FREE eBook

# LEARNING dplyr

Unaffiliated free eBook created from
**Stack Overflow contributors.**

#dplyr

# Table of Contents


About ... 1

Chapter 1: Getting started with dplyr ... 2

Remarks ... 2

Examples ... 2

Helper Functions ... 2

starts_with ... 2

ends_with ... 3

contains ... 3

matches ... 4

num_range ... 4

one_of ... 5

everything ... 5

Other Helpers ... 6

: ... 6

- ... 6

Any combination of helper functions ... 7

Basic Verbs ... 9

select ... 9

filter ... 9

between ... 11

slice ... 12

mutate ... 12

arrange ... 14

group_by ... 15

summarise ... 16

rename ... 17

Installation or Setup ... 17

Credits ... 19

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: dplyr

It is an unofficial and free dplyr 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 dplyr.



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

# Chapter 1: Getting started with dplyr

## Remarks

This section provides an overview of what dplyr is, and why a developer might want to use it.

It should also mention any large subjects within dplyr, and link out to the related topics. Since the Documentation for dplyr is new, you may need to create initial versions of those related topics.

## Examples

### Helper Functions

Helper functions are used in conjunction with `select` to identify variables to return. Unless otherwise noted, these functions expect a string as the first parameter `match`. Passing a vector or another object will generate an error.

```
library(dplyr)
library(nycflights13)
```

## starts_with

`starts_with` allows us to identify variables whose name begins with a string.

Returns all variables that begin with the letter "e".

```
planes %>% select(starts_with("e"))
```

```
## # A tibble: 3,322 × 2
##    engines    engine
##      <int>     <chr>
## 1        2 Turbo-fan
## 2        2 Turbo-fan
## 3        2 Turbo-fan
## 4        2 Turbo-fan
## 5        2 Turbo-fan
## 6        2 Turbo-fan
## 7        2 Turbo-fan
## 8        2 Turbo-fan
## 9        2 Turbo-fan
## 10       2 Turbo-fan
## # ... with 3,312 more rows
```

Set `ignore.case` parameter to FALSE for strict casing.

```
planes %>% select(starts_with("E", ignore.case = FALSE))
```

```
## # A tibble: 3,322 × 0
```

## ends_with

Return all variables that end with the letter "e".

```
planes %>% select(ends_with("e"))
```

```
## # A tibble: 3,322 × 2
##                       type    engine
##                      <chr>     <chr>
## 1  Fixed wing multi engine Turbo-fan
## 2  Fixed wing multi engine Turbo-fan
## 3  Fixed wing multi engine Turbo-fan
## 4  Fixed wing multi engine Turbo-fan
## 5  Fixed wing multi engine Turbo-fan
## 6  Fixed wing multi engine Turbo-fan
## 7  Fixed wing multi engine Turbo-fan
## 8  Fixed wing multi engine Turbo-fan
## 9  Fixed wing multi engine Turbo-fan
## 10 Fixed wing multi engine Turbo-fan
## # ... with 3,312 more rows
```

Set `ignore.case` parameter to FALSE for strict casing.

```
planes %>% select(ends_with("E", ignore.case = FALSE))
```

```
## # A tibble: 3,322 × 0
```

## contains

`contains` allows you to find any variables that contain a given string.

```
planes %>% select(contains("ea"))
```

```
## # A tibble: 3,322 × 2
##     year seats
##    <int> <int>
## 1   2004    55
## 2   1998   182
## 3   1999   182
## 4   1999   182
## 5   2002    55
## 6   1999   182
## 7   1999   182
## 8   1999   182
## 9   1999   182
## 10  1999   182
## # ... with 3,312 more rows
```

Set `ignore.case` parameter to FALSE for strict casing.

```
planes %>% select(contains("EA", ignore.case = FALSE))
```

```
## # A tibble: 3,322 × 0
```

# matches

`matches` is the only helper function that allows the use of regular expressions.

