Introduction to data

Jaya Veluri

Some define statistics as the field that focuses on turning information into knowledge. The first step in that process is to summarize and describe the raw information – the data. In this lab we explore flights, specifically a random sample of domestic flights that departed from the three major New York City airports in 2013. We will generate simple graphical and numerical summaries of data on these flights and explore delay times. Since this is a large data set, along the way you'll also learn the indispensable skills of data processing and subsetting.

Getting started

Load packages

In this lab, we will explore and visualize the data using the **tidyverse** suite of packages. The data can be found in the companion package for OpenIntro labs, **openintro**.

Let's load the packages.

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

The data

The Bureau of Transportation Statistics (BTS) is a statistical agency that is a part of the Research and Innovative Technology Administration (RITA). As its name implies, BTS collects and makes transportation data available, such as the flights data we will be working with in this lab.

First, we'll view the `nycflights` data frame. Type the following in your console to load the data:

```
data(nycflights)
```

The data set `nycflights` that shows up in your workspace is a *data matrix*, with each row representing an *observation* and each column representing a *variable*. R calls this data format a **data frame**, which is a term that will be used throughout the labs. For this data set, each *observation* is a single flight.

To view the names of the variables, type the command

```
names(nycflights)
```

```
##  [1] "year"      "month"     "day"       "dep_time"  "dep_delay" "arr_time"
##  [7] "arr_delay" "carrier"   "tailnum"   "flight"    "origin"    "dest"
## [13] "air_time"  "distance"  "hour"      "minute"
```

This returns the names of the variables in this data frame. The **codebook** (description of the variables) can be accessed by pulling up the help file:

```
?nycflights
```

Remember that you can use `glimpse` to take a quick peek at your data to understand its contents better.

```
glimpse(nycflights)
```

```
## Rows: 32,735
## Columns: 16
## $ year      <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, ~
## $ month     <int> 6, 5, 12, 5, 7, 1, 12, 8, 9, 4, 6, 11, 4, 3, 10, 1, 2, 8, 10~
## $ day       <int> 30, 7, 8, 14, 21, 1, 9, 13, 26, 30, 17, 22, 26, 25, 21, 23, ~
## $ dep_time  <int> 940, 1657, 859, 1841, 1102, 1817, 1259, 1920, 725, 1323, 940~
## $ dep_delay <dbl> 15, -3, -1, -4, -3, -3, 14, 85, -10, 62, 5, 5, -2, 115, -4, ~
## $ arr_time  <int> 1216, 2104, 1238, 2122, 1230, 2008, 1617, 2032, 1027, 1549, ~
## $ arr_delay <dbl> -4, 10, 11, -34, -8, 3, 22, 71, -8, 60, -4, -2, 22, 91, -6, ~
## $ carrier   <chr> "VX", "DL", "DL", "DL", "9E", "AA", "WN", "B6", "AA", "EV", ~
## $ tailnum   <chr> "N626VA", "N3760C", "N712TW", "N914DL", "N823AY", "N3AXAA", ~
## $ flight    <int> 407, 329, 422, 2391, 3652, 353, 1428, 1407, 2279, 4162, 20, ~
## $ origin    <chr> "JFK", "JFK", "JFK", "JFK", "LGA", "LGA", "EWR", "JFK", "LGA~
## $ dest      <chr> "LAX", "SJU", "LAX", "TPA", "ORF", "ORD", "HOU", "IAD", "MIA~
## $ air_time  <dbl> 313, 216, 376, 135, 50, 138, 240, 48, 148, 110, 50, 161, 87,~
## $ distance  <dbl> 2475, 1598, 2475, 1005, 296, 733, 1411, 228, 1096, 820, 264,~
## $ hour      <dbl> 9, 16, 8, 18, 11, 18, 12, 19, 7, 13, 9, 13, 8, 20, 12, 20, 6~
## $ minute    <dbl> 40, 57, 59, 41, 2, 17, 59, 20, 25, 23, 40, 20, 9, 54, 17, 24~
```

