Lab2 - Introduction to data

Jawaid Hakim

2022-09-12

Contents

Load required packages.

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

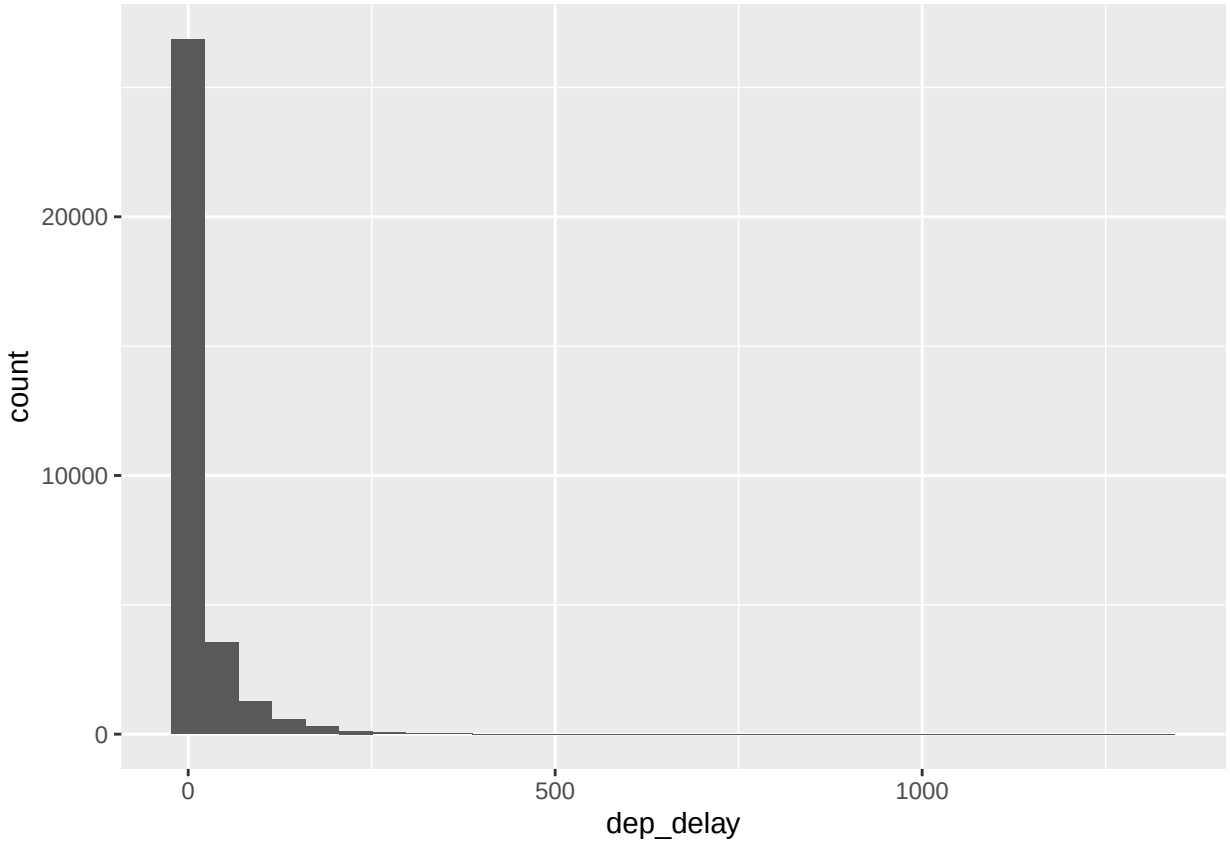
Load *nycflights* data.

