Lab2 - Introduction to data

Jawaid Hakim

2022-09-12

Contents

Load required packages.

```
library(tidyverse)
library(openintro)
```

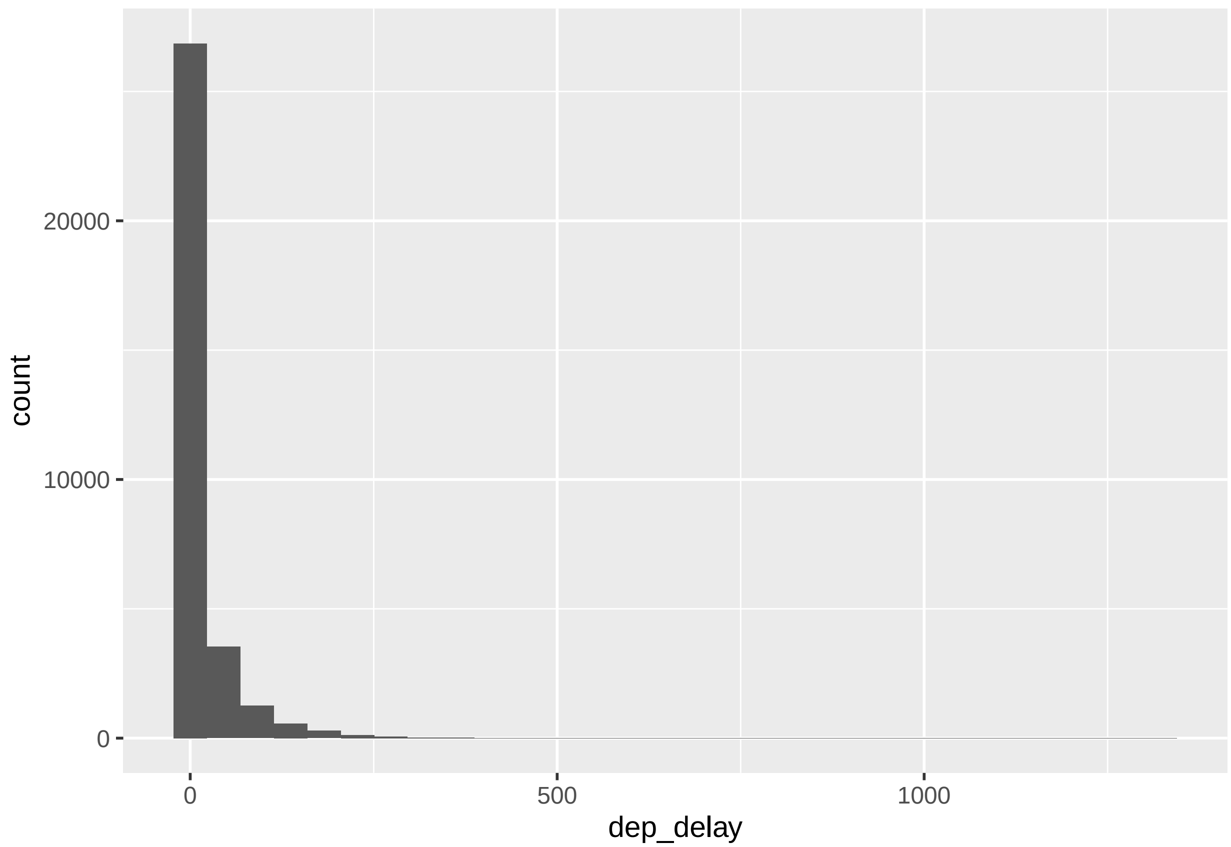
Load *nycflights* data.

```
data("nycflights")
```

