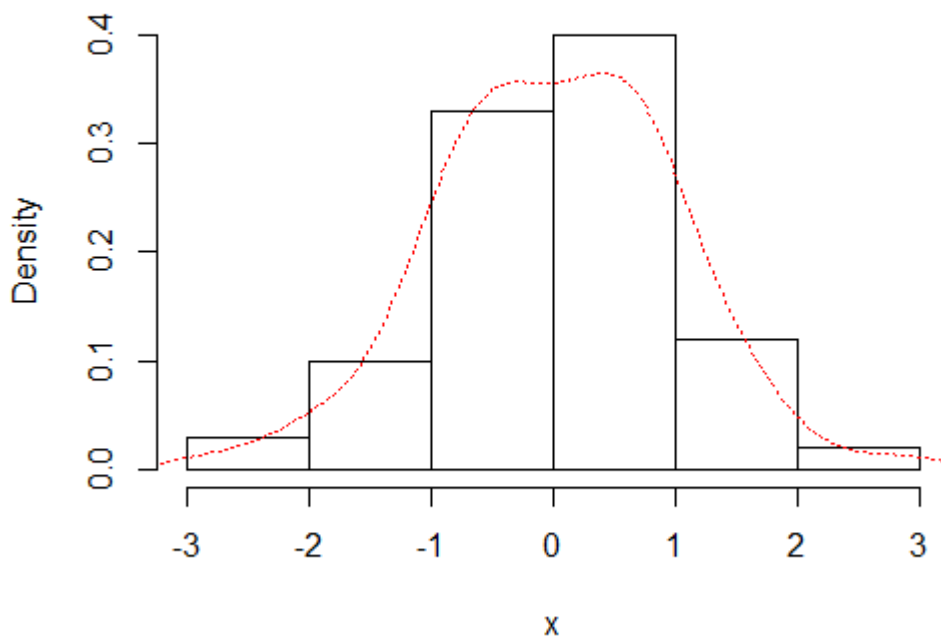
```
plot(density(rnorm(100)),main="Normal density",xlab="x")
```

## Normal density



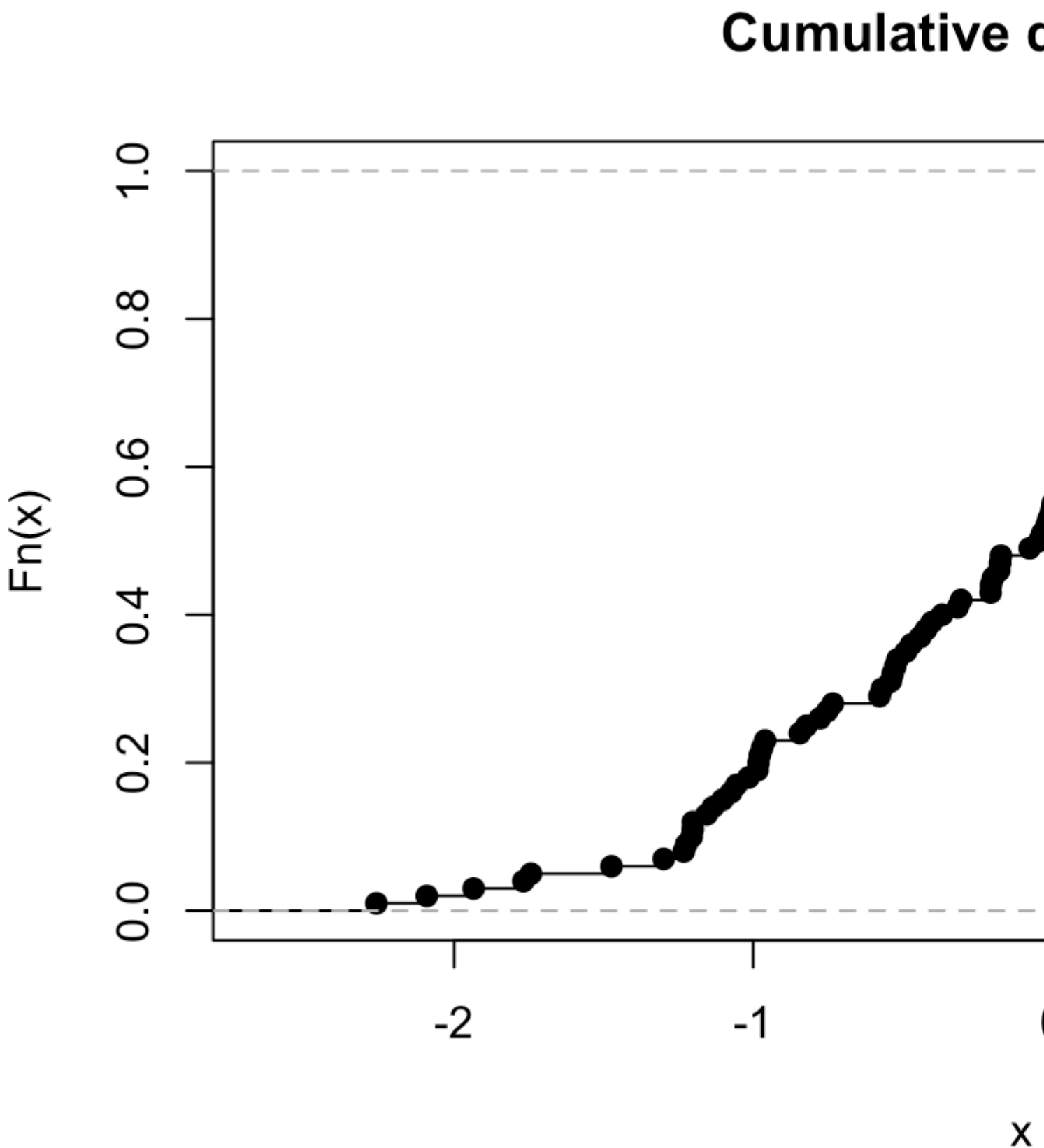
```
x=rnorm(100)
hist(x,prob=TRUE,main="Normal density + histogram")
lines(density(x),lty="dotted",col="red")
```

## Normal density + histogram



◦ `ecdf()` ◦

```
plot(ecdf(rnorm(100)),main="Cumulative distribution",xlab="x")
```



R\_Plots

•



◦

```
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)
```

y\_values x\_values y\_values

```
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
```

?plot()◦

•

```
#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.
```

•

```
hist(x = x_values) # Histogram for x vector
hist(x = x_values, breaks = 3) #use breaks to set the numbers of bars you want
```

•

```
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
```

<https://riptutorial.com/zh-TW/r/topic/1377/>

# 69: I / O Excel SAS SPSS Stata

## Examples

### rio

`rio` ◦ `import()` ◦ `URLimport()`

```
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()` `URLHTTPHTTPS` ◦ [riogithub](#) ◦

`import()`

```
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

#### Rexcel

|           |      |
|-----------|------|
| <b>R</b>  |      |
| XLSX      | Java |
| XLconnect | Java |
| openxlsx  | C ++ |
| readxl    | C ++ |
| RODBC     | ODBC |
| GDATA     | Perl |

JavaODBC ◦ R 64Java 64xlsxXLconnect ◦

`excel` ◦ ◦ `package::function` ◦ `openxlsxRTools` ◦

## xlsxexcel

```
library(xlsx)
```

◦

```
xlsx::read.xlsx("Book1.xlsx", sheetIndex=1)
xlsx::read.xlsx("Book1.xlsx", sheetName="Sheet1")
```

---

## XLconnectExcel

```
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])
```

XLConnectBook1.xlsx **Excel** ◦ **Excel** ◦ Book1.xlsxmyHeader myBodymyPcts ◦ R

```
Headerstyle <- XLConnect::getCellStyle(wb, "myHeader")
Bodystyle <- XLConnect::getCellStyle(wb, "myBody")
Pctsstyle <- XLConnect::getCellStyle(wb, "myPcts")
```

R◦

```
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◦ openxlsx◦

---

## openxlsxexcel

openxlsx**Excel**

```
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.
```

## Rsheet

```
openxlsx::read.xlsx("spreadsheet1.xlsx", sheet = 1)
```

```
openxlsx::read.xlsx("spreadsheet1.xlsx", sheet = "Sheet1")
```

openxlsx<sup>o</sup> detectDatesTRUE

```
openxlsx::read.xlsx("spreadsheet1.xlsx", sheet = "Sheet1", detectDates= TRUE)
```

---

## readxl

readxl<sup>o</sup> Excel<sup>o</sup>

```
library(readxl)
```

.xls.xlsx<sup>o</sup>

```
readxl::read_excel("spreadsheet1.xls")  
readxl::read_excel("spreadsheet2.xlsx")
```

.

```
readxl::read_excel("spreadsheet.xls", sheet = 1)  
readxl::read_excel("spreadsheet.xls", sheet = "summary")
```

col\_names = TRUE<sup>o</sup>

```
readxl::read_excel("spreadsheet.xls", sheet = 1, col_names = TRUE)
```

col\_types<sup>o</sup>

```
readxl::read_excel("spreadsheet.xls", sheet = 1, col_names = TRUE,  
                    col_types = c("text", "date", "numeric", "numeric"))
```

---

## RODBC

WindowsAccessACEJETODBC ExcelExcel<sup>o</sup> RODBCR<sup>o</sup> <sup>o</sup> Windows / PCJET / ACE.dll<sup>o</sup>

```
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)
```

## SQLExcel JOIN UNION ◦ JET / ACE SQL ◦ DML SELECT ◦

```
joindf <- sqlQuery(xlconn, "SELECT t1.*, t2.* FROM [Sheet1$] t1
                           INNER JOIN [Sheet2$] t2
                           ON t1.[ID] = t2.[ID]")

uniondf <- sqlQuery(xlconn, "SELECT * FROM [Sheet1$]
                           UNION
                           SELECT * FROM [Sheet2$]")
```

## ODBC

```
otherwkbkdf <- sqlQuery(xlconn, "SELECT * FROM
                                [Excel 12.0 Xml;HDR=Yes;
                                Database=C:\\Path\\To\\Other\\Workbook.xlsx].[Sheet1$];")
```

---

# gdataexcel

## Stata SPSS SAS

foreignhaven **Stata SPSS SAS** ◦ read ◦

```
# loading the packages
library(foreign)
library(haven)
library(readstata13)
library(Hmisc)
```

```
# reading Stata files with `foreign`
read.dta("path\to\your\data")
# reading Stata files with `haven`
read_dta("path\to\your\data")
```

foreign **stata.dta Stata 7-12** ◦ read.dta **13+** ◦ **Stata** readstata13 haven ◦ readstata13

```
# reading recent Stata (13+) files with `readstata13`
read.dta13("path\to\your\data")
```

## SPSS SAS

```
# 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
```

SAS Sci SAS SET read.fwf ◦ <https://github.com/ajdamico/SAS Sci>

write write.foreign() ◦ 2◦

```
# 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 `readstata13`
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")
```

read.spss SPSS

```
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 be 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.
```

Feather Apache Arrow Python R ◦ tibble data.frame ◦

```
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

```

alpha。 。 Feather。

I / O Excel SAS SPSS Stata <https://riptutorial.com/zh-TW/r/topic/5536/i---o-excel-sas-spss-stata->

# 70:

R◦ subsetting ◦

```
NA S[NANANA
```

```
NA ..
```

```
NA"" typeof(NA) ""◦ x[NA]x[as.logical(NA)] x[rep_len(as.logical(NA), length(x))] NA x◦
```

```
x <- 1:3
x[NA]
## [1] NA NA NA
```

“numeric”/“integer” NANANA

```
x[as.integer(NA)]
## [1] NA

x[c(NA, 1, NA, NA)]
## [1] NA 1 NA NA
```

```
[> length(x)NANULL◦ [[length(dim(x)) > 2
```

```
(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
```

“character”“names”

```
c(a = 1, b = 2)["c"]
## <NA>
## NA
list(a = 1, b = 2)["c"]
## <NA>
## NULL
```

?Extract ◦

## Examples



[

```
# create an example vector
v1 <- c("a", "b", "c", "d")

# select the third element
v1[3]
## [1] "c"
```

[°

```
v1 <- c("a", "b", "c", "d")

v1[c(1, 3)]
## [1] "a" "c"
```

◦ - ◦ **v1**v1[-1] ◦ ◦ v1[-c(1,3)] ◦

```
> v1[-1]
[1] "b" "c" "d"
> v1[-c(1,3)]
[1] "b" "d"
```

```
> v1=="c"
[1] FALSE FALSE TRUE FALSE
> which(v1=="c")
[1] 3
```

names

```
v <- 1:3
names(v) <- c("one", "two", "three")

v
## one two three
## 1 2 3

v["two"]
## two
## 2
```

[[1

```
v[[c(1, 2)]]
## Error in v[[c(1, 2)]] :
## attempt to select more than one element in vectorIndex

v[["two"]]
## [1] 2
```

◦ yx x[y] length(y) < length(x)ylength(x)

```
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)
```

[

```
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[2][<sup>o</sup> [[<sup>o</sup>

<sup>o</sup> [<sup>o</sup> length > 1[[["

```
l1[c(3, 1)]
## [[1]]
## [[1]][[1]]
## [1] 10
##
## [[1]][[2]]
## [1] 20
##
##
## [[2]]
```

```
## [1] 1 2 3
```

```
l1[[c(3, 1)]]  
## [1] 10
```

```
l1[[3]][[1]]  
## [1] 10
```

```
$[[ ]° $
```

```
l1$two  
## [1] "a" "b" "c"
```

```
$
```

```
l1$t  
## [1] "a" "b" "c"
```

```
[[
```

```
l1[["t"]]  
## NULL  
l1[["t", exact = FALSE]]  
## [1] "a" "b" "c"
```

```
options(warnPartialMatchDollar = TRUE) $""
```

```
l1$t  
## [1] "a" "b" "c"  
## Warning message:  
## In l1$t : partial match of 't' to 'two'
```

```
[° ° [i, j][i, j]i j° 1°
```

```
## 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]matij° i2j1° ij°
```

```
mat[ , 3]  
## row1 row2  
##    5    6  
  
mat[1, ]  
# col1 col2 col3  
#    1    3    5
```

```
mat[ , 'col1']
# row1 row2
#    1    2
```

◦ ◦ [ drop = FALSE

```
## 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"
```

```
mat[1:2, 2:3] ## A 2x2 matrix
#      col2 col3
# row1    3    5
# row2    4    6
```

**Nx2N**◦ (1st row, 1st column), (1st row, 3rd column), (2nd row, 3rd column), (2nd row, 1st column)

```
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
```

ind<sub>1</sub>mat ind<sub>2</sub>mat◦

◦

```
> 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
```

`[[ ]]`。

```
> 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
```

`ij`。

```
> 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
```

`is.data.frame`。

```
> is.vector(df3[, 2])
## TRUE
```

```
> is.data.frame(df3[2, ])
## TRUE

> is.data.frame(df3[, 2, drop = FALSE])
## TRUE
```

```
[1] R isTRUE(is.object(x)) - "";/[ [ ] ]
```

“data.frame” is.object(iris) [.data.frame][.data.frame]“”“ “data.frame”[ ] [.data.frame]

```
iris[invalidArgument, ]
## Error in `[.data.frame'(iris, invalidArgument, ) :
## object 'invalidArgument' not found
```

```
[
```

```
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
```

```
[.subset [ [ .subset2 "unclass(x)
```

```
.subset(x, c(3, 2, 4))
## [1] 3 2 4
```

```
> x <- 11:20
> x
[1] 11 12 13 14 15 16 17 18 19 20
```

R1<sub>x[1]11</sub> . x

```
> x[c(2,4,6)]
[1] 12 14 16
```

R

```
> x[c(-1,-3)]
[1] 12 14 15 16 17 18 19 20
```

TRUE

```
> x[c(rep(TRUE,5),rep(FALSE,5))]
[1] 11 12 13 14 15 16
```

```
> x[c(TRUE, FALSE)]
[1] 11 13 15 17 19
> x[c(TRUE, FALSE, FALSE)]
[1] 11 14 17 20
```

AB. + - / \* ^ . °

|   | B       |    |
|---|---------|----|
| + | A + B.  | AB |
| - | A - B.  | AB |
| / | A / B.  | AB |
| * | A * B.  | AB |
| ^ | a ^ - 1 | A  |

“” %\*% ° AB A %\*% B ° ncol() nrow() B

| nrow      | nrowA      | A     |
|-----------|------------|-------|
| Ncol      | NcolA      | A     |
| rownames  | rownamesA  | A     |
| colnames  | colnamesA  | A     |
| rowMeans  | rowMeansA  | A     |
| colMeans  | colMeansA  | A     |
| upper.tri | upper.triA | upper |
|           |            | A     |
| lower.tri | lower.triA | lower |
|           |            | A     |
| DET       | DETA       | A     |
|           | A          | A     |
|           | A          |       |
|           |            | A     |

| T         | A          | A           |
|-----------|------------|-------------|
|           | A          | A           |
| is.matrix | is.matrixA | TRUEFALSEA。 |
| as.matrix | as.matrixx | x           |

<https://riptutorial.com/zh-TW/r/topic/1686/>



# 71:

- `install.packages(pkgs=libreposmethod=destdir=dependencies=...`

|         |                                    |
|---------|------------------------------------|
| PKGS    | ◦ repos = NULL repos = NULL◦       |
| LIB     | ◦                                  |
|         | URL=NULL                           |
| DESTDIR |                                    |
|         | // import / suggest◦ repos = NULL◦ |
| ...     | 'download.file'OS XWindows◦        |

- 

## Examples

R ◦ R ◦ CRANR ◦

## CRAN

### CRAN

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

"dplyr"◦

`combine_c()`

```
install.packages(c("dplyr", "tidyr", "ggplot2"))
```

`install.packages(CRAN=getOption("repos") ◦ CRANrepos`

```
install.packages("dplyr", repos = "https://cloud.r-project.org/")
```

`repos ◦ ?install.packages ◦`

`data.table ◦ dependencies=TRUE`

```
install.packages("data.table", dependencies = TRUE)
```

# Bioconductor

Bioconductor Bioinformatics `biocLite`

```
## Try http:// if https:// URLs are not supported
source("https://bioconductor.org/biocLite.R")
biocLite()
```

◦ ◦ **Bioconductor** `RImmPort`

```
source("https://bioconductor.org/biocLite.R")
biocLite("RImmPort")
```

```
install.packages(path_to_source, repos = NULL, type="source")

install.packages("~/Downloads/dplyr-master.zip", repos=NULL, type="source")
```

`path_to_source` ◦

`ziptar.gz`

```
install.packages(file.choose(), repos=NULL)
```

---

## GUIRStudio

1 “ ” ◦

2 ◦

3 **Install From.zip; .tar.gz**

4 `crayon_1.3.1.zip`

---

`Rdevtools` `install_local()` ◦

```
library(devtools)
install_local("~/Downloads/dplyr-master.zip")
```

## GitHub

GitHub `devtools`

```
library(devtools)
install_github("authorName/repositoryName")
```

## github

```
devtools::install_github("tidyverse/ggplot2")
```

**master** ggplot2 ref googlewaydev\_general

```
devtools::install_github("SymbolixAU/googleway", ref = "dev_general")
```

ghit **github**

```
install.packages("ghit")
ghit::install_github("google/CausalImpact")
```

Github <http://www.github.com/settings/tokens/> install\_github

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")
```

```
ssl.verifypeer ssl.verifyhost
FALSE          FALSE
```

5. 

```
library(httr)
set_config(config(ssl_verifypeer = 0L))
```

“CA”

6. 

```
install_github("username/package_name", auth_token="abc")
```

GITHUB\_PAT

```
Sys.setenv(GITHUB_PAT = "access_token")
devtools::install_github("organisation/package_name")
```

GithubPAT.Rprofile . .

## CLI - pacman

pacmanR

pacmanp\_load pacman

```
p_load(data.table, dplyr, ggplot2)
```

```
library(requireinstall.packages)pacman
```

```
library(pacman)  
p_load(data.table, dplyr, ggplot2)
```

```
pacman::p_load(data.table, dplyr, ggplot2)
```

pacman°

pacmanpacman

```
if(!(require(pacman)) install.packages("pacman"))  
pacman::p_load(data.table, dplyr, ggplot2)
```

R°

```
R CMD build my_package
```

R° R°

```
unloadNamespace("my_package")  
library(my_package)
```

devtools° R

```
devtools::install()
```

°

<https://riptutorial.com/zh-TW/r/topic/1719/>

# 72: /

```

EXPR "try part" tryCatch ◦ readLines ◦ return explicitly "try part"
// etc /◦ AFAIU ;simpleCondition tryCatch ◦
""◦ finally = <expression> "try part"◦

```

## tryCatch

tryCatch(expr ◦ return(NA) warning?tryCatch error ◦ tryCatch◦

NA""yNA ◦ NULLy23lapply""/""NULL ◦ returnNULL ◦

""

urlsreadLines

```

Warning message:
  In file(con, "r") : cannot open file 'I'm no URL': No such file or directory

```

"" ◦ readLineswarn = FALSE ◦

```

suppressWarnings(readLines(con = url))

```

```

readLines(con = url, warn = FALSE)

```

## Examples

### tryCatch

URLHTML. ◦

### tryCatchtryCatch

```

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)
}

```

## URLURL

```

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"
)

```

```

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
```

```
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
```

<https://riptutorial.com/zh-TW/r/topic/4060/>

# 73:

## Examples

Logistic *probit**log-log*.

*logitlog-odds*. *logit*

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

-Inf; + Inf [01];

```
family = binomialglm family = binomialfamily = binomial(link="logit") ; logit
```

RMS.

```
url <- "http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/titanic.txt"
titanic <- read.csv(file = url, stringsAsFactors = FALSE)
```

o

```
titanic$age[is.na(titanic$age)] <- mean(titanic$age, na.rm = TRUE)
```

```
titanic.train <- glm(survived ~ pclass + sex + age,
                    family = binomial, data = titanic)
```

```
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
```

- ◦ ◦
- ◦ ◦
- **zWald zp**
  - **“”** ◦ ◦ I ◦
  - **0.1** ◦
  - ◦
  - ◦
  - **CI**<sub>confint</sub> ◦
- **AkaikeAIC**
  - **AIC** ◦
  - ◦ ◦

```
exp(coef(titanic.train)[3])
```

```
pclass3rd  
0.08797765
```

◦

```
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
```

◦

```
with(titanic.train, pchisq(null.deviance - deviance, df.null - df.residual  
, lower.tail = FALSE))  
[1] 1.892539e-111
```

p0

<https://riptutorial.com/zh-TW/r/topic/2892/>

# 74:

## Examples

RWilkinson-Rogers<sup>1</sup>。

lm formula ◦ formula◦

```
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
```

~ LHSRHS◦ formula~tilde-operator

```
form <- mpg ~ wt
class(form)
#[1] "formula"
```

formula~

```
form_mt <- formula(mpg ~ wt, env = mtcars)
```

wt◦ formula0-1 /0

```
coef(lm(mpg ~ 0 + wt, data = mtcars))
coef(lm(mpg ~ wt -1, data = mtcars))
```

abformulaa:b

```
coef(lm(mpg ~ wt:vs, data = mtcars))
```

formulaa + b + a:b◦ a\*b\*“”

```
coef(lm(mpg ~ wt*vs, data = mtcars))
```

\*

```
coef(lm(mpg ~ wt*vs*hp, data = mtcars))
```

7◦ ◦

◦ -

```
coef(lm(mpg ~ wt*vs*hp - wt:vs:hp, data = mtcars))
```

^

```
coef(lm(mpg ~ (wt + vs + hp) ^ 2, data = mtcars))
```

o

.o dataLHSo

```
coef(lm(mpg ~ ., data = mtcars))
```

10o o .?update.formulao

## 1. GN WilkinsonCE Rogerso o C Vol. 22No. 31973pp.392-399

$y \sim . . y$  o  $y \sim \text{var1} + \text{var2} + \text{var3} + \dots + \text{var15}$

$y \sim . ^ 2$  o  $y \sim \text{var1} + \text{var2} + \dots + \text{var15} + \text{var1}:\text{var2} + \text{var1}:\text{var3} + \text{var1}:\text{var4} \dots$  and so on

$y \sim \text{var1} + \text{var2} + \dots + \text{var15} + \text{I}(\text{var1}^2) + \text{I}(\text{var2}^2) + \text{I}(\text{var3}^2) \dots + \text{I}(\text{var15}^2)$  I(var^2)o

$y \sim \text{poly}(\text{var1}, \text{degree} = 2) + \text{poly}(\text{var2}, \text{degree} = 2) + \dots + \text{poly}(\text{var15}, \text{degree} = 2)$

$y \sim \text{poly}(\text{var1}, \text{var2}, \text{var3}, \dots, \text{var15}, \text{degree} = 2)$ o

$\text{poly}(\text{var1}, \text{degree} = 2)\text{var1} + \text{I}(\text{var1}^2)$  o

$\text{poly}()$ degree = 3 o

$\text{polyI}(\text{var}, 2)$ o  $\text{poly}()$ I(var, 2)  $\text{poly}()$  $\text{poly}()$ o

$y \sim .^2 + \text{I}(\text{var1}^2) + \text{I}(\text{var2}^2) + \dots + \text{I}(\text{var15}^2)$

```
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 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>
```

<https://riptutorial.com/zh-TW/r/topic/1061/>

# 75:

as.numericas.data.frameR。

## Examples

R。

```
x = 1:3
x
[1] 1 2 3
typeof(x)
#[1] "integer"

x[2] = "hi"
x
#[1] "1" "hi" "3"
typeof(x)
#[1] "character"
```

xinteger。 x[2] = "hi" xcharacterR。

<https://riptutorial.com/zh-TW/r/topic/9793/>

---

# 76:

## For. for

```
for ( [index] in [domain]){  
  [body]  
}
```

1. [index][domain]
2. [domain]
3. [body]

## for.

```
x <- 1:4  
cumulative_sum <- 0  
for (i in x){  
  cumulative_sum <- cumulative_sum + x[i]  
}  
cumulative_sum
```

---

## for. apply. for. forapply.

### “for. .

```
[output] <- [vector_of_length]  
for ([index] in [length_safe_domain]){  
  [output][index] <- [body]  
}
```

◦ "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  
}
```

x\_squaredx“numeric” ◦ seq\_along“” ◦ seq\_alongforseq\_along ◦ for (i in 1:length(x)) x01:00R ◦

applyapplyfor. for.

---

## For. .

formtcarsmtcars colMeans◦

```
column_mean_loop <- vector("numeric", length(mtcars))
for (k in seq_along(mtcars)){
  column_mean_loop[k] <- mean(mtcars[[k]])
}
```

forapply。

```
col_mean_fn <- function(x) mean(x)
column_mean_apply <- vapply(mtcars, col_mean_fn, numeric(1))
```

```
identical(column_mean_loop,
           unname(column_mean_apply)) /* vapply added names to the elements
                                       /* remove them for comparison
```

◦ apply。 apply。 ◦

## Examples

mtcars ◦

A

```
squared_deviance <- vector("list", length(mtcars))
for (i in seq_along(mtcars)){
  squared_deviance[[i]] <- (mtcars[[i]] - mean(mtcars[[i]]))^2
}
```

squared\_deviance<sup>11</sup>。

```
class(squared_deviance)
length(squared_deviance)
```

B

```
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
}
```

data.frame。 fordata.frame ◦

```
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
```

B.

## For

good for loop

1. for
2. for
3. \*apply
4. colMeans

;;

```
column_mean_poor <- NULL
for (i in 1:length(mtcars)){
  column_mean_poor[i] <- mean(mtcars[[i]])
}
```

```
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)
```

```
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
```

for for forR

for

vapplyR

colMeansvapply colMeansvapplyas.matrixmtcarsdata.frame

R while repeat



## while

while

```
while (condition) {  
  ## do something  
  ## in loop body  
}
```

condition° conditionTRUE conditionFALSE break;° forwhile°

```
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
```

while i <- i + 1°

---

breakwhile

```
iter <- 0  
while (TRUE) {  
  if (runif(1) < 0.25) {  
    break  
  } else {  
    iter <- iter + 1  
  }  
}  
iter  
#[1] 4
```

conditionTRUE bodybreak° iterPRNG°

---

## repeat

repeatwhile (TRUE) { ## something }

```
repeat ({  
  ## do something
```

```
## in loop body
})
```

{}()◦ repeat

```
iter <- 0
repeat ({
  if (runif(1) < 0.25) {
    break
  } else {
    iter <- iter + 1
  }
})
iter
#[1] 2
```

## break

break ◦

```
while (TRUE) {
  while (TRUE) {
    cat("inner loop\n")
    break
  }
  cat("outer loop\n")
}
```

◦

```
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))
    }
  }
}
```

break return◦ returnreturn()break

```
(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))
    }
}
})()
```

exit <<-

```
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
}
```

<https://riptutorial.com/zh-TW/r/topic/2201/>

# 77:

caretR ◦ ◦ ◦

caret ◦

## Examples

preProcess() ◦ x preProcess() ◦

preProcess()method ◦

- 1.
- 2.
3. Box-Cox / Yeo-Johnson /
- 4.
- 5.
- 6.
- 7.
8. PCA
9. ICA
- 10.

mtcars ◦

```
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)
```

<https://riptutorial.com/zh-TW/r/topic/4271/>

## 78:

### Examples

◦ `is.numeric()` ◦ `is.numeric()` `as.numeric()` `double` ◦

```
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
```

R ◦ 8 ◦ R ◦

```
is.double(1)
TRUE
is.double(1.0)
TRUE
is.double(1L)
FALSE
```

◦ L ◦ L ◦

```
typeof(1)
[1] "double"
class(1)
[1] "numeric"
typeof(1L)
[1] "integer"
class(1L)
[1] "integer"
```

◦ 84 ◦ ◦

```

# 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

```

<https://riptutorial.com/zh-TW/r/topic/9018/>

# 79: I / O.

- RMySQL
- RODBC

## Examples

### MySQL

#### RMySQLMySQLMariaDBR

```
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
```

#### 100,000 SQL

```
queryString <- "SELECT * FROM table1 limit 100000"
```

### MongoDB

#### MongoDBRMongoLite

```
# 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 }')
```

PASSWORDUSERNAMEHOSTNAME TweetCollectorTweets . Text .

. Text documents\$Text .

