kornel_kovacs_hw_3_nyflights

Kornel Kovacs

2019 10 01

Install the dataset if you don't have it

```
install.packages("nycflights13")
```

```
library(nycflights13)
flights
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```
View(flights)
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

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

Today, we'll cover

- `filter()`
- `arrange()`
- `select()`

Next week, we'll cover

- `mutate()`
- `summarise()`
- `group_by()`, which tells the other verbs to use the data by groups

All take as first argument a data frame (or tibble) and return a data frame (or tibble). Together they form the verbs of the tidyverse.

Filtering (choosing) rows with filter()

```
filter(flights, month = 1) # Produces an error
filter(flights, month == 1)
filter(flights, month == 1, day == 1)
filter(flights, dep_time == 517)
```

dplyr functions don't change the data frame that you give it. They return a new one.

1. Save the filtered data

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

```
## # A tibble: 842 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 832 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

2. Assign and print, use (varname <- ...)

```
(feb1 <- filter(flights, month == 2, day == 1))
```

```
## # A tibble: 926 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     2     1      456            500        -4      652
##  2  2013     2     1      520            525        -5      816
##  3  2013     2     1      527            530        -3      837
##  4  2013     2     1      532            540        -8     1007
##  5  2013     2     1      540            540         0      859
##  6  2013     2     1      552            600        -8      714
##  7  2013     2     1      552            600        -8      919
##  8  2013     2     1      552            600        -8      655
##  9  2013     2     1      553            600        -7      833
## 10  2013     2     1      553            600        -7      821
## # ... with 916 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

3. Check it really assigned

```
feb1
```

```
## # A tibble: 926 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     2     1      456            500        -4      652
##  2  2013     2     1      520            525        -5      816
##  3  2013     2     1      527            530        -3      837
##  4  2013     2     1      532            540        -8     1007
##  5  2013     2     1      540            540         0      859
##  6  2013     2     1      552            600        -8      714
##  7  2013     2     1      552            600        -8      919
##  8  2013     2     1      552            600        -8      655
##  9  2013     2     1      553            600        -7      833
## 10  2013     2     1      553            600        -7      821
## # ... with 916 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

Some notes on comparisons

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

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

```
sqrt(4)^2 == 4
```

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

```
(1/3)*3 == 1
```

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

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

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

```
1/(7^9)*7^9 == 1
```

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

In short, you can't rely on "It works because it works for what I tried".

For floating point comparisons, use near() to compare numbers

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

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

Multiple constraints |: is 'or' operator

```
(jan_feb <- filter(flights, month == 1 | month == 2))
```

```
## # A tibble: 51,955 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 51,945 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```
(not_jan <- filter(flights, !(month == 1)))
```

```
## # A tibble: 309,772 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013    10     1      447            500       -13      614
##  2  2013    10     1      522            517         5      735
##  3  2013    10     1      536            545        -9      809
##  4  2013    10     1      539            545        -6      801
##  5  2013    10     1      539            545        -6      917
##  6  2013    10     1      544            550        -6      912
##  7  2013    10     1      549            600       -11      653
##  8  2013    10     1      550            600       -10      648
##  9  2013    10     1      550            600       -10      649
## 10  2013    10     1      551            600        -9      727
## # ... with 309,762 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

Class exercise: How do we know these actually worked?

```
filter(not_jan, month == 1)
```

```
## # A tibble: 0 x 19
## # ... with 19 variables: year <int>, month <int>, day <int>,
## #   dep_time <int>, sched_dep_time <int>, dep_delay <dbl>, arr_time <int>,
## #   sched_arr_time <int>, arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>,
## #   distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
View(jan_feb)
unique(not_jan$month)
```

```
##  [1] 10 11 12  2  3  4  5  6  7  8  9
```

```
jan <- filter(flights, month == 1)
nrow(flights) == nrow(jan) + nrow(not_jan)
```

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

```
(jan_to_june1 <- filter(flights, month <= 6))
```

```
## # A tibble: 166,158 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
```

```
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 166,148 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```
jan_to_june2 <- filter(flights, month %in% c(1,2,3,4,5,6))
```

Check same number of observations

```
nrow(jan_to_june1) == nrow(jan_to_june2)
```

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

Class Exercise: What does this do?

