Assignment_2

AZM

8/12/2022

Library Load In

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.1
## v readr   2.1.2     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

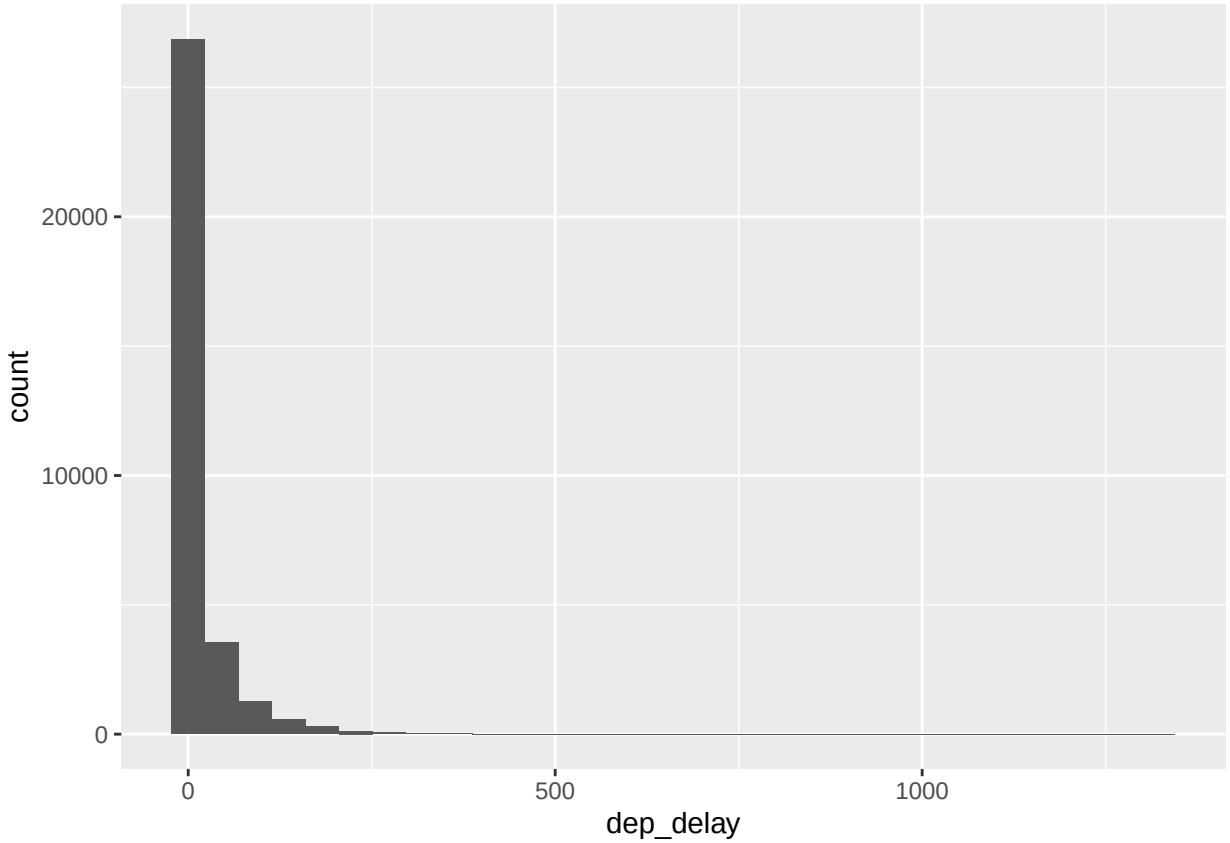
```
library(openintro)
```

```
## Loading required package: airports
## Loading required package: cherryblossom
## Loading required package: usdata
```