Return all variables with a name at least six alpha characters:

```
planes %>% select(matches("[[:alpha:]]{6,}"))
```

```
## # A tibble: 3,322 × 4
##    tailnum     manufacturer engines    engine
##      <chr>            <chr>   <int>     <chr>
## 1   N10156          EMBRAER       2 Turbo-fan
## 2   N102UW AIRBUS INDUSTRIE       2 Turbo-fan
## 3   N103US AIRBUS INDUSTRIE       2 Turbo-fan
## 4   N104UW AIRBUS INDUSTRIE       2 Turbo-fan
## 5   N10575          EMBRAER       2 Turbo-fan
## 6   N105UW AIRBUS INDUSTRIE       2 Turbo-fan
## 7   N107US AIRBUS INDUSTRIE       2 Turbo-fan
## 8   N108UW AIRBUS INDUSTRIE       2 Turbo-fan
## 9   N109UW AIRBUS INDUSTRIE       2 Turbo-fan
## 10  N110UW AIRBUS INDUSTRIE       2 Turbo-fan
## # ... with 3,312 more rows
```

Set `ignore.case` parameter to FALSE for strict casing.

# num_range

For this example I will generate a dummy dataframe with random values and sequential variable names.

```
set.seed(1)
df <- data.frame(x1 = runif(10),
                 x2 = runif(10),
                 x3 = runif(10),
                 x4 = runif(10),
                 x5 = runif(10))
```

`num_range` can be used to select a range of varaibles given a consistent `prefix`.

Select the variables 2:4 from `df`:

```
df %>% select(num_range('x', range = 2:4))
```

```
##           x2         x3        x4
## 1  0.2059746 0.93470523 0.4820801
## 2  0.1765568 0.21214252 0.5995658
## 3  0.6870228 0.65167377 0.4935413
```

```
## 4  0.3841037 0.12555510 0.1862176
## 5  0.7698414 0.26722067 0.8273733
## 6  0.4976992 0.38611409 0.6684667
## 7  0.7176185 0.01339033 0.7942399
## 8  0.9919061 0.38238796 0.1079436
## 9  0.3800352 0.86969085 0.7237109
## 10 0.7774452 0.34034900 0.4112744
```

# one_of

`one_of` can take a vector as the `match` parameter and returns each variable.

```
planes %>% select(one_of(c("tailnum", "model")))
```

```
## # A tibble: 3,322 × 2
##    tailnum     model
##      <chr>     <chr>
## 1   N10156 EMB-145XR
## 2   N102UW  A320-214
## 3   N103US  A320-214
## 4   N104UW  A320-214
## 5   N10575 EMB-145LR
## 6   N105UW  A320-214
## 7   N107US  A320-214
## 8   N108UW  A320-214
## 9   N109UW  A320-214
## 10  N110UW  A320-214
## # ... with 3,312 more rows
```

## everything

`everything` can be used to reposition variables in the dataframe.

Make `manufacturer` the first variable followed by all remaining variables.

```
planes %>% select(manufacturer, everything())
```

```
## # A tibble: 3,322 × 9
##        manufacturer tailnum  year                    type     model
##               <chr>   <chr> <int>                   <chr>     <chr>
## 1           EMBRAER  N10156  2004 Fixed wing multi engine EMB-145XR
## 2  AIRBUS INDUSTRIE  N102UW  1998 Fixed wing multi engine  A320-214
## 3  AIRBUS INDUSTRIE  N103US  1999 Fixed wing multi engine  A320-214
## 4  AIRBUS INDUSTRIE  N104UW  1999 Fixed wing multi engine  A320-214
## 5           EMBRAER  N10575  2002 Fixed wing multi engine EMB-145LR
## 6  AIRBUS INDUSTRIE  N105UW  1999 Fixed wing multi engine  A320-214
## 7  AIRBUS INDUSTRIE  N107US  1999 Fixed wing multi engine  A320-214
## 8  AIRBUS INDUSTRIE  N108UW  1999 Fixed wing multi engine  A320-214
## 9  AIRBUS INDUSTRIE  N109UW  1999 Fixed wing multi engine  A320-214
## 10 AIRBUS INDUSTRIE  N110UW  1999 Fixed wing multi engine  A320-214
## # ... with 3,312 more rows, and 4 more variables: engines <int>,
## #   seats <int>, speed <int>, engine <chr>
```

## Other Helpers

Though the `:` and `-` operators are not part of the `dplyr` package we can still use them to identify variables to return.

`:`

Define an inclusive range of variables to return.