```
mystery_filter <- filter(flights, !(arr_delay > 120 | dep_delay > 120))
mystery_filter2 <- filter(flights, arr_delay <= 120, dep_delay <= 120)
mystery_filter
```

```
## # A tibble: 316,050 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 316,040 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```
mystery_filter2
```

```
## # A tibble: 316,050 x 19
```

```
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 316,040 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

Vote:

1. All flights that started and landed 120 minutes late
2. All flights that started 120 minutes late or landed 120 minutes late
3. All flights that started less than 120 minutes late or landed less than 120 minutes late
4. All flights that started and landed less than 120 minutes late

3. All flights that started less than 120 minutes late or landed less than 120 minutes late

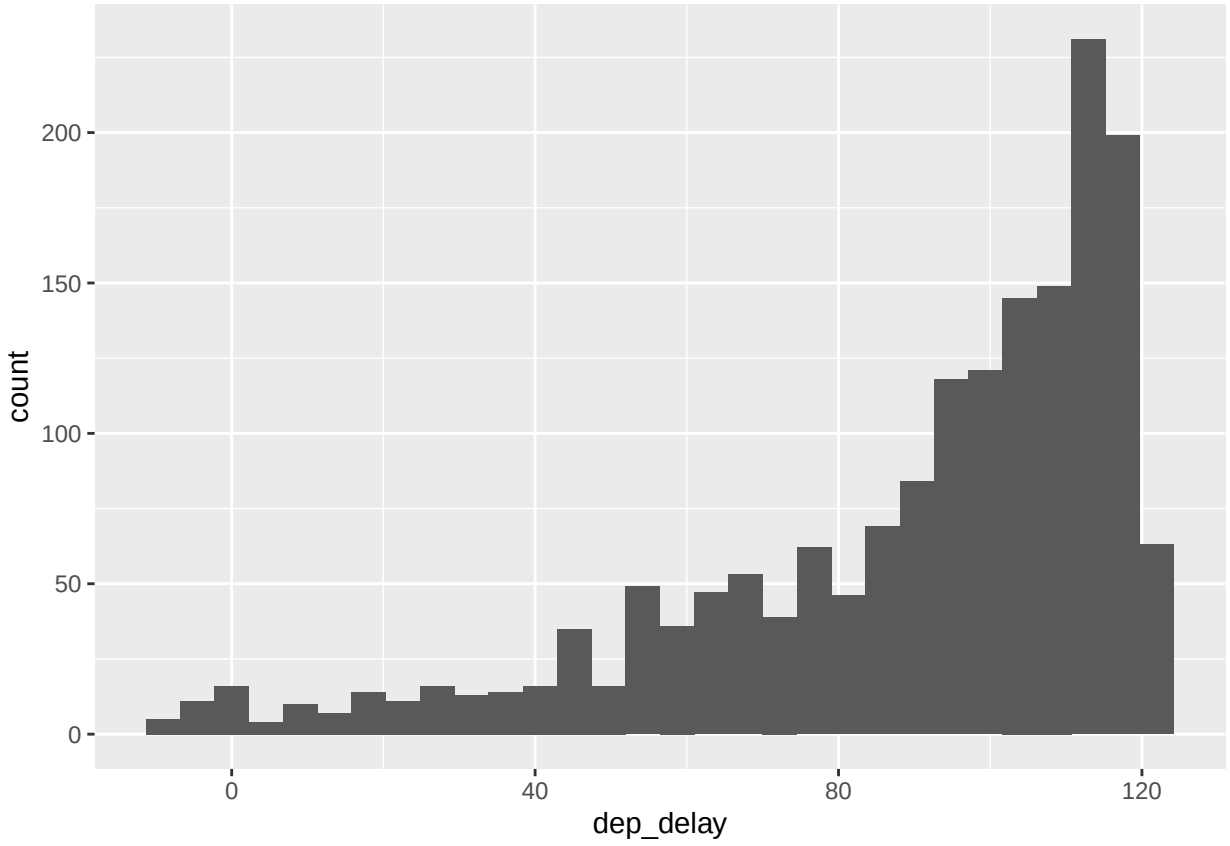
```
number3 <- filter(flights, arr_delay <= 120 | dep_delay <= 120)
number3 <- filter(flights, arr_delay < 120 | dep_delay < 120)
```