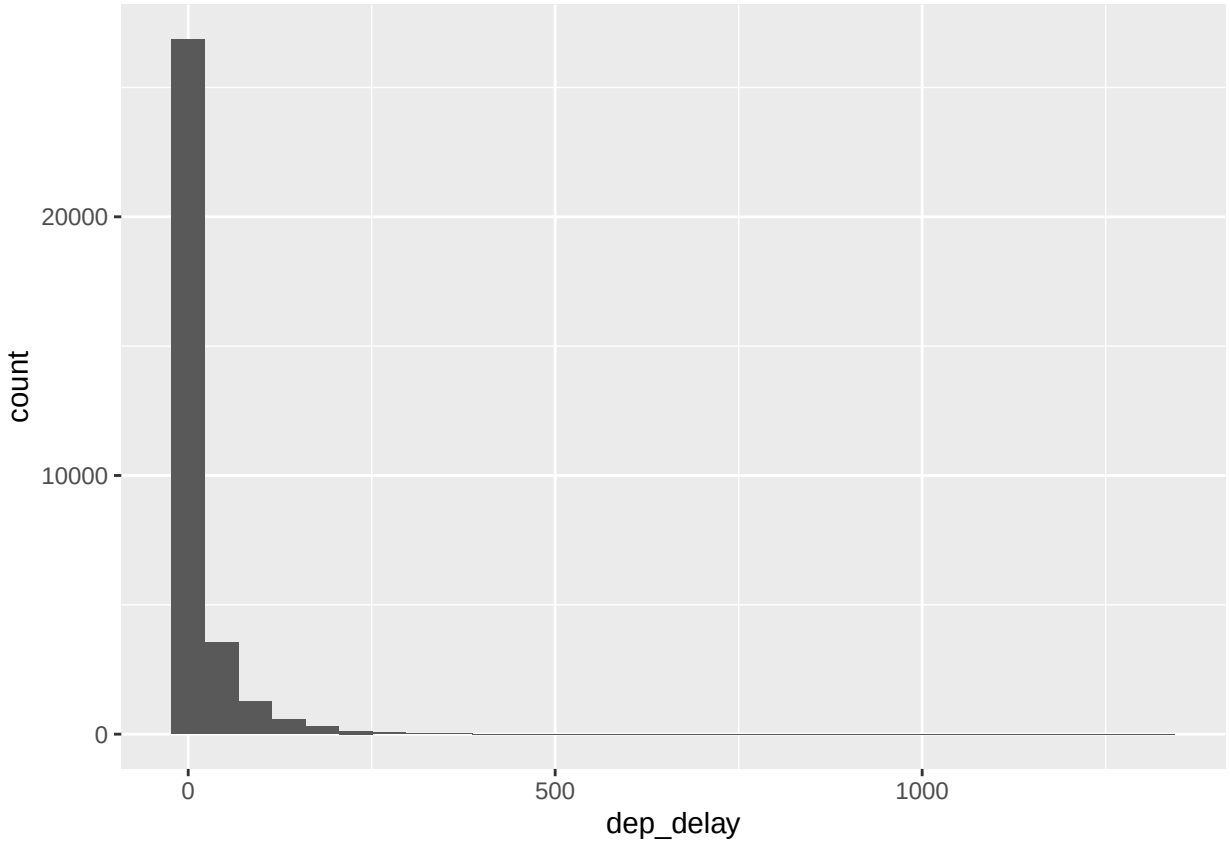
Analysis

Departure delays

Let's start by examing the distribution of departure delays of all flights with a histogram.

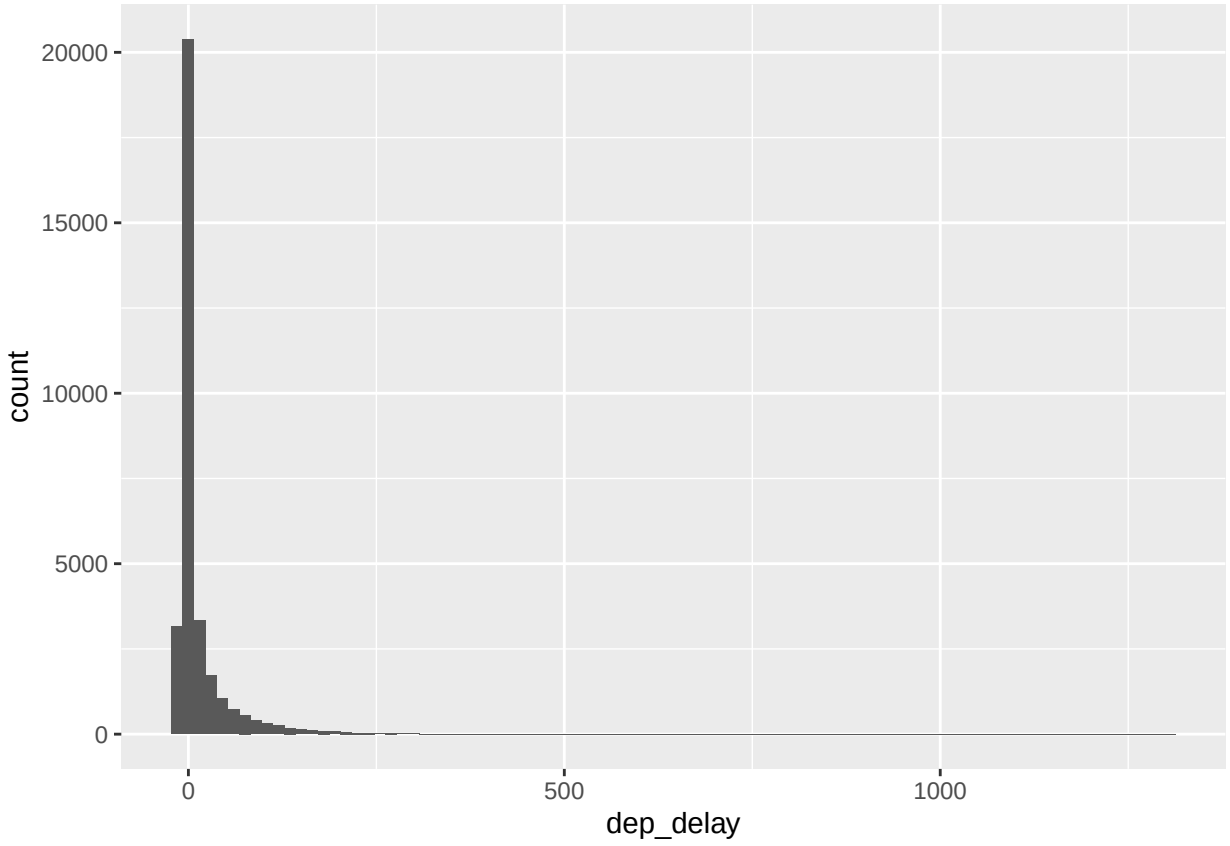
```
ggplot(data = nycflights, aes(x = dep_delay)) +
  geom_histogram()
```