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

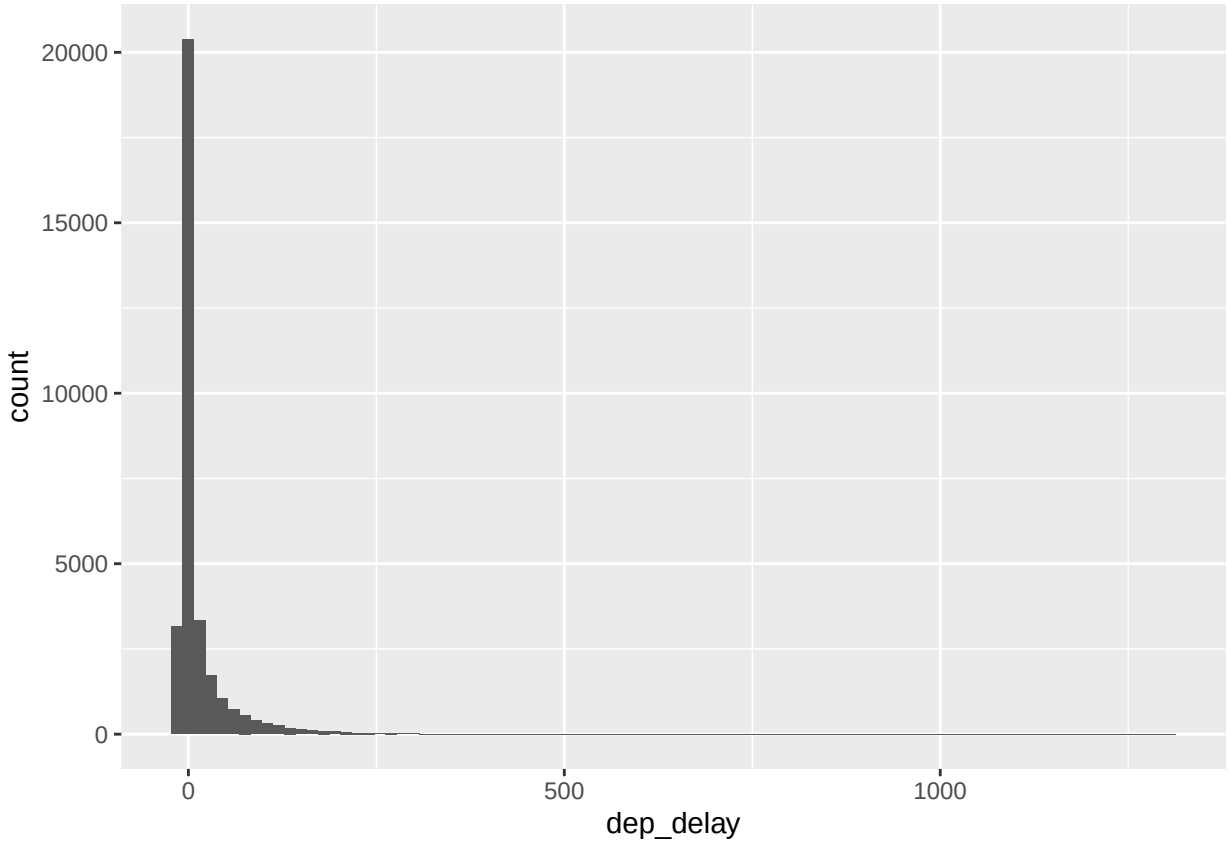
1 Exercise 1

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

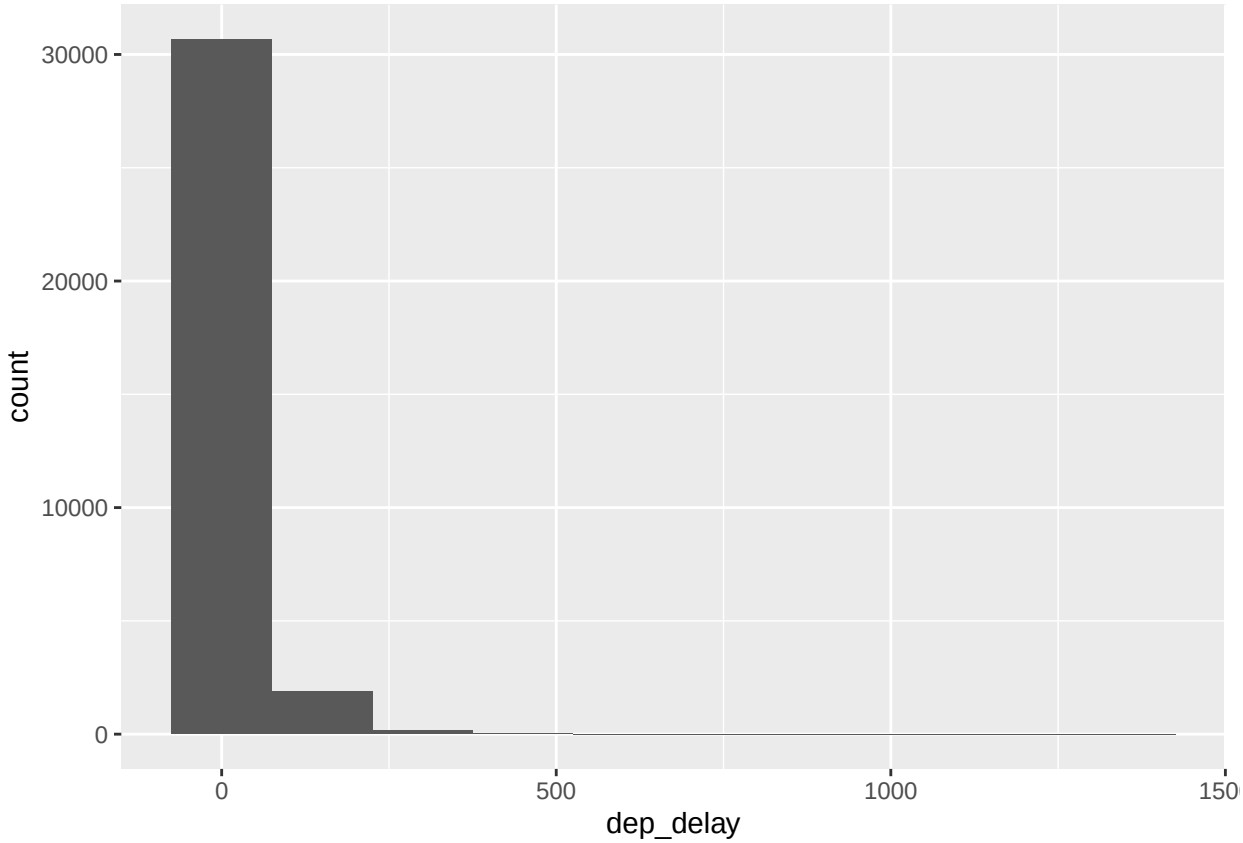
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
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 headed to SFO in February.

```
sfo_feb_flights <- nycflights %>%
    filter(grepl("SFO", dest, ignore.case = TRUE), month == 2) # Case insensitive filtering using grepl
NROW(sfo_feb_flights)
```