Class Exercise: get all flights that departed with less than 120 minutes delay, but arrived with more than 120 minutes delay.

```
dep_ok_arr_not <- filter(flights, dep_delay <= 120, arr_delay > 120)

ggplot(data = dep_ok_arr_not,
       mapping = aes(x = dep_delay)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



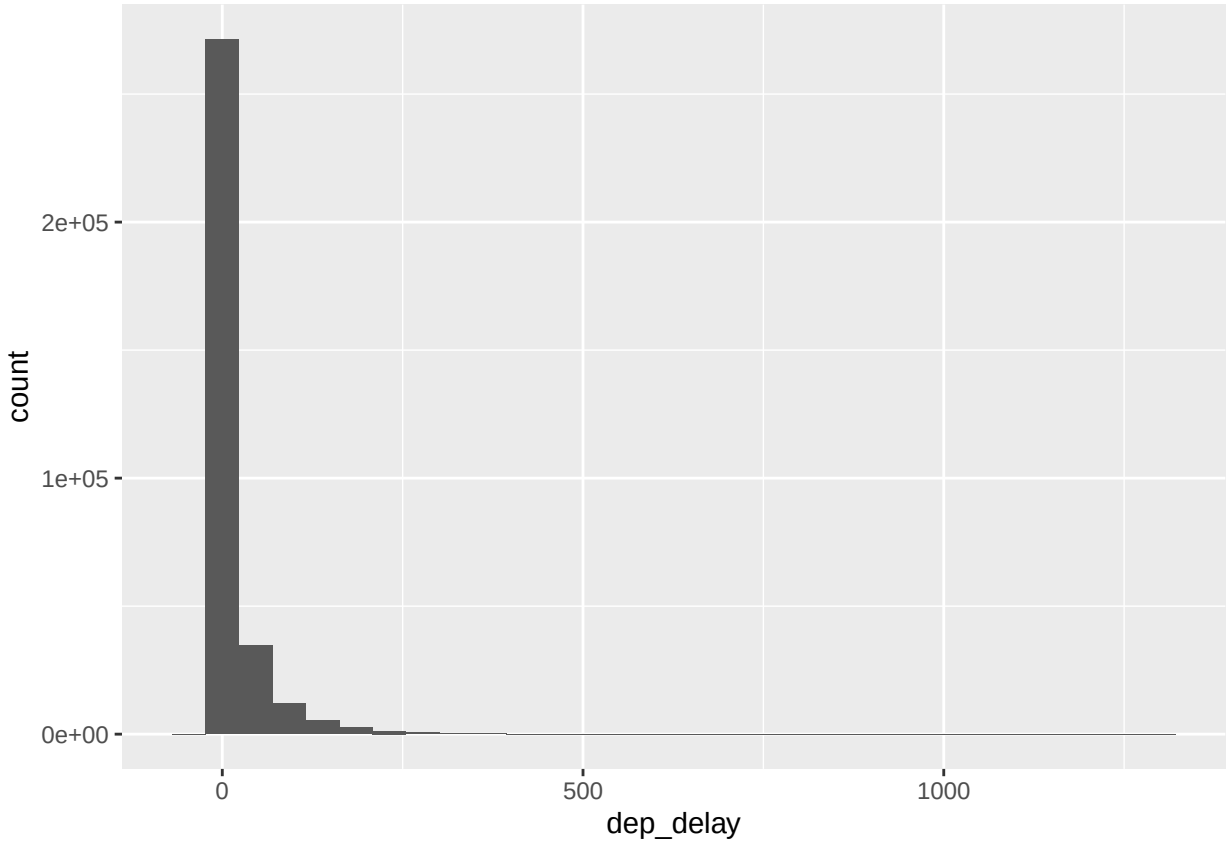
Let's look at the data to see what the departure was for planes that arrived

late but didn't start quite as late

```
ggplot(data = flights,
       mapping = aes(x = dep_delay)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