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

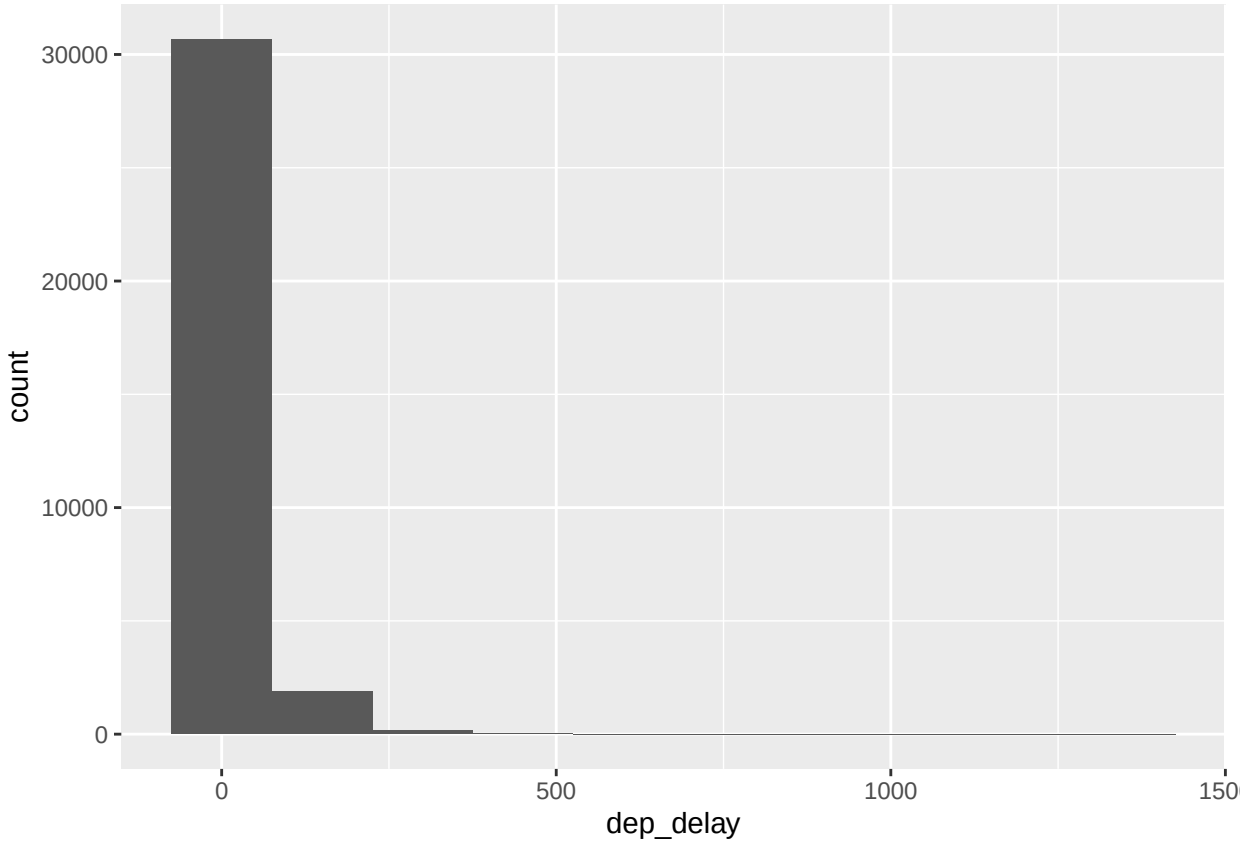


Histograms are generally a very good way to see the shape of a single distribution of numerical data, but that shape can change depending on how the data is split between the different bins. You can easily define the binwidth you want to use:

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




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



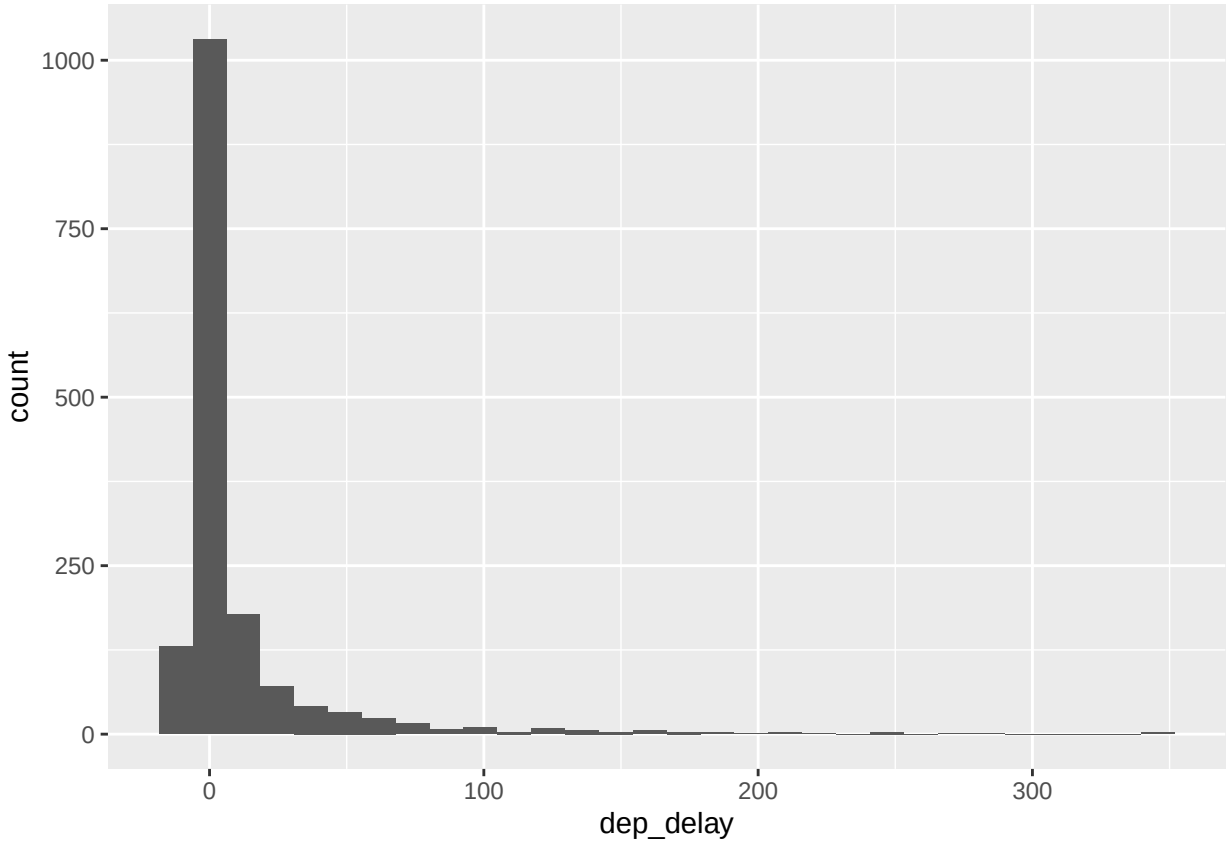
Exercise 1

Look carefully at these three histograms. How do they compare? Are features revealed in one that are obscured in another?

If you want to visualize only on delays of flights headed to Los Angeles, you need to first `filter` the data for flights with that destination (`dest == "LAX"`) and then make a histogram of the departure delays of only those flights.

```
lax_flights <- nycflights %>%
  filter(dest == "LAX")
ggplot(data = lax_flights, aes(x = dep_delay)) +
  geom_histogram()
```

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



You can also obtain numerical summaries for these flights:

```
lax_flights %>%
  summarise(mean_dd   = mean(dep_delay),
            median_dd = median(dep_delay),
            n         = n())
```

```
## # A tibble: 1 x 3
##   mean_dd median_dd     n
##     <dbl>     <dbl> <int>
## 1    9.78        -1  1583
```

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?

```
sfo_feb_flights <- nycflights %>%
  filter(dest == "SFO", month == 2)
glimpse(sfo_feb_flights)
```