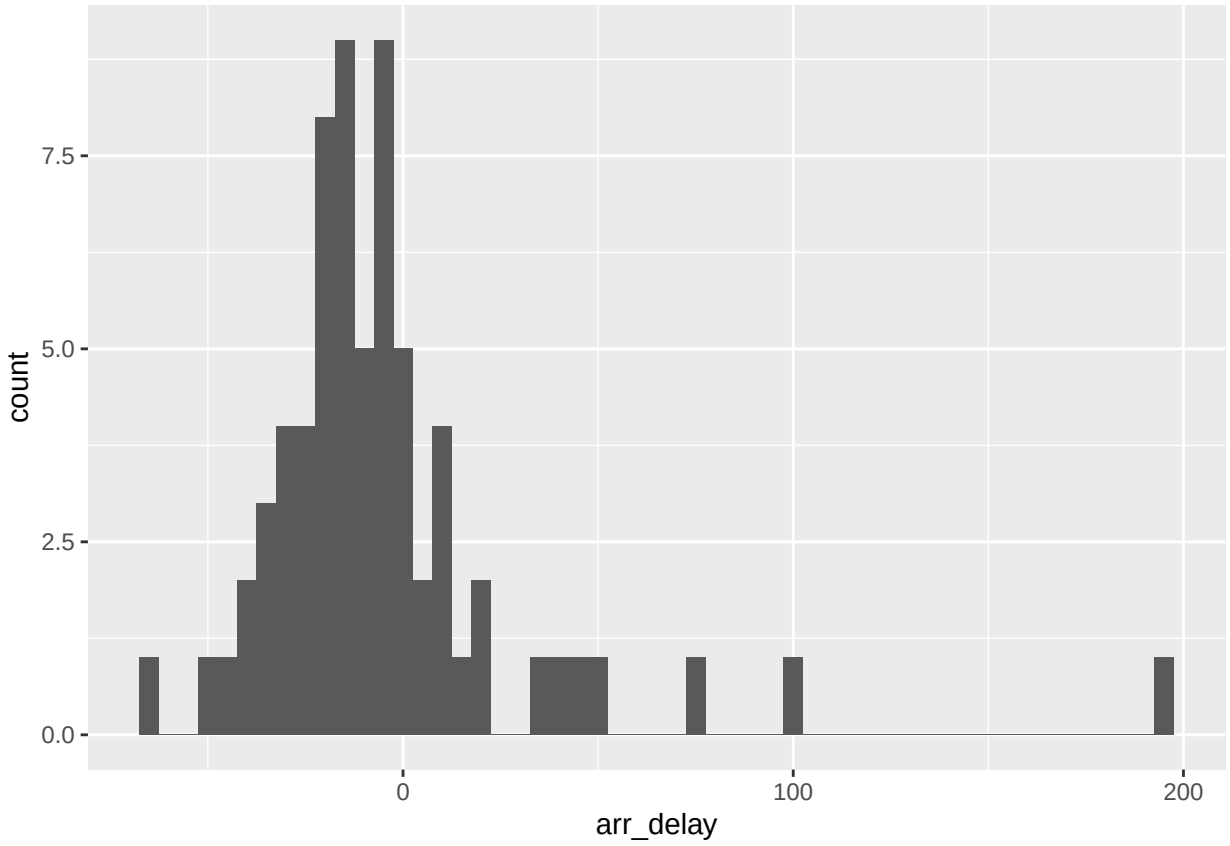
```
## [1] 68
```

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* 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 = 5)
```



Looks like the *arrival delay* for SFO bound flights is clustered around -50 and 25 with some large outliers to the right. If one assumes that arrival delays negatively impact the passenger experience, e.g. delays in departure times and missed connections, it seems to be good news for passengers on majority of flights to SFO!

Due to the large outliers in the data the *IQR* and *demian* are better descriptive stat than *mean*.

To verify our visual observation, lets compute summary stats. For this data the *median=-11*, *IQR=23.25*, *min=-66*, *max=196*. This is in line with the visual observations.

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

```
## # A tibble: 1 x 4
##   median_dd irq_dd min_dd max_dd
##       <dbl>  <dbl>  <dbl>  <dbl>
## 1       -11   23.2    -66    196
```

But is it good news for passengers from ALL NYC area airports? Let's find out. With smaller *median* and *IQR* 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_dela
```

```
## # A tibble: 2 x 5
##   origin median_dd iqr_dd min_dd max_dd
##   <chr>      <dbl>  <dbl>  <dbl>  <dbl>
## 1 EWR        -15.5   17.5    -35      7
## 2 JFK        -10.5   22.8    -66    196
```

4 Exercise 4

Calculate the median and interquartile range 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 largest spread between *min* and *max*. 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)) %>%
    arrange(desc(iqr_dd))
```

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

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 is impacted by large variability in data - i.e. a large *min* or *max* can cause significant change in the mean. Median is a more stable measure but does not indicate how large are the spread on either side of the median.

In conclusion, from the perspective of a passenger, *mean* would be the correct measure to look at for choosing the best month to travel.

Let's validate our intuition by looking at a summary of flight delays.

The *median* is in range *[-3, 0]* which tells us that half of the flights experienced no delay. But this is not very useful to passengers in deciding which months are best for travel.

Other the other hand, the *mean* clearly shows that *June, July, and December* experience the most significant delays (holiday seasons). *September* is the best month for travel (summer vacation is over, students are back in school).

```
nycflights %>%
  group_by(month) %>%
  summarise(mean_dd = mean(dep_delay),
            median_dd = median(dep_delay)) %>%
  arrange(desc(mean_dd))
```

```
## # A tibble: 12 x 3
##    month mean_dd median_dd
##    <int>   <dbl>     <dbl>
##  1     7   20.8          0
##  2     6   20.4          0
##  3    12   17.4          1
##  4     4   14.6         -2
##  5     3   13.5         -1
##  6     5   13.3         -1
##  7     8   12.6         -1
##  8     2   10.7         -2
##  9     1   10.2         -2
## 10     9    6.87        -3
## 11    11    6.10        -2
## 12    10    5.88        -3
```

Just for fun, let's filter out flights that did not experience delays. Now the *median* is more descriptive but the *mean* is still the better measure.