Return every variable from `year` to `manufacturer`:

```
planes %>% select(year:manufacturer)
```

```
## # A tibble: 3,322 × 3
##     year                   type     manufacturer
##    <int>                  <chr>            <chr>
## 1   2004 Fixed wing multi engine          EMBRAER
## 2   1998 Fixed wing multi engine AIRBUS INDUSTRIE
## 3   1999 Fixed wing multi engine AIRBUS INDUSTRIE
## 4   1999 Fixed wing multi engine AIRBUS INDUSTRIE
## 5   2002 Fixed wing multi engine          EMBRAER
## 6   1999 Fixed wing multi engine AIRBUS INDUSTRIE
## 7   1999 Fixed wing multi engine AIRBUS INDUSTRIE
## 8   1999 Fixed wing multi engine AIRBUS INDUSTRIE
## 9   1999 Fixed wing multi engine AIRBUS INDUSTRIE
## 10  1999 Fixed wing multi engine AIRBUS INDUSTRIE
## # ... with 3,312 more rows
```

Return multiple ranges of variables:

```
planes %>% select(c(year:manufacturer, seats:engine))
```

```
## # A tibble: 3,322 × 6
##     year                   type     manufacturer seats speed    engine
##    <int>                  <chr>            <chr> <int> <int>     <chr>
## 1   2004 Fixed wing multi engine          EMBRAER    55    NA Turbo-fan
## 2   1998 Fixed wing multi engine AIRBUS INDUSTRIE   182    NA Turbo-fan
## 3   1999 Fixed wing multi engine AIRBUS INDUSTRIE   182    NA Turbo-fan
## 4   1999 Fixed wing multi engine AIRBUS INDUSTRIE   182    NA Turbo-fan
## 5   2002 Fixed wing multi engine          EMBRAER    55    NA Turbo-fan
## 6   1999 Fixed wing multi engine AIRBUS INDUSTRIE   182    NA Turbo-fan
## 7   1999 Fixed wing multi engine AIRBUS INDUSTRIE   182    NA Turbo-fan
## 8   1999 Fixed wing multi engine AIRBUS INDUSTRIE   182    NA Turbo-fan
## 9   1999 Fixed wing multi engine AIRBUS INDUSTRIE   182    NA Turbo-fan
## 10  1999 Fixed wing multi engine AIRBUS INDUSTRIE   182    NA Turbo-fan
## # ... with 3,312 more rows
```

`-`

The `-` operator will remove a variable from a result set.

Return all variables with the exception of `type`:

```
planes %>% select(-type)
```

```
## # A tibble: 3,322 × 8
##    tailnum  year     manufacturer     model engines seats speed    engine
##      <chr> <int>            <chr>     <chr>   <int> <int> <int>     <chr>
## 1   N10156  2004          EMBRAER EMB-145XR       2    55    NA Turbo-fan
## 2   N102UW  1998 AIRBUS INDUSTRIE  A320-214       2   182    NA Turbo-fan
## 3   N103US  1999 AIRBUS INDUSTRIE  A320-214       2   182    NA Turbo-fan
## 4   N104UW  1999 AIRBUS INDUSTRIE  A320-214       2   182    NA Turbo-fan
## 5   N10575  2002          EMBRAER EMB-145LR       2    55    NA Turbo-fan
## 6   N105UW  1999 AIRBUS INDUSTRIE  A320-214       2   182    NA Turbo-fan
## 7   N107US  1999 AIRBUS INDUSTRIE  A320-214       2   182    NA Turbo-fan
## 8   N108UW  1999 AIRBUS INDUSTRIE  A320-214       2   182    NA Turbo-fan
## 9   N109UW  1999 AIRBUS INDUSTRIE  A320-214       2   182    NA Turbo-fan
## 10  N110UW  1999 AIRBUS INDUSTRIE  A320-214       2   182    NA Turbo-fan
## # ... with 3,312 more rows
```

You can also pass a vector of variable names to exclude from your result set.

```
planes %>% select(-c(type, engines:engine))
```

```
## # A tibble: 3,322 × 4
##    tailnum  year     manufacturer     model
##      <chr> <int>            <chr>     <chr>
## 1   N10156  2004          EMBRAER EMB-145XR
## 2   N102UW  1998 AIRBUS INDUSTRIE  A320-214
## 3   N103US  1999 AIRBUS INDUSTRIE  A320-214
## 4   N104UW  1999 AIRBUS INDUSTRIE  A320-214
## 5   N10575  2002          EMBRAER EMB-145LR
## 6   N105UW  1999 AIRBUS INDUSTRIE  A320-214
## 7   N107US  1999 AIRBUS INDUSTRIE  A320-214
## 8   N108UW  1999 AIRBUS INDUSTRIE  A320-214
## 9   N109UW  1999 AIRBUS INDUSTRIE  A320-214
## 10  N110UW  1999 AIRBUS INDUSTRIE  A320-214
## # ... with 3,312 more rows
```