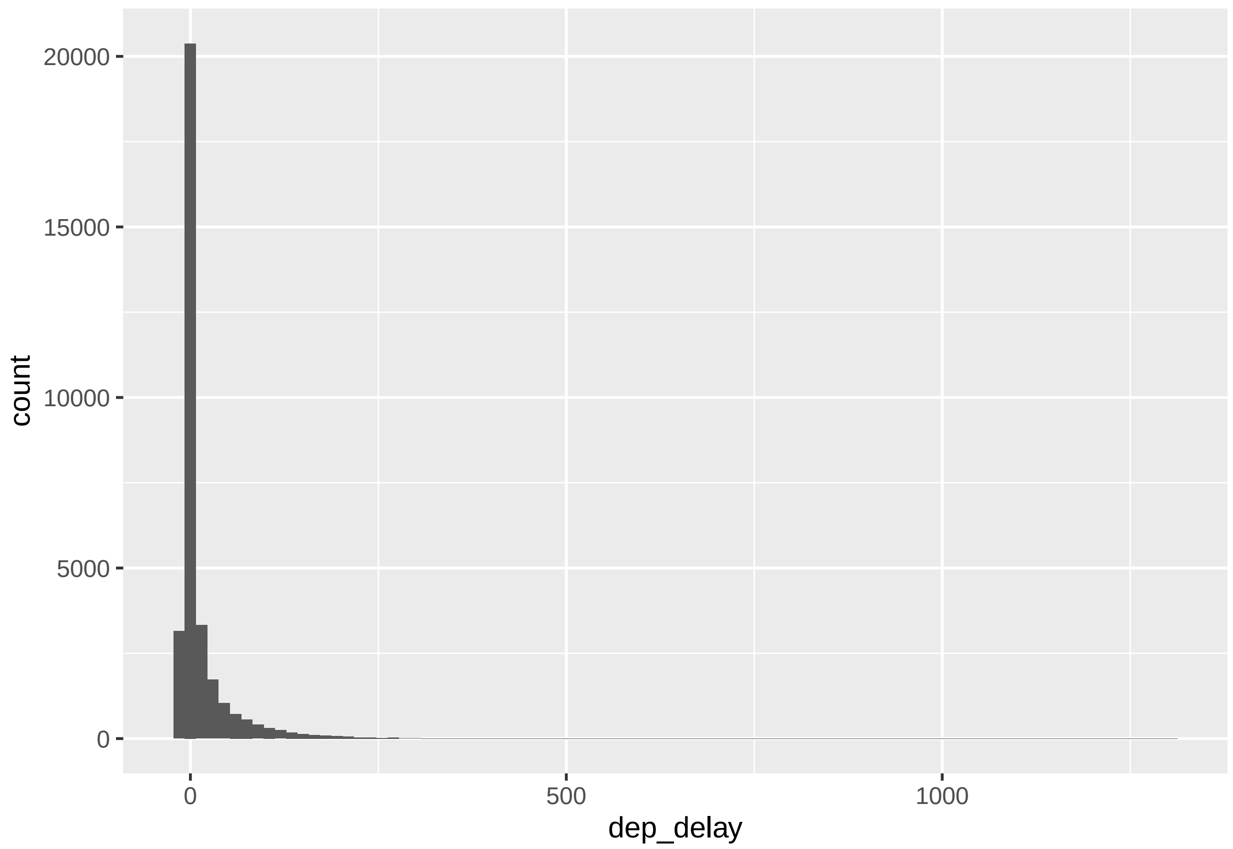
1 Exercise 1

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

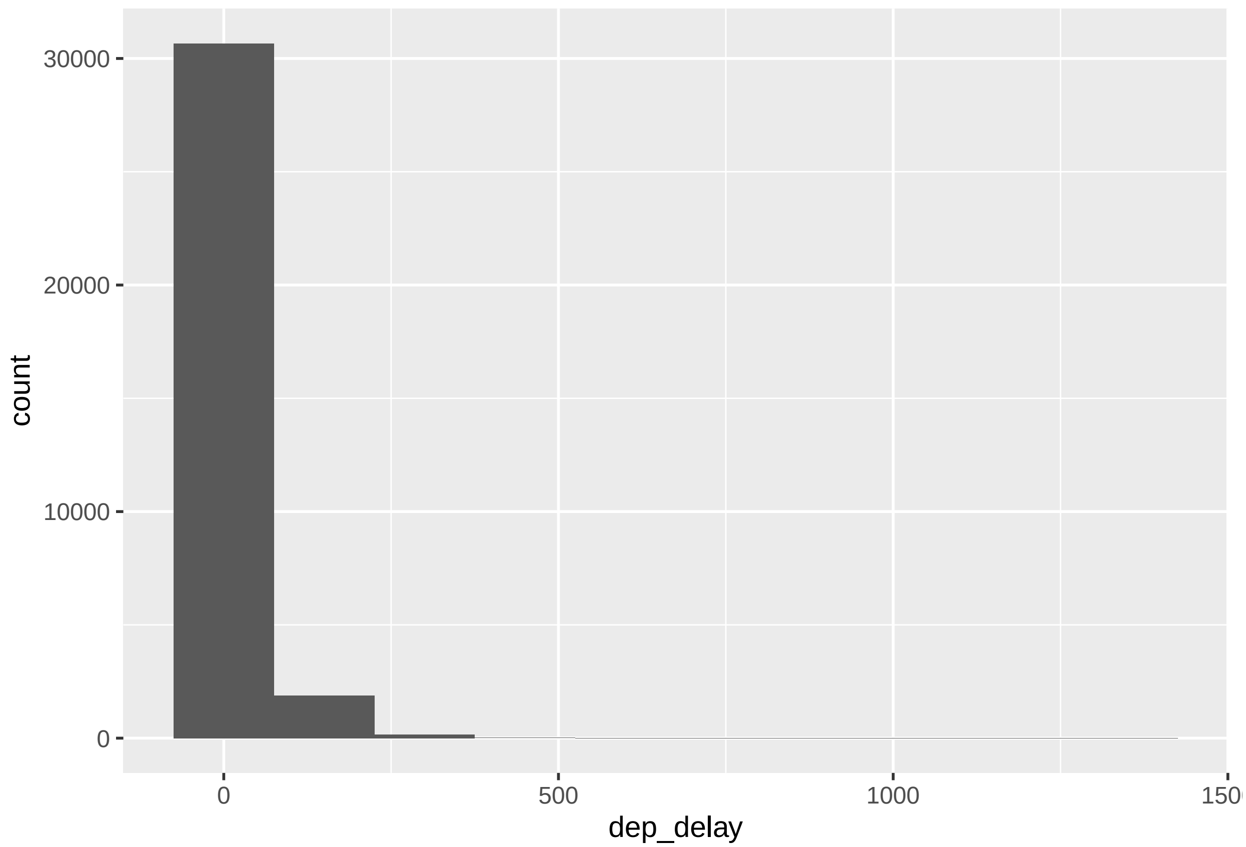
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



```
ggplot(data = nycflights, aes(x = dep_delay)) +
  geom_histogram(binwidth = 15)
```



```
ggplot(data = nycflights, aes(x = dep_delay)) +
  geom_histogram(binwidth = 150)
```



The *binwidth* parameter of *geom_histogrm* function has an impact on the *granualarity* (level of detail) of the resulting plot. First plot has (default) *binwidth=30*. The second plot, has *binwidth=15* and shows the fine-grained details of the underlying observations. The third plot has large bins, *binwidth=150*, and masks the underlying details.

Compared to smaller bins, larger bins mask the variations in the underlying observations.

2 Exercise 2

Create a new data frame that includes flights headed to SFO in February, and save this data frame as sfo_feb_flights. How many flights meet these criteria?

There were *68* flights departing for SFO in February.

```
sfo_feb_flights <- nycflights %>%
    filter(dest == "SFO", month == 2) # Case insensitive filtering using grepl
glimpse(sfo_feb_flights)
```