I / O. <https://riptutorial.com/zh-TW/r/topic/5537/i---o->

---

**80:**

R. R. R.

## Examples

R. ◦

<https://vincentarelbundock.github.io/Rdatasets/datasets.html>

---

---

1888. ◦

```
library(tidyverse)

swiss %>%
  ggplot(aes(x = Agriculture, y = Fertility,
             color = Catholic > 50))+
  geom_point()+
  stat_ellipse()
```

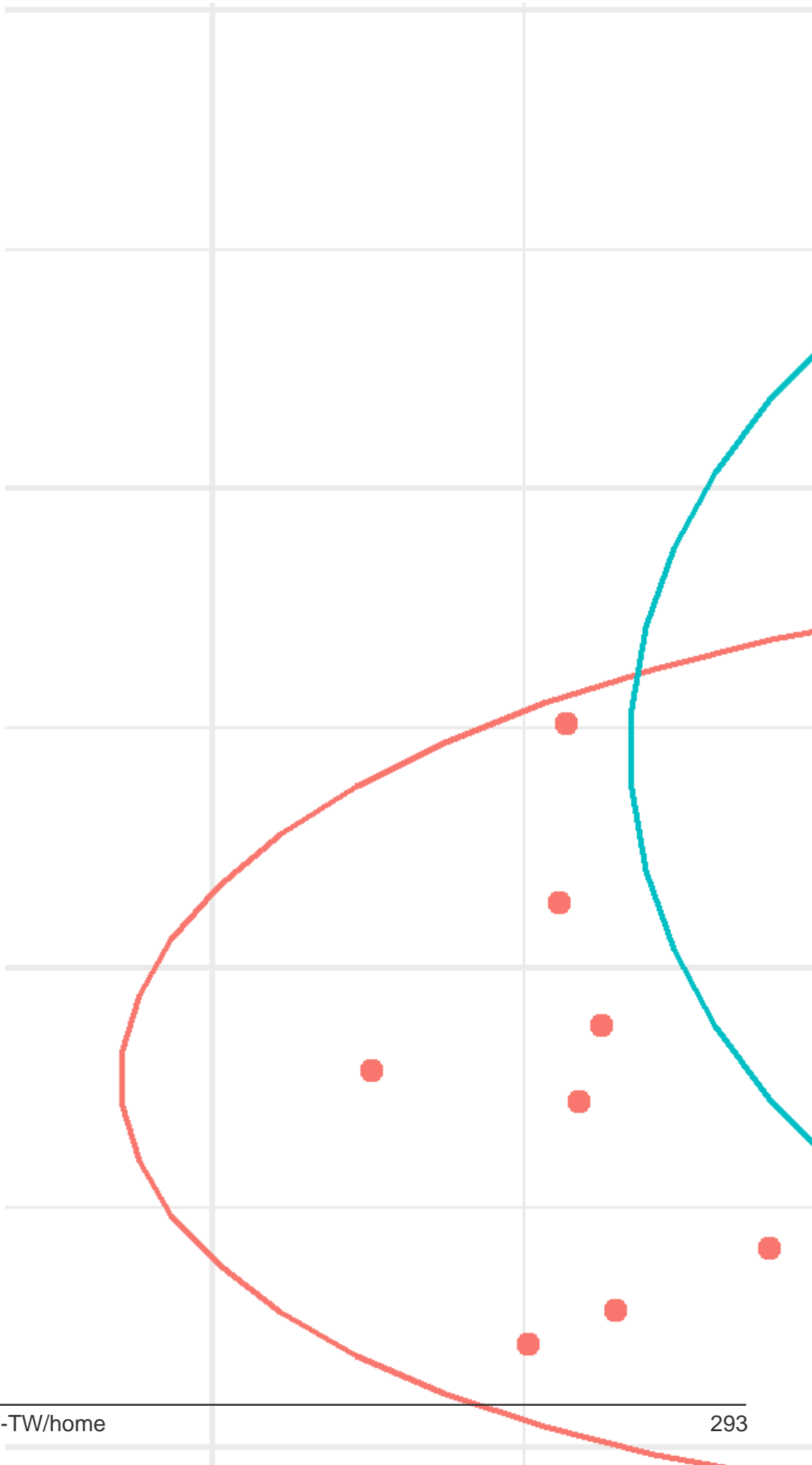


Fertility

110

90

70



# 81:

- `data.frame...row.names = NULL``check.rows = FALSE``check.names = TRUE``stringsAsFactors = default``stringsAsFactors`
- `as.data.frame(x, row.names = NULL, optional = FALSE, ...)` generic function
- `as.data.frame(x, ...stringsAsFactors = default.stringsAsFactors)``S3'`
- `as.data.frame(x, row.names = NULL, optional = FALSE, ...stringsAsFactors = default.stringsAsFactors)``S3'matrix'`
- `is.data.frame(x)`

## Examples

### data.frame

`data.frame` ◦ "" ◦ ◦

`data.frame`

```
> structure(list(character()), class = "data.frame")
NULL
<0 rows> (or 0-length row.names)
```

◦ `data.frame` ◦ `data.frame` a b

```
> structure(list(a = 1:3, b = letters[1:3]), class = "data.frame")
[1] a b
<0 rows> (or 0-length row.names)
```

`data.frame` ◦ 13

```
> 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
```

`data.frame` ◦ `nrow()` `ncol()` `dim()`

```
> 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
```

Rstructure() **data.frame** ◦ data.frame() ◦ ◦ data.frame()

```
> 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
```

as.data.frame() ◦ data.frame() **data.frame** data.frame() ◦

```
> m <- matrix(letters[1:9], nrow = 3)
> m
      [,1] [,2] [,3]
[1,] "a"  "d"  "g"
[2,] "b"  "e"  "h"
[3,] "c"  "f"  "i"
```

```
> 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
```

[ [ [ \$

◦

- data[rows, columns]matrix data[rows, columns]
  - 
  -
- list
  - data[columns]
  - data[[one\_column]]
- \$data\$column\_name

mtcars◦

**data[rows, columns]**

mtcars [ ]◦

```
# get the first row
mtcars[1, ]
# get the first five rows
mtcars[1:5, ]
```

```
# get the first column
mtcars[, 1]
# get the first, third and fifth columns:
mtcars[, c(1, 3, 5)]
```

- `mtcars[1, ]`
- `data.frame` ◦ `character` ◦ `numeric`

```
# get the mpg column
mtcars[, "mpg"]
# get the mpg, cyl, and disp columns
mtcars[, c("mpg", "cyl", "disp")]
```

```
mtcars["Mazda Rx4", ]
```

```
# 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")]
```

- ◦

```
## multiple columns returns a data frame
class(mtcars[, c("mpg", "cyl")])
# [1] "data.frame"
## single column returns a vector
class(mtcars[, "mpg"])
# [1] "numeric"
```

- `drop = FALSE` ◦ **R**

```
class(mtcars[, "mpg", drop = FALSE])
# [1] "data.frame"
```

- `drop = FALSE` ◦

`list` ◦ `[[]]` ◦

**data[columns]**

- 

```
mtcars["mpg"]
mtcars[c("mpg", "cyl", "disp")]
my_columns <- c("mpg", "cyl", "hp")
mtcars[my_columns]
```

`data[columns]` `data[, columns]` `data.frame` `list` `data.frame` ◦ `data.frame` `matrix` `data.frame` ◦

```
## 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"
```

```
data[[one_column]]
```

```
data.frame(list [[ ° °
```

```
# 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"]
```

```
$
```

```
$
```

```
# get the column "mpg"
mtcars$mpg
```

```
$°
```

```
$
```

```
$RStudio° $ °
```

```
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
```

```
$ R° $ °
```

```
$
```

```
# 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
```

```
°
```

```
mtcars[1, ] # first row
mtcars[-1, ] # everything but the first row
mtcars[-(1:10), ] # everything except the first 10 rows
```

<

```
# 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, ]
```

```
# 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

data.frames subset() transform() with() within() ◦

subset() data.frame

```
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
```

cyl == 6 mpg hp ◦ []

```
mtcars[mtcars$cyl == 6, c("mpg", "hp")]
```

transform() data.frame ◦ mpg2 mpg^2 mtcars data.frame

```
mtcars <- transform(mtcars, mpg2 = mpg^2)
```

...

with() within() data.frame\$[] ◦

airquality data.frame /

```
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
})

```

◦ ◦ data.frame ◦ ◦ :13◦

```

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"

```

◦

```

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

```

df1df2“x”◦ strclass◦

```

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

```

df1data.frame23“x”“y”◦ “x”“y”◦ ◦ stringsAsFactors

```

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”◦ “”◦ data.frame◦ data.frame(x = 1:3, y = 1:4)◦

R◦

```

mydataframe <- iris
str(mydataframe)

```

## do.call

do.call ◦ ◦

```
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
```

◦

```
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

data.frame data.frames RDBMS data.frames data.frames ◦ ◦

- stringsAsFactors FALSE ◦

◦

```
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)
})
```

columnsrows°

```
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
```

```
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       |

```
price_Elasticity > 0total_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        |

<https://riptutorial.com/zh-TW/r/topic/438/>

## Examples

### N-gram Word

tmWeb

```
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", "lifevantage", "protandim", "truescience", "company", "fiscal",
"nasdaq")

ds0.1g <- 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 RColorBrewer

1.

n-gram TDM n-gram

Rn-gram tm bi-gram n-gram

hacks tm tokenizer tau textcnt()

```
tokenize_ngrams <- function(x, n=3)
  return(rownames(as.data.frame(unclass(textcnt(x, method="string", n=n)))))
```

tmRWeka tokenizer

```
# BigramTokenize
BigramTokenizer <- function(x) NGramTokenizer(x, Weka_control(min = 2, max = 2))
```

1

```
# Create an n-gram Word Cloud -----
tdm.ng <- TermDocumentMatrix(ds5.1g, control = list(tokenize = BigramTokenizer))
dtm.ng <- DocumentTermMatrix(ds5.1g, control = list(tokenize = BigramTokenizer))

# Try removing sparse terms at a few different levels
tdm89.ng <- removeSparseTerms(tdm.ng, 0.89)
tdm9.ng <- removeSparseTerms(tdm.ng, 0.9)
tdm91.ng <- removeSparseTerms(tdm.ng, 0.91)
tdm92.ng <- removeSparseTerms(tdm.ng, 0.92)

notsparse <- tdm91.ng
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)

```



Hack-R. ◦

<https://riptutorial.com/zh-TW/r/topic/3579/>

# 83:

## Examples

### aov

aov() ◦

◦ aov() lm() Wilkinson-Rogers  $y \sim f y f$  ◦  $f$  aov() ANOVA ◦

aov() | ◦ ◦ ◦ ◦

||||| ◦ || ◦ || ◦

Type III of Squares ◦ III ◦

|||| Anova() ◦

---

mtcars ◦

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

### ANOVA

```
summary(mtCarsAnovaModel)
```

lm() ◦

```
coefficients(mtCarsAnovaModel)
```

### Anova

/|||| ◦ carAnova() ◦ || ◦ III ◦

Anova() lm() ◦

---

mtcars Type II Type III ◦

```
> 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

```

<https://riptutorial.com/zh-TW/r/topic/3610/>



---

# 84:

R;?Dates ?DateTimeClasses ?difftime"" °

---

- [POSIXct](#)

`POSIXct`1970-01-01 00:00:00 UTC `UNIX`° `Sys.Time()`°

- [POSIXlt](#)

° `strptime`°

- [Date](#)°

---

`POSIXct`tidyverse `UNIX`° `POSIXlt`°

```
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$hour # Hour 0-23
posixlt$mday # 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
```

- 
- `data.table` `IDateLTime`
- `fasttime`
- [lubridate](#)
- `nanotime`

## Examples

### R

```
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()` Olson / IANA

```
str(OlsonNames())
## chr [1:589] "Africa/Abidjan" "Africa/Accra" "Africa/Addis_Ababa" "Africa/Algiers"
"Africa/Asmara" "Africa/Asmera" "Africa/Bamako" ...
```

```
eom <- function(x, p=as.POSIXlt(x)) as.Date(modifyList(p, list(mon=p$mon + 1, mday=0)))
```

```
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
>
```

```
> eom('2000-01-01')
[1] "2000-01-31"
```

```
date <- as.Date("2017-01-20")
> as.POSIXlt(cut(date, "month"))
[1] "2017-01-01 EST"
```

num° mondate

```
moveNumOfMonths <- function(date, num) {  
  as.Date(mondate(date) + num)  
}
```

°

```
> moveNumOfMonths("2017-10-30", -1)  
[1] "2017-09-30"
```

```
> moveNumOfMonths("2017-10-30", -2)  
[1] "2017-08-30"
```

```
> moveNumOfMonths("2017-02-28", 2)  
[1] "2017-04-30"
```

24°

```
> moveNumOfMonths("2016-11-30", 2)  
[1] "2017-01-31"  
> moveNumOfMonths("2017-01-31", -2)  
[1] "2016-11-30"
```

1130

```
> moveNumOfMonths("2017-01-30", -2)  
[1] "2016-11-30"  
> moveNumOfMonths("2016-11-30", 2)  
[1] "2017-01-31"
```

131111°

<https://riptutorial.com/zh-TW/r/topic/1157/>

# 85: POSIXct/POSIXlt

R - POSIXct/POSIXlt - ?DateTimeClasses ◦

POSIXct◦

- lubridate

## Examples

```
# 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"
```

?strptime◦

POSIXct/POSIXlt;"".◦

```
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"
```

◦

```
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"
```

?strptime ◦

- ◦
- 0◦
- tz◦
- tz◦
  - CST"CST6CDT"America/Chicago"
- - R OlsonNames()
  - R system("cat \$R\_HOME/share/zoneinfo/zone.tab") system("cat \$R\_HOME/share/zoneinfo/zone.tab")
- IANA
  - tz
  - IANA TZ2016e

## /POSIXct

```
## 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"
```

as.difftime◦

```
as.POSIXct("2016-01-01") +
  as.difftime(3, units="hours") +
  as.difftime(14, units="mins") +
  as.difftime(15, units="secs")
# [1] "2016-01-01 03:14:15 AEDT"
```

/difftime()°

```
# using POSIXct objects
difftime(
  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() seq °

**POSIXct****POSIXlt** <https://riptutorial.com/zh-TW/r/topic/9027/-posixctposixlt->

# 86:

statsglm() ◦ [CRAN](#) ◦

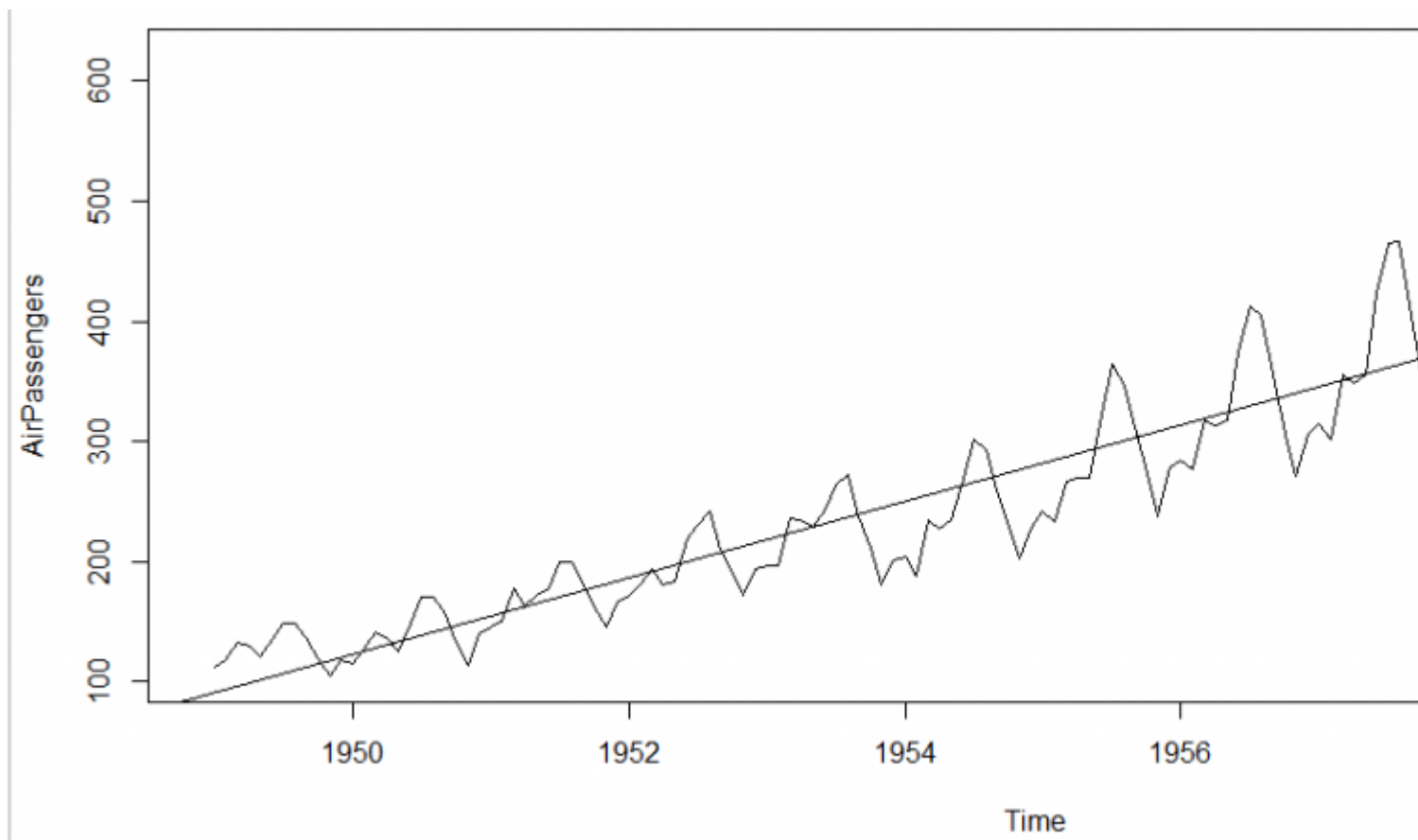
## Examples

```
data(AirPassengers)
class(AirPassengers)
```

1 “ts”

### EDA

```
plot(AirPassengers) # plot the raw data
abline(reg=lm(AirPassengers~time(AirPassengers))) # fit a trend line
```



### EDA

```
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  |

```

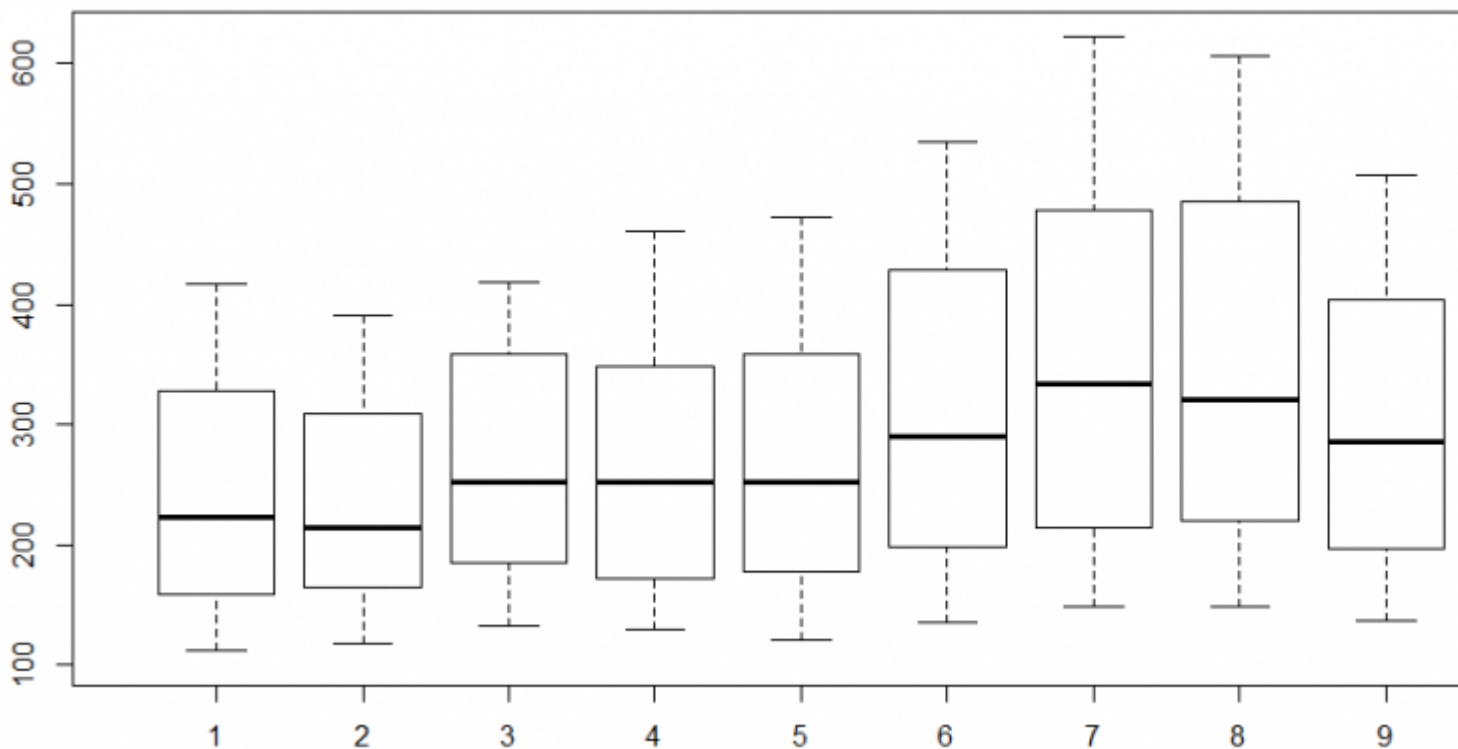
1952 1 2 3 4 5 6 7 8 9 10 11 12
1953 1 2 3 4 5 6 7 8 9 10 11 12
1954 1 2 3 4 5 6 7 8 9 10 11 12
1955 1 2 3 4 5 6 7 8 9 10 11 12
1956 1 2 3 4 5 6 7 8 9 10 11 12
1957 1 2 3 4 5 6 7 8 9 10 11 12
1958 1 2 3 4 5 6 7 8 9 10 11 12
1959 1 2 3 4 5 6 7 8 9 10 11 12
1960 1 2 3 4 5 6 7 8 9 10 11 12

```

```

boxplot(AirPassengers~cycle(AirPassengers)) #Box plot across months to explore seasonal
effects

```



## ts

ts° tsARIMA° 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))
```

ts

```
#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)
```

<https://riptutorial.com/zh-TW/r/topic/2701/>

# 87: R

## Examples

### Windows

WindowsR.

R.

RR. c:\stats\R. R.tm.

- 
- miniCRAN

- use `packageStatus`

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

`installr` `gui` `guiRguiRStudio`.

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

<https://www.r-statistics.com/tag/installr/> Windows [https //www.r-statistics.com/2015/06/--R-/](https://www.r-statistics.com/2015/06/--R-/)

o R.

R <https://riptutorial.com/zh-TW/r/topic/4088/r>

---

# 88: R

◦ R ◦ R ◦

## Examples

R

<https://cran.r-project.org/> ◦ ◦ ◦

### installr Package

**installr**

RStudio

```
install.packages("installr")  
library("installr")  
updateR()
```



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(library(in

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 executed.
Be sure to click 'next' as needed...
trying URL 'https://cran.rstudio.com/bin/windows/base/R-3.4.1-win.exe'
Content type 'application/x-msdos-program' length 78086510 bytes (74.5 MB)
downloaded 74.5 MB
```

Select Setup Language



Select the language to use during installation:

English

OK

“”。

RR。 R.



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/R40.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?



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

&gt; 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/
Content type 'application/x-msdos-program' length 7808
downloaded 74.5 MB
```

Question



Once your packages are copied to the new R installation, do you wish to KEEP the packages from the old installation?  
(if you choose 'NO' - you will erase your packages)

Rprofile.site



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/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?

R.



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(li

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 ex
  Be sure to click 'next' as needed...
trying URL 'https://cran.rstudio.com/bin/windows/base/R-3
Content type 'application/x-msdos-program' length 7808651
downloaded 74.5 MB
```

Question



Do you wish to update your packages in

Ye

R。

R

R

version

R <https://riptutorial.com/zh-TW/r/topic/10729/r>



# 89:

◦ - ◦ ◦

## Examples

### barplot

xy◦ ◦

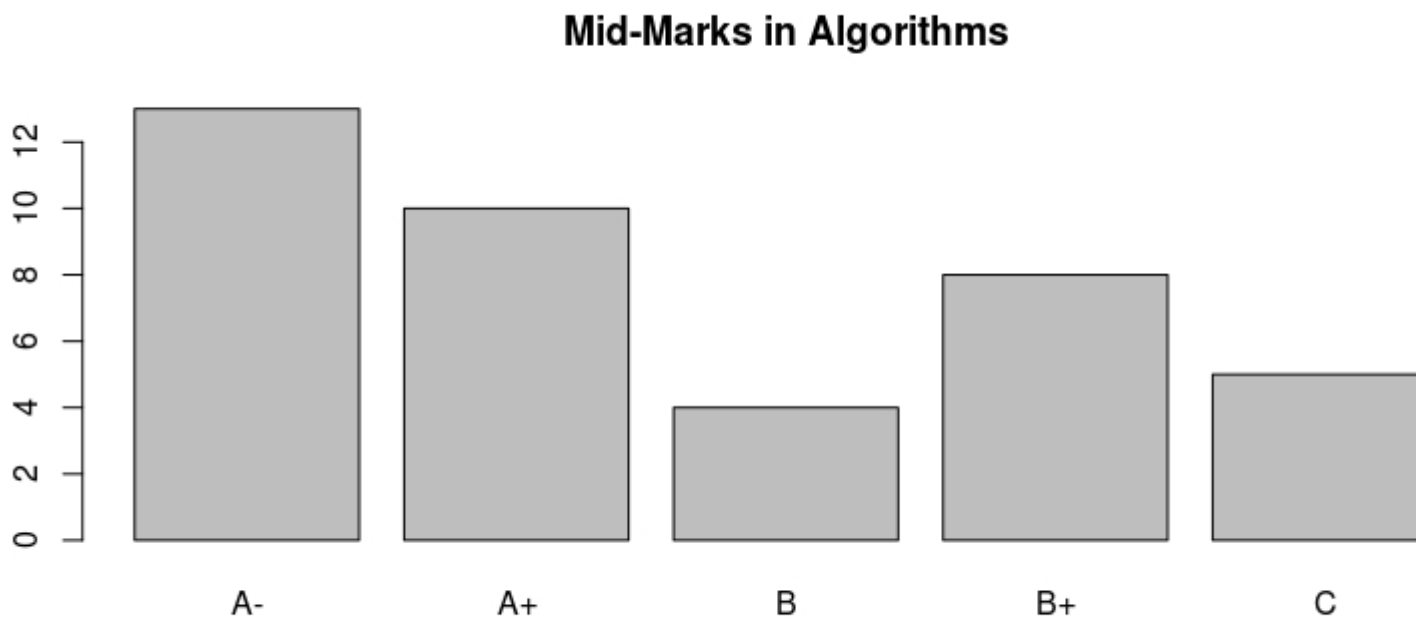
barplot() R◦ barplot()◦ Rheights◦ ◦

barplot()

```
> 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+"
>
```

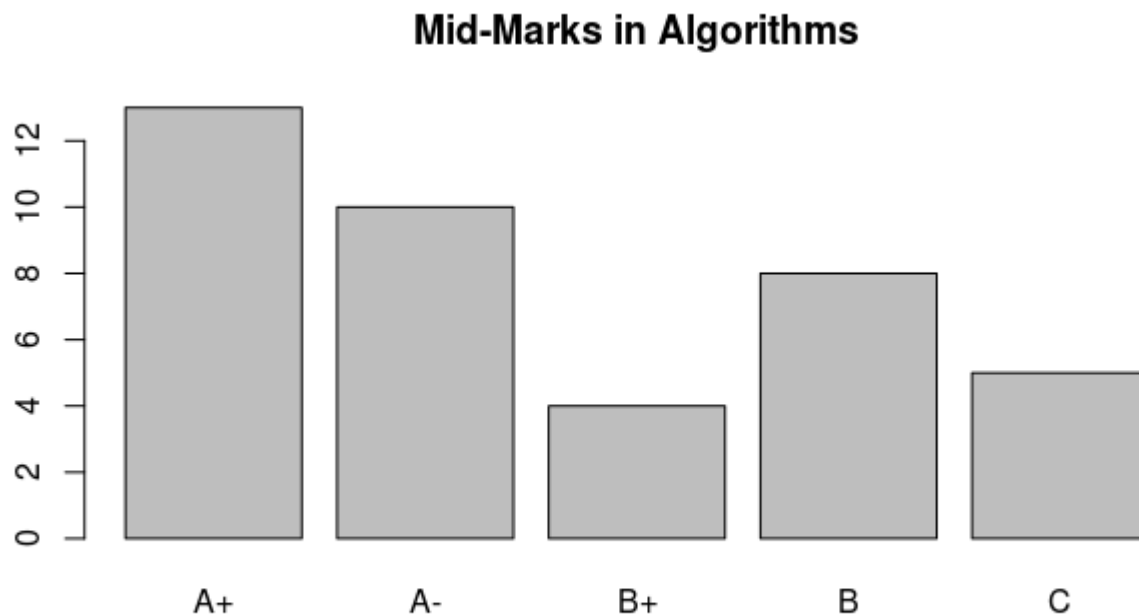
### Marks

```
> barplot(table(Marks),main="Mid-Marks in Algorithms")
```



`barplot`lexicographical orderX° names.arg °

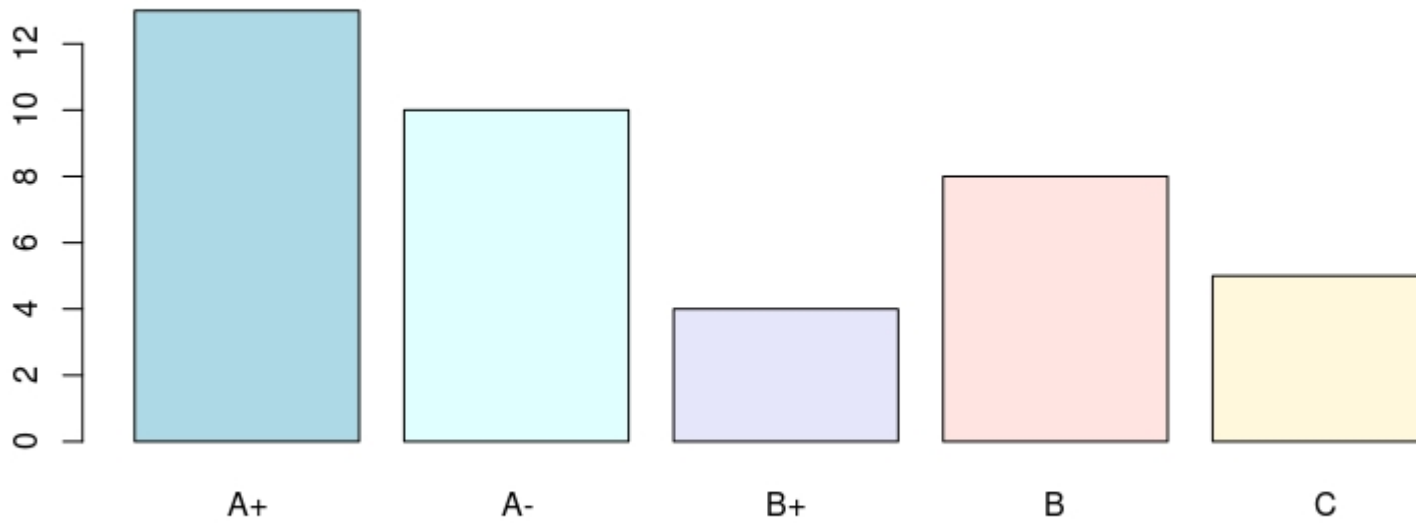
```
# plot to the desired horizontal axis labels  
> barplot(table(Marks),names.arg=grades ,main="Mid-Marks in Algorithms")
```



col=°

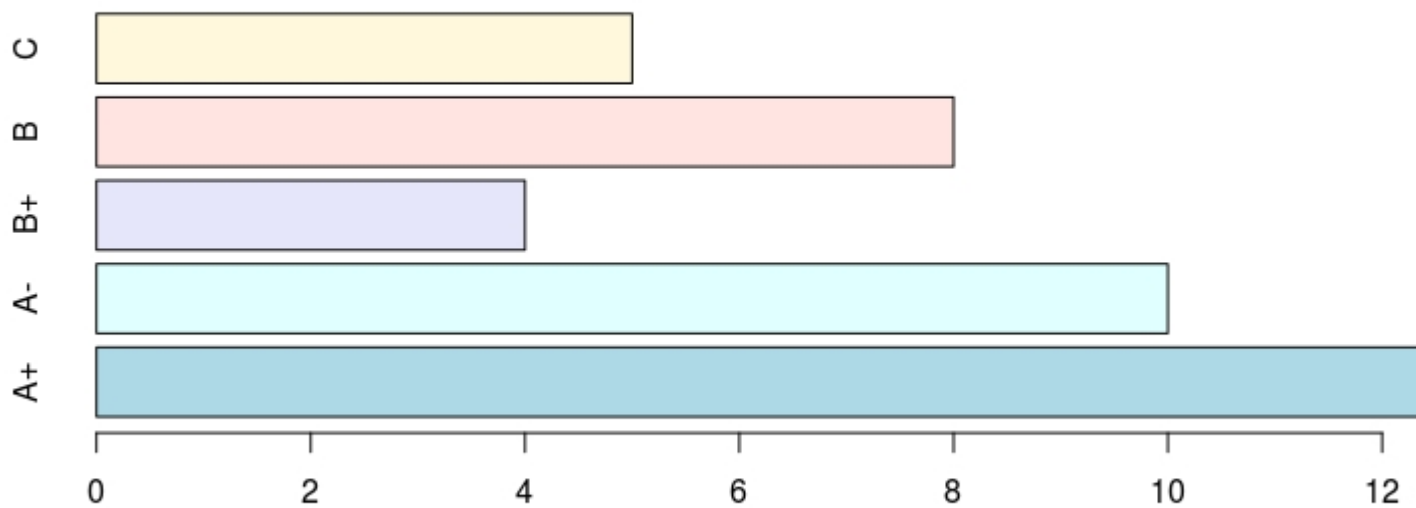
```
> barplot(table(Marks),names.arg=grades,col = c("lightblue",  
"lightcyan", "lavender", "mistyrose", "cornsilk"),  
main="Mid-Marks in Algorithms")
```

## Mid-Marks in Algorithms