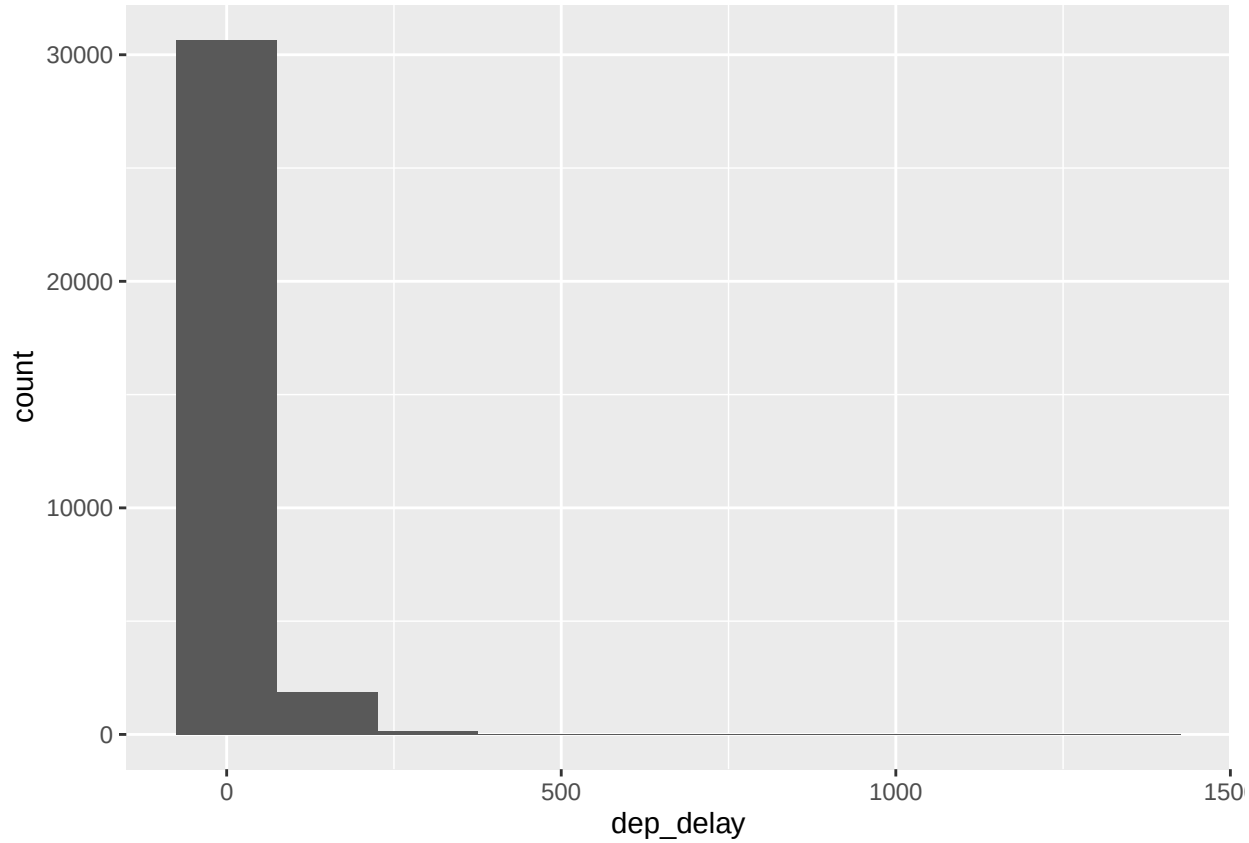
```
## Rows: 68
## Columns: 16
## $ year      <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, ~
```

```
## $ month     <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 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~
```

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.

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



```
sfo_feb_flights %>%
  group_by(origin) %>%
  summarise(median_dd = median(dep_delay), iqr_dd = IQR(dep_delay), n_flights = n())
```

```
## # A tibble: 2 x 4
##   origin median_dd iqr_dd n_flights
##   <chr>      <dbl>  <dbl>     <int>
## 1 EWR          0.5   5.75         8
## 2 JFK         -2.5  15.2         60
```

```
sfo_feb_flights %>%
  summarise(mean_ad = mean(arr_delay), median_ad = median(arr_delay), iqr_ad = IQR(arr_delay), n_flights
```

```
## # A tibble: 1 x 4
##   mean_ad median_ad iqr_ad n_flights
##     <dbl>     <dbl>  <dbl>     <int>
## 1    -4.5       -11   23.2        68
```

The histogram is right-skewed, so the standard deviation would not accurately represent the distribution of the data. The IQR on the other hand describes how the middle 50% of the data is distributed about the median. Both values can be found below.