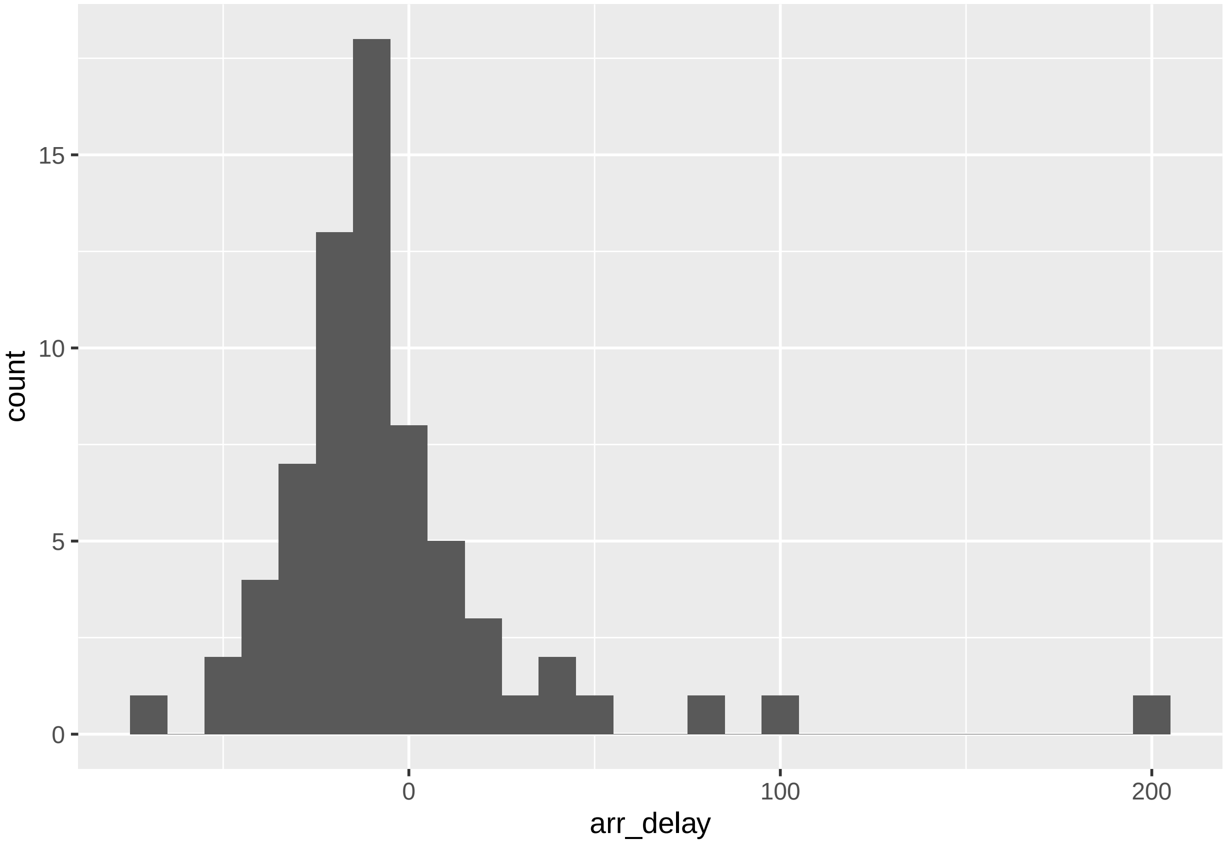
```
## Rows: 68
## Columns: 16
## $ year      <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, ~
## $ month     <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ~
## $ day       <int> 18, 3, 15, 18, 24, 25, 7, 15, 13, 8, 11, 13, 25, 20, 12, 27,~
## $ dep_time  <int> 1527, 613, 955, 1928, 1340, 1415, 1032, 1805, 1056, 656, 191~
## $ dep_delay <dbl> 57, 14, -5, 15, 2, -10, 1, 20, -4, -4, 40, -2, -1, -6, -7, 2~
## $ arr_time  <int> 1903, 1008, 1313, 2239, 1644, 1737, 1352, 2122, 1412, 1039, ~
## $ arr_delay <dbl> 48, 38, -28, -6, -21, -13, -10, 2, -13, -6, 2, -5, -30, -22,~
## $ carrier   <chr> "DL", "UA", "DL", "UA", "UA", "UA", "B6", "AA", "UA", "DL", ~
## $ tailnum   <chr> "N711ZX", "N502UA", "N717TW", "N24212", "N76269", "N532UA", ~
## $ flight    <int> 1322, 691, 1765, 1214, 1111, 394, 641, 177, 642, 1865, 272, ~
## $ origin    <chr> "JFK", "JFK", "JFK", "EWR", "EWR", "JFK", "JFK", "JFK", "JFK~
## $ dest      <chr> "SFO", "SFO", "SFO", "SFO", "SFO", "SFO", "SFO", "SFO", "SFO~
## $ air_time  <dbl> 358, 367, 338, 353, 341, 355, 359, 338, 347, 361, 332, 351, ~
## $ distance  <dbl> 2586, 2586, 2586, 2565, 2565, 2586, 2586, 2586, 2586, 2586, ~
## $ hour      <dbl> 15, 6, 9, 19, 13, 14, 10, 18, 10, 6, 19, 8, 10, 18, 7, 17, 1~
## $ minute    <dbl> 27, 13, 55, 28, 40, 15, 32, 5, 56, 56, 10, 33, 48, 49, 23, 2~
```

3 Exercise 3

Describe the distribution of the arrival delays of these flights using a histogram and appropriate summary statistics. Hint: The summary statistics you use should depend on the shape of the distribution.