```
> barplot(table(Marks),names.arg=grades,horiz=TRUE,col = c("lightblue",  
  "lightcyan", "lavender", "mistyrose", "cornsilk"),  
  main="Mid-Marks in Algorithms")
```

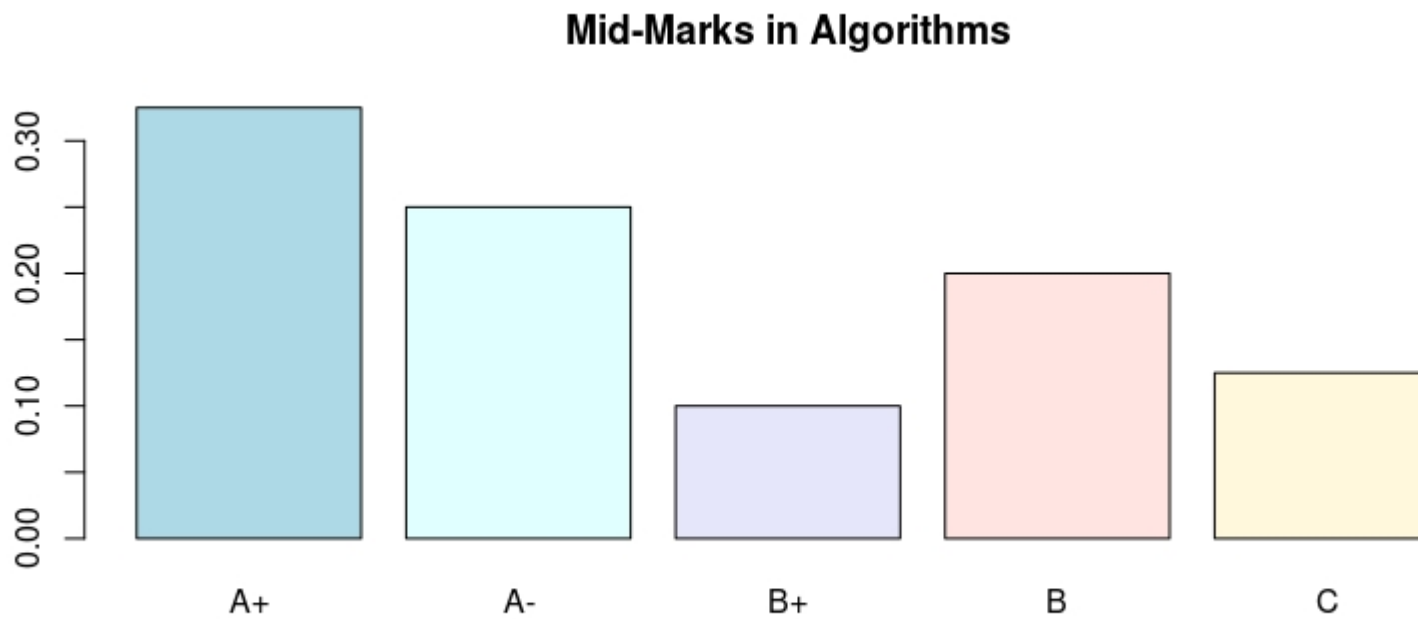
## Mid-Marks in Algorithms



y

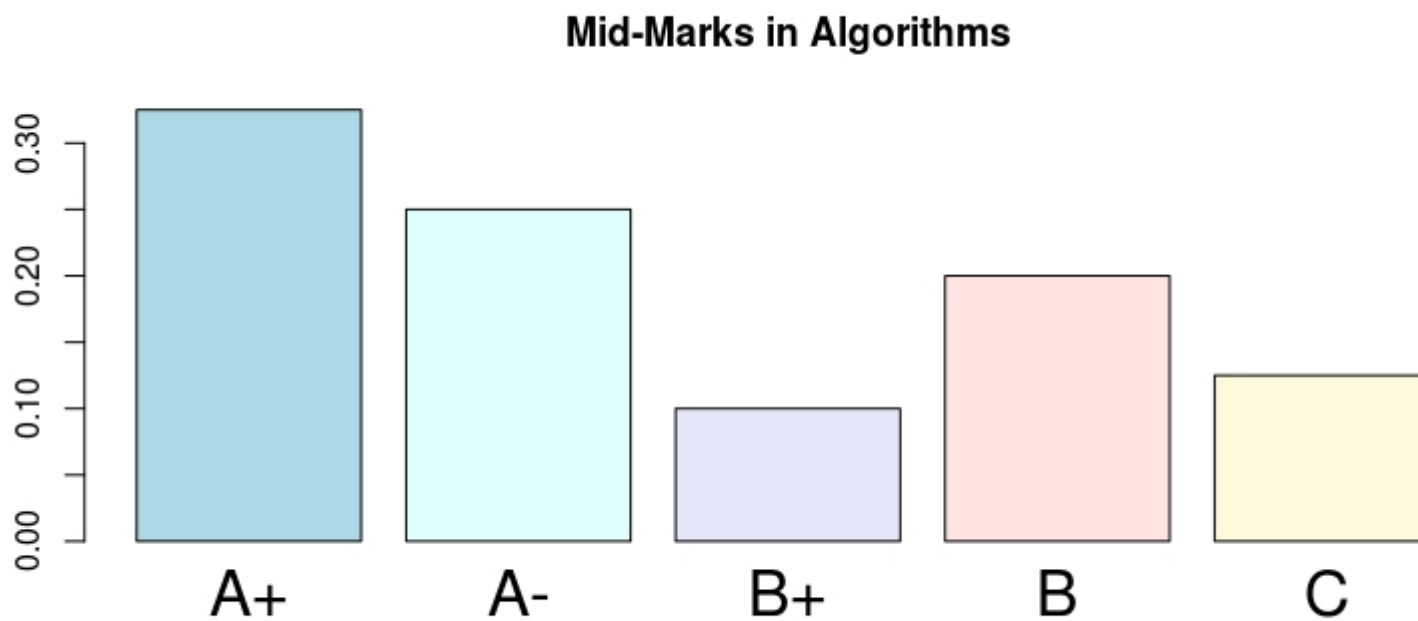
```
> barplot(prop.table(table(Marks)),names.arg=grades,col = c("lightblue",  
  "lightcyan", "lavender", "mistyrose", "cornsilk"),
```

```
main="Mid-Marks in Algorithms")
```



`cex.names` X°

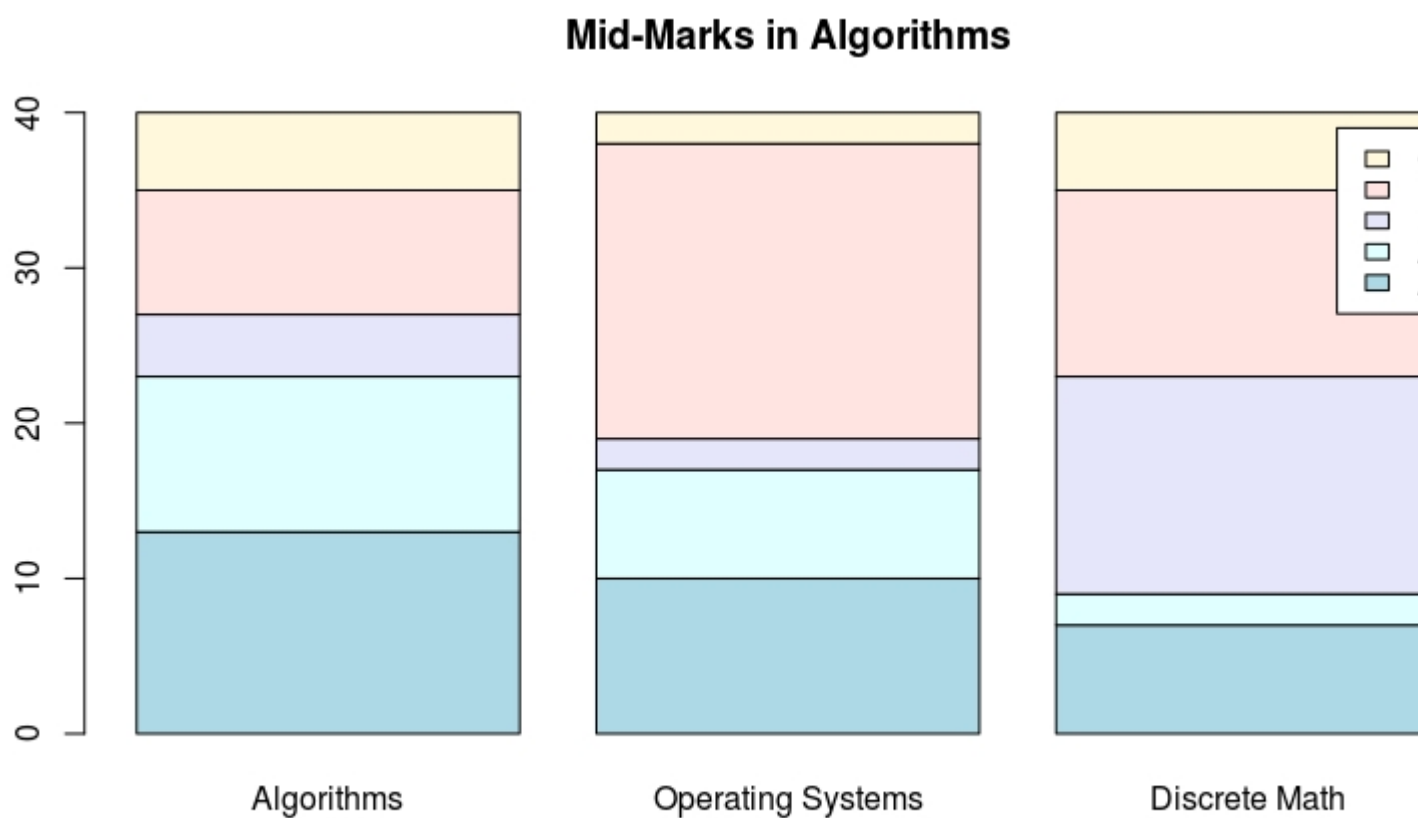
```
> barplot(prop.table(table(Marks)),names.arg=grades,col = c("lightblue",  
"lightcyan", "lavender", "mistyrose", "cornsilk"),  
main="Mid-Marks in Algorithms",cex.names=2)
```



barplot()heights◦ ◦

```
> 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
```

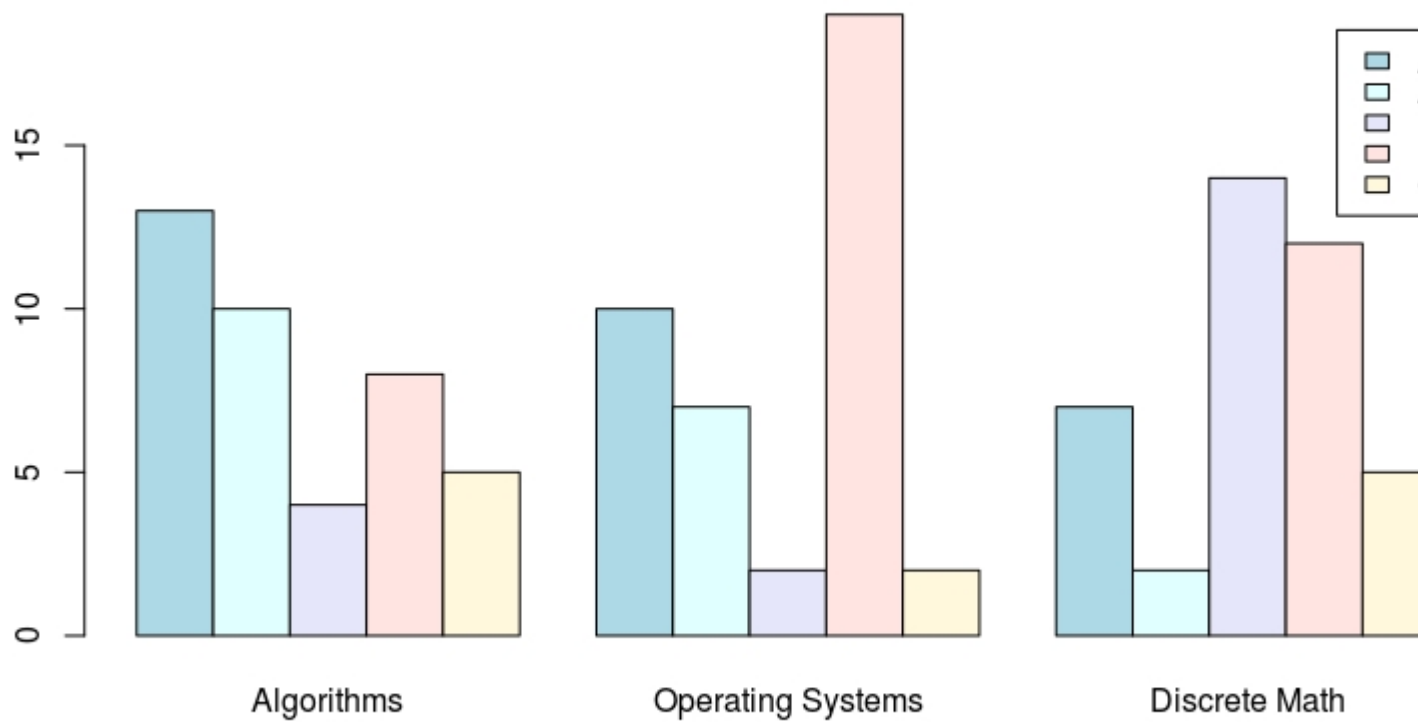
```
> barplot(gradTab,col = c("lightblue","lightcyan",
  "lavender", "mistyrose", "cornsilk"),legend.text = grades,
  main="Mid-Marks in Algorithms")
```



besides

```
> barplot(gradTab,beside = T,col = c("lightblue","lightcyan",
  "lavender", "mistyrose", "cornsilk"),legend.text = grades,
  main="Mid-Marks in Algorithms")
```

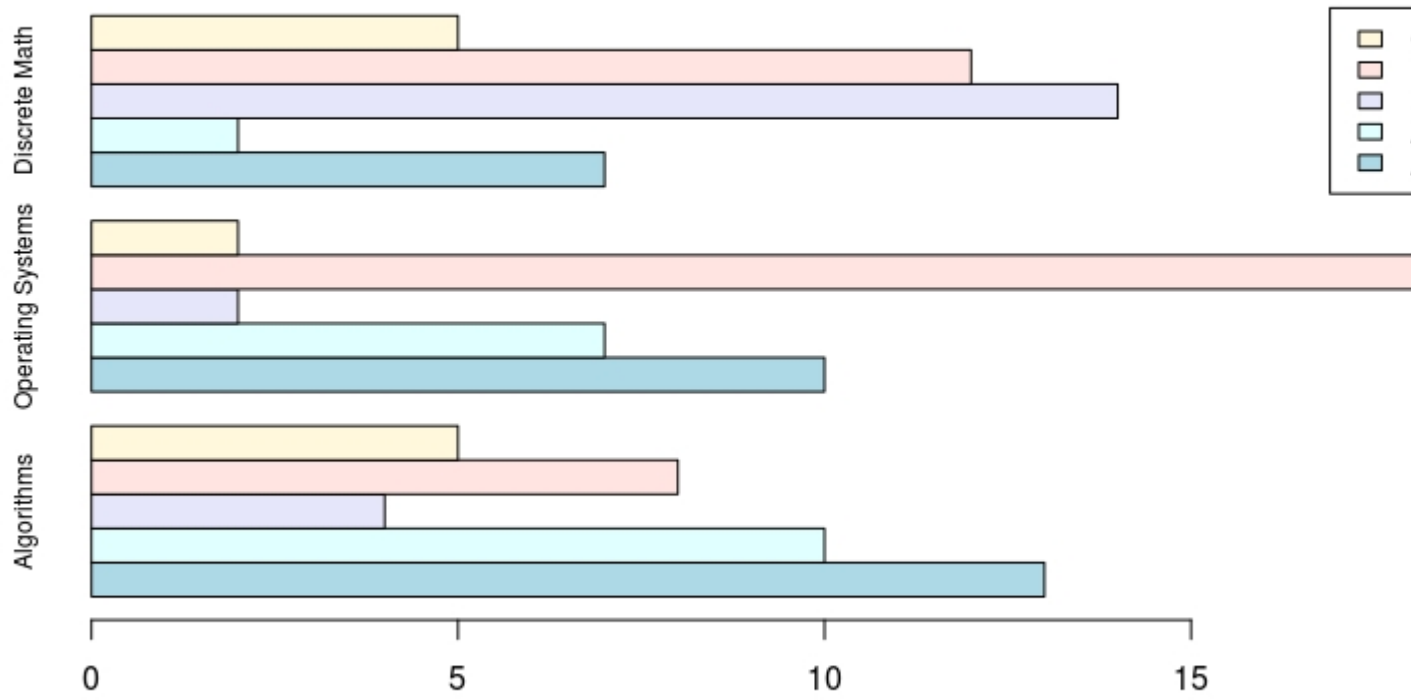
## Mid-Marks in Algorithms



horiz=T

```
> 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



<https://riptutorial.com/zh-TW/r/topic/8091/>

---

# 90:

◦

- `grep` "optional\_args"
- `grep1` "optional\_args"
- `gsub` "group1group2" "\ group" "subject"

---

`\1 \1 patternreplacementsubgsub` ◦

`grep` `sub` `regexpr` `Perl` `PCRE` `lookarounds` ◦ `perl=TRUE` ◦ [R](#) ◦

- 
- [stringi](#)
  - `stringr`

## Examples

```
# example data
test_sentences <- c("The quick brown fox quickly", "jumps over the lazy dog")
```

```
sub("brown","red", test_sentences)
#[1] "The quick red fox quickly"      "jumps over the lazy dog"
```

"fast" "fastly" ◦

```
sub("quick", "fast", test_sentences)
#[1] "The fast red fox quickly"        "jumps over the lazy dog"
```

`subgsub`

```
gsub("quick", "fast", test_sentences)
#[1] "The fast red fox fastly"         "jumps over the lazy dog"
```

◦

```
# example data
test_sentences <- c("The quick brown fox", "jumps over the lazy dog")
```

---

`grep1()` ◦ `TRUE / FALSE` "◦"



“fox”。

```
grepl("fox", test_sentences)
#[1] TRUE FALSE
```

grep ° “fox”。

```
grep("fox", test_sentences)
#[1] 1
```

```
# 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"fixed = TRUEgrepgrepl °

```
grep("fox", test_sentences, fixed = TRUE)
#[1] 1
```

grep **with** invert = TRUE ;-grep(...)!grepl(...)

grepl(pattern, x)grep(pattern, x) x pattern° pattern[1]x[1] pattern[2]x[2]°

greplTRUEFALSE ° ° summary

```
# example data
test_sentences <- c("The quick brown fox", "jumps over the lazy dog")

# find matches
matches <- grepl("fox", test_sentences)

# overview
summary(matches)
```

°

PCRE<sub>g</sub> °

R

- sub(pattern, replacement, text) **pattern**
- gsub(pattern, replacement, text) **sub**

- `regexr(pattern, text)` **pattern**
- `gregexpr(pattern, text)` ◦

```
set.seed(123)
teststring <- paste0(sample(letters, 20), collapse="")

# teststring
#[1] "htjuwakqxpgrsbncvyo"
```

```
sub("[aeiou]", "** HERE WAS A VOWEL** ", teststring)
#[1] "htj ** HERE WAS A VOWEL** wakqxpgrsbncvyo"

gsub("[aeiou]", "** HERE WAS A VOWEL** ", teststring)
#[1] "htj ** HERE WAS A VOWEL** w ** HERE WAS A VOWEL** kqxpgrsbncv ** HERE WAS A VOWEL** **
HERE WAS A VOWEL** "
```

```
regexr("[^aeiou][aeiou]+", teststring)
#[1] 3
#attr(,"match.length")
#[1] 2
#attr(,"useBytes")
#[1] TRUE
```

## 23 ju

```
gregexpr("[^aeiou][aeiou]+", teststring)
#[[1]]
#[1] 3 5 19
#attr(,"match.length")
#[1] 2 2 2
#attr(,"useBytes")
#[1] TRUE
```

`regmatches` **regexr** ◦

```
matches <- gregexpr("[^aeiou][aeiou]+", teststring)
regmatches(teststring, matches)
#[[1]]
#[1] "ju" "wa" "yo"
```

```
teststring2 <- "this is another string to match against"
regmatches(teststring2, matches)
#[[1]]
#[1] "is" "i" "ri"
```

**Perl lookarounds** `perl=TRUE` ◦

`grepl("fox", test_sentences)` `grepl("fox", test_sentences)` ◦ ◦

`perl = TRUE` ◦ `fixed = TRUE` ◦

```
# example data
test_sentences <- c("The quick brown fox", "jumps over the lazy dog")

grepl("fox", test_sentences, perl = TRUE)
#[1] TRUE FALSE
```

◦ grepl◦

```
searchCorpus <- function(corpus, pattern) {
  return(tm_index(corpus, FUN = function(x) {
    grepl(pattern, x, ignore.case = TRUE, perl = TRUE)
  })))
}
```

<https://riptutorial.com/zh-TW/r/topic/1123/>

# 91:

## Examples

Random Forest Breiman L. 2001. Random Forests. 455-532. Breiman randomForest Fortran R.

factor R iris

```
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
```

|          |            |
|----------|------------|
| X        |            |
| ŷ        | ◦ factor ◦ |
| ntree    | CART       |
| do.trace | i          |

<https://riptutorial.com/zh-TW/r/topic/8326/>

# 92:

R. ◦

Comprehensive R Archive NetworkCRAN ◦

## Examples

dplyr

```
help(package = "dplyr")
```

◦

dplyr

```
data(package = "dplyr")
```

◦

dplyr

```
library(dplyr)  
ls("package:dplyr")
```

◦ ◦

```
## 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'
```

```
search()
```

```
(.packages())
```

<https://riptutorial.com/zh-TW/r/topic/7408/>

# 93:

“” R?regex [Regex](#) SO/“”R-regexpattern

- "[AB]"AB.
- "[[:alpha:]]"
- "[[:lower:]]" " [az]"ú
- "[[:upper:]]" " [AZ]"Ú
- "[[:digit:]]" "0,1,2...9" [0-9]"

+ \*? - + \*0?01

- "^..."
- "...\$"

R

- R"\"RR\s \\s
- RUTF-8U[\U{1F600}][\U1F600]Rubyu

[reg101](#)R-script

[R Programming wikibook](#)

## Examples

```
string <- ' some text on line one;
and then some text on line two '
```

“” gsub

R 3.2.0

```
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"
```

## R 3.2.0

```
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      "
```

## R 3.2.0

```
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"
```

```
gsub(pattern = "\\s",
     replacement = "",
     x = string)

[1] "sometextonlineone;andthensometextonlinetwo"
```

`\t \r\n` ◦

## “YYYYMMDD”

YYYYMMDD 20170101\_results.csv ◦

```
\\d{4}(0[1-9]|1[012]) (0[1-9]|12)[0-9] | 3[01]
```

0000-9999 01-1201-31 ◦

```
> 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
```

20170229 **2017**.

```
> grepl("\\d{4}(0[1-9]|1[012])(0[1-9]|[12][0-9]|3[01])", "20170229")
[1] TRUE
```

```
is.Date <- function(x) {return(!is.na(as.Date(as.character(x), format = '%Y%m%d')))}
```

```
> is.Date(c("20170229", "20170101", 20170101))
[1] FALSE TRUE TRUE
```

regex **50Commonwealth / Territory** [www.50states.com](http://www.50states.com)

```
regex <-
"(A[LKSZR])|(C[AOT])|(D[EC])|(F[ML])|(G[AU])|(HI)|(I[DLNA])|(K[SY])|(LA)|(M[EHDAINSOT])|(N[EVHJMYCD])|(O[HA])|(P[RI])|(R[I])|(T[EX])|(U[VA])|(V[IR])|(W[YO])|(Z[ND])"
```

```
> 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
>
```

**50R-dataset** state.abb from state

```
> data(state)
> test %in% state.abb
[1] TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE
```

**50** TRUE AL, AZ, AR, FL °

```
us.phones.regex <- "^\\s*(\\+\\s*1(-?\\s+))*[0-9]{3}\\s*-?\\s*[0-9]{3}\\s*-?\\s*[0-9]{4}$"
```

+1-xxx-xxx-xxxx // +1-xxx-xxx-xx xx° - xxx xxx xxx xxxxxxxxxxxx ° +1°

```
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")
```

```
> grepl(us.phones.regex, phones.OK)
[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE
>
```



```
> grepl(us.phones.regex, phones.NOK)
[1] FALSE FALSE FALSE FALSE FALSE
>
```

- `\\s`

## R

### R"\" grep sub gsub 4

```
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"
```

◦ ◦ 4

```
gsub("\\n|   ", "\\t", x)
#[1] "a\tb" "c\t d" "e\tf"
```

## PerIPOSIX

R◦ POSIX-consistent; R\_perl = TRUE◦

/

perl = TRUE◦

- "(?<=A)B" BA"ABACADABRA""abacadabra""aBacadabra"◦

<https://riptutorial.com/zh-TW/r/topic/5748/-->

# 94:

## Examples

;

◦

```
exampleList1 <- list('a', 'b')
exampleList2 <- list(1, 2)
exampleList3 <- list('a', 1, 2)
```

str

```
str(exampleList1)
str(exampleList2)
str(exampleList3)
```

◦ []

```
# Returns List
exampleList3[1]
exampleList3[1:2]
```

[[

```
# Returns Character
exampleList3[[1]]
```

```
exampleList4 <- list(
  num = 1:3,
  numeric = 0.5,
  char = c('a', 'b')
)
```

◦

```
exampleList4[['char']]
```

\$

```
exampleList4$num
```

◦ \$

```
exampleList5 <- exampleList4[2:3]
```

```

exampleList4$num
# c(1, 2, 3)

exampleList5$num
# 0.5

exampleList5[['num']]
# NULL

```

◦

```

## 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

```

◦ list

```

l1 <- list(c(1, 2, 3), c("a", "b", "c"))
l1
## [[1]]
## [1] 1 2 3
##
## [[2]]
## [1] "a" "b" "c"

```

◦ ◦ ◦ names◦

```
names(l1)
## NULL
names(l1) <- c("vector1", "vector2")
l1
## $vector1
## [1] 1 2 3
##
## $vector2
## [1] "a" "b" "c"
```

◦

```
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"
```

“vec”“mat”◦

R◦ ;/

◦

```
# 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)
```

◦ ◦

## R1if。。

```
# 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]]
```

## data.frame。

### nmxn data.frame data.table。

```
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
```

## ◦ Azure MLR◦

```
> 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"
```

```
> paste(df$name[4], "is a", df$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"
```

◦

```
l <- list(df, number)
dataframe_container <- data.frame(out2 = as.integer(serialize(l, connection=NULL)))
```

```
#----- 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]]
```

```
> 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"
```

<https://riptutorial.com/zh-TW/r/topic/1365/>

---

# 95:

R. . . .

## Examples

### Vector1

```
set.seed(123)
Vector1 <- rnorm(20)
```

```
set.seed(123)
Vector1[sample(1:length(Vector1), 5)] <- NA
```

### is.naVector

```
Vector1 <- Vector1[!is.na(Vector1)]
```

### Vector1NA

### NAcomplete.cases

### airquality6NA

```
x <- head(airquality)
```

### Solar.RNA

```
x_no_NA <- x[complete.cases(x),]
```

### x\_no\_NANA

<https://riptutorial.com/zh-TW/r/topic/8165/>

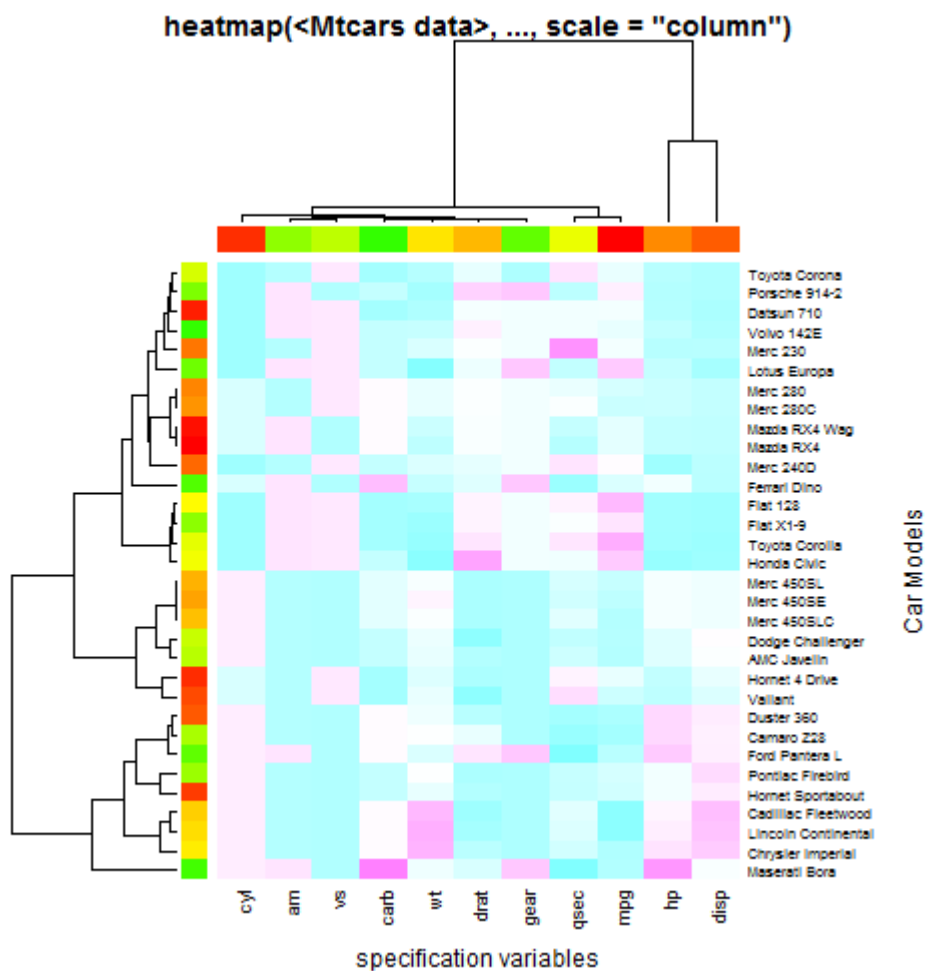
# 96:

## Examples



### 1

```
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\")")
```



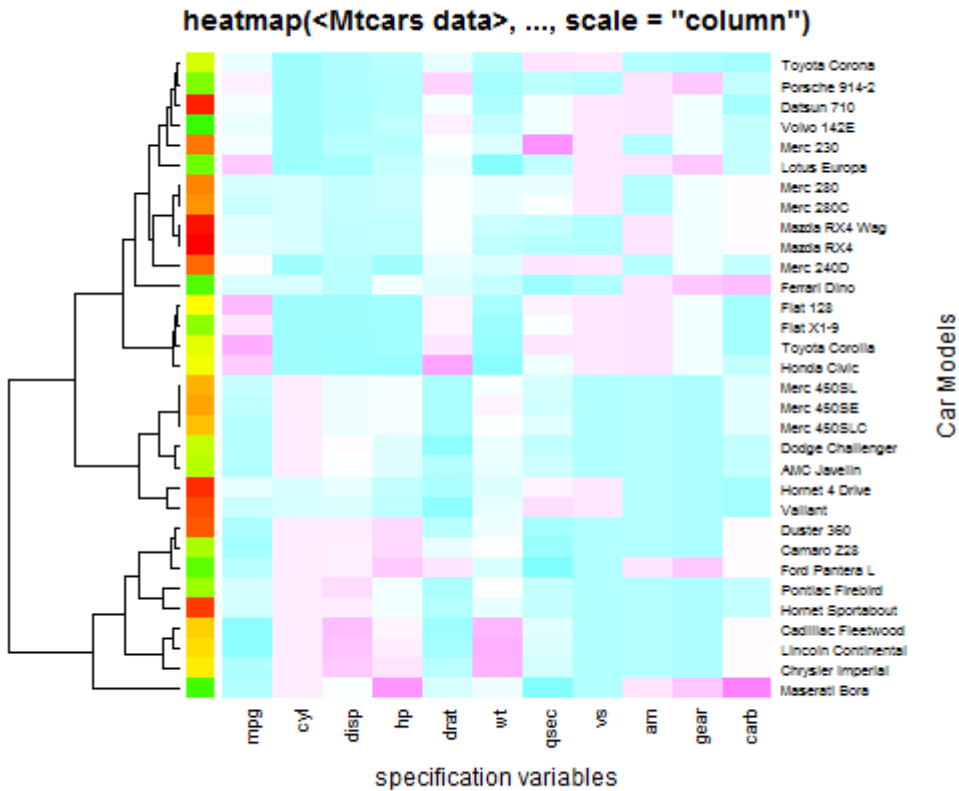
```
utils::str(hv) # the two re-ordering index vectors
# List of 4
# $ rowInd: int [1:32] 31 17 16 15 5 25 29 24 7 6 ...
# $ colInd: int [1:11] 2 9 8 11 6 5 10 7 1 4 ...
# $ Rowv : NULL
```