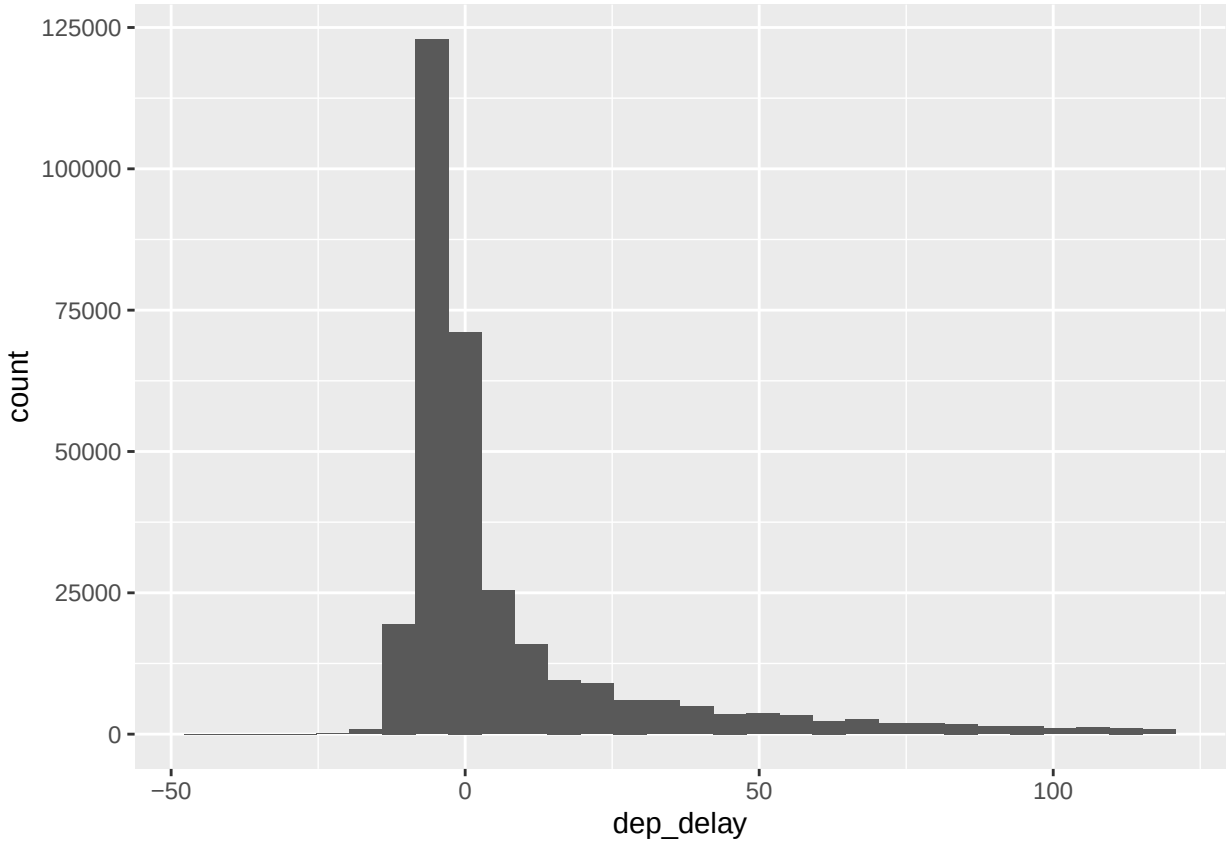
```
## Warning: Removed 8255 rows containing non-finite values (stat_bin).
```



Filter flights by those that had dep_delay <= 120, then plot histogram

```
dep_ok <- filter(flights, dep_delay <= 120)
ggplot(data = dep_ok,
       mapping = aes(x = dep_delay)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



NA: Not available

```
NA > 5
```

```
## [1] NA
```

```
10 == NA
```

```
## [1] NA
```

```
NA == NA
```

```
## [1] NA
```

```
FALSE & NA
```

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

```
TRUE & NA
```

```
## [1] NA
```

```
NA & FALSE
```

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

Let x be Mary's age. We don't know how old she is.

```
x <- NA
```

Let y be John's age. We don't know how old he is.

```
y <- NA
```

Are John and Mary the same age?

