Intro to data wrangling

nycflights13

```
library(nycflights13)

dim(flights)
[1] 336776     19

head(flights)
# A tibble: 6 × 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
# ... with 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 <dttm>
```

- Part of the `tidyverse`
- Grammar for wrangling data frames

select()

Extract columns by name

```
select(flights, year, month, day)
# A tibble: 336,776 × 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()

Reorder columns

```
select(flights, year, origin, dest, carrier, everything())
# A tibble: 336,776 × 19
    year origin  dest carrier month   day dep_time sched_dep_time dep_delay
   <int>  <chr> <chr>   <chr> <int> <int>    <int>          <int>     <dbl>
1   2013    EWR   IAH      UA     1     1      517            515         2
2   2013    LGA   IAH      UA     1     1      533            529         4
3   2013    JFK   MIA      AA     1     1      542            540         2
4   2013    JFK   BQN      B6     1     1      544            545        -1
5   2013    LGA   ATL      DL     1     1      554            600        -6
6   2013    EWR   ORD      UA     1     1      554            558        -4
7   2013    EWR   FLL      B6     1     1      555            600        -5
8   2013    LGA   IAD      EV     1     1      557            600        -3
9   2013    JFK   MCO      B6     1     1      557            600        -3
10  2013    LGA   ORD      AA     1     1      558            600        -2
# ... with 336,766 more rows, and 10 more variables: arr_time <int>,
#   sched_arr_time <int>, arr_delay <dbl>, flight <int>, tailnum <chr>,
#   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

`filter()`

Extract rows that meet a logical criteria

```
filter(flights, origin == "JFK")
# A tibble: 111,279 × 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      542            540         2      923            850
2   2013     1     1      544            545        -1     1004           1022
3   2013     1     1      557            600        -3      838            846
4   2013     1     1      558            600        -2      849            851
5   2013     1     1      558            600        -2      853            856
6   2013     1     1      558            600        -2      924            917
7   2013     1     1      559            559         0      702            706
8   2013     1     1      606            610        -4      837            845
9   2013     1     1      611            600        11      945            931
10  2013     1     1      613            610         3      925            921
# ... with 111,269 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 <dttm>
```

`filter()`

Extract rows that meet multiple logical criteria

```
filter(flights, month == 1 & day == 1)
# A tibble: 842 × 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 <dttm>
```

`filter()`

Extract rows that meet multiple logical criteria

```
filter(flights, carrier == "DL" | carrier == "WN")
# A tibble: 60,385 × 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      554            600        -6      812            837
2   2013     1     1      602            610        -8      812            820
3   2013     1     1      606            610        -4      837            845
4   2013     1     1      615            615         0      833            842
5   2013     1     1      629            630        -1      721            740
6   2013     1     1      653            700        -7      936           1009
7   2013     1     1      655            655         0     1021           1030
8   2013     1     1      655            700        -5     1037           1045
9   2013     1     1      655            700        -5     1002           1020
10  2013     1     1      657            700        -3      959           1013
# ... with 60,375 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 <dttm>
```

Logical operators

Operator	Explanation
`x == y`	x is exactly equal to y
`x %in% y`	x is an element of y
`x != y`	x is not equal to y
`x < y`	x is less than y
`x <= y`	x is less than or equal to y
`x > y`	x is greater than y
`x >= y`	x is greater than or equal to y

mutate(): add/modify a column

Compute new columns

```
mutate(flights,
       gain = arr_delay - dep_delay,
       speed = distance / air_time * 60)
```

```
# A tibble: 336,776 × 21
    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 13 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 <dttm>,
#   gain <dbl>, speed <dbl>
```

rename()

Rename columns

```
rename(flights, real_dep_time = dep_time)

# A tibble: 336,776 × 19
    year month   day real_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>
```

A smaller data frame

```
small_flights <- select(flights, carrier, origin, dest, month, day, arr_delay)
small_flights
# A tibble: 336,776 × 6
   carrier origin  dest month   day arr_delay
     <chr>  <chr> <chr> <int> <int>     <dbl>
1       UA    EWR   IAH     1     1        11
2       UA    LGA   IAH     1     1        20
3       AA    JFK   MIA     1     1        33
4       B6    JFK   BQN     1     1       -18
5       DL    LGA   ATL     1     1       -25
6       UA    EWR   ORD     1     1        12
7       B6    EWR   FLL     1     1        19
8       EV    LGA   IAD     1     1       -14
9       B6    JFK   MCO     1     1        -8
10      AA    LGA   ORD     1     1         8
# ... with 336,766 more rows
```

arrange()