```
# $ Colv : NULL
```

## 2

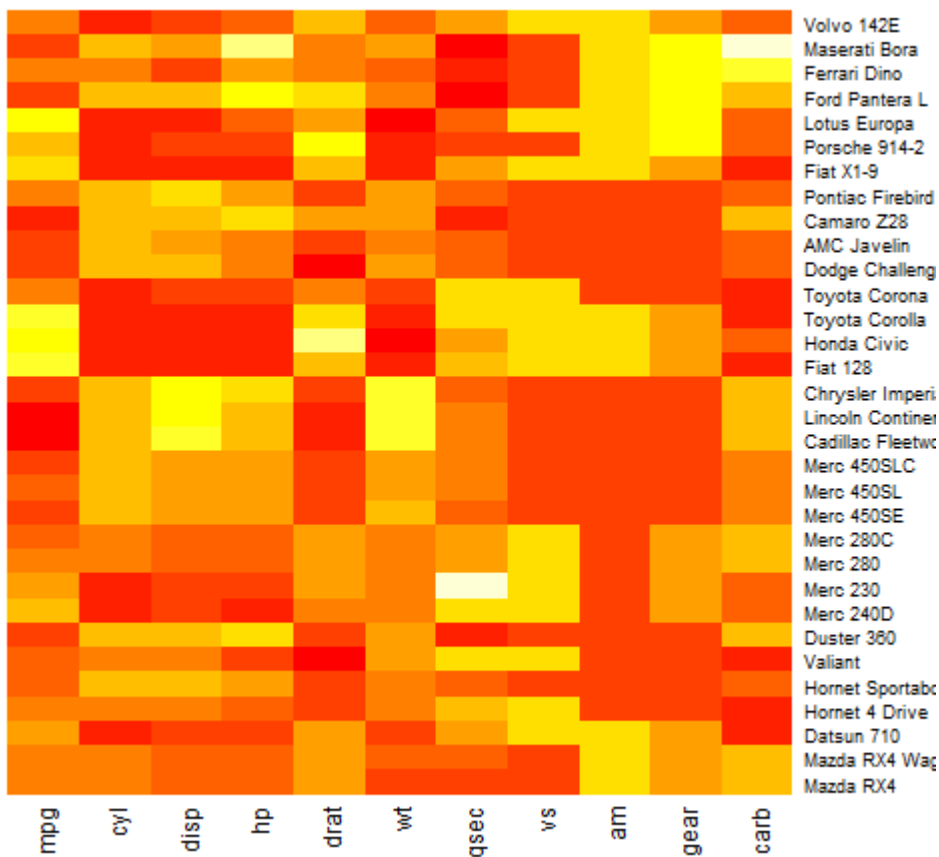
```
heatmap(x, Colv = NA, col = cm.colors(256), scale = "column",  
  RowSideColors = rc, margins = c(5,10),  
  xlab = "specification variables", ylab = "Car Models",  
  main = "heatmap(<Mtcars data>, ..., scale = \"column\")")
```



## 3“”

```
heatmap(x, Rowv = NA, Colv = NA, scale = "column",  
  main = "heatmap(*, NA, NA) ~ image(t(x))")
```

## heatmap(\*, NA, NA) ~= image(t(x))

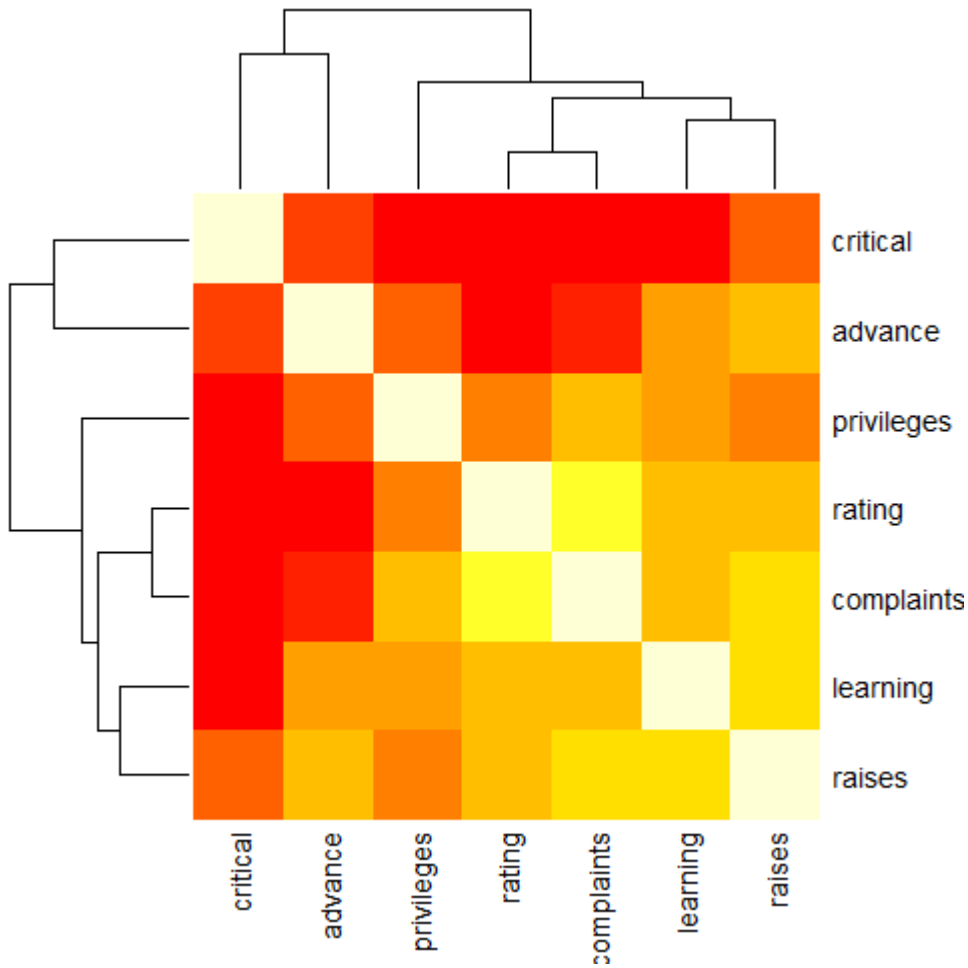


## 4

```

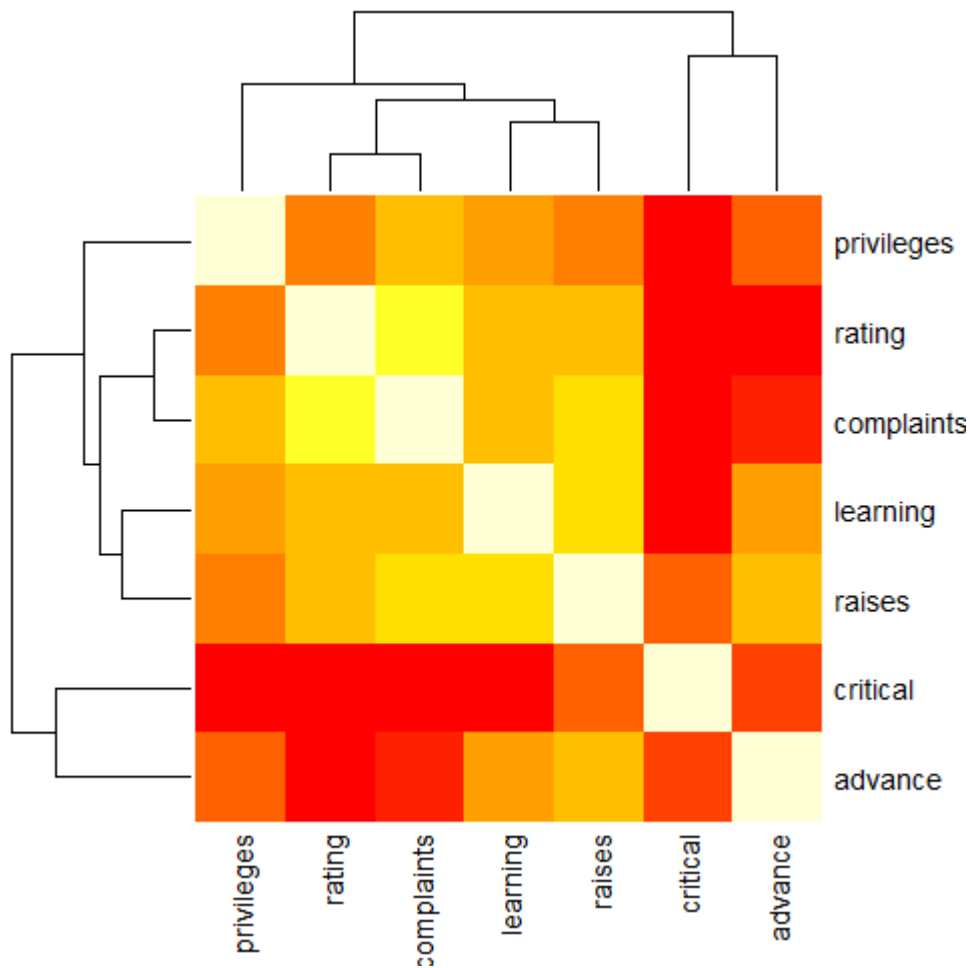
round(Ca <- cor(attendance), 2)
#           rating complaints privileges learning raises critical advance
# rating           1.00      0.83      0.43      0.62      0.59      0.16      0.16
# complaints      0.83      1.00      0.56      0.60      0.67      0.19      0.22
# privileges      0.43      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
#           r t c m p l r s c r 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))

```



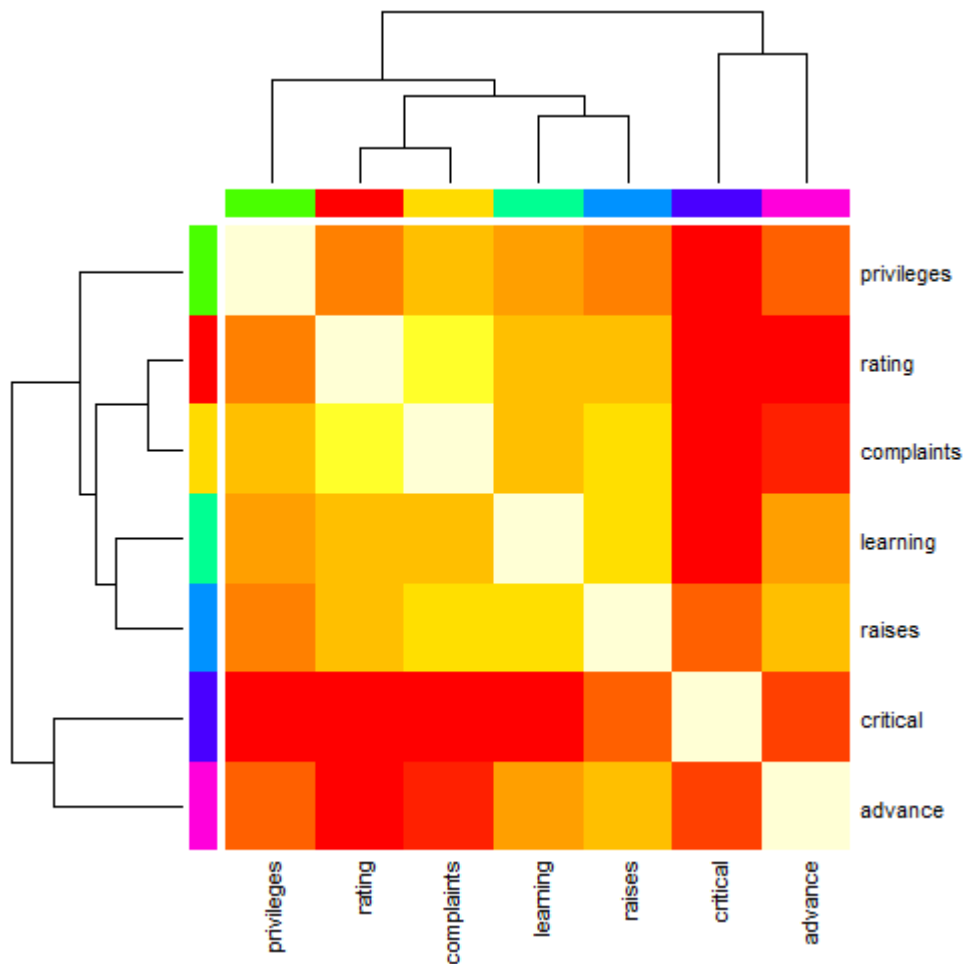
## 5

```
heatmap(Ca, Rowv = FALSE, symm = TRUE, margins = c(6,6))
```



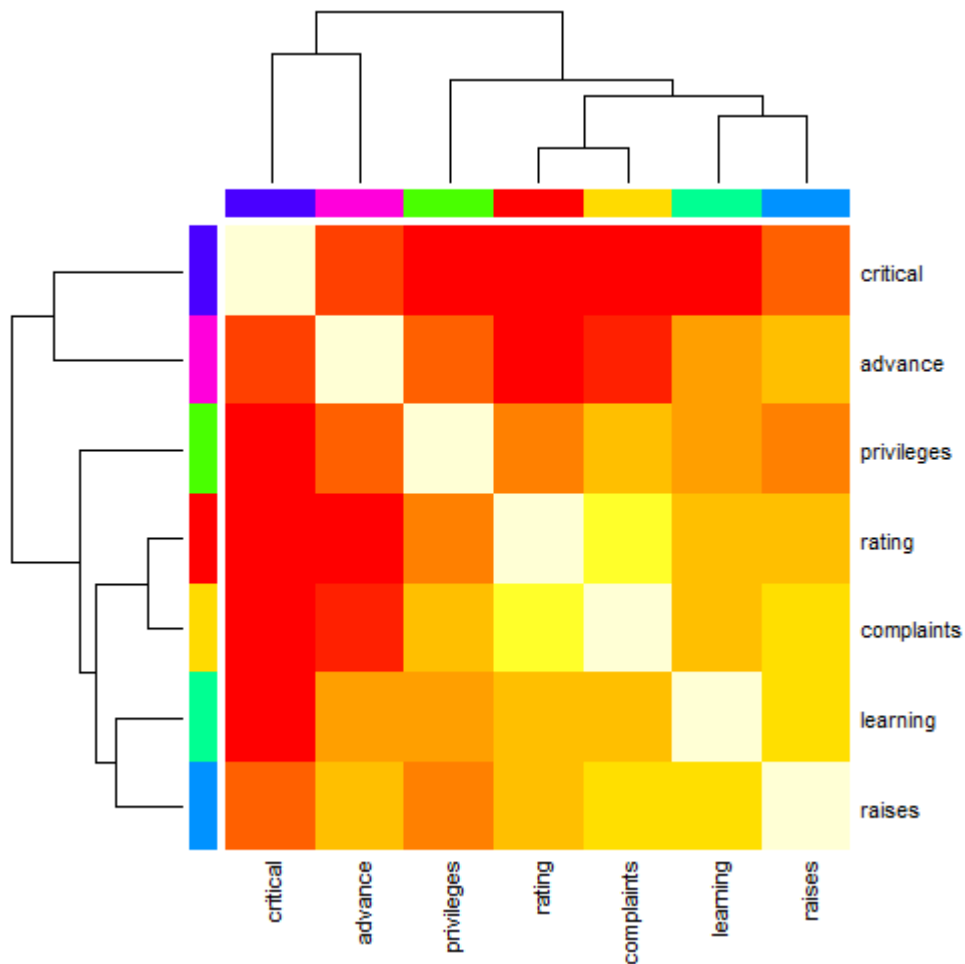
## 6

```
cc <- rainbow(nrow(Ca))
heatmap(Ca, Rowv = FALSE, symm = TRUE, RowSideColors = cc, ColSideColors = cc,
        margins = c(6,6))
```



## 7

```
heatmap(Ca,          symm = TRUE, RowSideColors = cc, ColSideColors = cc,
        margins = c(6,6))
```



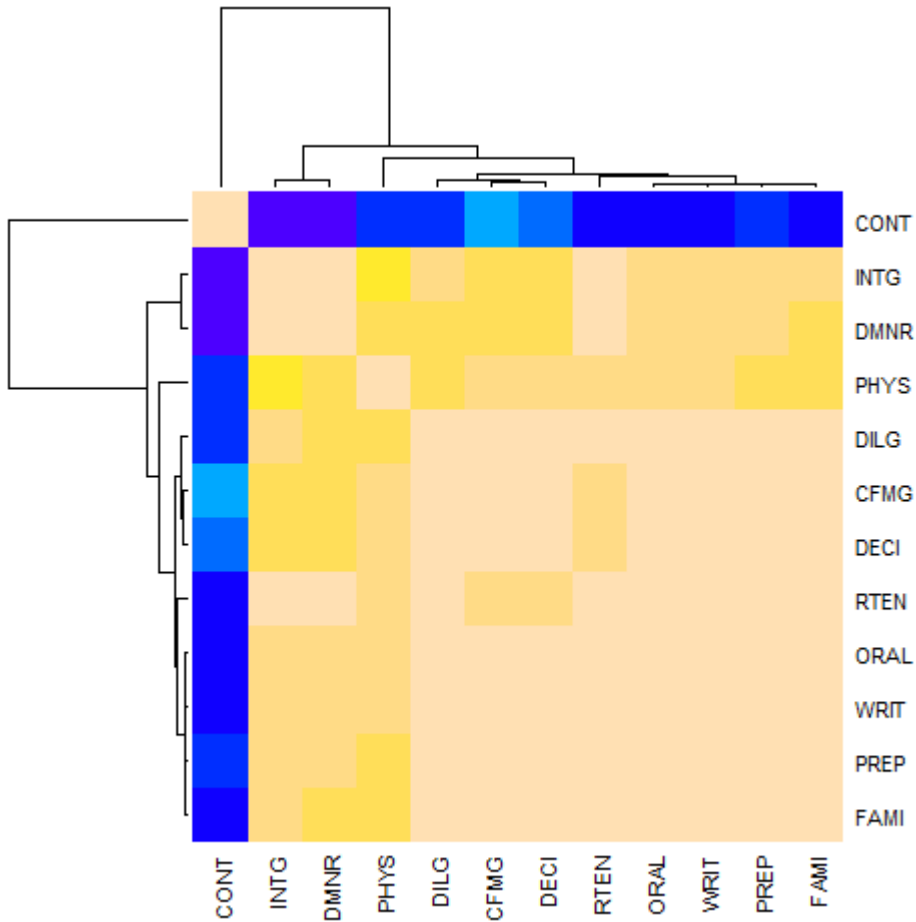
## 8cor

```

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"

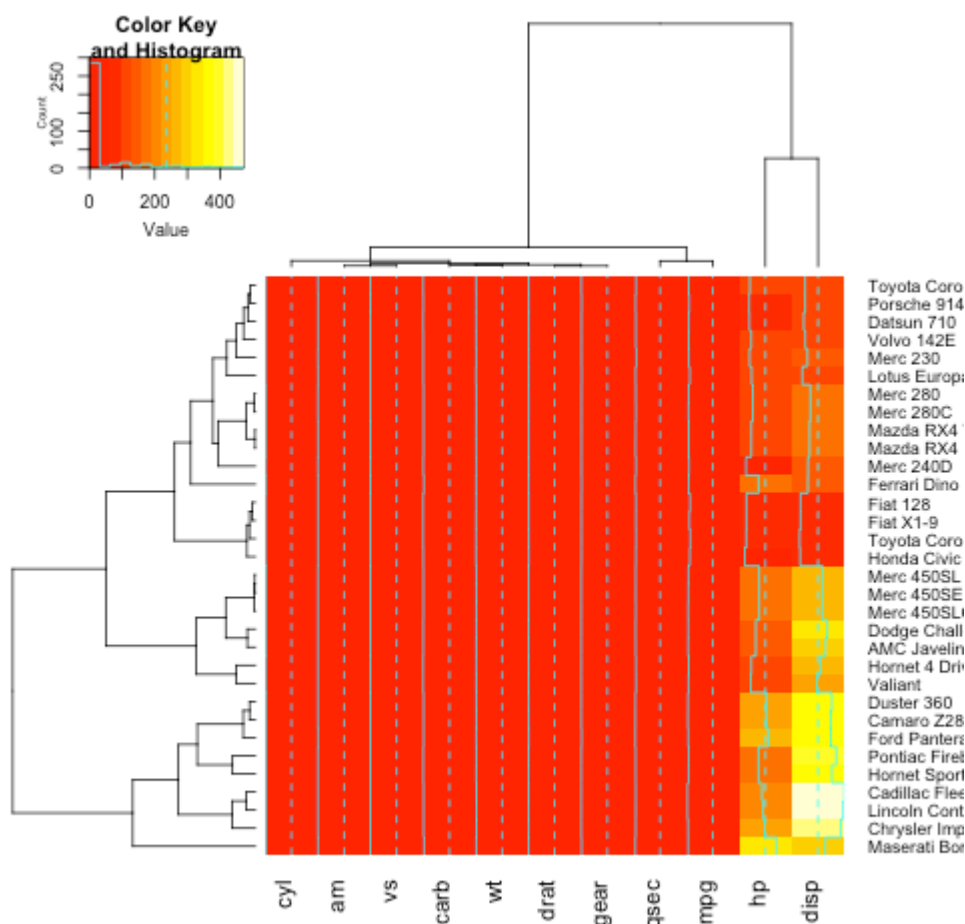
```

## heatmap

```
x <- as.matrix(mtcars)
```

heatmap.2 - heatmap

```
require(gplots)
heatmap.2(x)
```



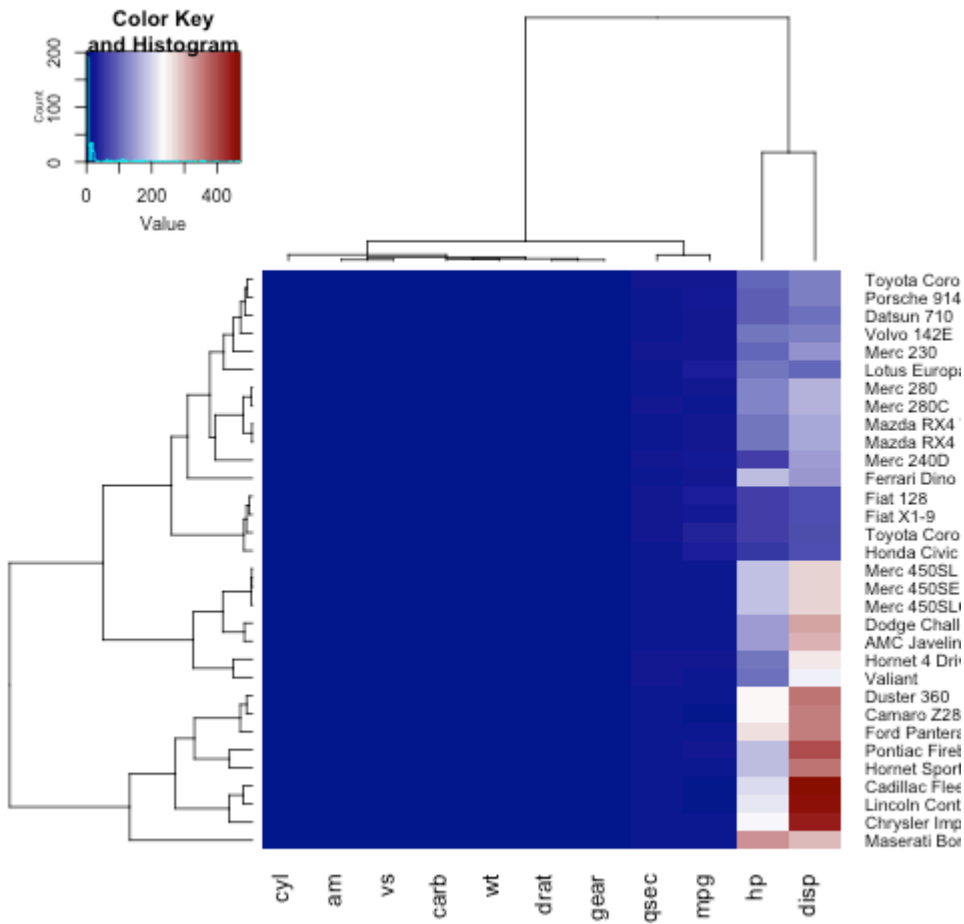
`XYmain xlabylab`

```
heatmap.2(x, main = "My main title: Overview of car features", xlab="Car features", ylab = "Car brands")
```



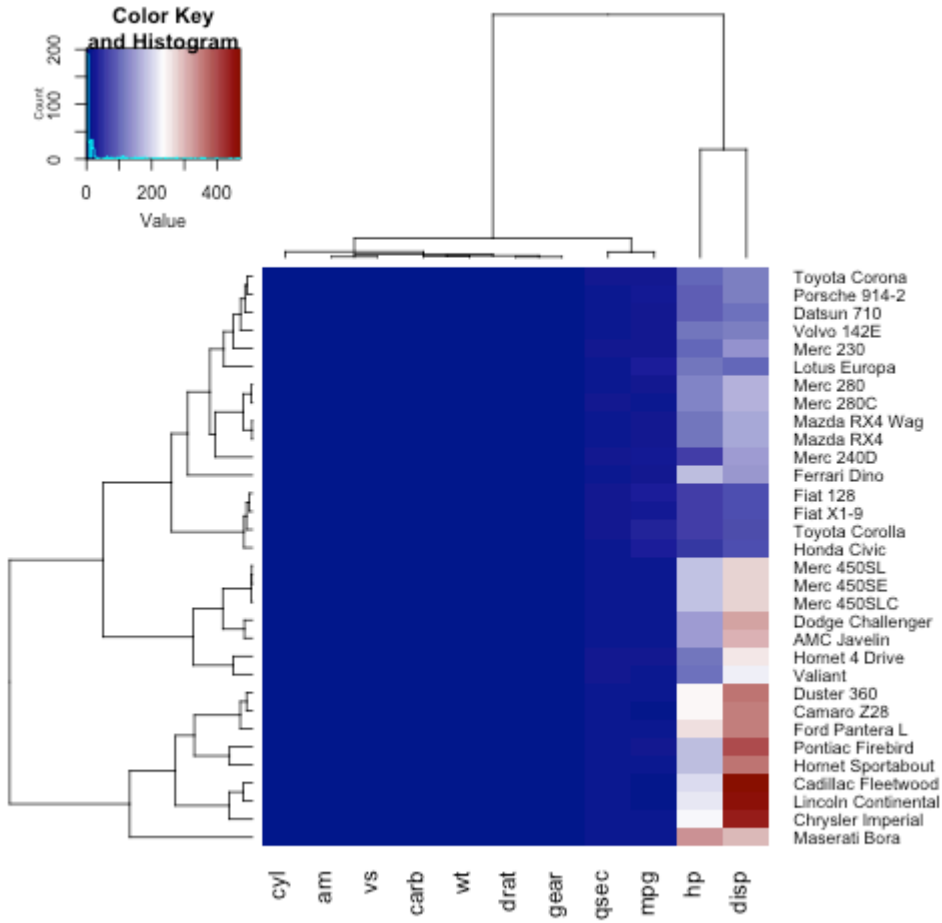
colorRampPalettecol

```
heatmap.2(x, trace="none", key=TRUE, Colv=FALSE, dendrogram = "row", col =  
colorRampPalette(c("darkblue", "white", "darkred"))(100))
```

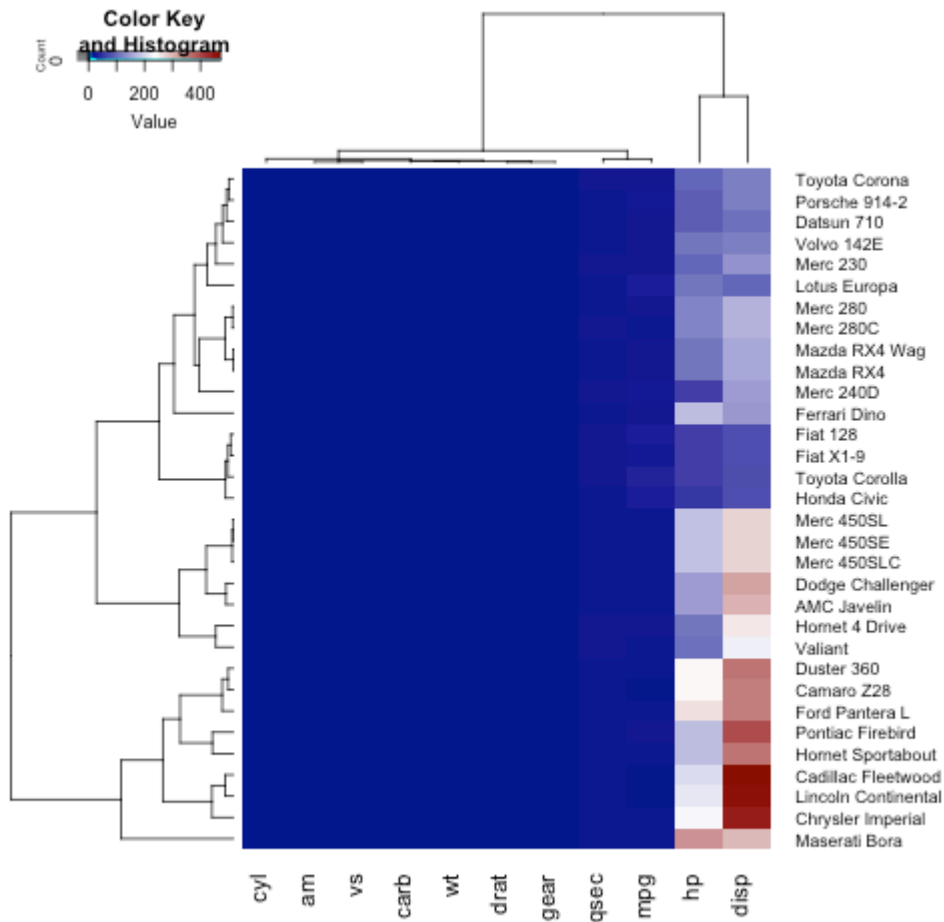


y margins

```
heatmap.2(x, trace="none", key=TRUE, col =  
colorRampPalette(c("darkblue", "white", "darkred"))(100), margins=c(5, 8))
```

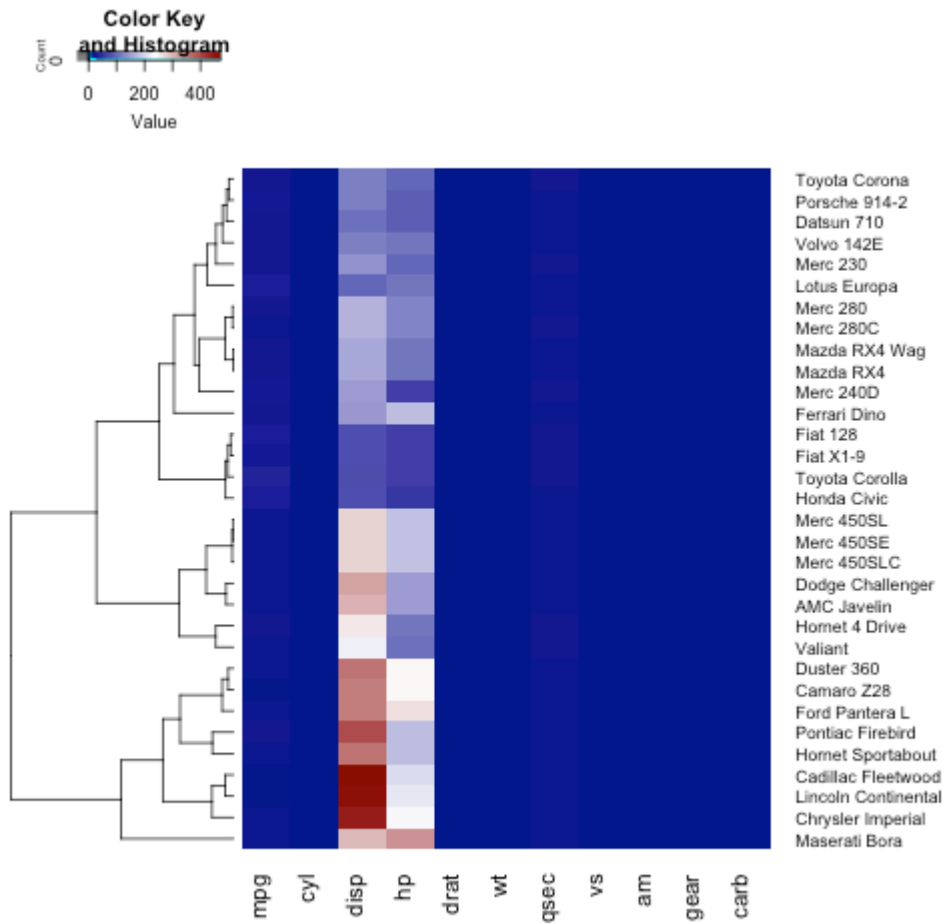


lheilwid



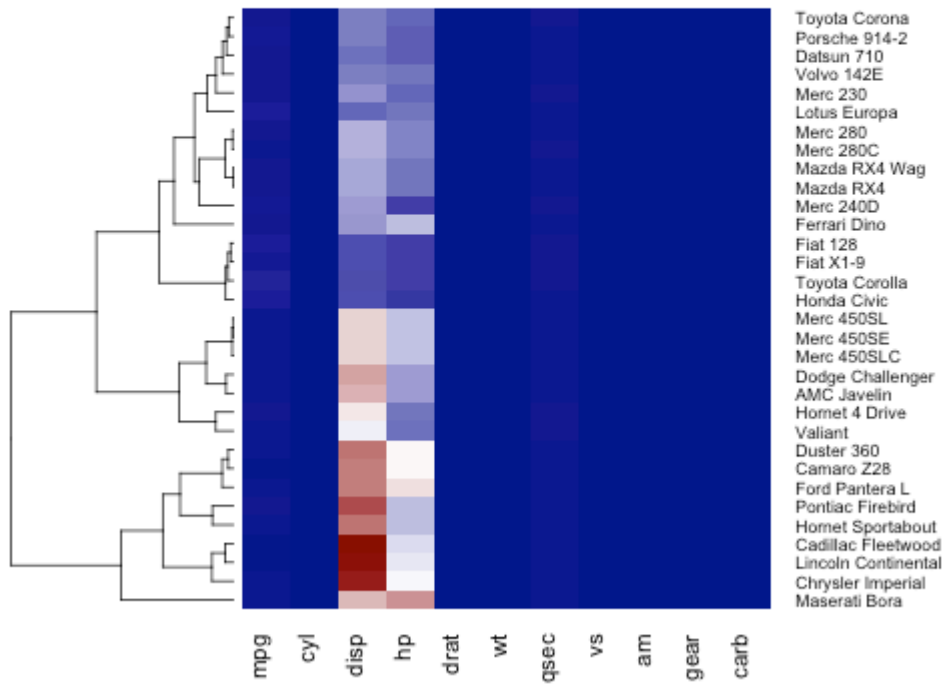
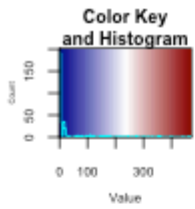
Colv=FALSE Rowv=FALSE dendrogram