```
x == y
```

```
## [1] NA
```

We don't know!

```
NA^0
```

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

```
0 * NA
```

```
## [1] NA
```

```
is.na(x)
```

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

```
df <- tibble(x = c(1, NA, 3))
df
```

```
## # A tibble: 3 x 1
##       x
##   <dbl>
## 1     1
## 2    NA
## 3     3
```

```r
filter(df, x > 1)
```

```
## # A tibble: 1 x 1
##       x
##   <dbl>
## 1     3
```

```r
filter(df, x > 1 | is.na(x))
```

```
## # A tibble: 2 x 1
##       x
##   <dbl>
## 1    NA
## 2     3
```

arrange()

```r
flights
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```r
arrange(flights, year, month, day)
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
```

```
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```
arrange(flights, dep_delay)
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013    12     7     2040           2123       -43       40
##  2  2013     2     3     2022           2055       -33     2240
##  3  2013    11    10     1408           1440       -32     1549
##  4  2013     1    11     1900           1930       -30     2233
##  5  2013     1    29     1703           1730       -27     1947
##  6  2013     8     9      729            755       -26     1002
##  7  2013    10    23     1907           1932       -25     2143
##  8  2013     3    30     2030           2055       -25     2213
##  9  2013     3     2     1431           1455       -24     1601
## 10  2013     5     5      934            958       -24     1225
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```
arrange(flights, desc(dep_delay))
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     9      641            900      1301     1242
##  2  2013     6    15     1432           1935      1137     1607
##  3  2013     1    10     1121           1635      1126     1239
##  4  2013     9    20     1139           1845      1014     1457
##  5  2013     7    22      845           1600      1005     1044
##  6  2013     4    10     1100           1900       960     1342
##  7  2013     3    17     2321            810       911      135
##  8  2013     6    27      959           1900       899     1236
##  9  2013     7    22     2257            759       898      121
## 10  2013    12     5      756           1700       896     1058
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```
arrange(df, x)
```

```
## # A tibble: 3 x 1
##       x
```

```
##   <dbl>
## 1     1
## 2     3
## 3    NA
```

```
arrange(df, desc(x))
```

```
## # A tibble: 3 x 1
##       x
##   <dbl>
## 1     3
## 2     1
## 3    NA
```

Class exercise (do at home): How can we get the missing values at the top?

```
head(arrange(flights, !is.na(desc(arr_delay))))
```

```
## # A tibble: 6 x 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
## 1  2013     1     1     1525           1530        -5     1934
## 2  2013     1     1     1528           1459        29     2002
## 3  2013     1     1     1740           1745        -5     2158
## 4  2013     1     1     1807           1738        29     2251
## 5  2013     1     1     1939           1840        59       29
## 6  2013     1     1     1952           1930        22     2358
## # ... with 12 more variables: sched_arr_time <int>, arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <dttm>
```

Fastest flight

```
colnames(flights)
```

```
##  [1] "year"           "month"          "day"            "dep_time"
##  [5] "sched_dep_time" "dep_delay"      "arr_time"       "sched_arr_time"
##  [9] "arr_delay"      "carrier"        "flight"         "tailnum"
## [13] "origin"         "dest"           "air_time"       "distance"
## [17] "hour"           "minute"         "time_hour"
```

```
arrange(flights, air_time)
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1    16     1355           1315        40     1442
##  2  2013     4    13      537            527        10      622
##  3  2013    12     6      922            851        31     1021
##  4  2013     2     3     2153           2129        24     2247
##  5  2013     2     5     1303           1315       -12     1342
##  6  2013     2    12     2123           2130        -7     2211
##  7  2013     3     2     1450           1500       -10     1547
##  8  2013     3     8     2026           1935        51     2131
##  9  2013     3    18     1456           1329        87     1533
## 10  2013     3    19     2226           2145        41     2305
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