# Any combination of helper functions

Select all variables between `type` and `speed` (inclusive) and exclude `manufacturer`.

```
planes %>% select(type:speed, -manufacturer)
```

```
## # A tibble: 3,322 × 5
##                        type     model engines seats speed
##                       <chr>     <chr>   <int> <int> <int>
## 1  Fixed wing multi engine EMB-145XR       2    55    NA
## 2  Fixed wing multi engine  A320-214       2   182    NA
## 3  Fixed wing multi engine  A320-214       2   182    NA
## 4  Fixed wing multi engine  A320-214       2   182    NA
## 5  Fixed wing multi engine EMB-145LR       2    55    NA
## 6  Fixed wing multi engine  A320-214       2   182    NA
## 7  Fixed wing multi engine  A320-214       2   182    NA
## 8  Fixed wing multi engine  A320-214       2   182    NA
## 9  Fixed wing multi engine  A320-214       2   182    NA
## 10 Fixed wing multi engine  A320-214       2   182    NA
```

```
## # ... with 3,312 more rows
```

Modify the previous statement to exclude `manufacturer` and `model`.

```
planes %>% select(type:speed, -c(manufacturer, model))
```

```
## # A tibble: 3,322 × 4
##                       type engines seats speed
##                      <chr>   <int> <int> <int>
## 1  Fixed wing multi engine       2    55    NA
## 2  Fixed wing multi engine       2   182    NA
## 3  Fixed wing multi engine       2   182    NA
## 4  Fixed wing multi engine       2   182    NA
## 5  Fixed wing multi engine       2    55    NA
## 6  Fixed wing multi engine       2   182    NA
## 7  Fixed wing multi engine       2   182    NA
## 8  Fixed wing multi engine       2   182    NA
## 9  Fixed wing multi engine       2   182    NA
## 10 Fixed wing multi engine       2   182    NA
## # ... with 3,312 more rows
```

You can use the same helper function more than once.

```
planes %>% select(starts_with("m"), starts_with("s"))
```

```
## # A tibble: 3,322 × 4
##        manufacturer     model seats speed
##               <chr>     <chr> <int> <int>
## 1           EMBRAER EMB-145XR    55    NA
## 2  AIRBUS INDUSTRIE  A320-214   182    NA
## 3  AIRBUS INDUSTRIE  A320-214   182    NA
## 4  AIRBUS INDUSTRIE  A320-214   182    NA
## 5           EMBRAER EMB-145LR    55    NA
## 6  AIRBUS INDUSTRIE  A320-214   182    NA
## 7  AIRBUS INDUSTRIE  A320-214   182    NA
## 8  AIRBUS INDUSTRIE  A320-214   182    NA
## 9  AIRBUS INDUSTRIE  A320-214   182    NA
## 10 AIRBUS INDUSTRIE  A320-214   182    NA
## # ... with 3,312 more rows
```

You can use multiple helper functions together:

```
planes %>% select(starts_with("m"), ends_with("l"))
```

```
## # A tibble: 3,322 × 2
##        manufacturer     model
##               <chr>     <chr>
## 1           EMBRAER EMB-145XR
## 2  AIRBUS INDUSTRIE  A320-214
## 3  AIRBUS INDUSTRIE  A320-214
## 4  AIRBUS INDUSTRIE  A320-214
## 5           EMBRAER EMB-145LR
## 6  AIRBUS INDUSTRIE  A320-214
## 7  AIRBUS INDUSTRIE  A320-214
## 8  AIRBUS INDUSTRIE  A320-214
```

```
## 9  AIRBUS INDUSTRIE  A320-214
## 10 AIRBUS INDUSTRIE  A320-214
## # ... with 3,312 more rows
```

### Basic Verbs

```
library(dplyr)
library(nycflights13)
```

There are several verbs most commonly used in `dplyr` to modify datasets.