Let's plot the *arrival delay* of flights to SFO. We will use a fine-grained *binwidth = 5* because there are relatively few observations (*68*) and we want to see detailed shape of the distribution.

```
ggplot(data = sfo_feb_flights, aes(x = arr_delay)) +
  geom_histogram(binwidth = 10)
```



The *arrival delay* for SFO bound flights has outliers on the right. Due to outliers in the observations, *IQR* and *median* are better descriptive stat than *mean*.

To verify our visual observation, lets compute summary stats. The *median=-11*, *IQR=23.25*, *mean=-4.5*, *min=-66*, *max=196*. This is in line with the visual observations about outliers.

```
sfo_feb_flights %>%
    summarise(median_dd = median(arr_delay),
              irq_dd = IQR(arr_delay),
              mean_dd = mean(arr_delay),
              min_dd = min(arr_delay),
              max_dd = max(arr_delay))
```

```
## # A tibble: 1 x 5
##   median_dd irq_dd mean_dd min_dd max_dd
##       <dbl>  <dbl>   <dbl>  <dbl>  <dbl>
## 1       -11   23.2    -4.5    -66    196
```

But is it good news for passengers from ALL NYC area airports? Let's find out. With smaller *IQR* and *min/max spread* for arrival delay, SFO bound passengers departing from *EWR* are likely to have a better overall flying experience than *JFK* passengers.

```
sfo_feb_flights %>%
    group_by(origin) %>%
    summarise(median_dd = median(arr_delay),
              iqr_dd = IQR(arr_delay),
              min_dd = min(arr_delay),
              max_dd = max(arr_delay),
              spread_dd = max_dd - min_dd)
```

```
## # A tibble: 2 x 6
##   origin median_dd iqr_dd min_dd max_dd spread_dd
##   <chr>      <dbl>  <dbl>  <dbl>  <dbl>     <dbl>
## 1 EWR        -15.5   17.5    -35      7        42
## 2 JFK        -10.5   22.8    -66    196       262
```

4 Exercise 4

Calculate the median and interquartile range (IQR) for arr_delays of flights in in the sfo_feb_flights data frame, grouped by carrier. Which carrier has the most variable arrival delays?

DL and UA are *tied* for highest IRQ, with VX a close third. Of the 2 carriers with the highest IQR, UA has the largest *min/max spread*. So UA is the carrier with the most variable arrival delays.

```
sfo_feb_flights %>%
    group_by(carrier) %>%
    summarise(
              iqr_dd = IQR(arr_delay),
              median_dd = median(arr_delay),
              min_dd = min(arr_delay),
              max_dd = max(arr_delay),
              spread_dd = max_dd - min_dd) %>%
    arrange(desc(iqr_dd))
```

```
## # A tibble: 5 x 6
##   carrier iqr_dd median_dd min_dd max_dd spread_dd
##   <chr>    <dbl>     <dbl>  <dbl>  <dbl>     <dbl>
## 1 DL        22       -15      -48     48        96
## 2 UA        22       -10      -35    196       231
## 3 VX        21.2     -22.5    -66     99       165
## 4 AA        17.5       5      -26     76       102
## 5 B6        12.2     -10.5    -18     11        29
```

5 Exercise 5

Suppose you really dislike departure delays and you want to schedule your travel in a month that minimizes your potential departure delay leaving NYC. One option is to choose the month with the lowest mean departure delay. Another option is to choose the month with the lowest median departure delay. What are the pros and cons of these two choices?

Mean Pro: It tells you what is the average amount of departure delay over the whole data set. Mean Con: It can be heavily by skewed by outliers in the observations.

Median Pro: Not impacted by outliers in observations, tell you that half of the time departure will be less (greater) than the median. Mean Con: Does not give a sense of how the data is distributed. For example, the median departure delay of observations *[0, 10, 20, 30, 500, 1000, 2000]* is *30*, but it is grossly misleading.

6 Exercise 6

If you were selecting an airport simply based on on-time departure percentage, which NYC airport would you choose to fly out of?

