

# Data Transformation with dplyr

Ismail Fadeli

August 27, 2021

## Data Transformation with dplyr

```
library(nycflights13)

library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --

## v ggplot2 3.3.5      v purrr 0.3.4
## v tibble 3.1.3       v dplyr 1.0.7
## v tidyr 1.1.3        v stringr 1.4.0
## v readr 2.0.1        v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()

flights

## # A tibble: 336,776 x 19
##   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>   <int>         <int>         <dbl>   <int>         <int>
## 1  2013     1     1     517             515           2       830           819
## 2  2013     1     1     533             529           4       850           830
## 3  2013     1     1     542             540           2       923           850
## 4  2013     1     1     544             545          -1      1004          1022
## 5  2013     1     1     554             600          -6       812           837
## 6  2013     1     1     554             558          -4       740           728
## 7  2013     1     1     555             600          -5       913           854
## 8  2013     1     1     557             600          -3       709           723
## 9  2013     1     1     557             600          -3       838           846
## 10 2013     1     1     558             600          -2       753           745
## # ... with 336,766 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dtm>
```

If we want to see all the data in flights: View(flights)

## Filter Rows with filter()

**filter()** allows you to subset observations based on their values. The first argument is the name of the dataframe. The second and the rest arguments are the expressions that filter the dataframe. For example, we can select all flights on January 1st with:

```
filter(flights, month == 1, day ==1)
```

```
## # A tibble: 842 x 19
##   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>   <int>         <int>         <dbl>   <int>         <int>
## 1  2013     1     1     517           515           2     830           819
## 2  2013     1     1     533           529           4     850           830
## 3  2013     1     1     542           540           2     923           850
## 4  2013     1     1     544           545          -1    1004          1022
## 5  2013     1     1     554           600          -6     812           837
## 6  2013     1     1     554           558          -4     740           728
## 7  2013     1     1     555           600          -5     913           854
## 8  2013     1     1     557           600          -3     709           723
## 9  2013     1     1     557           600          -3     838           846
## 10 2013     1     1     558           600          -2     753           745
## # ... with 832 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dtm>
```

To assign the filtered result into a new dataframe we can use an assignment:

```
jan1 <- filter(flights, month == 1, day == 1)
```

If you want to print the results and save them in a variable at the same time, you can use parentheses:

```
(dec25 <- filter(flights, month ==12, day == 25))
```

```
## # A tibble: 719 x 19
##   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>   <int>         <int>         <dbl>   <int>         <int>
## 1  2013    12    25     456           500          -4     649           651
## 2  2013    12    25     524           515           9     805           814
## 3  2013    12    25     542           540           2     832           850
## 4  2013    12    25     546           550          -4    1022          1027
## 5  2013    12    25     556           600          -4     730           745
## 6  2013    12    25     557           600          -3     743           752
## 7  2013    12    25     557           600          -3     818           831
## 8  2013    12    25     559           600          -1     855           856
## 9  2013    12    25     559           600          -1     849           855
## 10 2013    12    25     600           600           0     850           846
## # ... with 709 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dtm>
```

## Comparisons

The standard operations that R uses are: >, >=, <, <=, != (not equal), and == (equal).

```
sqrt(2) ^ 2 == 2
```

```
## [1] FALSE
```

```
1/49 *49 == 1
```

```
## [1] FALSE
```

```
near(sqrt(2) ^ 2, 2)
```

```
## [1] TRUE
```

```
near(1 / 49 * 49, 1)
```

```
## [1] TRUE
```

Boolean operators in R are: & is “and”, | is “or”, ! is “not”. The following code shows the months of November or December:

```
filter(flights, month == 11 | month == 12)
```

```
## # A tibble: 55,403 x 19
```

```
##   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>   <int>         <int>         <dbl>   <int>         <int>
## 1  2013    11     1       5           2359           6       352           345
## 2  2013    11     1      35           2250          105       123           2356
## 3  2013    11     1     455           500           -5       641           651
## 4  2013    11     1     539           545           -6       856           827
## 5  2013    11     1     542           545           -3       831           855
## 6  2013    11     1     549           600          -11       912           923
## 7  2013    11     1     550           600          -10       705           659
## 8  2013    11     1     554           600           -6       659           701
## 9  2013    11     1     554           600           -6       826           827
## 10 2013    11     1     554           600           -6       749           751
## # ... with 55,393 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dtm>
```