## select

Select `tailnum`, `type`, `model` variables from the dataframe `planes`:

```
select(planes, tailnum, type, model)
```

```
## # A tibble: 3,322 × 3
##    tailnum                type     model
##      <chr>                <chr>     <chr>
## 1   N10156 Fixed wing multi engine EMB-145XR
## 2   N102UW Fixed wing multi engine  A320-214
## 3   N103US Fixed wing multi engine  A320-214
## 4   N104UW Fixed wing multi engine  A320-214
## 5   N10575 Fixed wing multi engine EMB-145LR
## 6   N105UW Fixed wing multi engine  A320-214
## 7   N107US Fixed wing multi engine  A320-214
## 8   N108UW Fixed wing multi engine  A320-214
## 9   N109UW Fixed wing multi engine  A320-214
## 10  N110UW Fixed wing multi engine  A320-214
## # ... with 3,312 more rows
```

Rewrite the statement above with the forward-pipe operator (`%>%`) from the magrittr package:

```
planes %>% select(tailnum, type, model)
```

```
## # A tibble: 3,322 × 3
##    tailnum                type     model
##      <chr>                <chr>     <chr>
## 1   N10156 Fixed wing multi engine EMB-145XR
## 2   N102UW Fixed wing multi engine  A320-214
## 3   N103US Fixed wing multi engine  A320-214
## 4   N104UW Fixed wing multi engine  A320-214
## 5   N10575 Fixed wing multi engine EMB-145LR
## 6   N105UW Fixed wing multi engine  A320-214
## 7   N107US Fixed wing multi engine  A320-214
## 8   N108UW Fixed wing multi engine  A320-214
## 9   N109UW Fixed wing multi engine  A320-214
## 10  N110UW Fixed wing multi engine  A320-214
## # ... with 3,312 more rows
```

## filter

`filter` rows based on crieria.

Return a dataset where `manufacturer` is "EMBRAER":

```
planes %>% filter(manufacturer == "EMBRAER")
```

```
## # A tibble: 299 × 9
##    tailnum  year                  type manufacturer     model engines
##      <chr> <int>                 <chr>        <chr>     <chr>   <int>
## 1   N10156  2004 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 2   N10575  2002 Fixed wing multi engine      EMBRAER EMB-145LR       2
## 3   N11106  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 4   N11107  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 5   N11109  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 6   N11113  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 7   N11119  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 8   N11121  2003 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 9   N11127  2003 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 10  N11137  2003 Fixed wing multi engine      EMBRAER EMB-145XR       2
## # ... with 289 more rows, and 3 more variables: seats <int>, speed <int>,
## #   engine <chr>
```

Return a dataset where `manufacturer` is "EMBRAER" and `model` is "EMB-145XR":

```
planes %>%
  filter(manufacturer == "EMBRAER", model == "EMB-145XR")
```

```
## # A tibble: 104 × 9
##    tailnum  year                  type manufacturer     model engines
##      <chr> <int>                 <chr>        <chr>     <chr>   <int>
## 1   N10156  2004 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 2   N11106  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 3   N11107  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 4   N11109  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 5   N11113  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 6   N11119  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 7   N11121  2003 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 8   N11127  2003 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 9   N11137  2003 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 10  N11140  2003 Fixed wing multi engine      EMBRAER EMB-145XR       2
## # ... with 94 more rows, and 3 more variables: seats <int>, speed <int>,
## #   engine <chr>
```

The statement above is the same as writing an "AND" condition.

```
planes %>% filter(manufacturer == "EMBRAER" & model == "EMB-145XR")
```

```
## # A tibble: 104 × 9
##    tailnum  year                  type manufacturer     model engines
##      <chr> <int>                 <chr>        <chr>     <chr>   <int>
## 1   N10156  2004 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 2   N11106  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 3   N11107  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 4   N11109  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 5   N11113  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
```

```
## 6   N11119  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 7   N11121  2003 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 8   N11127  2003 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 9   N11137  2003 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 10  N11140  2003 Fixed wing multi engine      EMBRAER EMB-145XR       2
## # ... with 94 more rows, and 3 more variables: seats <int>, speed <int>,
## #   engine <chr>
```

Use the pipe (|) character for "OR" conditions:

```
planes %>% filter(manufacturer == "EMBRAER" | model == "EMB-145XR")
```

```
## # A tibble: 299 × 9
##    tailnum  year                    type manufacturer     model engines
##      <chr> <int>                   <chr>        <chr>     <chr>   <int>
## 1   N10156  2004 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 2   N10575  2002 Fixed wing multi engine      EMBRAER EMB-145LR       2
## 3   N11106  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 4   N11107  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 5   N11109  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 6   N11113  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 7   N11119  2002 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 8   N11121  2003 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 9   N11127  2003 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 10  N11137  2003 Fixed wing multi engine      EMBRAER EMB-145XR       2
## # ... with 289 more rows, and 3 more variables: seats <int>, speed <int>,
## #   engine <chr>
```