Let's assume a flight that leaves 5 minutes after scheduled departure is considered delayed. Let's compute on-time departures.

```
nycflights <- nycflights %>%
    mutate(dep_type = ifelse(dep_delay <= 5, "on time", "delayed"))
```

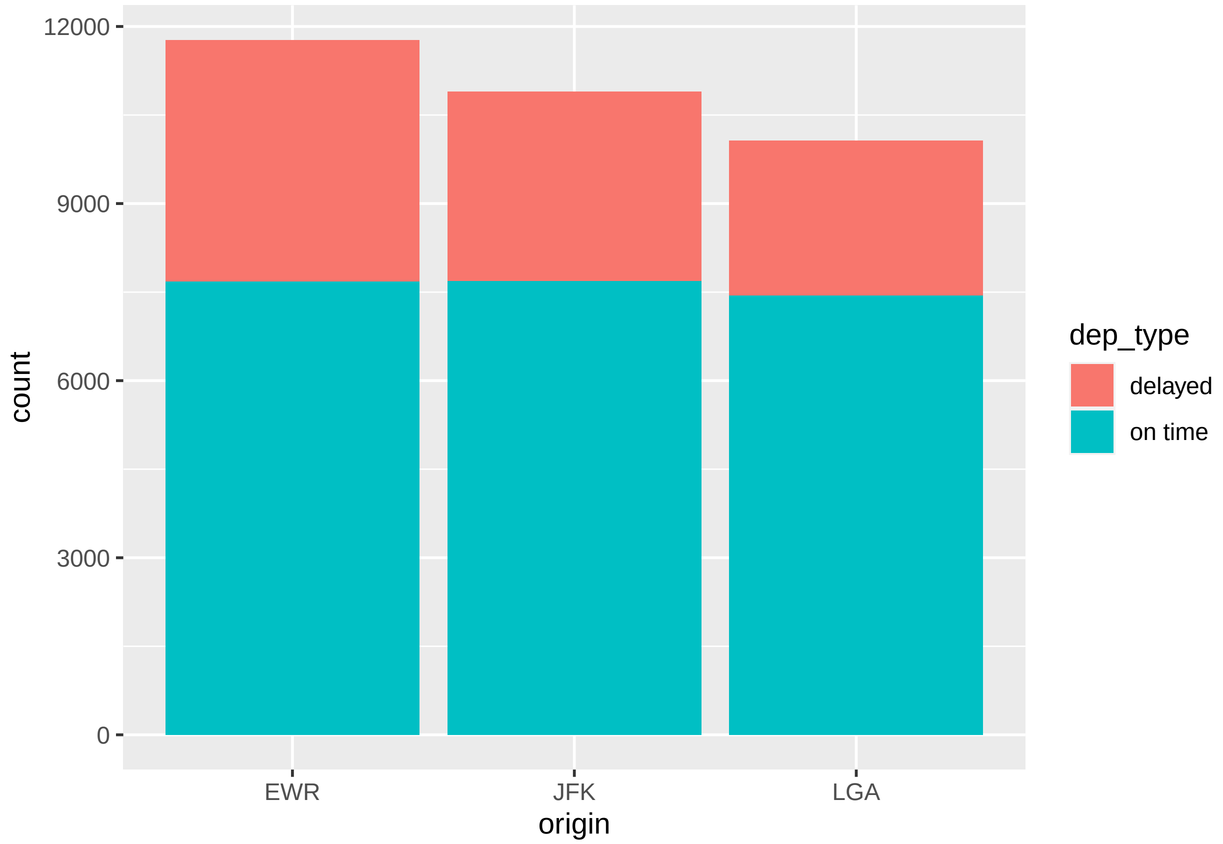
Now we compute the on-time departure percentage for all airports. Based on the on-time departure percentages, I would fly out of (in order): LGA, JFK, EWR.

```
nycflights %>%
    group_by(origin) %>%
    summarise(ot_dep_rate = sum(dep_type == "on time") / n()) %>%
    arrange(desc(ot_dep_rate))
```

```
## # A tibble: 3 x 2
##   origin ot_dep_rate
##   <chr>        <dbl>
## 1 LGA          0.739
## 2 JFK          0.705
## 3 EWR          0.652
```

A picture is worth a thousand words. A quick visual inspection shows that LGA has best proportion of flights leaving on time.

```
ggplot(data = nycflights, aes(x = origin, fill = dep_type)) +
  geom_bar()
```



7 Exercise 7

Mutate the data frame so that it includes a new variable that contains the average speed, avg_speed traveled by the plane for each flight (in mph). Hint: Average speed can be calculated as distance divided by number of hours of travel, and note that air_time is given in minutes.