```
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))
```



parcex.main, cex.lab, cex.axis

```
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))
```



<https://riptutorial.com/zh-TW/r/topic/4814/>

## 97:

- `<- readlineprompt = ""`
- `name <- readlineprompt = ""`

## Examples

R

RR。

readline

```
name <- readline(prompt = "What is your name?")
```

。

```
result <- readline(prompt = "What is the result of 1+1?")
while(result!=2){
  readline(prompt = "Wrong answer. What is the result of 1+1?")
}
```

。

as.numeric

```
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?")
}
```

<https://riptutorial.com/zh-TW/r/topic/5098/>

# 98:

## Examples

### randomForestSRC

o

CRAN randomForestSRC o

```
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
```

-

```
survivalR.lungsurvreg() survfit() predict°
```

2sex

```
require(survival)
s <- with(lung, Surv(time, status))

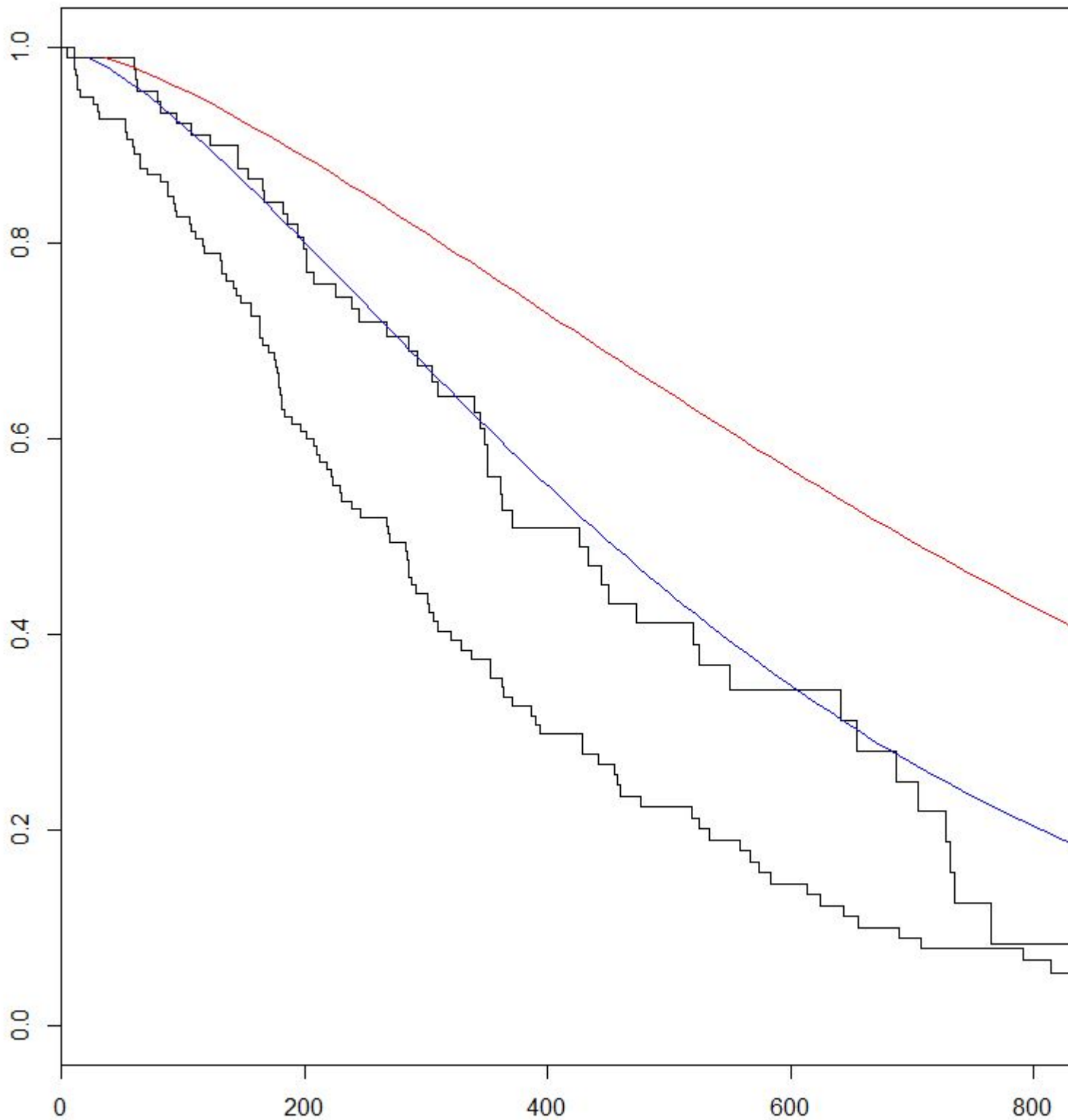
sWei <- survreg(s ~ as.factor(sex)+age+ph.ecog+wt.loss+ph.karno, dist='weibull', data=lung)

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")
```

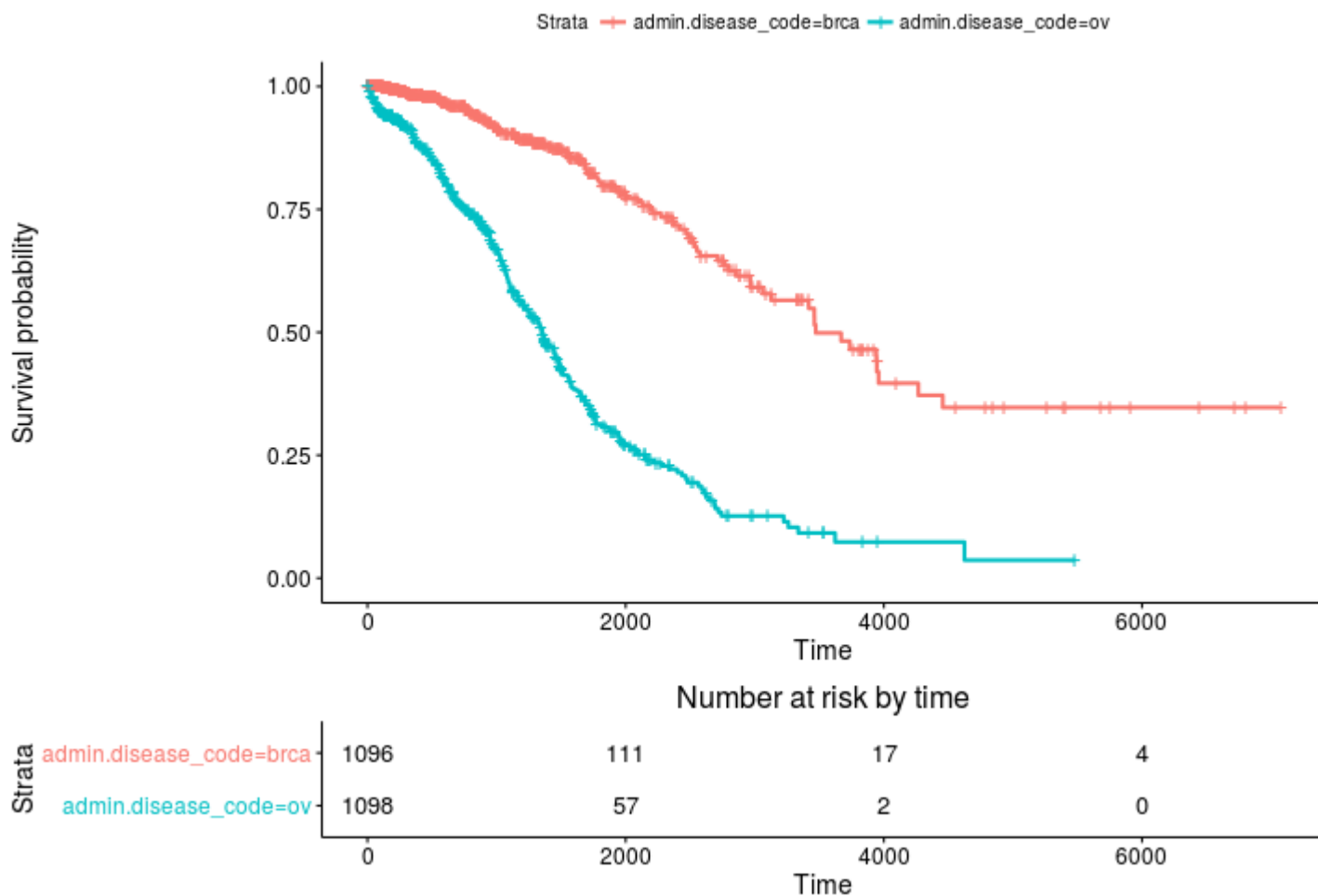




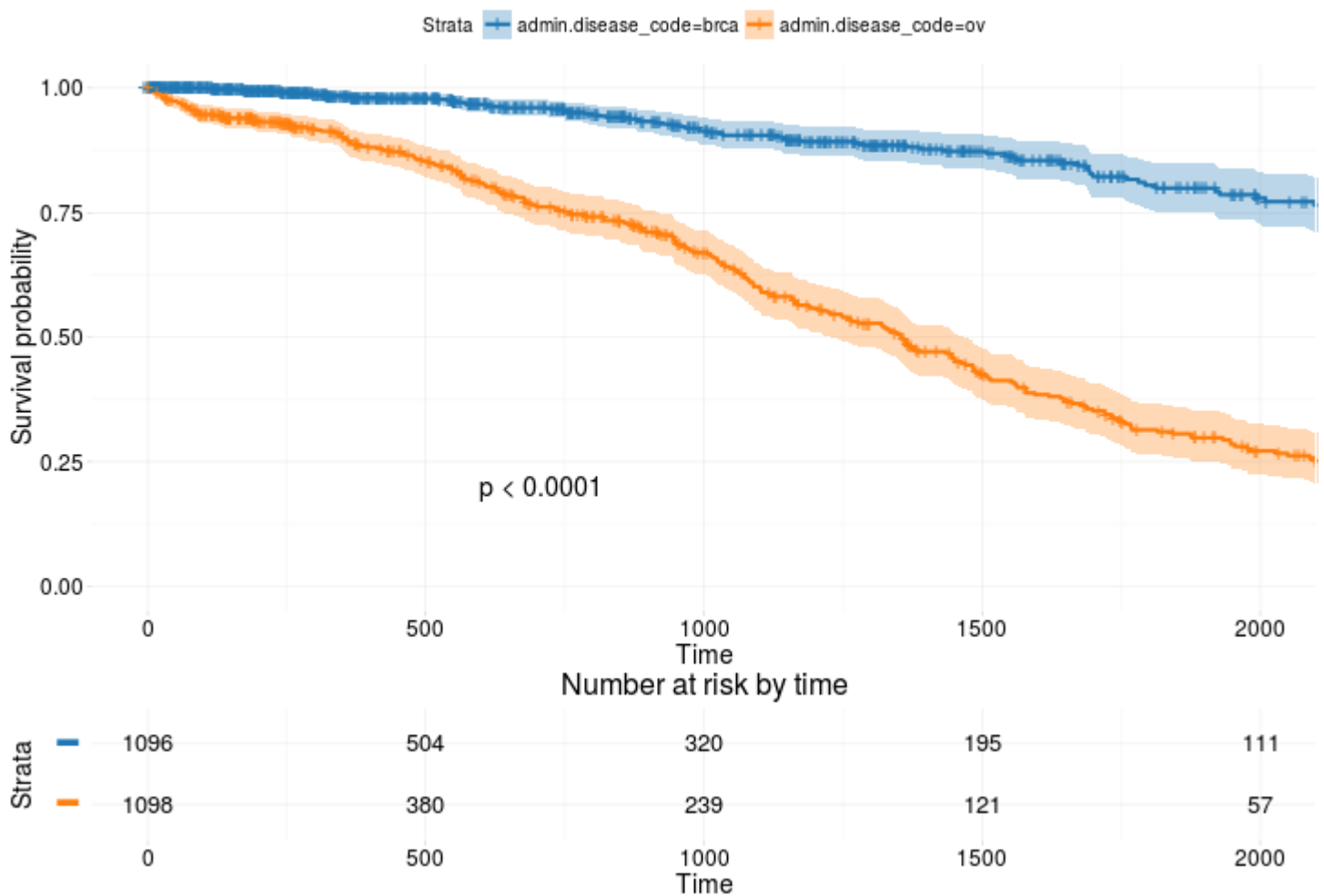
## Kaplan Meier

```
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)
```



```
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
)
```



<http://r-addict.com/2016/05/23/Informative-Survival-Plots.html>

<https://riptutorial.com/zh-TW/r/topic/3788/>

---

# 99: R

◦ ◦

## Examples

◦ tweeter ◦ ◦ 20175

20175

1

2

---

# R

R

```
library("devtools")
library("twitter")
library("ROAuth")
```

```
api_key <- XXXXXXXXXXXXXXXXXXXXXXX
api_secret <- XXXXXXXXXXXXXXXXXXXXXXX
access_token <- XXXXXXXXXXXXXXXXXXXXXXX
access_token_secret <- XXXXXXXXXXXXXXXXXXXXXXX
```

```
setup_twitter_oauth(api_key, api_secret)
```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXtweeter◦

◦

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

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

1000""◦

◦ sapply◦

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

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

head◦

```
head(coffee_tweets)
```

[R https://riptutorial.com/zh-TW/r/topic/10086/r](https://riptutorial.com/zh-TW/r/topic/10086/r)

# 100:

## Examples

◦ ◦ matrix◦

```
matrix(data = 1:6, nrow = 2, ncol = 3)
##      [,1] [,2] [,3]
## [1,]  1   3   5
## [2,]  2   4   6
```

16◦ data nrowncol◦ ◦ byrow

```
matrix(data = 1:6, nrow = 2, ncol = 3, byrow = TRUE)
##      [,1] [,2] [,3]
## [1,]  1   2   3
## [2,]  4   5   6
```

- ◦

```
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"
```

◦ ◦ rownamescolnames ◦ NULL◦ ◦

```
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
```

◦ ◦

class isas<sup>1</sup>◦

```
class(mat1)
## [1] "matrix"
```

```
is.matrix(mat1)
## [1] TRUE
as.vector(mat1)
## [1] 1 4 2 5 3 6
```

<https://riptutorial.com/zh-TW/r/topic/9019/>

# 101:

## Examples

XY

R.

XY. XY.

rgdal° **R**Spatial\*DataFrame \*Points LinesPolygons °

[OpenGeocode](#) °

CSV ° rgdal °

```
setwd("D:/GeocodeExample/")
library(rgdal)
```

CSVdata.frameR

```
xy <- read.csv("worldcities.csv", stringsAsFactors = FALSE)
```

°

```
head(xy)
str(xy)
```

“-33.532” ° SpatialPointsDataFrame()numeric ° °

```
xy$latitude <- as.numeric(xy$latitude)
xy$longitude <- as.numeric(xy$longitude)
```

NA ° °

```
xy <- xy[!is.na(xy$longitude),]
```

XY. CRS.

```
xySPoints <- SpatialPointsDataFrame(coords = c(xy[,c("longitude", "latitude")]),
proj4string = CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"),
data = xy
)
```

°



```
plot(xySPoints, pch = ".")
```



**.shp**

---

## rgdal

`rgdalreadOGR()` [ESRIR](#)◦

```
library(rgdal)
shp <- readORG(dsn = "/path/to/your/file", layer = "filename")
```

`dsn/layer.shp`

---

**shapefile** [raster](#) [shapefile](#)

```
library(raster)
shp <- shapefile("path/to/your/file.shp")
```

`rgdal import`◦

---

## TMAP

`tmaprgdal::readORG`◦

```
library(tmap)
sph <- read_shape("path/to/your/file.shp")
```

<https://riptutorial.com/zh-TW/r/topic/2093/>

# 102:

R: `+function (e1, e2) .Primitive("+")e1e2`

```
`+` <- function(e1, e2) {e1-e2}
> 3+10
[1] -7
```

## Examples

```
3+1:5
```

```
[1] 4 5 6 7 8
```

:+

- `3+1:5`
- `3+c(1, 2, 3, 4, 5)`
- `c(4, 5, 6, 7, 8)`

R( )

```
(3+1):5
```

R

```
[1] 4 5
```

。

1.

+-

```
> 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
```

```
> 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.

o

```
> 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

```
> sum(A)
# [1] 5.7
> sum(-A)
# [1] -5.7
> sum(A[-n]) + A[n]
# [1] 5.7
```

R° o o o

```
> 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
```

```
> 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°

```
> 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
```

<https://riptutorial.com/zh-TW/r/topic/4389/>

## 103: >

magrittr dplyr

◦

- `lhs>rhs` `rhs(lhs)`
- `lhs>rsa = 1` `rhs(lhs, a = 1)` `rhs(lhs, a = 1)`
- `lhs>rsa = 1b` `rhs(a = 1, b = lhs)` `rhs(a = 1, b = lhs)`
- `lhs lhs <- rhs(lhs) lhs<>rhs`
- `lhs$rsa#tipefor` `with(lhs, rhs(lhs$a))`
- `lhsT>rhspipe` `{ rhs(lhs); lhs }`

| LHS      | RHS      |
|----------|----------|
| magrittr | magrittr |

%>%

magrittrdplyrmagrittr [tidyverse tidyverse](#) “API”

magrittr%<>% %\$%T>% ◦ + [ [◦

+ \* ^ & %in% ?'%>%!help('%>%') pkg:magrittr

[RStudio](#) `Ctrl+Shift+M` *WindowsLinux* `Cmd+Shift+M` *Mac* ◦

◦

- 
- `object %>% rm()` `object`

## Examples

%>% magrittr ◦ **LHSRHS** ◦

```
library(magrittr)

1:10 %>% mean
# [1] 5.5

# is equivalent to
mean(1:10)
```

```
# [1] 5.5
```

◦ ◦ years◦

```
years <- factor(2008:2012)

# nesting
as.numeric(as.character(years))

# piping
years %>% as.character %>% as.numeric
```

LHSRHS.◦

```
# 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"
```

◦

```
. %>% RHS
```

```
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
```

functions

```
read_year
# Functional sequence with the following components:
#
# 1. as.character(.)
# 2. as.Date(.)
# 3. year(.)
#
# Use 'functions' to extract the individual functions.
```

```
read_year[[2]]
# function (.)
# as.Date(.)
```

◦



```
magrittr%<>% magrittrrhs◦ <-◦ %<>%
```

```
library(magrittr)
library(dplyr)

df <- mtcars
```

```
df <- df %>% select(1:3) %>% filter(mpg > 20, cyl == 6)
```

```
df %>% select(1:3) %>% filter(mpg > 20, cyl == 6) -> df
```

df

```
df %<>% select(1:3) %>% filter(mpg > 20, cyl == 6)
```



```
%%$R◦ data|> data.frame|> dplyr◦
```

```
%%$%◦ data.frame|> 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
```

```
%>%data.framefilter() %$%cor.test() ◦
```

```
with()R◦
```

## dplyr + ggplot2

```
%>%dplyr + ggplot◦ EDA◦ ggplot◦
```

```
library(dplyr)
library(ggplot)

diamonds %>%
  filter(depth > 60) %>%
  group_by(cut) %>%
  summarize(mean_price = mean(price)) %>%
  ggplot(aes(x = cut, y = mean_price)) +
  geom_bar(stat = "identity")
```

T>

R◦

```
%T>% tee lhs◦ tee %>% lhs rhs /◦
```

```
◦ %>%T>% all_letters NULL◦
```

```
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
```

save().load()◦ ◦

```
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")
```

> <https://riptutorial.com/zh-TW/r/topic/652/---gt--->

## 104:

- `boxplotx...` generic function
- `boxplot = NULL...na.action = NULL##"`
- `boxplotx...range = 1.5width = NULLvarwidth = FALSEnotch = FALSEoutline = TRUEnames plot = TRUEborder = par("fg"col = NULLlog = " "pars = listboxwex = 0.8staplewex = 0.5outwex = 0.5horizontal = FALSEadd = FALSEat = NULL## Default S3 method`

|                        | R                             |
|------------------------|-------------------------------|
|                        | <code>y~grpygrp</code>        |
|                        | <code>data.frame</code> list  |
|                        | ◦                             |
| <code>na.action</code> | <code>NA</code> ◦ ◦           |
| <code>boxwex</code>    | ◦ ◦                           |
|                        | <code>TRUE</code> boxplot ◦ ◦ |
|                        | <code>col</code> null ◦ ◦     |

## Examples

### `boxplot{graphics}`

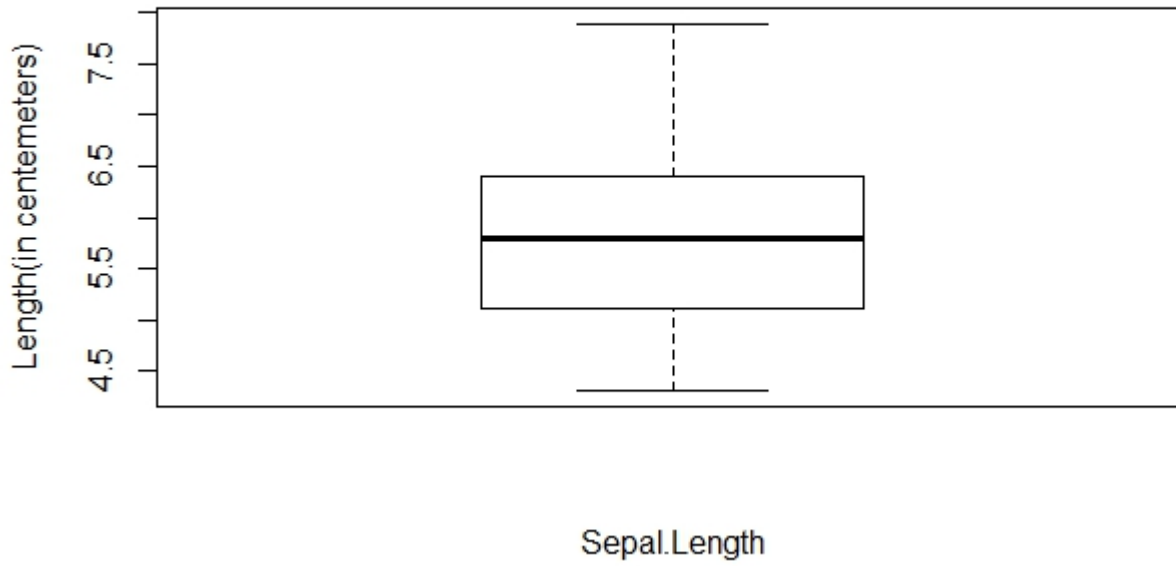
```
boxplot()iris
```

```
> 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
```

## `boxplotSepal.Length`

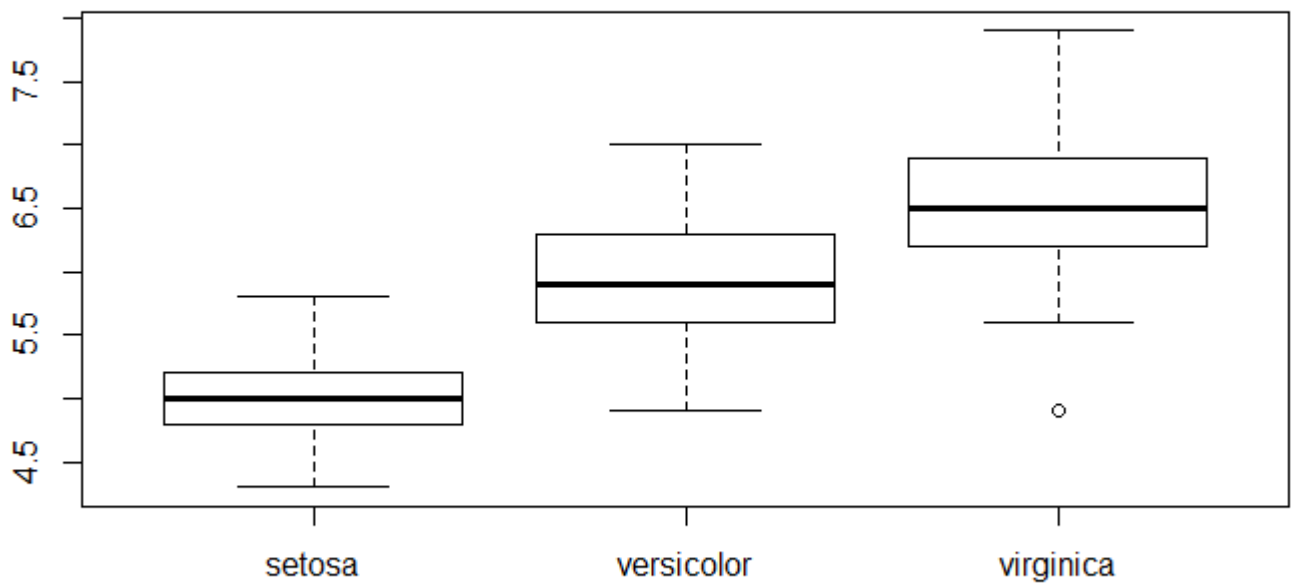
```
boxplot(iris[,1],xlab="Sepal.Length",ylab="Length(in centemeters)",
        main="Summary Charateristics of Sepal.Length(Iris Data)")
```

## Summary Characteristics of Sepal.Length(Iris Data)



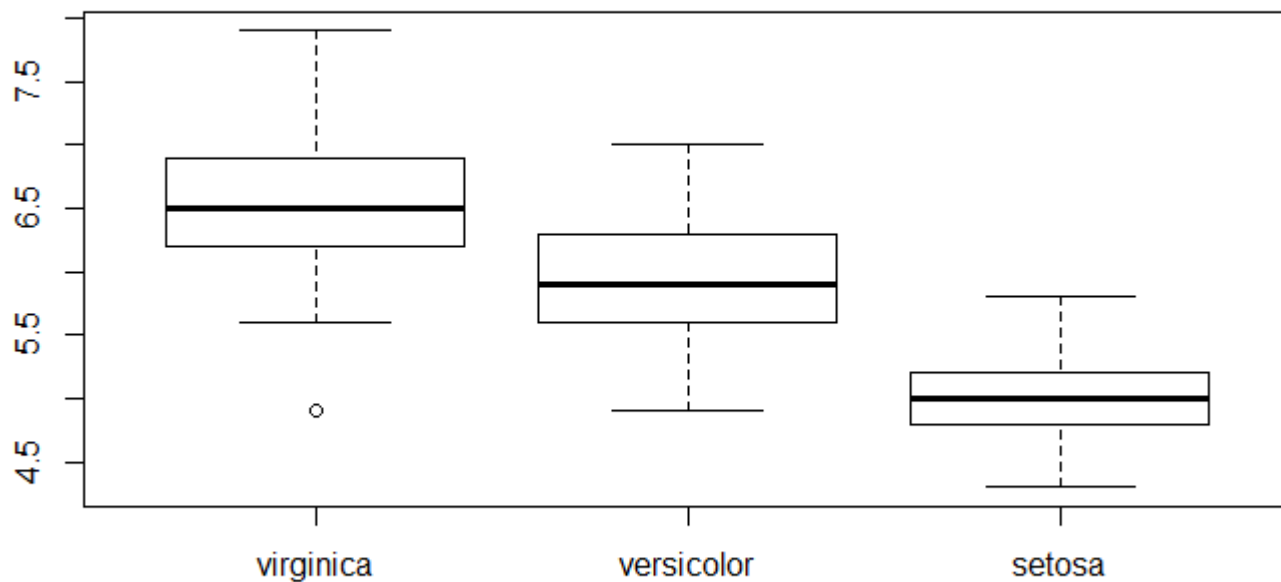
## Boxplot

```
boxplot(Sepal.Length~Species,data = iris)
```



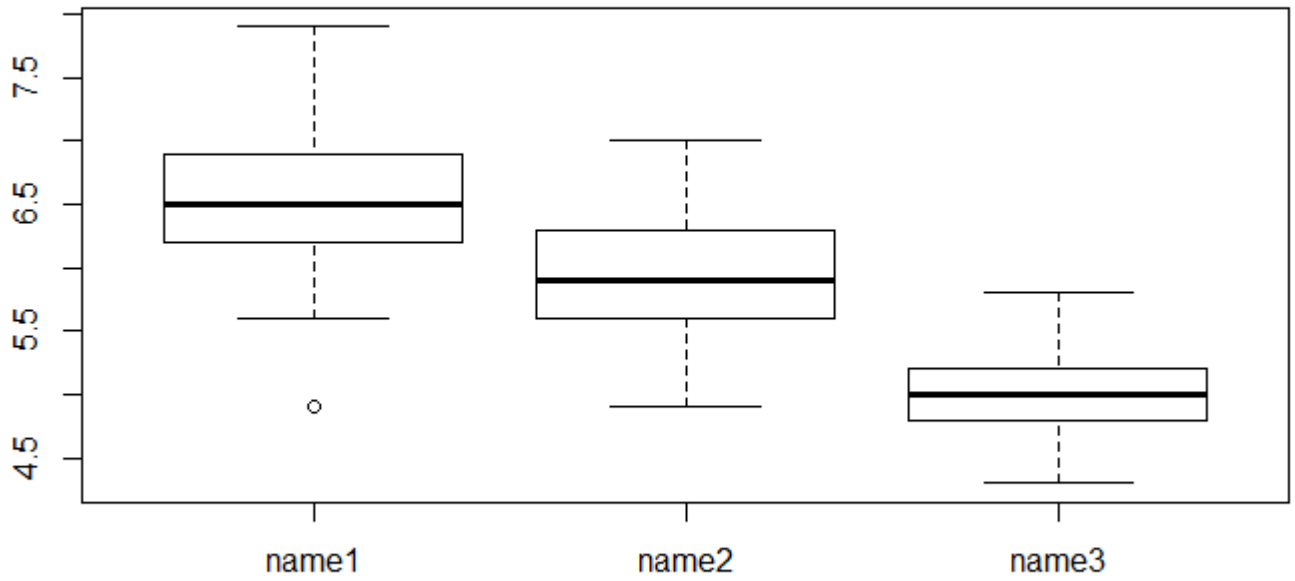
```
virginica - versicolor - setosa
```

```
newSpeciesOrder <- factor(iris$Species, levels=c("virginica","versicolor","setosa"))  
boxplot(Sepal.Length~newSpeciesOrder,data = iris)
```



Names°

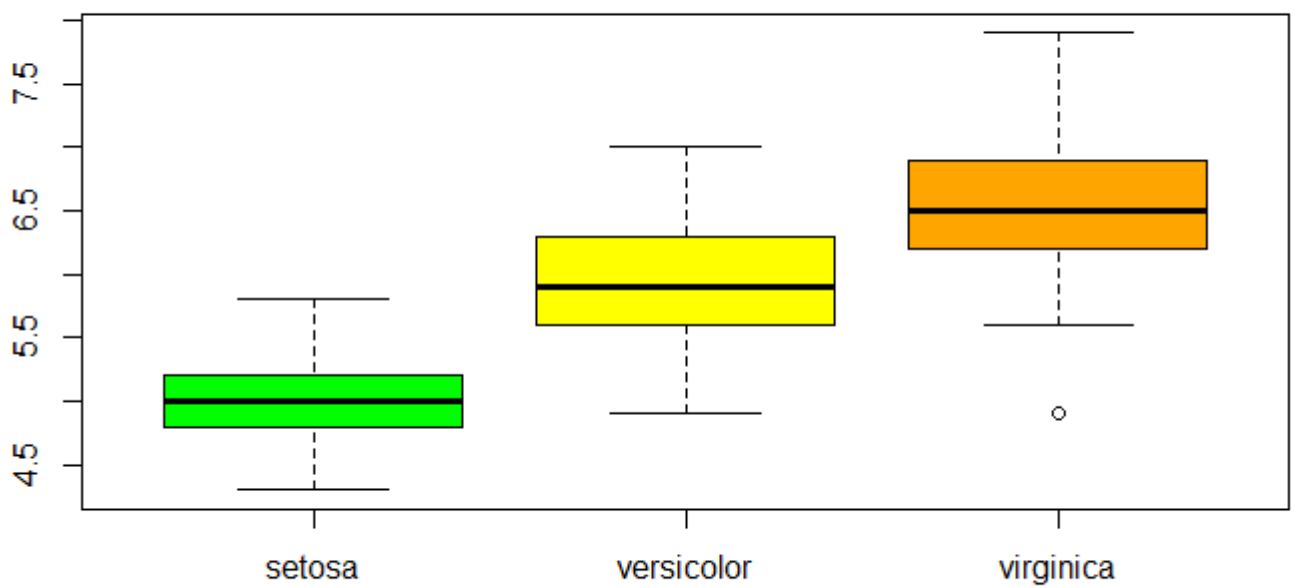
```
boxplot(Sepal.Length~newSpeciesOrder,data = iris,names= c("name1","name2","name3"))
```