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?

```
sfo_feb_flights %>%
  group_by(carrier) %>%
  summarise(median_ad = median(arr_delay), iqr_ad = IQR(arr_delay), n_flights = n())
```

```
## # A tibble: 5 x 4
##   carrier median_ad iqr_ad n_flights
##   <chr>       <dbl>  <dbl>     <int>
## 1 AA            5     17.5        10
## 2 B6          -10.5   12.2         6
## 3 DL          -15     22          19
## 4 UA          -10     22          21
## 5 VX          -22.5   21.2        12
```

The carriers DL and UA are tied for having the most variable arrival delays because their interquartile ranges are tied for the highest at 22.00. This suggests that they exhibit the greatest variation in arrival delays for the middle 50% of their data.

Departure delays by month

Which month would you expect to have the highest average delay departing from an NYC airport?

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

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

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: This represents the overall average departure delay, taking into account the effect of each delay and giving an idea as to how the data is distributed. Con: This can be skewed by outliers.

Median Pro: It takes the middle value of the entire data set, so outliers do not skew the median. Con: It fails to represent how the data is distributed.

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

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

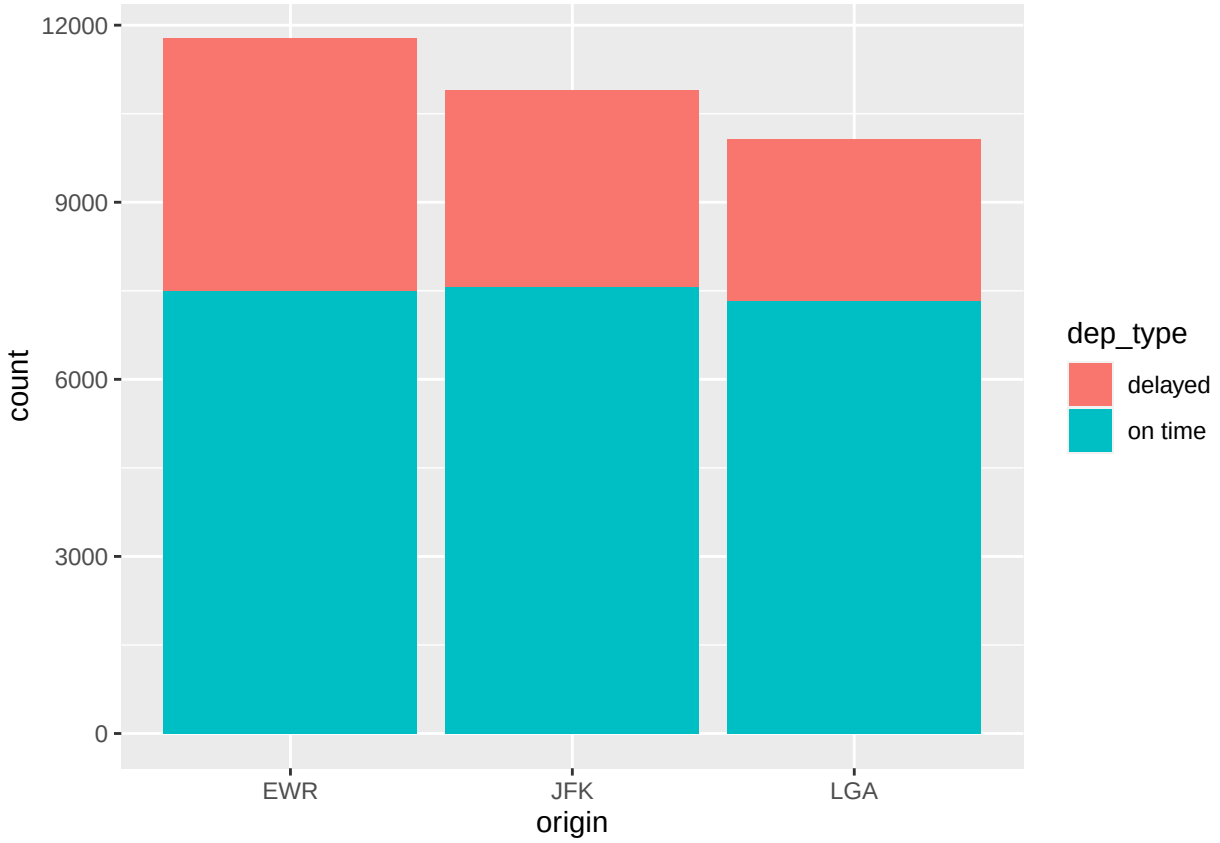
```
## # A tibble: 3 x 2
##   origin ot_dep_rate
##   <chr>        <dbl>
## 1 LGA          0.728
## 2 JFK          0.694
## 3 EWR          0.637
```

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?

You can also visualize the distribution of on on time departure rate across the three airports using a segmented bar plot.