Computation: group by *flight*, summsarise sums of *air_time* and *distance*, and finally compute the *ave_speed* (making sure to convert *air_time* from *minutes* to *hours*).

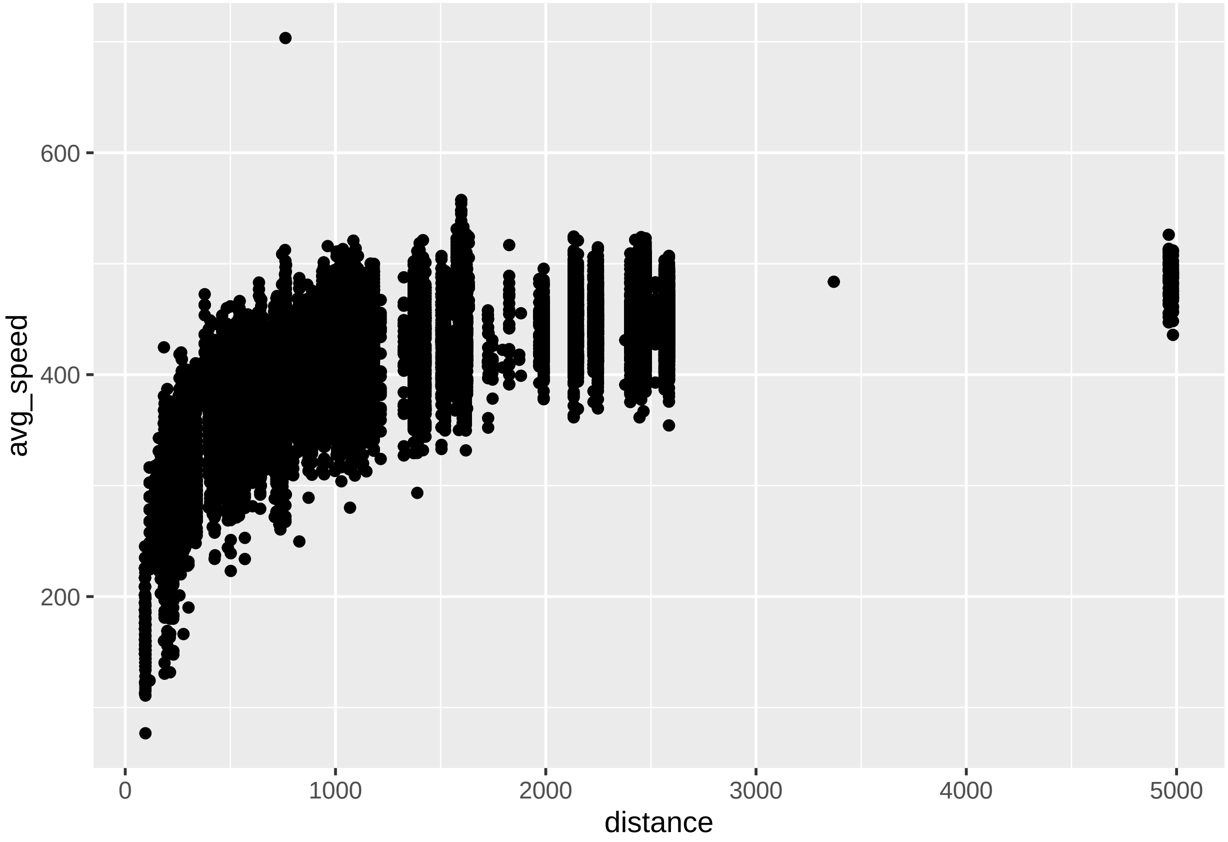
```
nycflights <- nycflights %>%
    group_by(flight) %>%
    mutate(avg_speed = 60 * (distance / air_time)) %>%  # Compute avg_speed mph
    arrange(desc(flight))
```

8 Exercise 8

Make a scatterplot of avg_speed vs. distance. Describe the relationship between average speed and distance. Hint: Use geom_point().