—

col

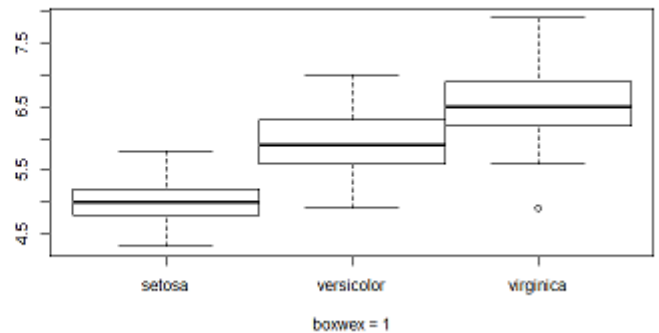
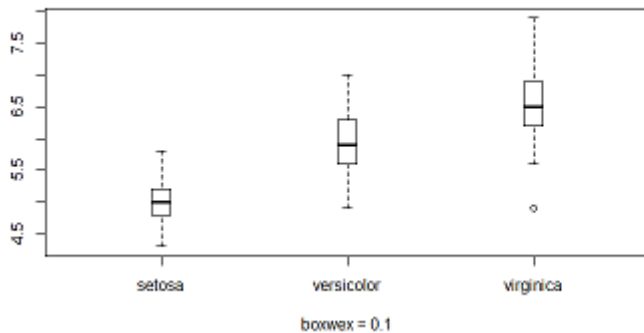
```
boxplot(Sepal.Length~Species,data = iris,col=c("green","yellow","orange"))
```



boxwex ◦

```
boxplot(Sepal.Length~Species,data = iris,boxwex = 0.1)
```

```
boxplot(Sepal.Length~Species,data = iris,boxwex = 1)
```



**plot=FALSE**

parameter plotFALSE ◦

```
> 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"
```

**boxplot**◦

- boxlty -
- boxlwd -
- boxcol -
- boxfill - box

- medlty - ""
  - medlwd - width
  - medcol -
  - medpch - NA
  - medcex -
  - medbg -
- 

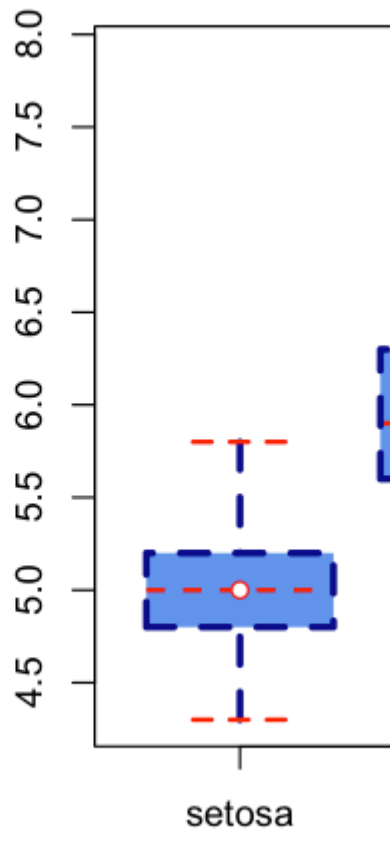
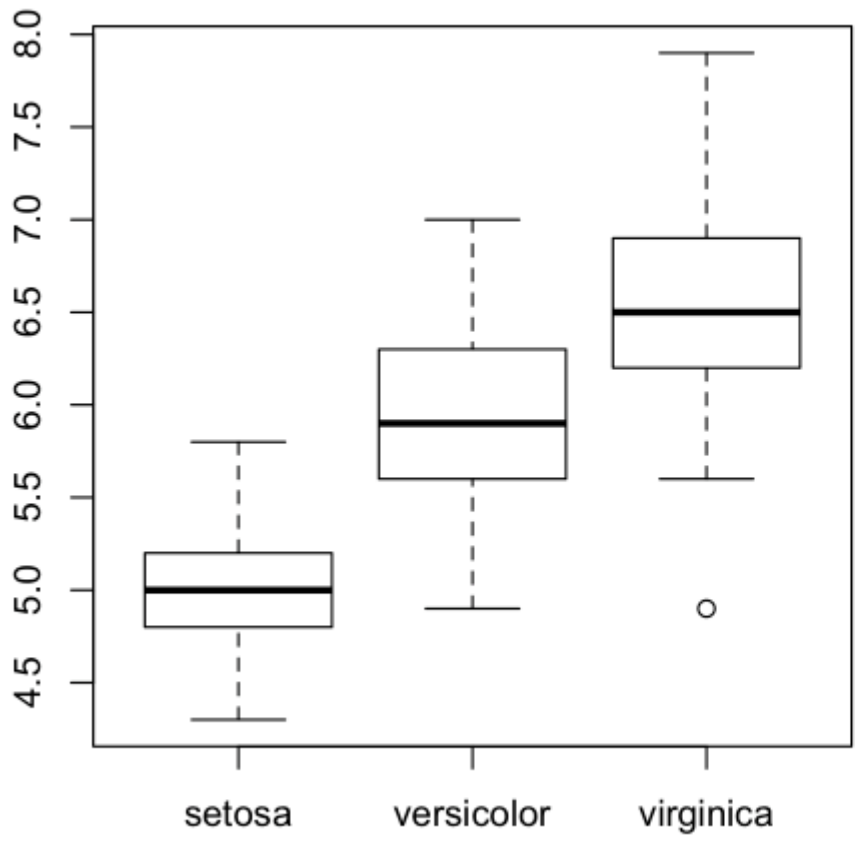
- whisklty - whisker line type
  - whisklwd -
  - whiskcol -
- 

- staplelty -
  - staplelwd -
  - staplecol -
- 

- outlty - ""
  - outlwd -
  - outcol -
  - outpch - NA
  - outcex -
  - outbg -
- 

```
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
        )
```





<https://riptutorial.com/zh-TW/r/topic/1005/>

# 105:

## Examples

combn

```
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"
```

expand.grid

```
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
```

outer

```
# 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"
```

---

```
choose(length(LETTERS), 5)
[1] 65780
```

---

```
length(letters)^5
[1] 11881376
```

<https://riptutorial.com/zh-TW/r/topic/5836/>

---

# 106: R

## Examples

### RCurl

#### imdb

```
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
```

[R https://riptutorial.com/zh-TW/r/topic/4336/r](https://riptutorial.com/zh-TW/r/topic/4336/r)

# 107:

- `lmna.actionmethod = "qr" model = TRUE x = FALSE y = FALSE qr = TRUE singular.ok = TRUE contrasts = NULL offset..`

|             |  |
|-------------|--|
|             | <i>Wilkinson-Rogers</i> ; response ~ ... where ...data   |
|             | data   |
| na.action   | NA ?na.action  |
|             | ◦ "qr" "model.frame" model=TRUE  |
| X           |  |
| $\hat{y}$   |  |
| QR          | QR   |
| singular.ok | NA   |
|             | ;<br>?model.matrix.default contrasts.arg<br>options() contrasts contrast? contrasts<br>? contrasts |
|             | ◦ ◦ ?model.offset  |
| ...         | lm.fit() lm.wfit()   |

## Examples

### mtcars

mtcars 32 ◦ help(mtcars) ◦

mpg wt

```
plot(mpg ~ wt, data = mtcars, col=2)
```

lm

```
fit <- lm(mpg ~ wt, data = mtcars)
```

~ "mpg ~ wt" mpgw

```
summary(fit)
```

```
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
```

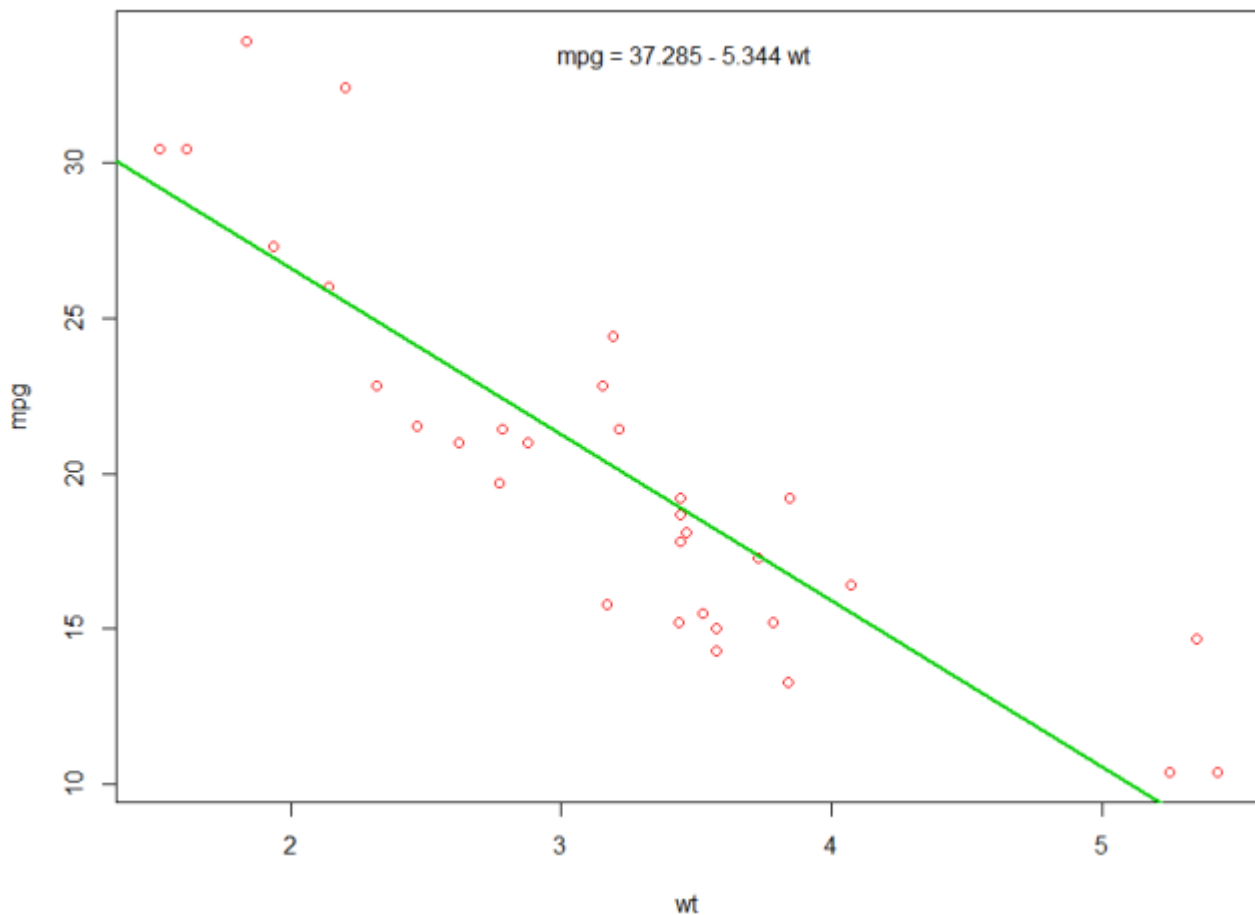
- $\text{wt} \cdot \text{mpg}$
- $\text{p}$
- $R^2$

```
mpg
```

```
abline(fit,col=3,lwd=2)
```

```
◦ coef ◦ paste0+/- ◦ mtextmtext
```

```
bs <- round(coef(fit), 3)
lmlab <- paste0("mpg = ", bs[1],
              ifelse(sign(bs[2])==1, " + ", " - "), abs(bs[2]), " wt ")
mtext(lmlab, 3, line=-2)
```



mtcars

```
fit <- lm(mpg ~ wt, data = mtcars)
```

```
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))))))
```

rsquare

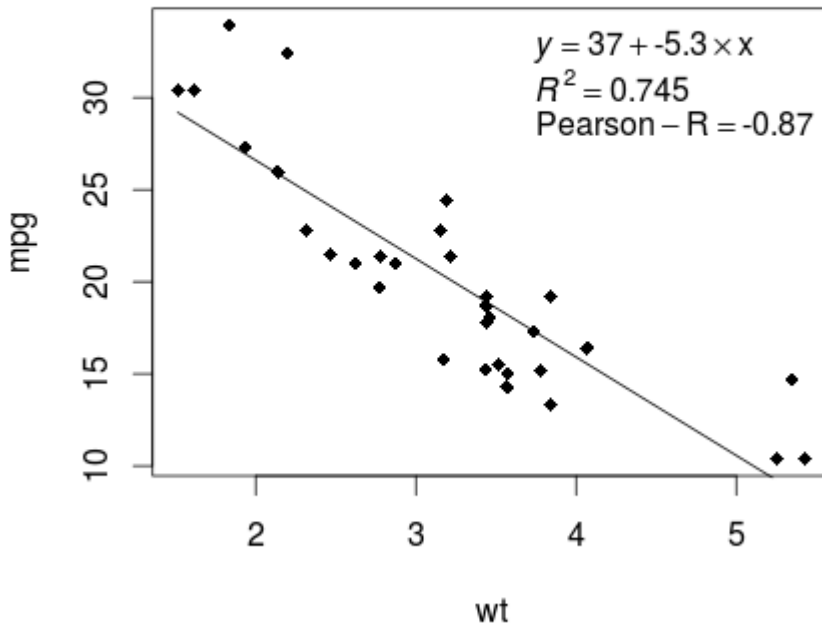
```
rp = vector('expression',3)
rp[1] = substitute(expression(italic(y) == MYOTHERVALUE3 + MYOTHERVALUE4 %*% x),
list(MYOTHERVALUE3 = format(fit$coefficients[1], digits = 2),
MYOTHERVALUE4 = format(fit$coefficients[2], digits = 2)))[2]
rp[2] = substitute(expression(italic(R)^2 == MYVALUE),
list(MYVALUE = 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')
```

RMSE

```
rp = vector('expression',10)
```

r[1] ....r[10]



- ◦
- ◦ ◦ ◦
- - IPW ◦ ◦

lm() ◦ surveysvyglm() ◦ survey ◦ lm()svyglm()gaussian() ◦ ◦

```
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))
```

```
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
progvillm   -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

```

## 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 *
progvillm   -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

```

◦ ◦ ◦

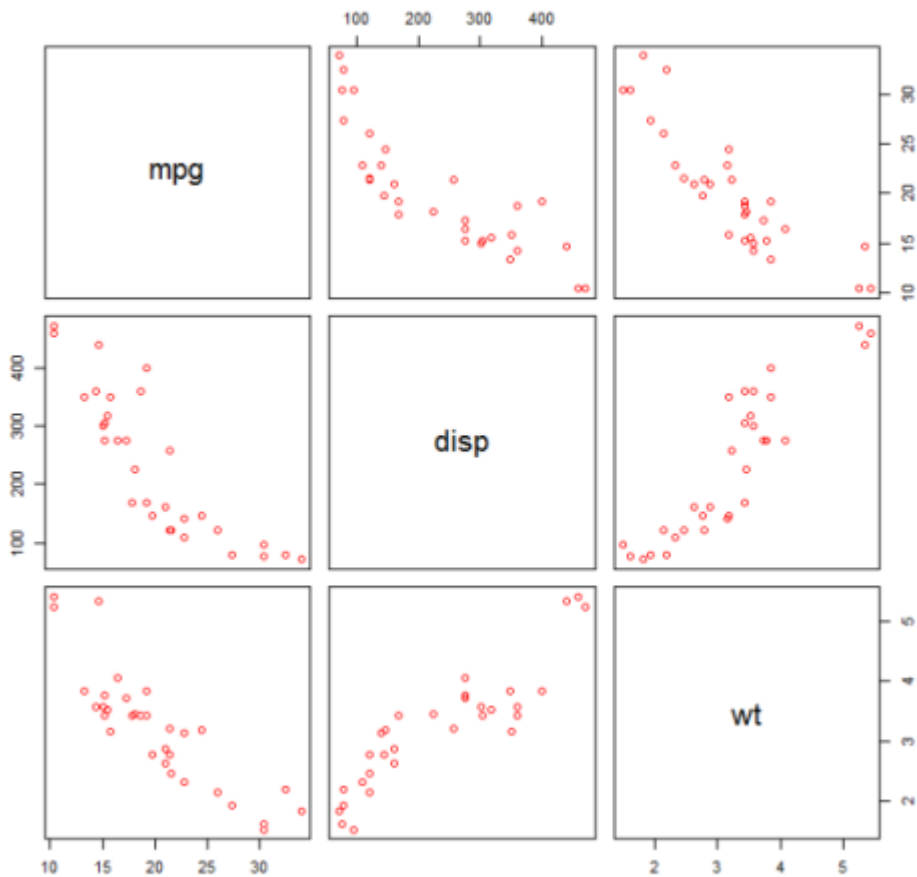
mtcars◦ [mtcars](#)◦

mpg Miles / gallon disp Displacementcu.in◦ wt Weight1000 lbs◦ mpgdisp◦

```

plot(mtcars[,c("mpg", "disp", "wt")])

```



disp°

```
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
```

I(disp^2) ° R^2°

```
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
```

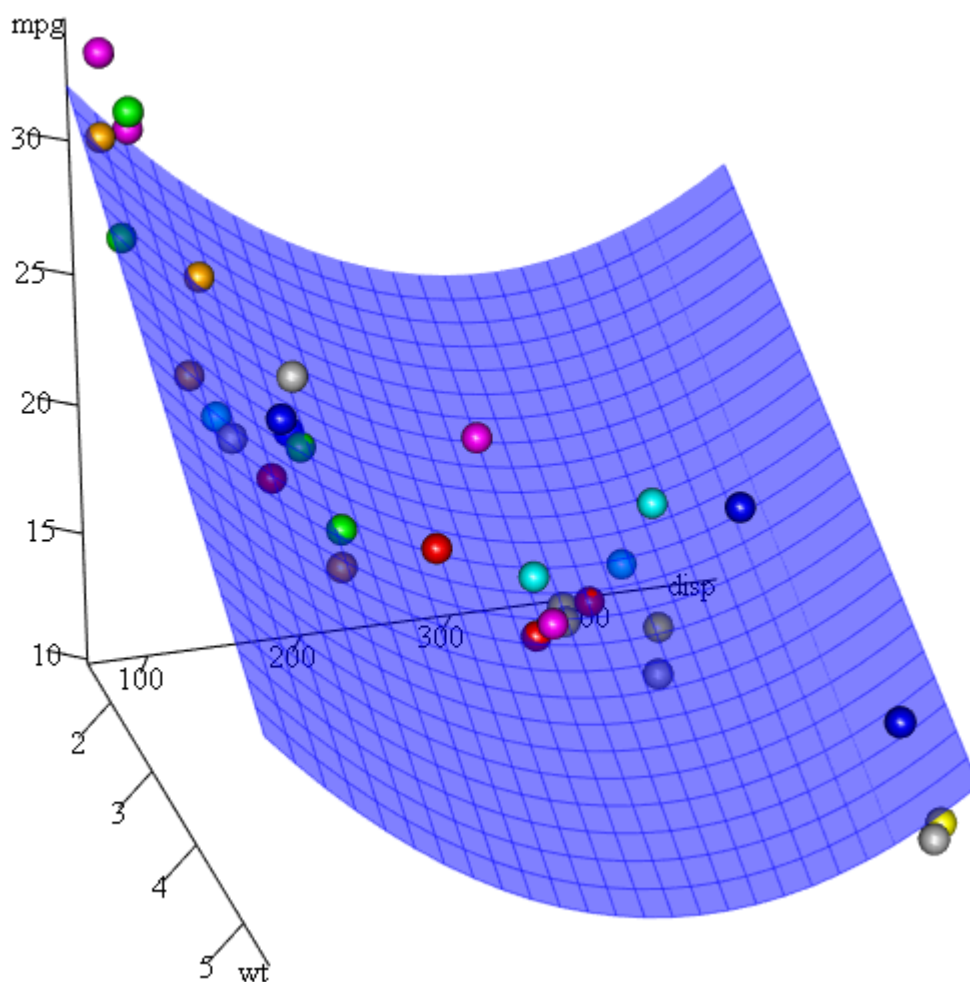
```
mpg = 41.4020-3.4179*wt-0.0824*disp+0.0001277*disp^2
```

```
raw=TRUE poly help(ploy)help(ploy) °
```

```
summary(lm(mpg ~ wt+poly(disp, 2, raw=TRUE),mtcars))
```

```
R3D° Fit3dp3d°
```

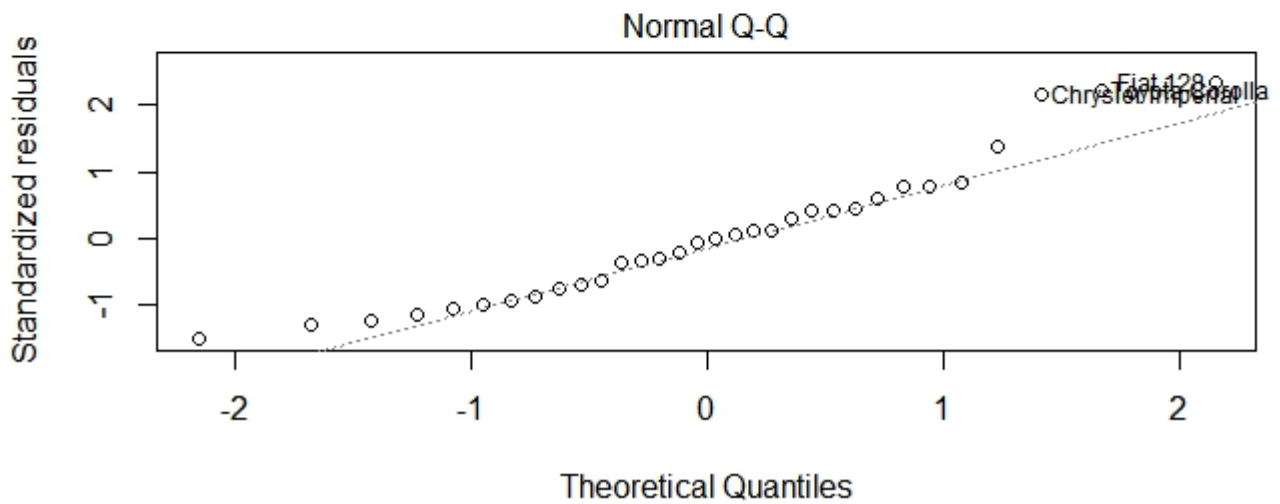
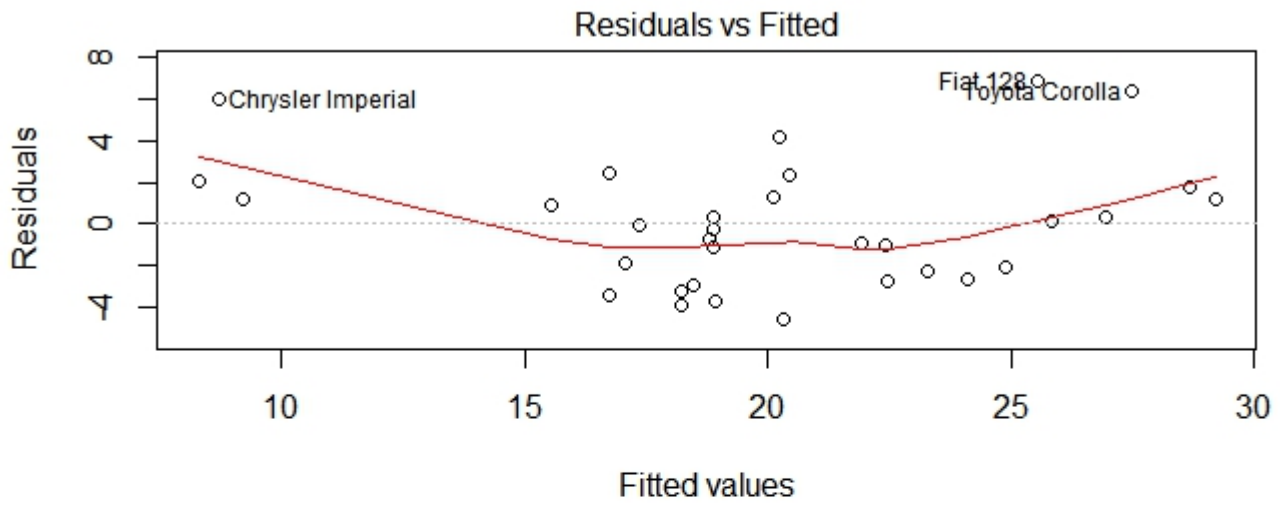
```
library(p3d)
Init3d(family="serif", cex = 1)
Plot3d(mpg ~ disp+wt, mtcars)
Axes3d()
Fit3d(fit1)
```



```
° °
```

```
# fit the model
fit <- lm(mpg ~ wt, data = mtcars)
#
par(mfrow=c(2,1))
# plot model object
```

```
plot(fit, which =1:2)
```



1. mpg wt . . . .

2. . . QQ. .

''

```
predict(mtcars
```

```
my_md1 <- lm(mpg ~ disp, data=mtcars)
my_md1

Call:
lm(formula = mpg ~ disp, data = mtcars)

Coefficients:
```

```
(Intercept)      disp
 29.59985      -0.04122
```

◦

```
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_mdl, newdf)
      1      2      3      4      5
18.96635 26.66946 26.47987 23.62366 13.11381
```

◦ disp◦

## 1. data.frame

```
predict(my_mdl, newdata)
Error in eval(predvars, data, env) :
  numeric 'envir' arg not of length one
```

```
2. newdf2 <- data.frame(newdata)
predict(my_mdl, newdf2)
Error in eval(expr, envir, enclos) : object 'disp' not found
```

y◦ newdf'mpg'disp'◦

```
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
```

<https://riptutorial.com/zh-TW/r/topic/801/-->

# 108:

NA

NA  $\circ$  `sqrt(-1)` NaN  $\circ$

## Examples

`anyNA`; `is.na`; `is.na`

```
vec <- c(1, 2, 3, NA, 5)

anyNA(vec)
# [1] TRUE
is.na(vec)
# [1] FALSE FALSE FALSE TRUE FALSE
```

`is.na` FALSE = 0 TRUE = 1

```
sum(is.na(vec))
# [1] 1
```

`colSums`; `is.na` NA

```
colSums(is.na(airquality))
#   Ozone Solar.R   Wind   Temp   Month   Day
#     37       7     0       0     0     0
```

[nanian](#) [github](#) CRAN

## NA

`read.*R` "NA"  $\circ$  NA  $\circ$  . - empty NA  $\circ$  `read.*na.strings` R/NA

```
read.csv("name_of_csv_file.csv", na.strings = "-")
```

NA

```
read.csv('missing.csv', na.strings = c('.', '-'))
```

NA `write.csv`  $\circ$   $\circ$

## NA

NA logical

```
class(NA)
```

```
#[1] "logical"
```

NA

```
x <- c(1, NA, 1)
class(x[2])
#[1] "numeric"
```

NANA\_character\_ NA\_integer\_ NA\_real\_NA\_complex\_ ° NA\_integer\_;

```
class(Sys.Date()[NA_integer_])
# [1] "Date"
```

## TRUE / FALSE/NA

NANANA ° NA OR TRUETRUETRUE NA OR FALSENANATRUEFALSE

```
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.
```

NANA °

```
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>
```

---

NA ◦ **SPSS**99◦

```
num.vec <- c(1, 2, 3, 99, 5)
num.vec
## [1] 1 2 3 99 5
```

NA

```
num.vec[num.vec == 99] <- NA
```

```
is.na<-◦ ?is.na
```

```
is.na<-◦ ◦
```

```
is.na(num.vec) <- num.vec == 99
```

```
num.vec
## [1] 1 2 3 NA 5
```

---

```
num.vec[!is.na(num.vec)]
num.vec[complete.cases(num.vec)]
na.omit(num.vec)
## [1] 1 2 3 5
```

---

```
mean(num.vec) # returns: [1] NA
```

na.rmNA

```
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
```

---

R1m na.action◦ na.omit options (na.action = 'na.exclude') R◦

na.action na.action

```
lm(y2 ~ y1, data = anscombe, na.action = 'na.exclude')
```

<https://riptutorial.com/zh-TW/r/topic/3388/>



# 109:

R。 R。

## Examples

R

aggregate

```
aggregate(formula, function, data)
```

。

```
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)
```

## OUTPUT

```
> 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

### dplyr::group\_by() summarize()

```

# 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()

```

## OUTPUT

```

> 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

**data.table** `dt[i, j, by]` “*idtj*” `dt` `list().().().()`

```

# 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]

```

## OUTPUT

```

> # 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
```

<https://riptutorial.com/zh-TW/r/topic/10792/>

# 110:

NLP.

## Examples

◦ ◦ ◦ ◦

◦ `Rtm` ◦

```
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))
```

`tmCorpus Corpus CorpusVectorSource VectorSource` ◦ `tm_corpus` ◦

```
str(tm_corpus)
List of 4
 $ 1:List of 2
  ..$ content: chr "drugs hospitals doctors"
  ..$ meta :List of 7
  .. ..$ author : chr(0)
  .. ..$ datetimestamp: 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]
```

`Corpus Corpus` ◦ `tmtm_map apply` ◦

```
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)
```

```
tdm <- TermDocumentMatrix(tm_corpus)
```

```
<<TermDocumentMatrix (terms: 8, documents: 4)>>
Non-/sparse entries: 12/20
Sparsity : 62%
```

```
Maximal term length: 9
Weighting           : term frequency (tf)
```

```
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
```

- environmentenviron - 44.

/.

<https://riptutorial.com/zh-TW/r/topic/10119/>

# 111:

ArimaRarima°

ARIMA° ° ARIMA°

ARIMA°

## Examples

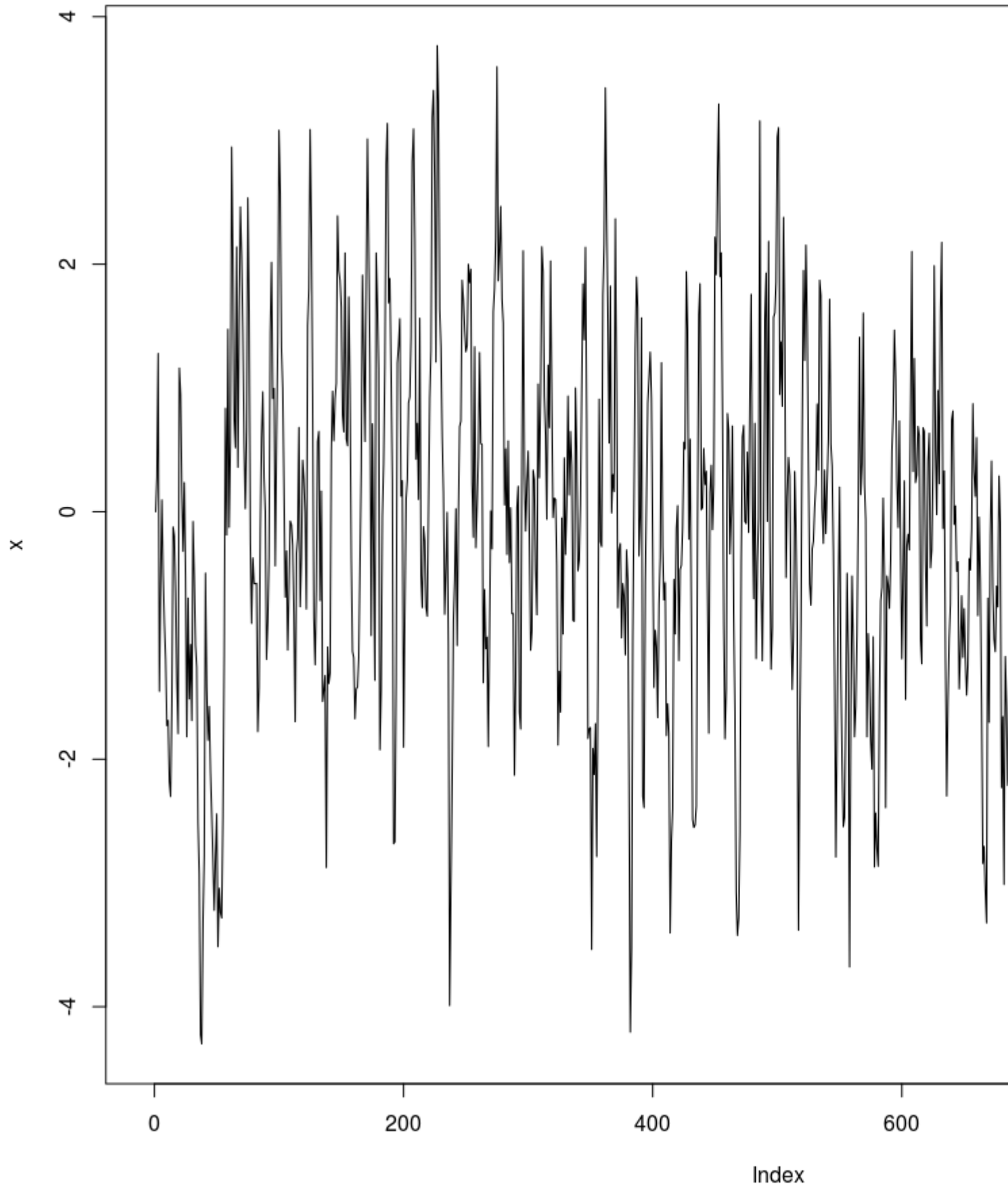
### ArimaAR1

$$x_t = .7x_{t-1} + \epsilon \quad \epsilon \sim N(0,1)$$

