Lecture 9: The `%>%` Pipe Operator

STAT 450, Fall 2021

`dplyr` functions for data wrangling:

- `select()` take a subset of the columns (variables)
- `filter()` take a subset of the rows (observations)
- `arrange()` reorder the rows
- `mutate()` creates new variables that are functions of existing variables

The function names are *verbs* that describe the type of action each function performs on the data.

Last time we discussed how to use these function one-at-a-time. Today we will discuss the pipe operator `%>%` which can be used to combine a sequence of `dplyr` operations.

Flights Data Set

The `flights` data set contains information on all the flights that departed from New York City in 2013.

```
library(tidyverse)
library(nycflights13)
```

```
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 <dttm>
```

Type `help(flights)` to read the documentation on this data set.

Exercise 1 (review): Use `filter()` to find all flights that

- Had an arrival delay of two or more hours
- Were operated by United (UA), American (AA), or Delta (DL)

Exercise 2: How many flights have a missing value for `dep_time`? What other variables have missing values? What might these rows represent? (Hint: use `is.na()`)

Using the pipe %>%

The pipe %>% allows us to combine a sequence of operations using dplyr.

Select columns by name:

```
flights %>% select(year:day, origin, dest)
```

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

Subset all flights on Dec 25:

```
flights %>% filter(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 <dttm>
```

Select columns and then subset flights on Dec 25:

```
flights %>%
  select(year:day, origin, dest) %>%
  filter(month == 12, day == 25)
```

```
## # A tibble: 719 x 5
##     year month   day origin dest 
##    <int> <int> <int> <chr>  <chr>
##  1  2013    12    25 EWR    CLT  
##  2  2013    12    25 EWR    IAH  
##  3  2013    12    25 JFK    MIA  
##  4  2013    12    25 JFK    BQN  
##  5  2013    12    25 LGA    ORD  
##  6  2013    12    25 LGA    DTW  
##  7  2013    12    25 LGA    ATL  
##  8  2013    12    25 LGA    FLL  
##  9  2013    12    25 EWR    FLL  
## 10  2013    12    25 JFK    MCO  
## # ... with 709 more rows
```

Select columns and then use `mutate()` to add a new column with the `speed` of the aircraft in miles per hour.

```
flights %>%
  select(year:day, distance, air_time) %>%
  mutate(speed = distance / air_time * 60)
```

```
## # A tibble: 336,776 x 6
##     year month   day distance air_time speed
##    <int> <int> <int>    <dbl>    <dbl> <dbl>
##  1  2013     1     1     1400      227  370.
##  2  2013     1     1     1416      227  374.
##  3  2013     1     1     1089      160  408.
##  4  2013     1     1     1576      183  517.
##  5  2013     1     1      762      116  394.
##  6  2013     1     1      719      150  288.
##  7  2013     1     1     1065      158  404.
##  8  2013     1     1      229       53  259.
##  9  2013     1     1      944      140  405.
## 10  2013     1     1      733      138  319.
## # ... with 336,766 more rows
```

Exercise 3:

(a) Run the following code. Which three variables get selected by `contains("dep")`, and how are they related?

```
flights %>%
  select(year:day, carrier, contains("dep")) %>%
  filter(carrier == "UA")
```

(b) Next, add another pipe with the `arrange()` function to identify the UA flights with the longest departure delays.

(c) Similarly, identify the UA flights with the longest arrival delays.