select()

```
select(flights, year, month, day)
```

```
## # A tibble: 336,776 x 3
##     year month   day
##    <int> <int> <int>
##  1  2013     1     1
##  2  2013     1     1
##  3  2013     1     1
##  4  2013     1     1
##  5  2013     1     1
##  6  2013     1     1
##  7  2013     1     1
##  8  2013     1     1
##  9  2013     1     1
## 10  2013     1     1
## # ... with 336,766 more rows
```

```
select(arrange(flights, air_time), air_time, origin, dest)
```

```
## # A tibble: 336,776 x 3
##    air_time origin dest
##       <dbl> <chr>  <chr>
##  1       20 EWR    BDL
##  2       20 EWR    BDL
##  3       21 EWR    BDL
##  4       21 EWR    PHL
##  5       21 EWR    BDL
##  6       21 EWR    PHL
##  7       21 LGA    BOS
##  8       21 JFK    PHL
```

```
##  9       21 EWR    BDL
## 10       21 EWR    BDL
## # ... with 336,766 more rows
```

That's tedious to write. Hence the pipe.

```
flights %>%
  arrange(air_time) %>%
  select(air_time, origin, dest)
```

```
## # A tibble: 336,776 x 3
##    air_time origin dest
##       <dbl> <chr>  <chr>
##  1       20 EWR    BDL
##  2       20 EWR    BDL
##  3       21 EWR    BDL
##  4       21 EWR    PHL
##  5       21 EWR    BDL
##  6       21 EWR    PHL
##  7       21 LGA    BOS
##  8       21 JFK    PHL
##  9       21 EWR    BDL
## 10       21 EWR    BDL
## # ... with 336,766 more rows
```

Notice that the data doesn't have to be mentioned, and the first argument should not have to be provided

```
select(flights, year:day)
```

```
## # A tibble: 336,776 x 3
##     year month   day
##    <int> <int> <int>
##  1  2013     1     1
##  2  2013     1     1
##  3  2013     1     1
##  4  2013     1     1
##  5  2013     1     1
##  6  2013     1     1
##  7  2013     1     1
##  8  2013     1     1
##  9  2013     1     1
## 10  2013     1     1
## # ... with 336,766 more rows
```

```
flights %>% select(year:day)
```

```
## # A tibble: 336,776 x 3
##     year month   day
##    <int> <int> <int>
##  1  2013     1     1
##  2  2013     1     1
##  3  2013     1     1
##  4  2013     1     1
##  5  2013     1     1
##  6  2013     1     1
##  7  2013     1     1
##  8  2013     1     1
##  9  2013     1     1
## 10  2013     1     1
## # ... with 336,766 more rows
```

```
colnames(flights)
```

```
##  [1] "year"           "month"          "day"            "dep_time"
##  [5] "sched_dep_time" "dep_delay"      "arr_time"       "sched_arr_time"
##  [9] "arr_delay"      "carrier"        "flight"         "tailnum"
## [13] "origin"         "dest"           "air_time"       "distance"
## [17] "hour"           "minute"         "time_hour"
```

dropping cols

```
select(flights, -(year:day))
```

```
## # A tibble: 336,776 x 16
##    dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
##       <int>          <int>     <dbl>    <int>          <int>     <dbl>
##  1      517            515         2      830            819        11
##  2      533            529         4      850            830        20
##  3      542            540         2      923            850        33
##  4      544            545        -1     1004           1022       -18
##  5      554            600        -6      812            837       -25
##  6      554            558        -4      740            728        12
##  7      555            600        -5      913            854        19
##  8      557            600        -3      709            723       -14
##  9      557            600        -3      838            846        -8
## 10      558            600        -2      753            745         8
## # ... with 336,766 more rows, and 10 more variables: carrier <chr>,
## #   flight <int>, tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>,
## #   distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

Some helper functions

```
select(flights, starts_with("arr"))
```

```
## # A tibble: 336,776 x 2
##    arr_time arr_delay
##       <int>     <dbl>
##  1      830        11
##  2      850        20
##  3      923        33
##  4     1004       -18
##  5      812       -25
##  6      740        12
##  7      913        19
##  8      709       -14
##  9      838        -8
## 10      753         8
## # ... with 336,766 more rows
```