```
nycflights %>%
  group_by(month) %>%
  filter(dep_delay > 0) %>%
  summarise(mean_dd = mean(dep_delay),
            median_dd = median(dep_delay)) %>%
  arrange(desc(median_dd))
```

```
## # A tibble: 12 x 3
##    month mean_dd median_dd
##    <int>   <dbl>     <dbl>
##  1     4    46.4        25
##  2     6    49.1        24
##  3     7    47.0        24
##  4     5    40.0        22
##  5     3    40.8        20
##  6    12    38.2        20
##  7     2    34.2        18
##  8     8    38.3        17
##  9     1    34.8        15
## 10    11    28.3        15
```

```
## 11     9    36.9      14.5
## 12    10    31.5      14
```

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(ot_dep = ifelse(dep_delay <= 5, TRUE, FALSE))
```

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 of preference): LGA, then JFK, and finally EWR.

```
nycflights %>%
    group_by(origin) %>%
    summarise(ot_dep_rate = sum(ot_dep == TRUE) / 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
```

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*).

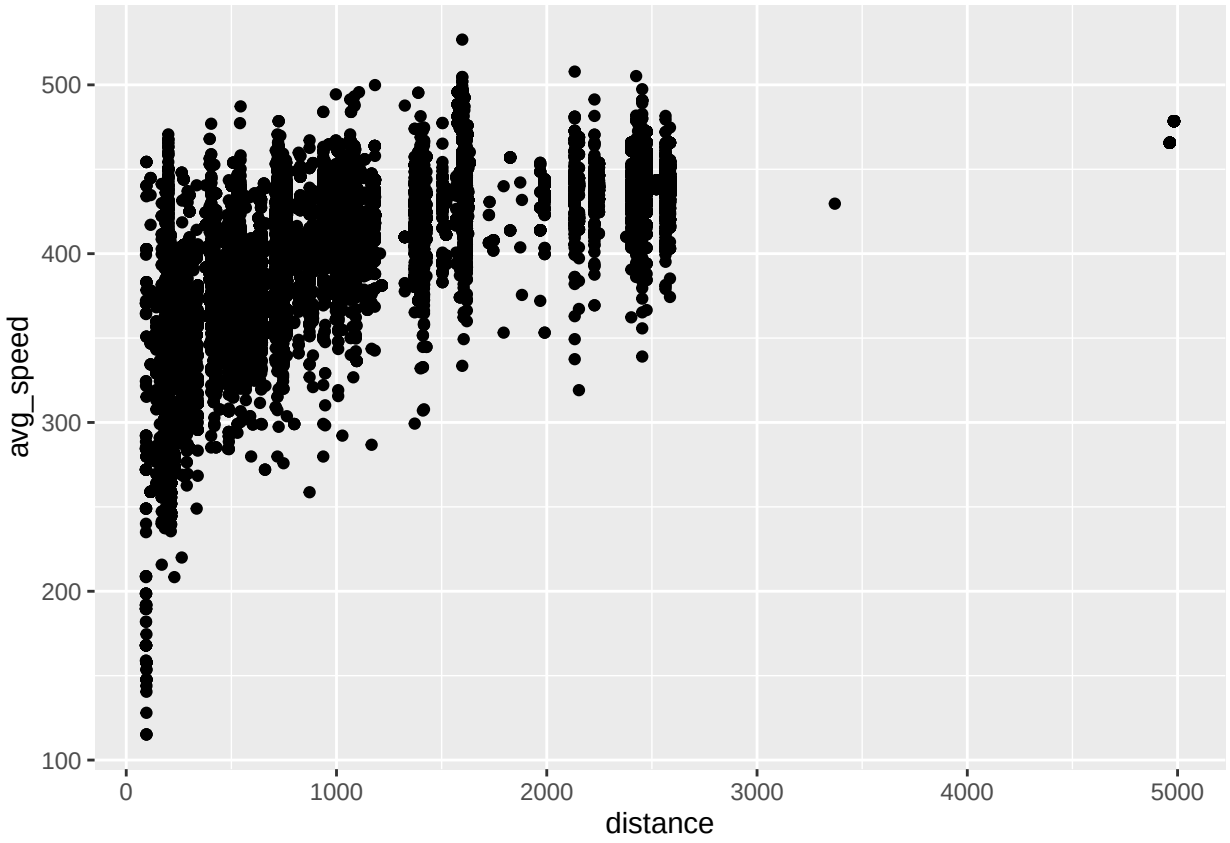
```
#nycflights1 <- nycflights %>%
#    group_by(flight) %>%                                    # Group by flight
#    summarise(total_air_time = sum(air_time) / 60,          # Sum air_time (in hours) and dist
#              total_distance = sum(distance)) %>%
#    mutate(avg_speed = total_distance / total_air_time) %>%  # Compute avg_speed mph
#    arrange(desc(flight))

nycflights1 <- nycflights %>%
    group_by(flight) %>%
    mutate(total_air_time_hrs = sum(air_time) / 60,          # Sum air_time (in hours) and dist
           total_distance = sum(distance),
           avg_speed = total_distance / total_air_time_hrs) %>%
    arrange(desc(flight))
```