Use `grepl` in combination with `filter` for pattern-matching conditions.

```
planes %>% filter(grepl("^172.", model))
```

```
## # A tibble: 3 × 9
##   tailnum  year                    type manufacturer model engines seats
##     <chr> <int>                   <chr>        <chr> <chr>   <int> <int>
## 1  N378AA  1963 Fixed wing single engine       CESSNA  172E       1     4
## 2  N621AA  1975 Fixed wing single engine       CESSNA  172M       1     4
## 3  N737MQ  1977 Fixed wing single engine       CESSNA  172N       1     4
## # ... with 2 more variables: speed <int>, engine <chr>
```

## between

Return all rows where `year` is `between` 2004 and 2005:

```
planes %>% filter(between(year, 2004, 2005))
```

```
## # A tibble: 354 × 9
##    tailnum  year                    type manufacturer     model engines
##      <chr> <int>                   <chr>        <chr>     <chr>   <int>
## 1   N10156  2004 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 2   N11155  2004 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 3   N11164  2004 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 4   N11165  2004 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 5   N11176  2004 Fixed wing multi engine      EMBRAER EMB-145XR       2
```

```
## 6   N11181  2005 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 7   N11184  2005 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 8   N11187  2005 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 9   N11189  2005 Fixed wing multi engine      EMBRAER EMB-145XR       2
## 10  N11191  2005 Fixed wing multi engine      EMBRAER EMB-145XR       2
## # ... with 344 more rows, and 3 more variables: seats <int>, speed <int>,
## #   engine <chr>
```

# slice

`slice` returns only rows by the given index.

Return the first five rows of data (same as the base `head` function):

```
planes %>% slice(1:5)
```

```
## # A tibble: 5 × 9
##   tailnum  year                  type     manufacturer     model engines
##     <chr> <int>                 <chr>            <chr>     <chr>   <int>
## 1  N10156  2004 Fixed wing multi engine          EMBRAER EMB-145XR       2
## 2  N102UW  1998 Fixed wing multi engine AIRBUS INDUSTRIE  A320-214       2
## 3  N103US  1999 Fixed wing multi engine AIRBUS INDUSTRIE  A320-214       2
## 4  N104UW  1999 Fixed wing multi engine AIRBUS INDUSTRIE  A320-214       2
## 5  N10575  2002 Fixed wing multi engine          EMBRAER EMB-145LR       2
## # ... with 3 more variables: seats <int>, speed <int>, engine <chr>
```

Return the 1st, 3rd, and 5th rows of data:

```
planes %>% slice(c(1, 3, 5))
```

```
## # A tibble: 3 × 9
##   tailnum  year                  type     manufacturer     model engines
##     <chr> <int>                 <chr>            <chr>     <chr>   <int>
## 1  N10156  2004 Fixed wing multi engine          EMBRAER EMB-145XR       2
## 2  N103US  1999 Fixed wing multi engine AIRBUS INDUSTRIE  A320-214       2
## 3  N10575  2002 Fixed wing multi engine          EMBRAER EMB-145LR       2
## # ... with 3 more variables: seats <int>, speed <int>, engine <chr>
```

Return the first and last rows:

```
planes %>% slice(c(1, nrow(planes)))
```

```
## # A tibble: 2 × 9
##   tailnum  year                  type                  manufacturer
##     <chr> <int>                 <chr>                         <chr>
## 1  N10156  2004 Fixed wing multi engine                       EMBRAER
## 2  N999DN  1992 Fixed wing multi engine MCDONNELL DOUGLAS CORPORATION
## # ... with 5 more variables: model <chr>, engines <int>, seats <int>,
## #   speed <int>, engine <chr>
```

# mutate

`mutate` can add new variables or modify existing variables.