```
select(flights, -starts_with("arr"))
```

```
## # A tibble: 336,776 x 17
##     year month   day dep_time sched_dep_time dep_delay sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>          <int>
##  1  2013     1     1      517            515         2            819
##  2  2013     1     1      533            529         4            830
##  3  2013     1     1      542            540         2            850
##  4  2013     1     1      544            545        -1           1022
##  5  2013     1     1      554            600        -6            837
##  6  2013     1     1      554            558        -4            728
##  7  2013     1     1      555            600        -5            854
##  8  2013     1     1      557            600        -3            723
##  9  2013     1     1      557            600        -3            846
## 10  2013     1     1      558            600        -2            745
## # ... with 336,766 more rows, and 10 more variables: carrier <chr>,
## #   flight <int>, tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>,
## #   distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
select(flights, ends_with("hour"))
```

```
## # A tibble: 336,776 x 2
##     hour time_hour
##    <dbl> <dttm>
##  1     5 2013-01-01 05:00:00
##  2     5 2013-01-01 05:00:00
##  3     5 2013-01-01 05:00:00
##  4     5 2013-01-01 05:00:00
##  5     6 2013-01-01 06:00:00
##  6     5 2013-01-01 05:00:00
##  7     6 2013-01-01 06:00:00
##  8     6 2013-01-01 06:00:00
```

```
##  9     6 2013-01-01 06:00:00
## 10     6 2013-01-01 06:00:00
## # ... with 336,766 more rows
```

```
select(flights, -contains("time"))
```

```
## # A tibble: 336,776 x 13
##     year month   day dep_delay arr_delay carrier flight tailnum origin
##    <int> <int> <int>     <dbl>     <dbl> <chr>    <int> <chr>   <chr>
##  1  2013     1     1         2        11 UA        1545 N14228  EWR
##  2  2013     1     1         4        20 UA        1714 N24211  LGA
##  3  2013     1     1         2        33 AA        1141 N619AA  JFK
##  4  2013     1     1        -1       -18 B6         725 N804JB  JFK
##  5  2013     1     1        -6       -25 DL         461 N668DN  LGA
##  6  2013     1     1        -4        12 UA        1696 N39463  EWR
##  7  2013     1     1        -5        19 B6         507 N516JB  EWR
##  8  2013     1     1        -3       -14 EV        5708 N829AS  LGA
##  9  2013     1     1        -3        -8 B6          79 N593JB  JFK
## 10  2013     1     1        -2         8 AA         301 N3ALAA  LGA
## # ... with 336,766 more rows, and 4 more variables: dest <chr>,
## #   distance <dbl>, hour <dbl>, minute <dbl>
```

Function for renaming columns

```
rename(flights, destination = dest)
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, destination <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

Hard to see if it worked, so...

```
flights %>% rename(destination = dest) %>% select(year:day, destination)
```

```
## # A tibble: 336,776 x 4
##     year month   day destination
##    <int> <int> <int> <chr>
##  1  2013     1     1 IAH
##  2  2013     1     1 IAH
##  3  2013     1     1 MIA
##  4  2013     1     1 BQN
##  5  2013     1     1 ATL
##  6  2013     1     1 ORD
##  7  2013     1     1 FLL
##  8  2013     1     1 IAD
##  9  2013     1     1 MCO
## 10  2013     1     1 ORD
## # ... with 336,766 more rows
```

Moving some columns to the start

```
select(flights, origin, dest, everything())
```

```
## # A tibble: 336,776 x 19
##    origin dest   year month   day dep_time sched_dep_time dep_delay
##    <chr>  <chr> <int> <int> <int>    <int>          <int>     <dbl>
##  1 EWR    IAH    2013     1     1      517            515         2
##  2 LGA    IAH    2013     1     1      533            529         4
##  3 JFK    MIA    2013     1     1      542            540         2
##  4 JFK    BQN    2013     1     1      544            545        -1
##  5 LGA    ATL    2013     1     1      554            600        -6
##  6 EWR    ORD    2013     1     1      554            558        -4
##  7 EWR    FLL    2013     1     1      555            600        -5
##  8 LGA    IAD    2013     1     1      557            600        -3
##  9 JFK    MCO    2013     1     1      557            600        -3
## 10 LGA    ORD    2013     1     1      558            600        -2
## # ... with 336,766 more rows, and 11 more variables: arr_time <int>,
## #   sched_arr_time <int>, arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```