```
#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')
```





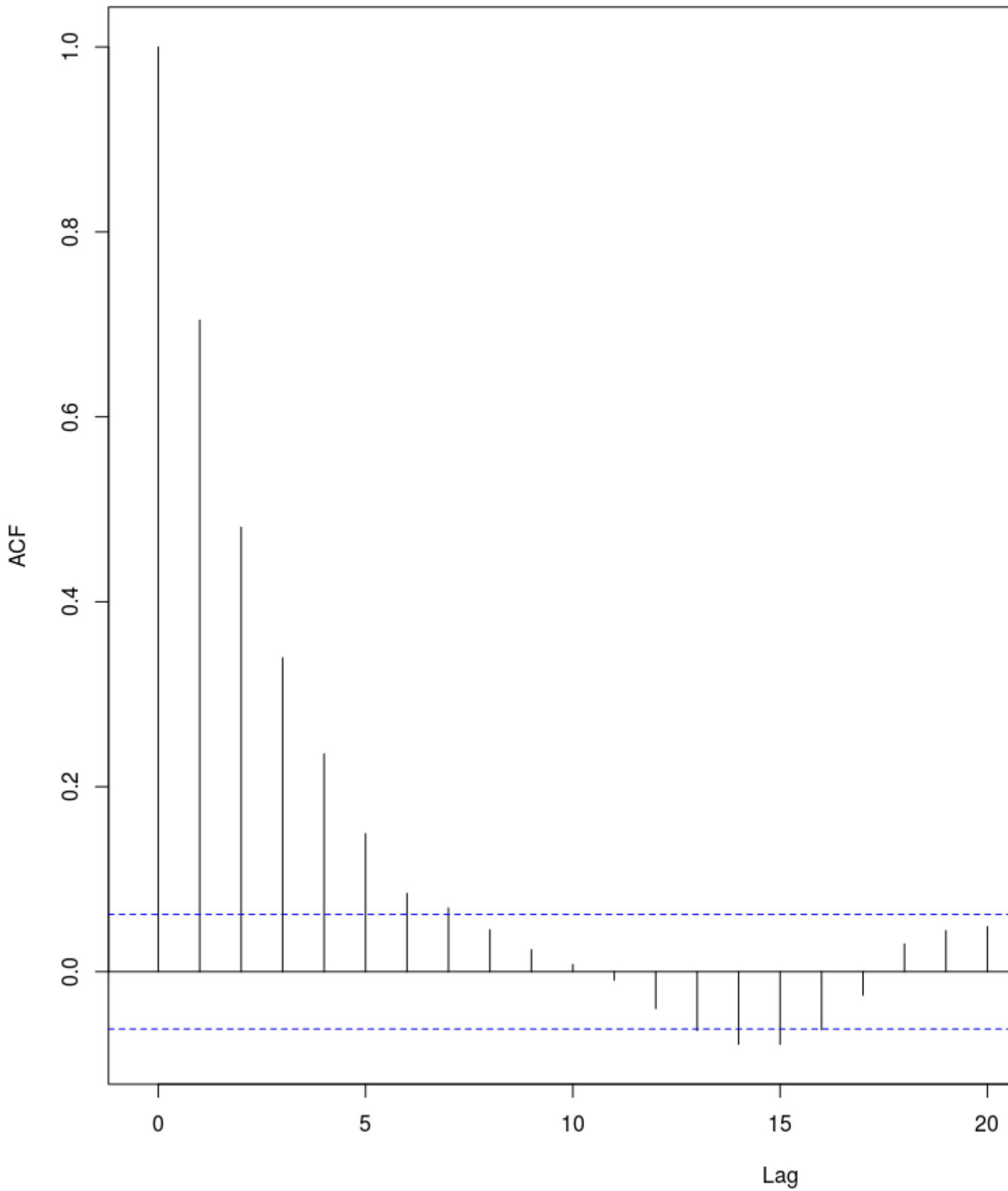
## 10MA0Arima

```
#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
```

```
fit$coef[1]
#      ar1
# 0.7040085

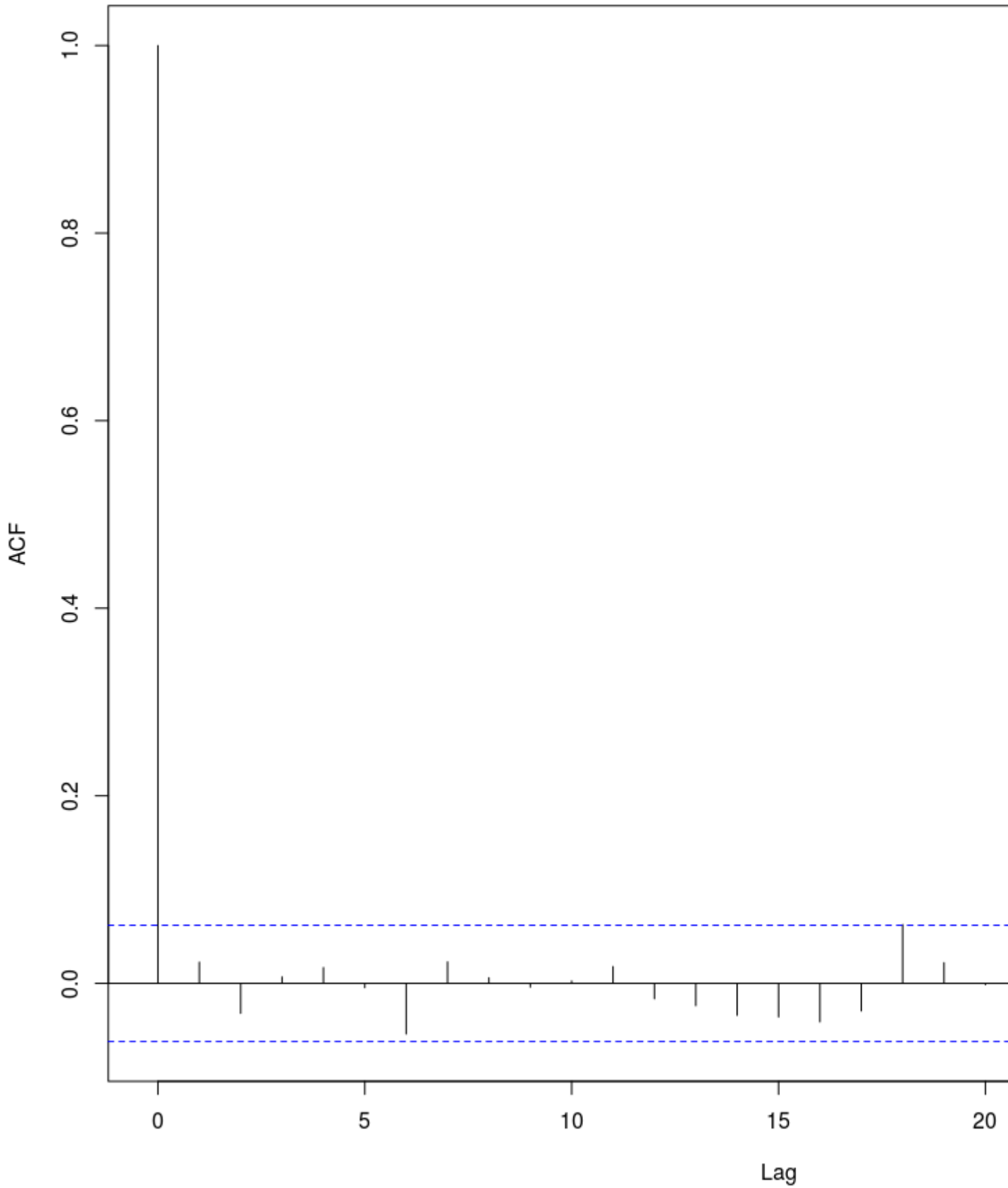
#Verify that the model eliminates the autocorrelation
acf(x)
```

# Series 1



```
acf(fit$resid)
```

### Series fit\$resid



```

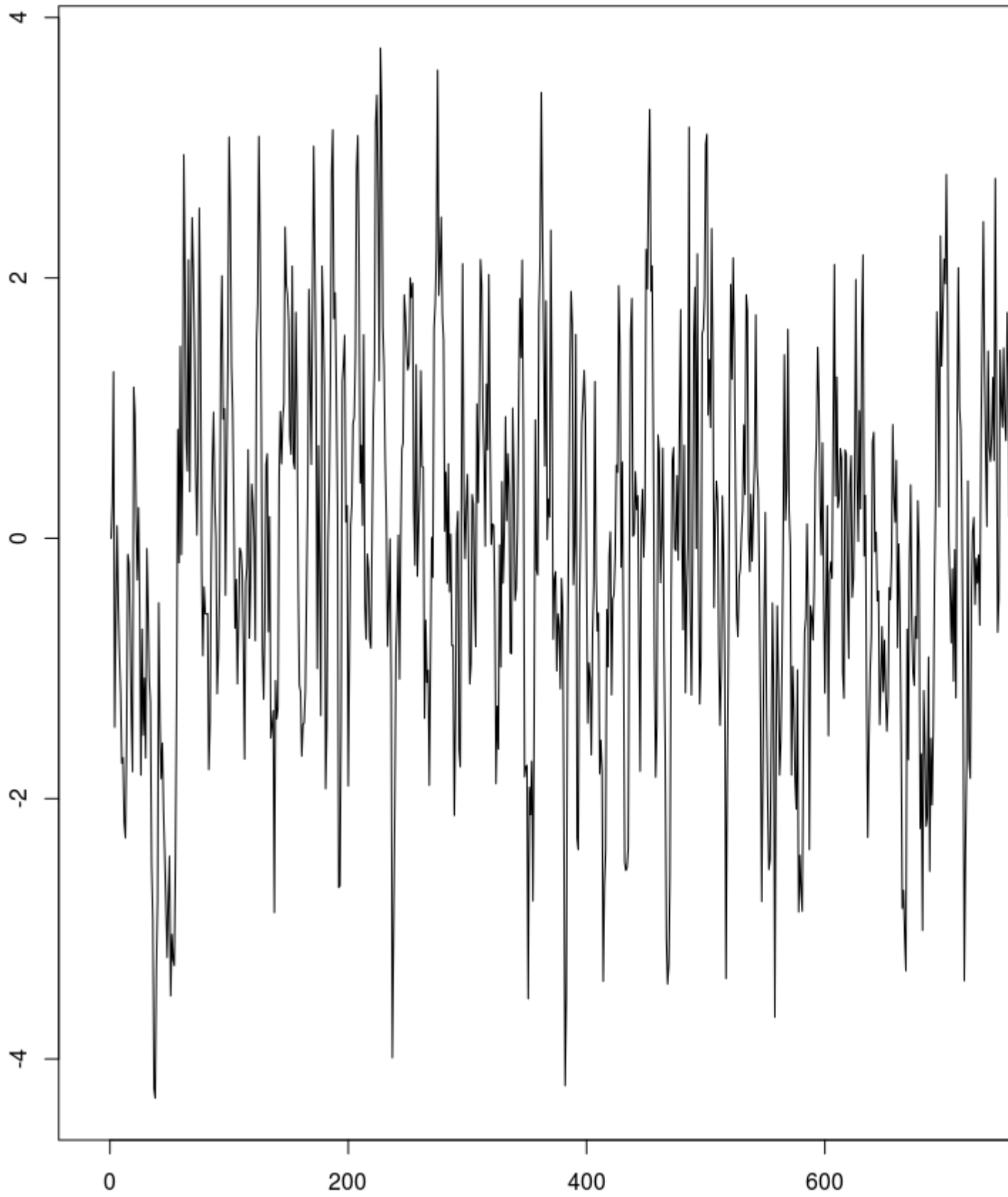
#Forecast 10 periods
fcst <- forecast(fit, h = 100)
fcst
  Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
1001    0.282529070 -0.9940493  1.559107 -1.669829  2.234887
1002    0.173976408 -1.3872262  1.735179 -2.213677  2.561630
1003    0.097554408 -1.5869850  1.782094 -2.478726  2.673835
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



<https://riptutorial.com/zh-TW/r/topic/1725/>

# 112:

- R“”。

```
dat <- c(1, 2, 2, 2, 3, 1, 4, 4, 1, 1)
```

- 1;32s;3s;◦ R。

- Rrleiddata.table ◦

## Examples

`rle`

- 

```
dat <- c(1, 2, 2, 2, 3, 1, 4, 4, 1, 1)
```

rle

```
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
```

r\$values

```
r$values
# [1] 1 2 3 1 4 1
```

1231。

r\$lengths

```
r$lengths
# [1] 1 3 1 1 2 2
```

1123。

R

- 

```
(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
```

x213211 y x1.5,46

## Rrlex

```
(r <- rle(dat$x))
# Run Length Encoding
# lengths: int [1:3] 2 3 1
# values : num [1:3] 1 2 1
```

length(r\$lengths) r\$lengths rep

```
(run.id <- rep(seq_along(r$lengths), r$lengths))
# [1] 1 1 2 2 2 3
```

tapplyidy

```
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

data.table

```
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
```

x213211 y x1.5,46

data.table rleididid

```
rleid(DT$x)
# [1] 1 1 2 2 2 3
```

IDy

```
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
```

## ◦ 1000100

```
set.seed(144)
dat <- sample(rep(0:1, c(1, 1e5)), 1e7, replace=TRUE)
table(dat)
#      0      1
# 103 9999897
```

## 1000

```
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
```

52,81810219,32910。 2074141000。 rle.dfwrite.csv◦

## ◦ rep valueslengths

```
decompressed <- rep(rle.df$values, rle.df$lengths)
```

```
identical(decompressed, dat)
# [1] TRUE
```

## inverse.rle

```
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
```

dat

```
identical(dat.inv, dat)
# [1] TRUE
```

<https://riptutorial.com/zh-TW/r/topic/1133/>

# 113: + eval

parse°

eval°

## Examples

°

```
# 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
```

+ eval <https://riptutorial.com/zh-TW/r/topic/5746/-plus-eval>

---

# 114:

“”。

- 
- - 
  - [strsplit](#)
  -

## Examples

`is.character()`。 `as.character()`。

```
x <- "The quick brown fox jumps over the lazy dog"
class(x)
[1] "character"
is.character(x)
[1] TRUE
```

NA。

```
as.numeric("2")
[1] 2
as.numeric("fox")
[1] NA
Warning message:
NAs introduced by coercion
```

<https://riptutorial.com/zh-TW/r/topic/9017/>

# 115: RODE

- odeytimesfuncparmsmethod...

|       |       |
|-------|-------|
|       |       |
| ÿ     | ODE   |
|       | ;     |
| FUNC  | ODE   |
| PARMS | func  |
|       | lsoda |

- ““Lorenz”

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

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

## Examples

LorenzXYZ.

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

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

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

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

abc

$$a = -8/3$$

$$b = -10$$

$$c = 28$$

```
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)

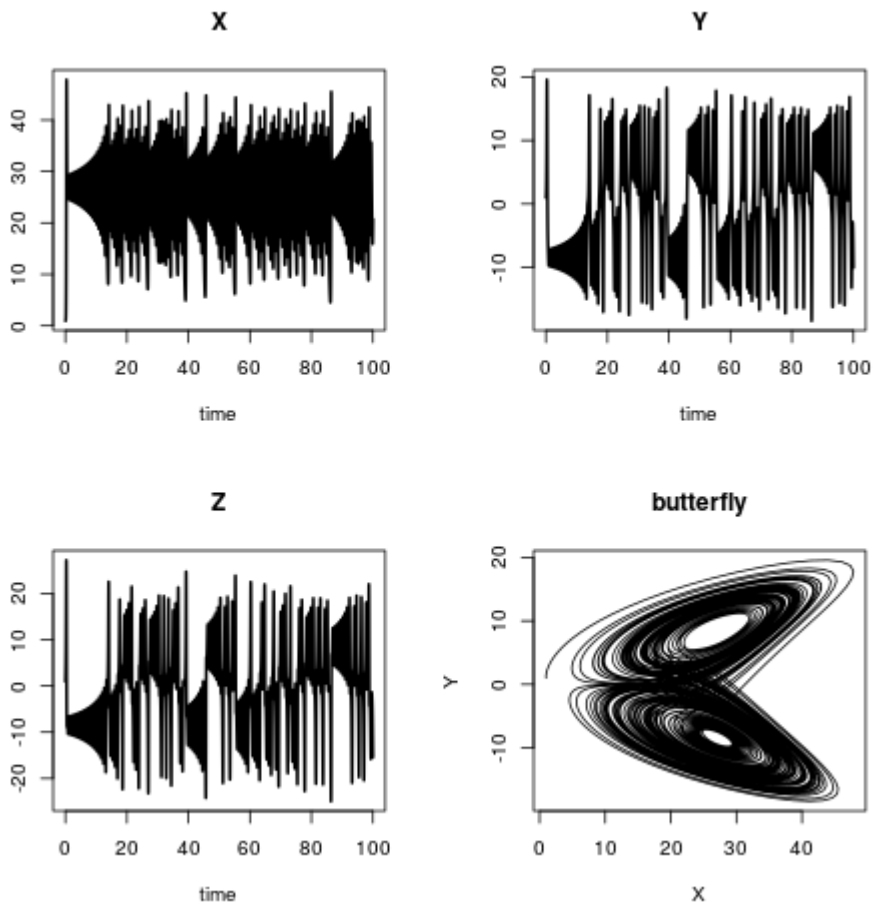
## -----
## Solve the ODEs
## -----

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")

```



## Lotka-Volterra

```
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
## -----
```

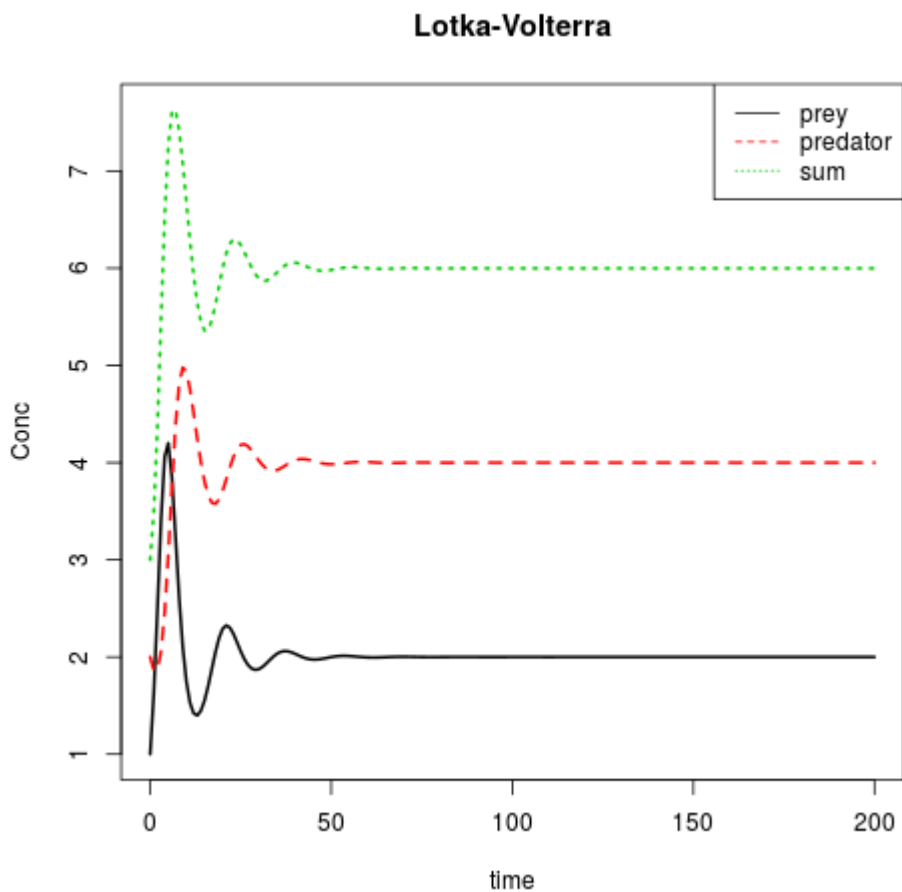
```

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)

```



## ODE - R

```

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)))
  })
}

```

## ODE - C

```

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]]

```

## ODE - fortran

```

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]]

```

## ODE -

### ODE - RODE - CODE - fortran.

```

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)
}

```

```

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

```

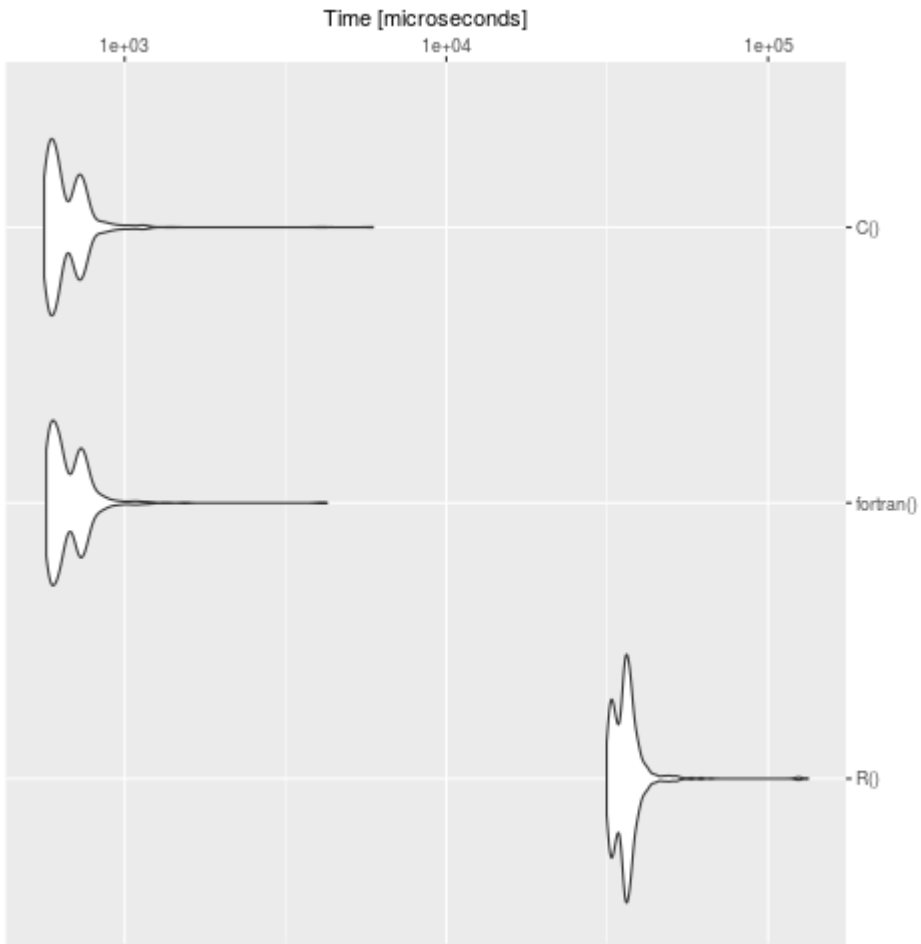
```

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   |



RCfortran ◦ ◦ c0deODERC ◦

RODE <https://riptutorial.com/zh-TW/r/topic/7448/rode>

# 116:

◦ R: `set` `setdiff` `intersect` `union` `setequal` `%in%` ◦ `v %in% S` `Sv` ◦

Rcpp ◦

## Examples

R

```
v = "A"
w = c("A", "A")
```

◦ R

```
setequal(v, w)
# TRUE
```

```
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
```

?`union` ◦

`%in%` ◦

```
v = "A"
w = c("A", "A")
```

```
w %in% v
# TRUE TRUE
```

```
v %in% w
# TRUE
```

◦

`%in%` `TRUE` `FALSE`

```
c(1, NA) %in% c(1, 2, 3, 4)
# TRUE FALSE
```

?`%in%`。

“”

**xyxY** 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
```

**data.frame**。 “” 。 unique lapplydo.call

```
m = do.call(expand.grid, lapply(list(X, Y), unique))

#   Var1 Var2
# 1    1    4
# 2    2    4
# 3    1    5
# 4    2    5
```

---

f(x,y)

```
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
```

outer

```
uX = unique(X)
uY = unique(Y)

outer(setNames(uX, uX), setNames(uY, uY), `*`)

#   4  5
# 1 4  5
# 2 8 10
```

◦

//

unique **drop**

```
x = c(2, 1, 1, 2, 1)

unique(x)
# 2 1
```

◦

duplicated

```
duplicated(x)
# FALSE FALSE TRUE TRUE TRUE
```

anyDuplicated(x) > 0L◦

/

```
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
```

◦

<https://riptutorial.com/zh-TW/r/topic/1383/>



# 117:

## Examples

browser() R

browser() R

|    |           |
|----|-----------|
|    |           |
| C  |           |
| F  |           |
| ñ  |           |
|    | Step Into |
| [R | “resume”  |
| Q  |           |

```
toDebug <- function() {  
  a = 1  
  b = 2  
  
  browser()  
  
  for(i in 1:100) {  
    a = a * b  
  }  
}  
  
toDebug()
```

```
Called from: toDebug  
Browser[1]>
```

```
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()

```
mtcars %>% group_by(cyl) %>% {browser() }
```

debug°

```
debug(mean)
mean(1:3)
```

° undebug°

```
undebug(mean)
mean(1:3)
```

debugonce °

```
debugonce(mean)
mean(1:3)
mean(1:3)
```

<https://riptutorial.com/zh-TW/r/topic/1695/>

# 118:

•

## Examples

R printcat R R

```
print("Hello World")
#[1] "Hello World"
cat("Hello World\n")
#Hello World
```

◦ `xx <- "Hello World"` print ◦

cat ◦ 1

```
cat(c("hello", "world", "\n"))
#hello world
```

\n

```
cat("Hello World")
#Hello World>
```

◦ RStudio ◦

print "" ◦ "Hello World" cat ◦ [1]

```
print("Hello World")
#[1] "Hello World"
```

R ◦ `sprint(s)` print("Hello World")

```
s <- "Hello World"
s
#[1] "Hello World"
```

```
"Hello World"
#[1] "Hello World"
```

c() ◦ **oncatenate** c print() cat

```
print(c("Hello World", "Here I am. "))
#[1] "Hello World" "Here I am. "
```

c() ◦ paste ◦ R ◦ R[1] ◦

```
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 ◦

print“numeric”“logical”

```
print(1)
#[1] 1
print(TRUE)
#[1] TRUE
```

**SO** ◦ catprint ◦ print()invisiblefooprint(foo) ◦ **RREPL**“- - -” ◦ cat ◦ print()**cat ()** ◦ message()  
warning() ;cat ◦ ◦

```
message("hello world")
#hello world
suppressMessages(message("hello world"))
```

◦ **Rmap-reduce** ◦ ◦ ◦

file()“r”

```
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
```

```
line <- readLines(conn, n=1, warn=FALSE)
```

connn=1 ◦ n**10,201020** ◦ n=-1 ◦

;writeLines(),cat() ◦ ◦ file()

```
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
```

```
writeLines("text",conn2, sep = "\n")
```

**Base R** ◦ ◦

```
system("top -a -b -n 1", intern = TRUE)
system2("top", "-a -b -n 1", stdout = TRUE)
```

◦

```
[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          "
[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                "

```

UNIX `top -a -b -n 1` ◦ ◦

devtools ◦ ◦

```
devtools::system_output("top", "-a -b -n 1")
```

`data.table` `fread` `shell` `read.table` ◦ `data.table` `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

```

`fread` **6** ◦

```
check.names = TRUE %CPU %MEM TIME+
```

<https://riptutorial.com/zh-TW/r/topic/5541/>

# 119:

## Examples

R `"R"` R `az Az 0-9 / . _`

- `list('.11' = "a")`

```
list( '.11' = "a")
#$`.11`
#[1] "a"
```

- `...ls()`

◦

foobar foo.bar foo\_bar .foobar

R `<-` operator =◦ ◦ `a<-1` `a <- 1` `a < -1` ◦

```
> foo <- 42
> fooEquals = 43
```

foo42 ◦ foo42 fooEquals43 ◦

```
> foo
[1] 42
> fooEquals
[1] 43
```

x

```
> (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.
```

->◦

```
> 5 -> x
> x
[1] 5
>
```

R ◦ 1 ◦

-

- `v <- c(2, 3, 7, 10) v2 <- c("a", "b", "c") v <- c(2, 3, 7, 10) v2 <- c("a", "b", "c")`◦
- ◦ `a <- matrix(data = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12), nrow = 4, ncol = 3, byrow = F)`◦ ◦ `a[1,2][1]` 5 ◦
- `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))`◦ ◦ `mylist$resultsmylist[[6]]`◦ `mylist[6]` **R**◦ `mylist[[6]][2]` **19**  
`mylist[6][2]`◦
- **data.frame**◦ ◦ `exam <- data.frame(matr = as.character(c(45020, 45679, 46789, 43126, 42345, 47568, 45674)), res_S = c(30, 19, 29, NA, 25, 26, 27), res_O = c(3, 3, 1, NA, 3, 2, NA), res_TOT = c(30,22,30,NA,28,28,27))`◦ `exam$matr` `exam[, 'matr']``exam[1]` `exam[,1]`◦  
`exam['rowname', ]``exam[1,]`◦ **rownames-attribute**

◦ ?Syntax?Syntax ◦ **R**◦

```
> 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))
```

```
> 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
```

```
> 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.....** ◦ `c + f`◦

## Matrix

```
> Z+W # matrix + matrix #(componentwise)
> Z*W # matrix* matrix#(Standard product is always componentwise)
```

## 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
```

“”

## R°

```
> foo <- 'foo'
> .foo <- 'bar'
```

ls°

```
> ls()
[1] "foo"
```

all.names = TRUE“private”

```
> ls(all.names = TRUE)
[1] ".foo"      "foo"
```

<https://riptutorial.com/zh-TW/r/topic/9013/>



# 120:

◦ ◦

## Examples

◦

```
x <- 1

foo <- function(x) {
  y <- 3
  z <- x + y
  return(z)
}

y
```

'y'

◦

```
foo <- function(x) {
  x <- 2
  y <- 3
  z <- x + y
  return(z)
}

foo(1)
x
```

1

◦

```
foo <- function() {
  y <- 3
  z <- x + y
  return(z)
}

foo()
```

4

◦

```
bar <- function() {
  z <- x + y
  return(z)
}
```

```

}

foo <- function() {
  y <- 3
  z <- bar()
  return(z)
}

foo()

```

## bar'y'

```

foo <- function() {

  bar <- function() {
    z <- x + y
    return(z)
  }

  y <- 3
  z <- bar()
  return(z)
}

foo()

```

## 4

<<-◦ bar()y◦

```

bar <- function() {
  z <- x + y
  return(z)
}

foo <- function() {
  y <<- 3
  z <- bar()
  return(z)
}

foo()

```

## 4

◦ ◦

R◦ ◦

package:basepackage:base◦

```

e1 <- new.env(parent = baseenv())
e2 <- new.env(parent = e1)

```

◦

```
assign("a", 3, envir = e1)
  get("a", envir = e1)
  get("a", envir = e2)
```

3

3

e2e1 e1e2 a3 ◦ e2ae1a◦

```
assign("a", 2, envir = e2)
  get("a", envir = e2)
  get("a", envir = e1)
```

3

2

on.exit()◦

◦ ◦

```
new_plot <- function(...) {
  old_pars <- par(mar = c(5,4,4,2) + .1, mfrow = c(1,1))
  on.exit(par(old_pars))
  plot(...)
}
```

◦ “”◦ ◦ ◦

```
library(plyr)
library(dplyr)
```

'dplyr'

'packageplyr'

descfailwithidmutaterenamesummarizesummarize

'packagestats'

'packagebase'

setdiffsetequalunion

package::function()◦

<https://riptutorial.com/zh-TW/r/topic/3138/>

---

# 121:

file.path ◦

dir ◦

## Examples

R ◦ ◦

◦ ◦

---

◦ *getwd()*?*setwd* ◦

```
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)
  )
```