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



LGA has the best time departure percentage of 72.8%. Also, the segmented bar plot below shows that LGA has the best proportion of flights leaving on time. Thus, I would choose LGA.

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.
```

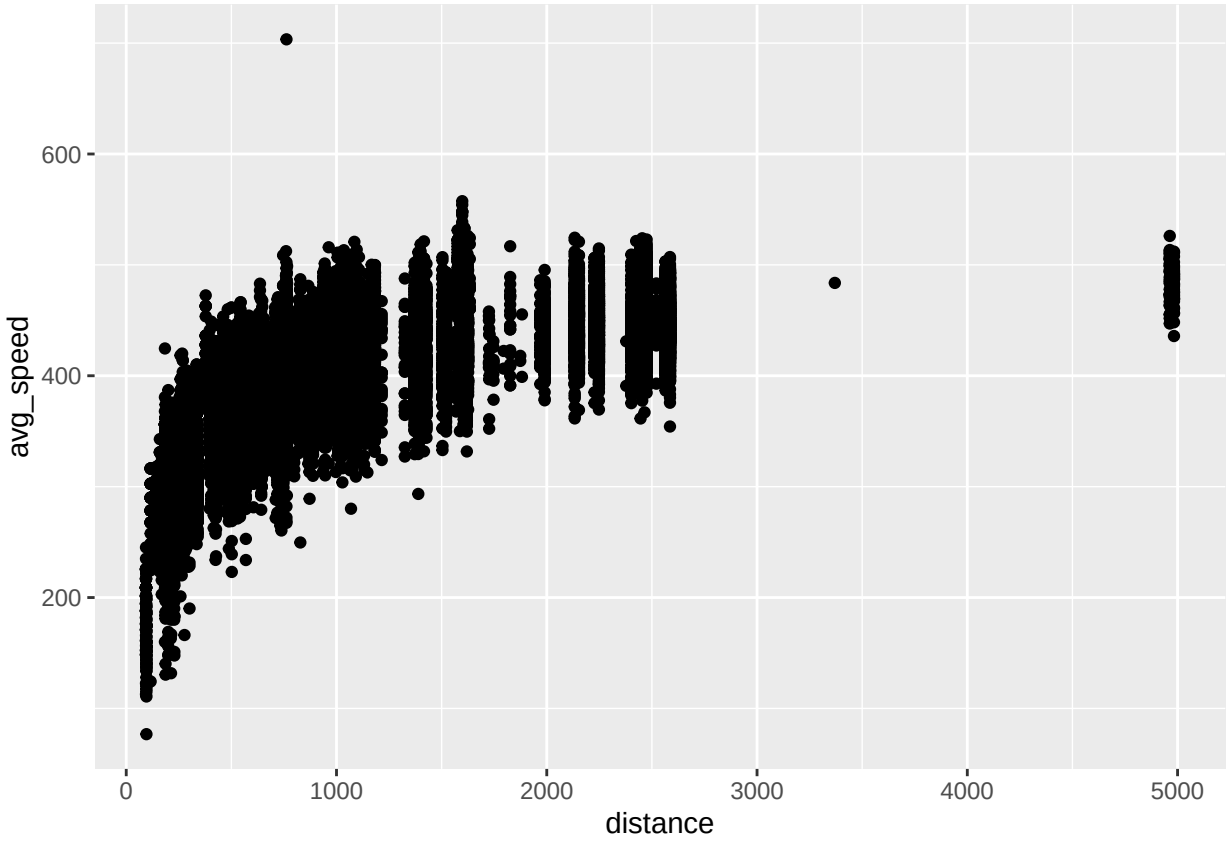
```
nycflights <- nycflights %>%
  mutate(avg_speed = 60*(distance / air_time))