Add a dummy variable, `engine.dummy` with a default value of 0:

```
planes %>%
  mutate(engine.dummy = 0) %>%
  select(engine, engine.dummy)
```

```
## # A tibble: 3,322 × 2
##       engine engine.dummy
##        <chr>        <dbl>
## 1  Turbo-fan            0
## 2  Turbo-fan            0
## 3  Turbo-fan            0
## 4  Turbo-fan            0
## 5  Turbo-fan            0
## 6  Turbo-fan            0
## 7  Turbo-fan            0
## 8  Turbo-fan            0
## 9  Turbo-fan            0
## 10 Turbo-fan            0
## # ... with 3,312 more rows
```

Using `dplyr::if_else`, add `engine.dummy` set to 1 if `engine` == "Turbo-fan", otherwise set `engine.dummy` to 0:

```
planes %>%
  mutate(engine.dummy = if_else(engine == "Turbo-fan", 1, 0)) %>%
  select(engine, engine.dummy)
```

```
## # A tibble: 3,322 × 2
##       engine engine.dummy
##        <chr>        <dbl>
## 1  Turbo-fan            1
## 2  Turbo-fan            1
## 3  Turbo-fan            1
## 4  Turbo-fan            1
## 5  Turbo-fan            1
## 6  Turbo-fan            1
## 7  Turbo-fan            1
## 8  Turbo-fan            1
## 9  Turbo-fan            1
## 10 Turbo-fan            1
## # ... with 3,312 more rows
```

Convert `planes$engine` to a factor.

```
planes %>%
  mutate(engine = as.factor(engine)) %>%
  select(engine)
```

```
## # A tibble: 3,322 × 1
##       engine
##       <fctr>
## 1  Turbo-fan
```

```
## 2  Turbo-fan
## 3  Turbo-fan
## 4  Turbo-fan
## 5  Turbo-fan
## 6  Turbo-fan
## 7  Turbo-fan
## 8  Turbo-fan
## 9  Turbo-fan
## 10 Turbo-fan
## # ... with 3,312 more rows
```

# arrange

Use `arrange` to sort your dataframe.

Arrange `planes` by `year`:

```
planes %>% arrange(year)
```

```
## # A tibble: 3,322 × 9
##    tailnum  year                     type manufacturer       model engines
##      <chr> <int>                    <chr>        <chr>       <chr>   <int>
## 1   N381AA  1956  Fixed wing multi engine      DOUGLAS      DC-7BF       4
## 2   N201AA  1959 Fixed wing single engine       CESSNA         150       1
## 3   N567AA  1959 Fixed wing single engine  DEHAVILLAND OTTER DHC-3       1
## 4   N378AA  1963 Fixed wing single engine       CESSNA        172E       1
## 5   N575AA  1963 Fixed wing single engine       CESSNA  210-5(205)       1
## 6   N14629  1965  Fixed wing multi engine       BOEING     737-524       2
## 7   N615AA  1967  Fixed wing multi engine        BEECH      65-A90       2
## 8   N425AA  1968 Fixed wing single engine        PIPER   PA-28-180       1
## 9   N383AA  1972  Fixed wing multi engine        BEECH        E-90       2
## 10  N364AA  1973  Fixed wing multi engine       CESSNA        310Q       2
## # ... with 3,312 more rows, and 3 more variables: seats <int>,
## #   speed <int>, engine <chr>
```

`arrange planes` by `year desc`:

```
planes %>% arrange(desc(year))
```

```
## # A tibble: 3,322 × 9
##    tailnum  year                     type manufacturer    model engines
##      <chr> <int>                    <chr>        <chr>    <chr>   <int>
## 1   N150UW  2013 Fixed wing multi engine       AIRBUS A321-211       2
## 2   N151UW  2013 Fixed wing multi engine       AIRBUS A321-211       2
## 3   N152UW  2013 Fixed wing multi engine       AIRBUS A321-211       2
## 4   N153UW  2013 Fixed wing multi engine       AIRBUS A321-211       2
## 5   N154UW  2013 Fixed wing multi engine       AIRBUS A321-211       2
## 6   N155UW  2013 Fixed wing multi engine       AIRBUS A321-211       2
## 7   N156UW  2013 Fixed wing multi engine       AIRBUS A321-211       2
## 8   N157UW  2013 Fixed wing multi engine       AIRBUS A321-211       2
## 9   N198UW  2013 Fixed wing multi engine       AIRBUS A321-211       2
## 10  N199UW  2013 Fixed wing multi engine       AIRBUS A321-211       2
## # ... with 3,312 more rows, and 3 more variables: seats <int>,
## #   speed <int>, engine <chr>
```

# group_by

`group_by` allows you to perform operations on a dataframe by subsets without extracting the subset.

```
df <- planes %>% group_by(manufacturer, model)
```