## CSV ◦

---

◦ 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
```

*str*(file\_contents)?*rbind*?*lapply*?*lapply* ◦

---

?*read.table*?*write.table* ◦

- R
- - CSV
  - TSV
  - 
  -
- - SAS
  - SPSS
  - 
  -
- - MySQL
  - SQLite
  - PostgreSQL

<https://riptutorial.com/zh-TW/r/topic/5543/>

# 122:

subgsub ;

## Examples

pattern replacement

“surnameforename”“forename surname”。

```
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"
```

。

```
sub("^(.+),\\s(.+)", "\\1", strings)
# [1] "Sigg"      "Holt"      "Ortega"    "De La Torre" "Perkins"
```

。

```
2,14,14,14,19
```

```
2,14,19
```

gsub

```
gsub("(\\d+) (, \\1)+", "\\1", "2,14,14,14,19")
[1] "2,14,19"
```

```
> gsub("(\\d+) (, \\1)+", "\\1", "2,14,14,14,19,19,20,21")
[1] "2,14,19,20,21"
```

1. (\\d+) 1。 \\ \\\"'。 \\d [0-9]。

2. , ,

3. \\1 1。。

```
one,two,two,three,four,four,five,six
```

```
\d \w\d \w° [a-zA-Z0-9_]
```

```
> gsub("(\\w+) (, \\1)+", "\\1", "one,two,two,three,four,four,five,six")  
[1] "one,two,three,four,five,six"  
>
```

◦

<https://riptutorial.com/zh-TW/r/topic/9219/>

---

# 123: R

R `source(.)R`

`/file("stdin") stdout() stderr() stdin() R`

## Examples

R

---

## R

R Windows`R.exe Rscript Rscript.exe R`

**R Shebang line** `#!/usr/bin/env Rscript` Windows

R“hist.png”

```
#!/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()
```

R Shebang `cat("....\n") "file("stdin") scan read.table read.csv ... png(.) dev.off() jpeg(.)pdf(.) dev.off() .`

---

## R

## Linux/

`chmod +x PATH/TO/SCRIPT/SCRIPTNAME.R`

```
chmod +x PATH/TO/SCRIPT/SCRIPTNAME.R
```



o

```
"C:\Program Files\R-XXXXXXX\bin\Rscript.exe" "%~dp0\XXXXXXX.R" %*
```

\*.txt\*.bat o notepad Word "FILENAME.bat"o ""o

XXX...

- R
- o

```
"C:\...\Rscript.exe"WindowsRscript.exeo "%~dp0\XXX.R"RscriptR %~dp0o %*Ro
```

Ro Ro

## littlerR

[littler](#) [little](#) [r](#) [cran](#) [littler](#) [r](#) [Linux](#) [MacOS](#)o

## R

```
install.packages("littler")
```

r

```
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.
```

r

```
ln -s /home/*USER*/R/x86_64-pc-linux-gnu-library/3.4/littler/bin/r /usr/local/bin/r
```

## apt-getDebianUbuntu

```
sudo apt-get install littler
```

## .r`littler`

`littler` r Ro

```
# 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()
```

shebang ◦ hist.r

```
r hist.r
```

## *shebangedlittler*

shebanglittlerR

```
#!/usr/bin/env r
```

◦ `chmod +X /path/to/script.rR`◦

**R** <https://riptutorial.com/zh-TW/r/topic/9937/r>

# 124:

◦

TRUE FALSE NA; ◦ R TF TRUE FALSE TF ◦

## Examples

! | & xor() && || ◦ ifcond ◦

|     |  |        |
|-----|--|--------|
|     |  |        |
|     |  | X      |
|     |  | xy     |
| &&  |  | x && y |
|     |  | x   y  |
|     |  | x    y |
| XOR |  | XY     |

|| TRUE ◦ ◦ FALSE && FALSE ◦

```
> x <- 5
> x > 6 || stop("X is too small")
Error: X is too small
> x > 3 || stop("X is too small")
[1] TRUE
```

is.logical() ◦

as.logical() ◦

```
> x <- 2
> z <- x > 4
> z
[1] FALSE
> class(x)
[1] "numeric"
> as.logical(z)
[1] TRUE
```

as.numeric() double ◦ NANANA ◦

## NAs

o

```
> TRUE & NA
[1] NA
> FALSE & NA
[1] FALSE
> TRUE || NA
[1] TRUE
> FALSE || NA
[1] NA
```

<https://riptutorial.com/zh-TW/r/topic/9016/>

# 125: tidyr

tidyr gather spread ◦

◦

## Examples

### spread

```
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
```

“”

```
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
```

“”

```
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

```

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

```

"

```

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

```

tidyr <https://riptutorial.com/zh-TW/r/topic/9195/tidyr>

---

# 126:

## Examples

ShinyRStudioR。

Shiny

- .R
- ui.Rserver.R。
- **ui**。
- **server**。

---

```
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)
```

---

**ui.R**

```
library(shiny)

# Define UI for application
shinyUI(fluidPage(
  # Application title
  titlePanel("Hello World!")
))
```

**server.R**

```
library(shiny)

# Define server logic
shinyServer(function(input, output){})
```

。

- selected0
-

## HTML。

```
library(shiny)

ui <- fluidPage(
  radioButtons("radio",
    label = HTML('<FONT color="red"><FONT size="5pt">Welcome</FONT></FONT><br>
<b>Your favorite color is red ?</b>'),
    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 !"

◦ ◦

```
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"
```

- 
- 
- selectedNULL
- 
- 

HTML。

。

```
library(shiny)

ui <- fluidPage(
  selectInput("id_selectInput",
             label = HTML('<B><FONT size="3">What is your favorite color ?</FONT></B>'),
             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 ?

 red  green  blue  
 yellow

```
[1] "red" "green" "blue"
```

- 
- 
- selectedNULL
- TRUEFALSE
-

- selectizeTRUEFALSEselectize.js

HTML。

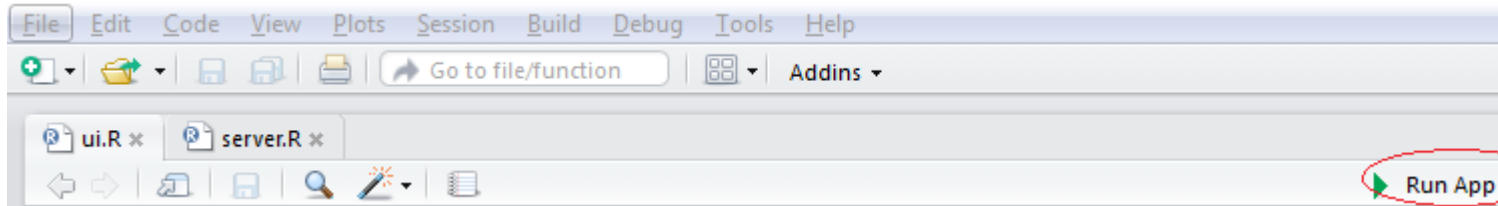
◦ ui.Rserver.R◦

# 1.

ui.Rserver.R◦ shinyApp() Shiny◦

```
shinyApp("path_to_the_folder_containing_the_files")
```

ui.Rserver.RRstudio“”Rstudio ◦



Shiny ApprunApp() ◦

# 2.

RshinyApp() ◦

- 

```
library(shiny)

ui <- fluidPage() #Create the ui
server <- function(input, output){} #create the server

shinyApp(ui = ui, server = server) #run the App
```

- appFileShiny.R

```
shinyApp(appFile="path_to_my_R_file_containig_the_app")
```

actionButton

checkboxGroupInput

checkboxInput

dateInput

dateRangeInput

|              |
|--------------|
| FileInput    |
| numericInput |
| selectInput  |
| sliderInput  |
| textInput    |

```

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)

```

`debug()` `debugonce()` [Shiny](#) `browser()` [Shiny](#) `browser()`

`server.R`

Showcase

- `display.mode = "showcase"` [Shiny app](#) `runApp("MyApp", display.mode = "showcase")`
- [Shiny app](#) `DESCRIPTION` `DisplayMode: Showcase`

## Reactive Log Visualizer

[Reactive Log Visualizer](#) `ReactiveLogVisualizerOptions(shiny.reactlog=TRUE)` `server.R`  
[Reactive Log Visualizer](#) `WindowsCtrl + F3` `MacCommand + F3` [Reactive Log Visualizer](#)

<https://riptutorial.com/zh-TW/r/topic/2044/>

# 127:

R. . .

```
rand = r * "runif" "run if" R.
```

## Examples

sampleurn.

```
set.seed = sample.set.seed
```

sample.

```
set.seed(1251)
sample(x = 10)

[1] 7 1 4 8 6 3 10 5 2 9
```

sample1x. . .

```
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
```

sample. . .

```
set.seed(7043)
sample(x = LETTERS, size = 7)
```

```
[1] "S" "P" "J" "F" "Z" "G" "R"
```

size $\times$  x $\times$

```
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'
```

◦

replacesample ◦ replace FALSE ◦ TRUE ◦

```
set.seed(5062)
sample(x = c("A", "B", "C", "D"), size = 8, replace = TRUE)

[1] "D" "C" "D" "B" "A" "A" "A" "A"
```

sample ◦ "" ◦ 20 ◦ 1/3 ◦

```
set.seed(6472)
sample(x = c("Red", "Blue", "Green"),
       size = 20,
       replace = TRUE)
```

211 ◦ xRed ◦ probsample ◦

prob ◦ 1/214 ◦

```
set.seed(28432)
sample(x = c("Red", "Blue", "Green"),
       size = 20,
       replace = TRUE,
       prob = c(0.50, 0.25, 0.25))
```

probprob1 ◦ 211 ◦

```
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
```

◦

replace FALSE prob ◦ prob ◦ size ◦

```
set.seed(21741)  
sample(x = c("Red", "Blue", "Green"),  
size = 2,  
replace = FALSE,  
prob = c(0.8,0.19,0.01))
```

◦ 80191◦

◦ 20191◦ 9519/2051/20◦

set.seed ◦ **R** set.seed set.seed ◦

```
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
```

<https://riptutorial.com/zh-TW/r/topic/9574/>



# 128:

## Examples

5

```
sample(5)
# [1] 4 5 3 1 2
```

```
sample(10:15)
# [1] 11 15 12 10 14 13
```

pracma

```
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
```

R `set.seed()`

```
> sample(1:10,5)
[1] 6 9 2 7 10
> sample(1:10,5)
[1] 7 6 1 2 10
```

```
> 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
```

1

```
> set.seed(1)
> sample(letters,2)
[1] "g" "j"
> set.seed(1)
> sample(letters,2)
[1] "g" "j"
```

```
rexp() rexp()
```

```
> 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
```

5.

## 010

```
runif(5, min=0, max=10)
[1] 2.1724399 8.9209930 6.1969249 9.3303321 2.4054102
```

## 01

```
rnorm(5, mean=0, sd=1)
[1] -0.97414402 -0.85722281 -0.08555494 -0.37444299 1.20032409
```

## 100.5

```
rbinom(5, size=10, prob=0.5)
[1] 4 3 5 2 3
```

## 0.2

```
rgeom(5, prob=0.2)
[1] 14 8 11 1 3
```

## 3105

```
rhyper(5, m=3, n=10, k=5)
[1] 2 0 1 1 1
```

## 100.8

```
rnbinom(5, size=10, prob=0.8)
[1] 3 1 3 4 2
```

## $\lambda^2$

```
rpois(5, lambda=2)
[1] 2 1 2 3 4
```

## 1.5

```
rexp(5, rate=1.5)
[1] 1.8993303 0.4799358 0.5578280 1.5630711 0.6228000
```

## 01

```
rlogis(5, location=0, scale=1)
[1] 0.9498992 -1.0287433 -0.4192311 0.7028510 -1.2095458
```

## 15

```
rchisq(5, df=15)
[1] 14.89209 19.36947 10.27745 19.48376 23.32898
```

## a = 1b = 0.5Beta

```
rbeta(5, shape1=1, shape2=0.5)
[1] 0.1670306 0.5321586 0.9869520 0.9548993 0.9999737
```

## Gamma3= 0.5

```
rgamma(5, shape=3, scale=0.5)
[1] 2.2445984 0.7934152 3.2366673 2.2897537 0.8573059
```

## Cauchy01

```
rcauchy(5, location=0, scale=1)
[1] -0.01285116 -0.38918446 8.71016696 10.60293284 -0.68017185
```

## 01

```
rlnorm(5, meanlog=0, sdlog=1)
[1] 0.8725009 2.9433779 0.3329107 2.5976206 2.8171894
```

## Weibull0.51

```
rweibull(5, shape=0.5, scale=1)
```

```
[1] 0.337599112 1.307774557 7.233985075 5.840429942 0.005751181
```

## Wilcoxon1020。

```
rwilcox(5, 10, 20)
[1] 111 88 93 100 124
```

## 53

```
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
```

<https://riptutorial.com/zh-TW/r/topic/1578/>

# 129:

RandomForest. [Wikipedia](#). RrandomForest. [CRAN](#).

## Examples

```
##### 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

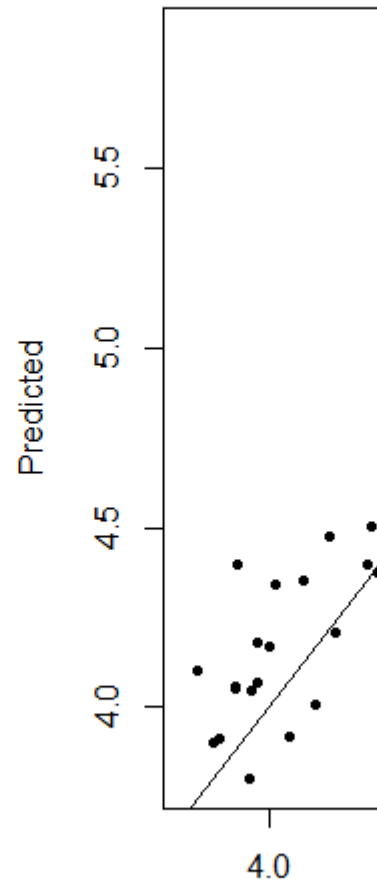
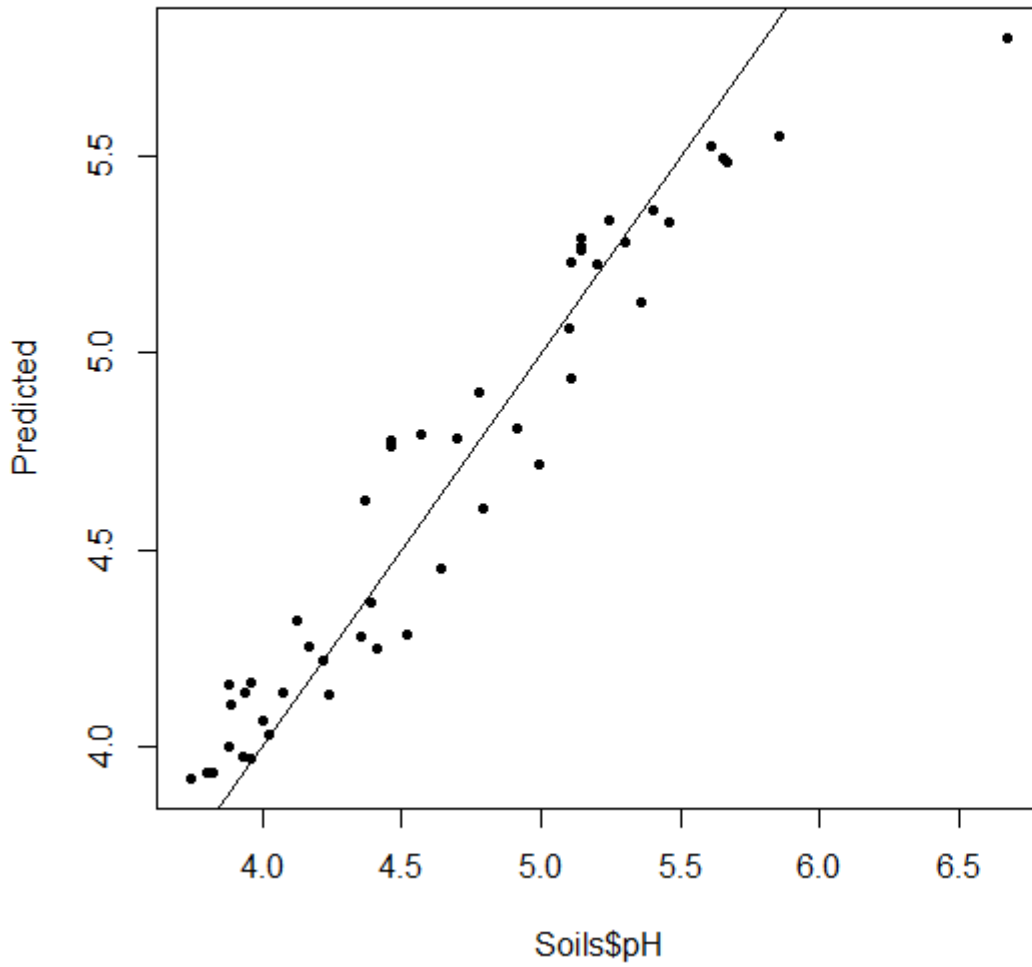
## Naive Bayes for comparison
library(e1071)
S_NB = naiveBayes(Soils[,6:14], Soils[,4])
Gp_NB = predict(S_NB, Soils[,6:14], type="class")
length(which(Gp_NB != Soils$Gp))          ## 6 Errors
```

## RF.

```
#####
## 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



<https://riptutorial.com/zh-TW/r/topic/8088/>

# 130:

Dplyr `NSE` `SE` 。

`summarise()` `summarise_()` 。

`lazyeval` `NSE` 。

## Examples

### dplyr

`NSE` 。

`dplyr` `lazyeval`

```
library(dplyr)
library(lazyeval)
```

### *NSE*

```
filter(mtcars, cyl == 8)
filter(mtcars, cyl < 6)
filter(mtcars, cyl < 6 & vs == 1)
```

### *SE*

```
filter_(mtcars, .dots = list(~ cyl == 8))
filter_(mtcars, .dots = list(~ cyl < 6))
filter_(mtcars, .dots = list(~ cyl < 6, ~ vs == 1))
```

### *NSE*

```
summarise(mtcars, mean_disp = mean(displ))
summarise(mtcars, mean_disp = mean(displ))
```

### *SE*

```
summarise_(mtcars, .dots = lazyeval::interp(~ mean(x), x = quote(displ)))
summarise_(mtcars, .dots = setNames(list(lazyeval::interp(~ mean(x), x = quote(displ))),
"mean_disp"))
summarise_(mtcars, .dots = list("mean_disp" = lazyeval::interp(~ mean(x), x = quote(displ))))
```

### *NSE*

```
mutate(mtcars, displ_l = displ / 61.0237)
```

SE

```
mutate_(
  .data = mtcars,
  .dots = list(
    "displ_l1" = lazyeval::interp(
      ~ x / 61.0237, x = quote(displ)
    )
  )
)
```

<https://riptutorial.com/zh-TW/r/topic/9365/>



# 131:

◦ `class-attribute` ◦ `class-attribute` ◦ `mode` ◦ `class` ◦ `class` ◦ `class<-` ◦ `function` ◦ S3S ◦ S4

“” ◦ `class` ◦ `mode` ◦ “” ◦ `typeof` ◦ `mode` ◦ `“class”` ◦ “”

## Examples

R ◦ `c()` ◦

```
c(1, 2, 3)
## [1] 1 2 3
c(TRUE, TRUE, FALSE)
## [1] TRUE TRUE FALSE
c("a", "b", "c")
## [1] "a" "b" "c"
```

`c()` ◦

```
x <- c(1, 2, 5)
y <- c(3, 4, 6)
z <- c(x, y)
z
## [1] 1 2 5 3 4 6
```

“”

R ◦ `class()` ◦ `str()` ◦

```
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"
```

`iris` ◦ `class` ◦ `data.frame` ◦ `str()` ◦ `Species` ◦ `str()` ◦ `class()` ◦

R ◦ ◦ `?atomic` ◦ `"logical"` ◦ `"integer"` ◦ `"numeric"` ◦ (synonym `"double"`) ◦ `"complex"` ◦ `"character"` ◦ `"raw"` ◦ `class`

```
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"
```

## R<sup>'''</sup>

```
mylist <- list( A = c(5,6,7,8), B = letters[1:10], CC = list( 5, "Z") )
```

- `f <- function(x) list(xplus = x + 10, xsq = x^2)`

```
f(7)
# $xplus
# [1] 17
#
# $xsq
# [1] 49
```

- ◦

```
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
```

## R<sup>'''</sup> -

<https://riptutorial.com/zh-TW/r/topic/3563/>

| S. No |                   | Contributors  |
|-------|-------------------|---|
| 1     | R                 | <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     | *                 | <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     | data.table        | <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>   |
| 5     | Date              | <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>  |
| 6     | 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>  |
| 7     | 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>  |
| 8     | GPU               | <a href="#">cdeterman</a>   |
| 9     | JSON              | <a href="#">SymbolixAU</a>  |
| 10    | lubridate         | <a href="#">alistaire</a> , <a href="#">Angelo</a> , <a href="#">Frank</a> , <a href="#">gitblame</a> , <a href="#">Hendrik</a> , <a href="#">scoa</a>  |
| 11    | Meta              | <a href="#">Frank</a> , <a href="#">Gregor</a> , <a href="#">Stephen Leppik</a> , <a href="#">Steve_Corrin</a>  |
| 12    | R MarkdownRStudio | <a href="#">dmail</a>   |
| 13    | R memento         | <a href="#">Lovy</a>  |

|    |                 |  |
|----|-----------------|--|
| 14 | RCP             | <a href="#">Artem Klevtsov</a> , <a href="#">coatless</a> , <a href="#">Dirk Eddelbuettel</a>  |
| 15 | RESTful R       | <a href="#">YCR</a>  |
| 16 | RMarkdownknitr  | <a href="#">Martin Schmelzer</a> , <a href="#">YCR</a>   |
| 17 | RMD             | <a href="#">J_F</a> , <a href="#">RamenChef</a>  |
| 18 | RODBC           | <a href="#">akrun</a> , <a href="#">Hack-R</a> , <a href="#">Parfait</a> , <a href="#">Tim Coker</a>   |
| 19 | roxygen2        | <a href="#">DeveauP</a> , <a href="#">PAC</a>  |
| 20 | R               | <a href="#">Alexey Shiklomanov</a>   |
| 21 | R -             | <a href="#">Joy</a>  |
| 22 | R               | <a href="#">Jon Ericson</a> , <a href="#">rcorty</a>   |
| 23 | R               | <a href="#">Axeman</a> , <a href="#">David Arenburg</a> , <a href="#">snaut</a>  |
| 24 | RLaTeXknitr     | <a href="#">JHowIX</a>   |
| 25 | RI / 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>  |
| 26 | R               | <a href="#">Pankaj Sharma</a>  |
| 27 | Spark APISparkR | <a href="#">Maximilian Kohl</a>  |
| 28 | sqldf           | <a href="#">Hack-R</a> , <a href="#">Miha</a>  |
| 29 | strsplit        | <a href="#">Imo</a>  |
| 30 | tidyverse       | <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>  |
| 31 | Web             | <a href="#">alistaire</a> , <a href="#">Dave2e</a>   |
| 32 | xgboost         | <a href="#">Hack-R</a>   |
| 33 |                 | <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>   |
| 34 |                 | <a href="#">Ben Bolker</a> , <a href="#">Glen Moutrie</a> , <a href="#">Jav</a> , <a href="#">SymbolixAU</a> , <a href="#">USER_1</a>  |
| 35 | CSVTSV          | <a href="#">a.powell</a> , <a href="#">Aaghaz Hussain</a> , <a href="#">abhiior</a> , <a href="#">Alex</a> , <a href="#">alistaire</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</a> |

|    |                            |   |
|----|----------------------------|---|
|    |                            | <a href="#">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>  |
| 36 | <a href="#">data.table</a> | <a href="#">Sun Bee</a>   |
| 37 | <a href="#">devtools</a>   | <a href="#">Frank</a> , <a href="#">Lovy</a>  |
| 38 | <a href="#">hclust</a>     | <a href="#">Frank</a> , <a href="#">G5W</a> , <a href="#">Tal Galili</a>  |
| 39 | <a href="#">igraph</a>     | <a href="#">Boysenb3rry</a>   |
| 40 | <a href="#">RMarkdown</a>  | <a href="#">ikashnitsky</a> , <a href="#">Karolis Koncevičius</a> , <a href="#">Martin Schmelzer</a>  |
| 41 | <a href="#">S4</a>         | <a href="#">David Leal</a>  |
| 42 | <a href="#">stringi</a>    | <a href="#">bartektartanus</a> , <a href="#">FisherDisinformation</a>   |
| 43 | <a href="#">texreg</a>     | <a href="#">Frank</a> , <a href="#">ikashnitsky</a>   |
| 44 |                            | <a href="#">Hack-R</a>  |
| 45 |                            | <a href="#">Frank</a> , <a href="#">loki</a>  |
| 46 | <a href="#">I / O.</a>     | <a href="#">Frank</a> , <a href="#">loki</a>  |
| 47 |                            | <a href="#">Jason</a>   |
| 48 |                            | <a href="#">Frank</a>   |
| 49 |                            | <a href="#">FisherDisinformation</a> , <a href="#">Frank</a> , <a href="#">L.V.Rao</a> , <a href="#">tenCupMaximum</a>  |
| 50 |                            | <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>   |
| 51 |                            | <a href="#">Ben Bolker</a>  |
| 52 |                            | <a href="#">akrun</a>   |
| 53 |                            | <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>                              |
| 54 |                            | <a href="#">Karolis Koncevičius</a>   |
| 55 |                            | <a href="#">egnha</a> , <a href="#">josliber</a>  |
| 56 | <a href="#">HashMap</a>    | <a href="#">nrussell</a> , <a href="#">russellpierce</a>  |
| 57 | <a href="#">R</a>          | <a href="#">Charmgoggles</a> , <a href="#">Frank</a> , <a href="#">ikashnitsky</a>  |
| 58 |                            | <a href="#">Frank</a> , <a href="#">USER_1</a>  |
| 59 |                            | <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>   |
| 60 |                         | <a href="#">ikashnitsky</a> , <a href="#">munirbe</a>  |
| 61 | R                       | <a href="#">AkselA</a> , <a href="#">ikashnitsky</a> , <a href="#">kaksat</a>  |
| 62 |                         | <a href="#">catastrophic-failure</a> , <a href="#">Jeff</a>  |
| 63 | <>                      | <a href="#">RobertMc</a>   |
| 64 |                         | <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>  |
| 65 |                         | <a href="#">4444</a> , <a href="#">AkselA</a> , <a href="#">alastaire</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>  |
| 66 | I / Oshapefile          | <a href="#">Alex</a> , <a href="#">Frank</a> , <a href="#">ikashnitsky</a>   |
| 67 |                         | <a href="#">Stephen Leppik</a> , <a href="#">tenCupMaximum</a>   |
| 68 |                         | <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>  |
| 69 | I / OExcelSASSPSS Stata | <a href="#">42-</a> , <a href="#">Alex</a> , <a href="#">alastaire</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>  |
| 70 |                         | <a href="#">42-</a> , <a href="#">Agriculturist</a> , <a href="#">alexis_laz</a> , <a href="#">alastaire</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>   |
| 71 |                         | <a href="#">Aaghaz Hussain</a> , <a href="#">akraf</a> , <a href="#">alko989</a> , <a href="#">Andrew Brēza</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> |
| 72 | /                       | <a href="#">Rappster</a>   |
| 73 |                         | <a href="#">Ben Bolker</a> , <a href="#">YCR</a>   |
| 74 |                         | <a href="#">42-</a> , <a href="#">Axeman</a> , <a href="#">Qaswed</a> , <a href="#">Sathish</a>  |
| 75 |                         | <a href="#">d.b</a>  |
| 76 |                         | <a href="#">Benjamin</a> , <a href="#">David Arenburg</a> , <a href="#">nrussell</a> , <a href="#">Robert</a> , <a href="#">Steve_Corrin</a>   |

|    |                |  |
|----|----------------|--|
| 77 |                | <a href="#">highBandWidth</a> , <a href="#">Steve_Corrin</a>   |
| 78 |                | <a href="#">Frank</a> , <a href="#">Steve_Corrin</a>   |
| 79 | I / O.         | <a href="#">Frank</a> , <a href="#">JHowIX</a> , <a href="#">SommerEngineering</a>   |
| 80 |                | <a href="#">ikashnitsky</a>  |
| 81 |                | <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> |
| 82 |                | <a href="#">Hack-R</a>   |
| 83 |                | <a href="#">Ben Bolker</a> , <a href="#">DataTx</a> , <a href="#">kneijenhuijs</a>   |
| 84 |                | <a href="#">AkselA</a> , <a href="#">alitaire</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>  |
| 85 | POSIXctPOSIXIt | <a href="#">AkselA</a> , <a href="#">alitaire</a> , <a href="#">coatless</a> , <a href="#">Frank</a> , <a href="#">MichaelChirico</a> , <a href="#">SymbolixAU</a> , <a href="#">thelatemail</a>   |
| 86 |                | <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>   |
| 87 | R              | <a href="#">Eric Lecoutre</a>  |
| 88 | R              | <a href="#">dmail</a>  |
| 89 |                | <a href="#">L.V.Rao</a>  |
| 90 |                | <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>   |
| 91 |                | <a href="#">loki</a>   |
| 92 |                | <a href="#">Frank</a> , <a href="#">Sowmya S. Manian</a>   |
| 93 |                | <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>   |
| 94 |                | <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>  |
| 95 |                | <a href="#">Derek Corcoran</a>   |
| 96 |                | <a href="#">AndreyAkinshin</a> , <a href="#">Nanami</a>  |

|     |        |   |
|-----|--------|---|
| 97  |        | Ashish, DeveauP   |
| 98  |        | 42-, Axeman, Hack-R, Marcin Kosiński  |
| 99  | R      | Umberto   |
| 100 |        | dayne, Frank  |
| 101 |        | beetroot, ikashnitsky, loki, maRtin   |
| 102 |        | Batanichek, FisherDisinformation, Matt Sandgren, Robert, russellpierce, Tensibai  |
| 103 | >      | 42-, Alexandru Papiu, Alihan Zihna, alistaire, AndreyAkinshin, Artem Klevtsov, Atish, Axeman, Benjamin, Carlos Cinelli, CMichael, DrPositron, Franck Dernoncourt, Frank, Gal Dreiman, Gavin Simpson, Gregor, ikashnitsky, James McCalden, Kay Brodersen, Matt, polka, RamenChef, Ryan Hilbert, Sam Firke, seasmith, Shawn Mehan, Simplans, Spacedman, SymbolixAU, thelatemail, tomw, TriskalJM, user2100721 |
| 104 |        | Carlos Cinelli, Christophe D., Karolis Koncevičius, L.V.Rao   |
| 105 |        | Frank, Karolis Koncevičius  |
| 106 | R      | Pankaj Sharma   |
| 107 |        | Amstell, Ben Bolker, Carl, Carlos Cinelli, David Robinson, fortune_p, Frank, highBandWidth, ikashnitsky, jaySf, Robert, russellpierce, thelatemail, USER_1, WAF   |
| 108 |        | Amit Kohli, Artem Klevtsov, Axeman, Eric Lecoutre, Frank, Gregor, Jaap, kitman0804, Imo, seasmith, Steve_Corrin, theArun, user2100721   |
| 109 |        | Florian, Frank  |
| 110 |        | CptNemo   |
| 111 |        | Andrew Bryk, Steve_Corrin   |
| 112 |        | Frank, josliber, Psidom   |
| 113 | + eval | YCR   |
| 114 |        | Frank, Steve_Corrin   |
| 115 | RODE   | J_F   |
| 116 |        | DeveauP, FisherDisinformation, Frank  |



|     |       |  |
|-----|-------|--|
| 117 |       | James Elderfield, russellpierce  |
| 118 |       | 42-, 4444, abhiieor, cdrini, dotancohen, Frank, Gregor, kdopen, Rich Scriven, Thomas, Uwe                  |
| 119 |       | 42-, Ale, Axeman, Craig Vermeer, Frank, L.V.Rao, Imckeogh  |
| 120 |       | Artem Klevtsov, K.Daisey, RamenChef  |
| 121 |       | Frank  |
| 122 |       | Alex, David Leal, Frank  |
| 123 | R     | akraf, herbaman  |
| 124 |       | 42-, Frank, Gregor, L.V.Rao, Steve_Corrin  |
| 125 | tidyr | Charmgoggles, Frank, Jeromy Anglim, SymbolixAU, user2100721  |
| 126 |       | alistaire, CClair, Christophe D., JvH, russellpierce, SymbolixAU, tuomastik, zx8754                        |
| 127 |       | TARehman   |
| 128 |       | bartektartanus, FisherDisinformation, Karolis Koncevičius, Miha, mnoronha                                  |
| 129 |       | G5W  |
| 130 |       | PAC  |
| 131 |       | 42-, AkselA, David Heckmann, dayne, Frank, Gregor, Jaap, kneijenhuijs, L.V.Rao, Nathan Werth, Steve_Corrin |