8 Exercise 8

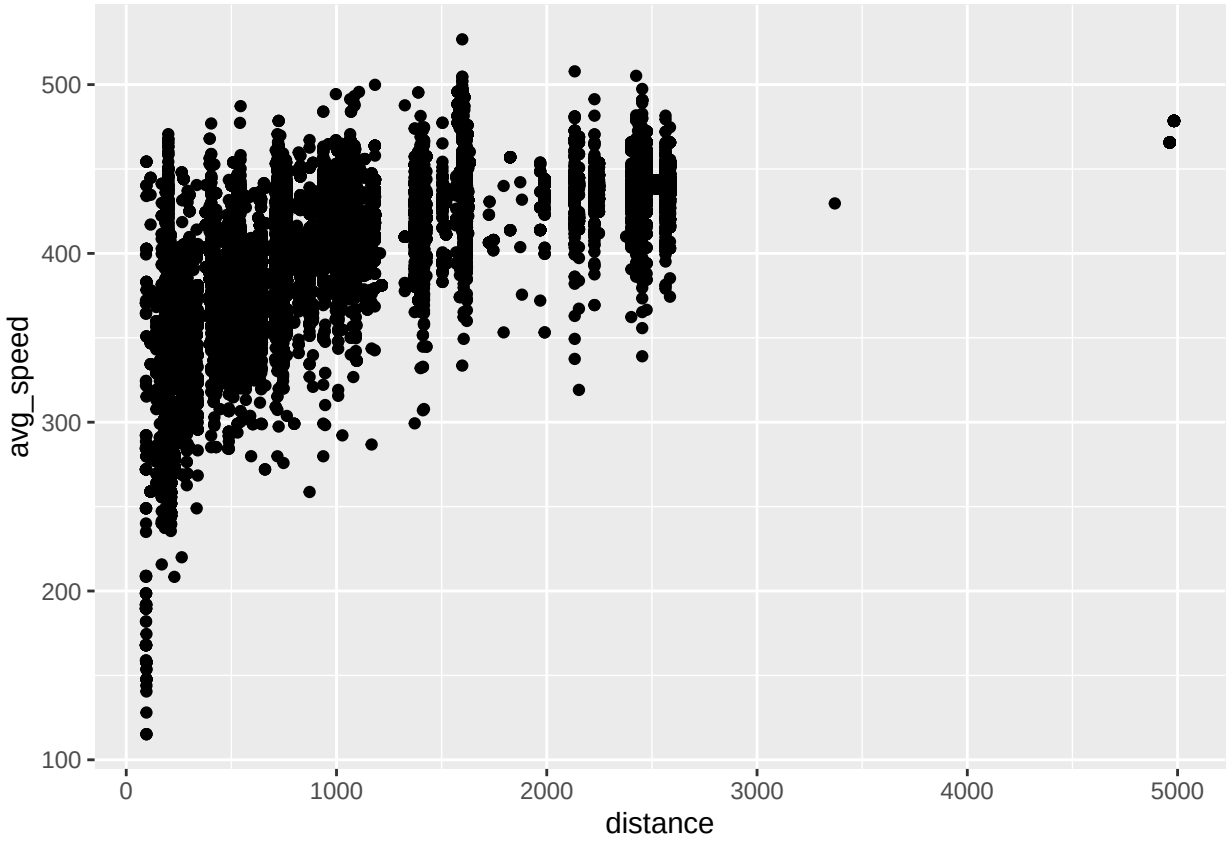
Make a scatterplot of avg_speed vs. distance. Describe the relationship between average speed and distance. Hint: Use geom_point().