The returned dataframe may not appear grouped. However, the `class` and `attributes` of the dataframe will confirm it is.

```
class(df)
```

```
## [1] "grouped_df" "tbl_df"     "tbl"        "data.frame"
```

```
attributes(df)$vars
```

```
## [[1]]
## manufacturer
##
## [[2]]
## model
```

```
head(attributes(df)$labels, n = 5L)
```

```
##   manufacturer    model
## 1   AGUSTA SPA    A109E
## 2       AIRBUS A319-112
## 3       AIRBUS A319-114
## 4       AIRBUS A319-115
## 5       AIRBUS A319-131
```

If you wish to add grouping elements to the dataframe without removing existing grouping elements, use the `add` parameter set to TRUE (set to FALSE by default):

```
df <- df %>% group_by(type, year, add = TRUE)
```

```
class(df)
```

```
## [1] "grouped_df" "tbl_df"     "tbl"        "data.frame"
```

```
attributes(df)$vars
```

```
## [[1]]
## manufacturer
##
## [[2]]
## model
##
## [[3]]
```

```
## type
##
## [[4]]
## year
```

```
head(attributes(df)$labels, n = 5L)
```

```
##   manufacturer    model                   type year
## 1   AGUSTA SPA    A109E              Rotorcraft 2001
## 2       AIRBUS A319-112 Fixed wing multi engine 2002
## 3       AIRBUS A319-112 Fixed wing multi engine 2005
## 4       AIRBUS A319-112 Fixed wing multi engine 2006
## 5       AIRBUS A319-112 Fixed wing multi engine 2007
```

If you want to remove grouping use `ungroup`.

```
df <- df %>% ungroup()
```

```
class(df)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```
attributes(df)$vars
```

```
## NULL
```

```
attributes(df)$labels
```

```
## NULL
```

# summarise

`summarise` is used to perform calculations on a dataset either as a whole or by groups.

Find the `mean` number of of `seats` per `manufacturer`?

```
planes %>%
  group_by(manufacturer) %>%
  summarise(Mean = mean(seats))
```

```
## # A tibble: 35 × 2
##              manufacturer     Mean
##                     <chr>    <dbl>
## 1              AGUSTA SPA   8.0000
## 2                  AIRBUS 221.2024
## 3        AIRBUS INDUSTRIE 187.4025
## 4   AMERICAN AIRCRAFT INC   2.0000
## 5      AVIAT AIRCRAFT INC   2.0000
## 6  AVIONS MARCEL DASSAULT  12.0000
## 7           BARKER JACK L   2.0000
```

```
## 8                   BEECH   9.5000
## 9                    BELL   8.0000
## 10                 BOEING 175.1877
## # ... with 25 more rows
```

`summarise` will not return variables that are not explicitly grouped or included in summary functions. If you want to add another variable you must pass it as a predicate to `group_by` or `summarise`.

```
planes %>%
  group_by(year, manufacturer) %>%
  summarise(Mean = mean(seats))
```

```
## Source: local data frame [164 x 3]
## Groups: year [?]
##
##     year manufacturer  Mean
##    <int>        <chr> <dbl>
## 1   1956      DOUGLAS   102
## 2   1959       CESSNA     2
## 3   1959  DEHAVILLAND    16
## 4   1963       CESSNA     5
## 5   1965       BOEING   149
## 6   1967        BEECH     9
## 7   1968        PIPER     4
## 8   1972        BEECH    10
## 9   1973       CESSNA     6
## 10  1974 CANADAIR LTD     2
## # ... with 154 more rows
```

# rename

`rename` a variable:

```
planes %>%
  rename(Mfr = manufacturer) %>%
  names()
```

```
## [1] "tailnum" "year"    "type"    "Mfr"     "model"   "engines" "seats"
## [8] "speed"   "engine"
```

## Installation or Setup

To install `dplyr` simply type in the R console.

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

And then to load `dplyr`, type

```
library("dplyr")
```

It's also possible to install the latest development version from Github with:

```
if (packageVersion("devtools") < 1.6) {
  install.packages("devtools")
}
devtools::install_github("hadley/lazyeval")
devtools::install_github("hadley/dplyr")
```

You may want to install the data packages used in most examples:

```
install.packages(c("nycflights13", "Lahman")).
```

Read Getting started with dplyr online: http://www.riptutorial.com/dplyr/topic/4254/getting-started-with-dplyr

# Credits

| S. No | Chapters | Contributors |
|---|---|---|
| 1 | Getting started with dplyr | Community, Daniel Falbel, theArun, timtrice |