glimpse(nycflights)
```

```
## Rows: 32,735
## Columns: 18
## $ year      <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, ~
## $ month     <int> 6, 5, 12, 5, 7, 1, 12, 8, 9, 4, 6, 11, 4, 3, 10, 1, 2, 8, 10~
## $ day       <int> 30, 7, 8, 14, 21, 1, 9, 13, 26, 30, 17, 22, 26, 25, 21, 23, ~
## $ dep_time  <int> 940, 1657, 859, 1841, 1102, 1817, 1259, 1920, 725, 1323, 940~
## $ dep_delay <dbl> 15, -3, -1, -4, -3, -3, 14, 85, -10, 62, 5, 5, -2, 115, -4, ~
## $ arr_time  <int> 1216, 2104, 1238, 2122, 1230, 2008, 1617, 2032, 1027, 1549, ~
## $ arr_delay <dbl> -4, 10, 11, -34, -8, 3, 22, 71, -8, 60, -4, -2, 22, 91, -6, ~
## $ carrier   <chr> "VX", "DL", "DL", "DL", "9E", "AA", "WN", "B6", "AA", "EV", ~
## $ tailnum   <chr> "N626VA", "N3760C", "N712TW", "N914DL", "N823AY", "N3AXAA", ~
## $ flight    <int> 407, 329, 422, 2391, 3652, 353, 1428, 1407, 2279, 4162, 20, ~
## $ origin    <chr> "JFK", "JFK", "JFK", "JFK", "LGA", "LGA", "EWR", "JFK", "LGA~
## $ dest      <chr> "LAX", "SJU", "LAX", "TPA", "ORF", "ORD", "HOU", "IAD", "MIA~
## $ air_time  <dbl> 313, 216, 376, 135, 50, 138, 240, 48, 148, 110, 50, 161, 87,~
## $ distance  <dbl> 2475, 1598, 2475, 1005, 296, 733, 1411, 228, 1096, 820, 264,~
## $ hour      <dbl> 9, 16, 8, 18, 11, 18, 12, 19, 7, 13, 9, 13, 8, 20, 12, 20, 6~
## $ minute    <dbl> 40, 57, 59, 41, 2, 17, 59, 20, 25, 23, 40, 20, 9, 54, 17, 24~
## $ dep_type  <chr> "delayed", "on time", "on time", "on time", "on time", "on t~
## $ avg_speed <dbl> 474.4409, 443.8889, 394.9468, 446.6667, 355.2000, 318.6957, ~
```

Exercise 8

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

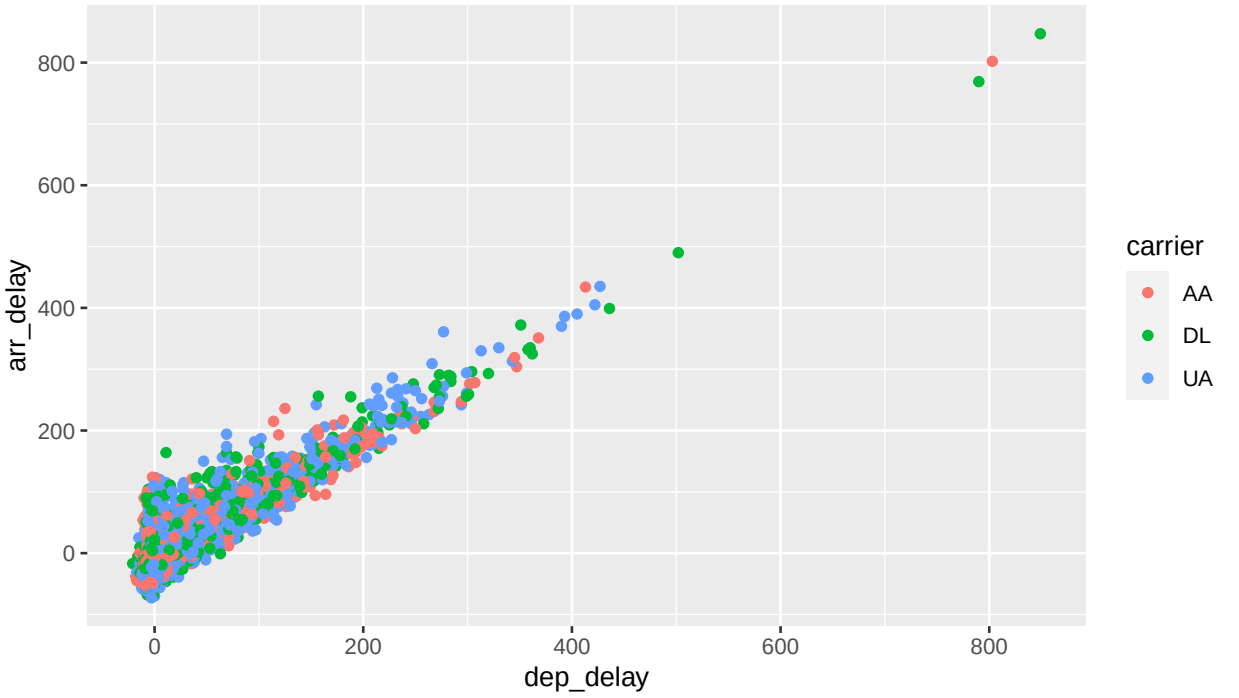
```
ggplot(data = nycflights, aes(x = distance, y = avg_speed)) + geom_point()
```



From the scatter plot below, we see that as distance increases, the average speed increases as well. The relationship appears to be logarithmic i.e. v = Alog(d) + B.

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 ‘color‘ed 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.
```



Based off the scatter plot below, the cutoff point for departure delays you can still REASONABLY expect to arrive at your destination on time is slightly ahead of schedule, roughly first five minutes before departure time. We see that with a 60 minute late departure time, it has still been possible to arrive at the destination on time, but this is extremely rare