Average speed is directly proportional to distance. Longer (shorter) distance flights have faster (slower) average speed. The relationship **is not** liner, the average speed tapers off as the distances get greater.

```
nycflights %>%
    ggplot(aes(x = distance, y = avg_speed)) + geom_point()
```



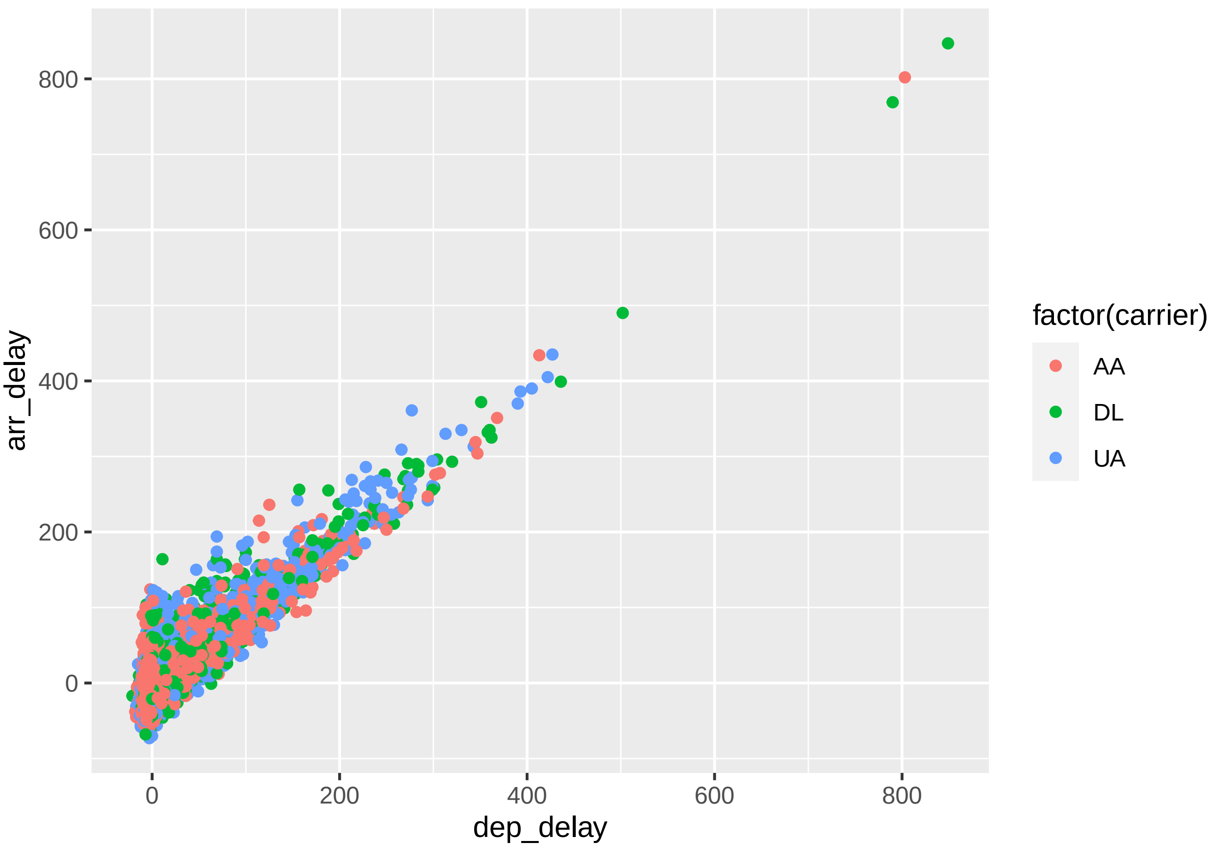
9 Exercise 9

Replicate the following plot. Hint: The data frame plotted only contains flights from American Airlines, Delta Airlines, and United Airlines, and the points are colored by carrier. Once you replicate the plot, determine (roughly) what the cutoff point is for departure delays where you can still expect to get to your destination on time.

From the plot the maximum departure delay, for still getting to the destination on time, is approximately 60 minutes (look at the *horizontal line* where *arr_delay = 0*. Move along the line to the right, until the last point on line).

However, for these 3 airlines, a majority of flights departing exactly on time (*dep_delay = 0*) nonetheless arrived late (*arr_delay > 0*). Assuming an arrival delay of [10-20] minutes is acceptable, flights would have to depart approximately 10 minutes *before* scheduled departure to arrive ‘on time’.

```
nycflights %>%
    filter(carrier %in% c('AA', 'DL', 'UA')) %>%
    ggplot(aes(x = dep_delay, y = arr_delay)) + geom_point(aes(color = factor(carrier)))
```


arr_delay
800
600
400
200
0
dep_delay
0
200
400
600
800
factor(carrier)
AA
DL
UA