Average speed is higher for longer distance flights. In addition, range (min - max) of average speed range of average is greater for *smaller* distances, although this might be because of larger number of observations for smaller distances.

```
nycflights1 %>%
    group_by(flight) %>%
    ggplot(aes(x = distance, y = avg_speed)) + geom_point()
```




```
nycflights1 %>%
    ggplot(aes(x = distance, y = avg_speed)) + geom_point()
```



9 Exercise 9

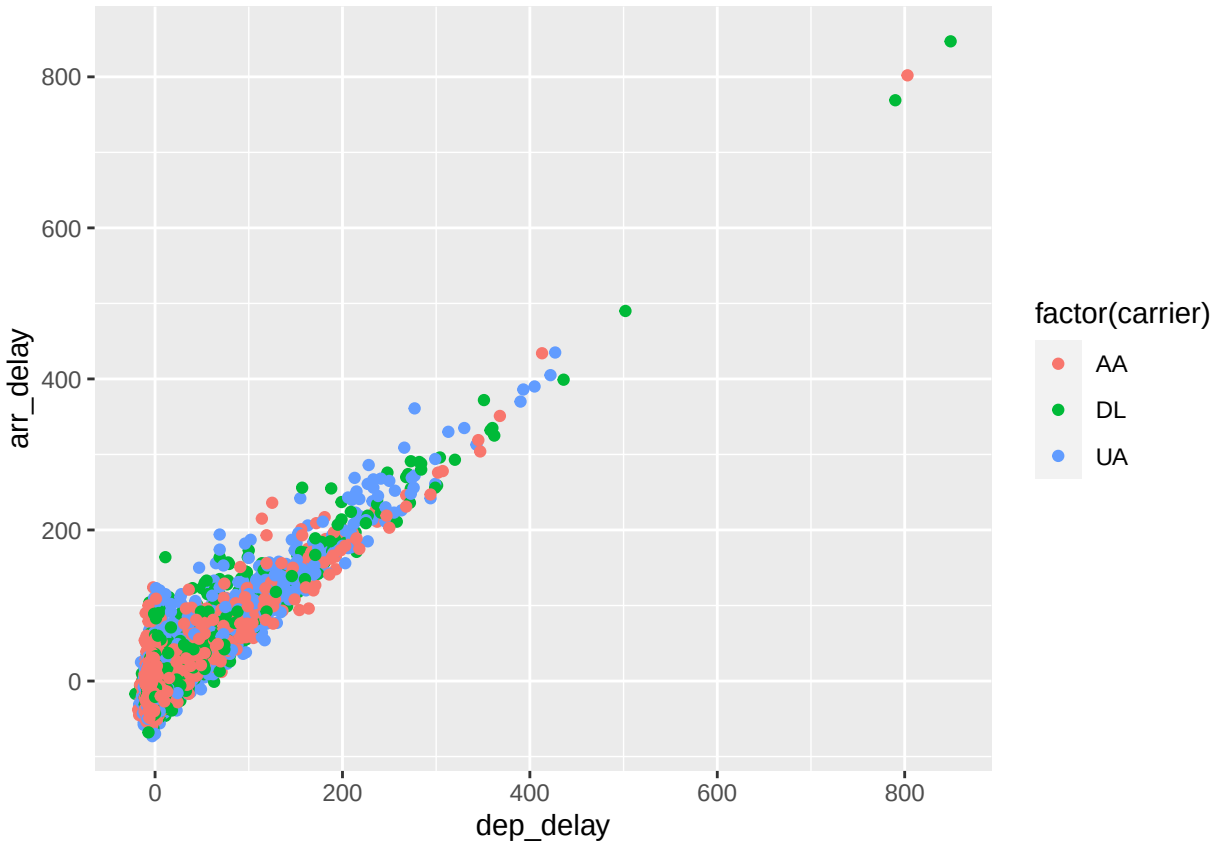
Replicate the following plot. Hint: The data frame plotted only contains flights from American Airlines, Delta Airlines, and United Airlines, and the points are colored by carrier. Once you replicate the plot, determine (roughly) what the cutoff point is for departure delays where you can still expect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

Let's filter dataset for the desired carriers.

```
nycflights2 <- nycflights1 %>%
    filter(carrier %in% c('AA', 'DL', 'UA'))
```

From the plot the maximum departure delay for still getting to the destination on time is approximately 60 minutes (greatest *dep_delay* where *arr_delay = 0*).

```
nycflights2 %>%
    ggplot(aes(x = dep_delay, y = arr_delay)) + geom_point(aes(color = factor(carrier)))
```


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