Order the rows by the value of a column (low to high)

```
arrange(small_flights, carrier)
# A tibble: 336,776 × 6
   carrier origin  dest month   day arr_delay
     <chr>  <chr> <chr> <int> <int>     <dbl>
1       9E    JFK   MSP     1     1        11
2       9E    JFK   IAD     1     1        -2
3       9E    JFK   BUF     1     1        -2
4       9E    JFK   SYR     1     1        -1
5       9E    JFK   ROC     1     1        -5
6       9E    JFK   BWI     1     1        -5
7       9E    JFK   ORD     1     1         5
8       9E    JFK   IND     1     1        13
9       9E    JFK   BNA     1     1        -8
10      9E    JFK   BOS     1     1       -33
# ... with 336,766 more rows
```

arrange()

Use desc() to order from high to low

```
arrange(small_flights, desc(carrier))
# A tibble: 336,776 × 6
   carrier origin  dest month   day arr_delay
     <chr>  <chr> <chr> <int> <int>     <dbl>
1       YV    LGA   IAD     1     3       -20
2       YV    LGA   IAD     1     3       -23
3       YV    LGA   IAD     1     4       -13
4       YV    LGA   IAD     1     4        75
5       YV    LGA   IAD     1     6       -15
6       YV    LGA   IAD     1     7       -18
7       YV    LGA   IAD     1     7        -1
8       YV    LGA   IAD     1     8       -22
9       YV    LGA   IAD     1     8         5
10      YV    LGA   IAD     1     9       -16
# ... with 336,766 more rows
```

arrange()

```
arrange(small_flights, carrier, dest)
# A tibble: 336,776 × 6
   carrier origin  dest month   day arr_delay
     <chr>  <chr> <chr> <int> <int>     <dbl>
1       9E    JFK   ATL     1     4        -4
2       9E    JFK   ATL     1     7       -43
3       9E    JFK   ATL     1     8         6
4       9E    JFK   ATL     1     9        NA
5       9E    JFK   ATL     1    10       -15
6       9E    JFK   ATL     1    11        19
7       9E    JFK   ATL     1    12       -16
8       9E    JFK   ATL     1    13        -2
9       9E    JFK   ATL     1    14        55
10      9E    JFK   ATL     1    15        32
# ... with 336,766 more rows
```

arrange()

```
arrange(small_flights, carrier, dest)
# A tibble: 336,776 × 6
   carrier origin  dest month   day arr_delay
     <chr>  <chr> <chr> <int> <int>     <dbl>
1       9E    JFK   ATL     1     4        -4
2       9E    JFK   ATL     1     7       -43
3       9E    JFK   ATL     1     8         6
4       9E    JFK   ATL     1     9        NA
5       9E    JFK   ATL     1    10       -15
6       9E    JFK   ATL     1    11        19
7       9E    JFK   ATL     1    12       -16
8       9E    JFK   ATL     1    13        -2
9       9E    JFK   ATL     1    14        55
10      9E    JFK   ATL     1    15        32
# ... with 336,766 more rows
```

summarize()

Compute table summaries

```
summarize(flights,
          N = n(),
          mean = mean(dep_delay, na.rm = TRUE),
          sd = sd(dep_delay, na.rm = TRUE),
          min = min(dep_delay, na.rm = TRUE),
          Q1 = quantile(dep_delay, probs = 0.25, na.rm = TRUE),
          median = median(dep_delay, na.rm = TRUE),
          Q3 = quantile(dep_delay, probs = 0.75, na.rm = TRUE),
          max = max(dep_delay, na.rm = TRUE))
# A tibble: 1 × 8
       N     mean       sd   min    Q1 median    Q3   max
   <int>    <dbl>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>
1 336776 12.63907 40.21006   -43    -5     -2    11  1301
```

Your turn

- Work on the `dplyr` exercises

For next time

- Complete the Data Camp tutorial (linked through Moodle)