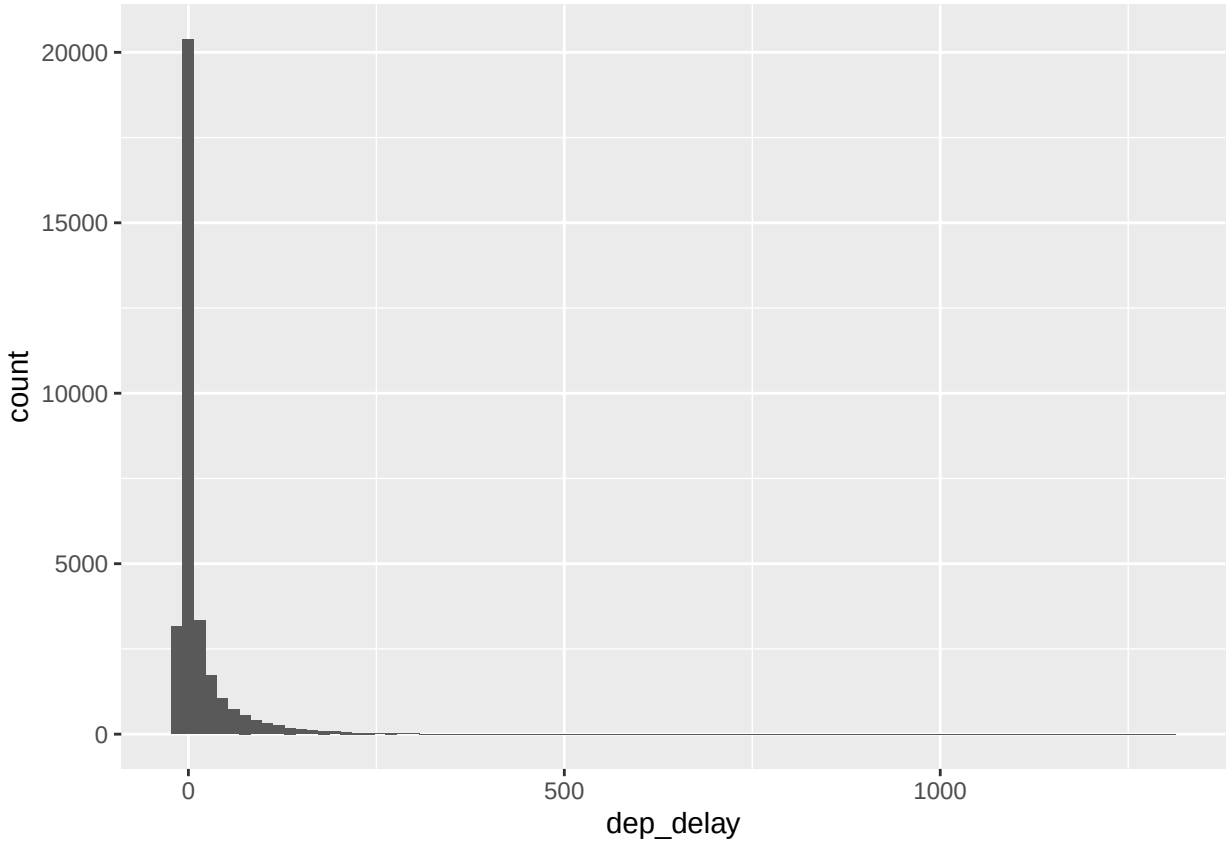
Let's Start charting

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

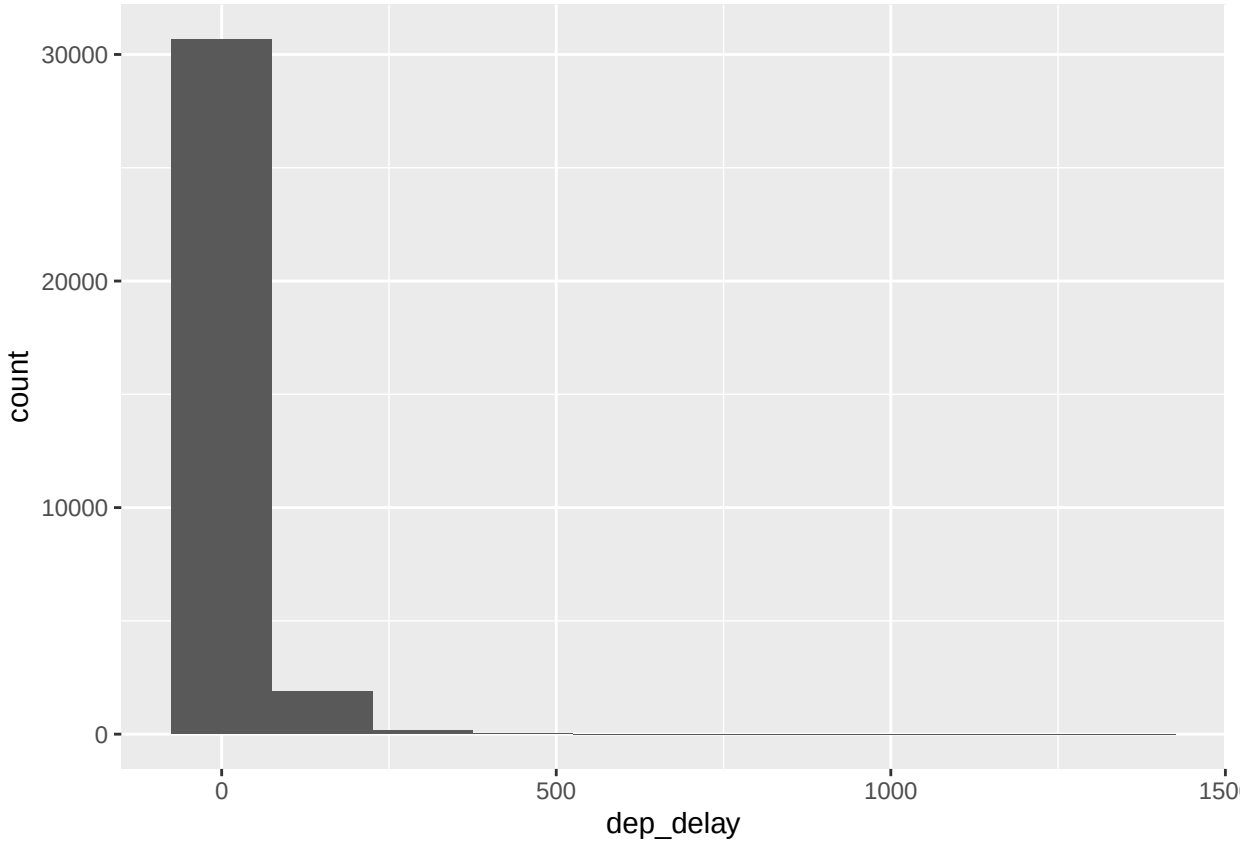
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




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




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



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

Practically each chart displays different bucketing. One unique observable feature is the number of flights have a negative departure delay time, implying they departed earlier than previously expected, which is simply obfuscated in other charts.

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" | origin == "SFO", month == 2)
sfo_feb_flights
```

```
## # A tibble: 68 x 16
##     year month   day dep_time dep_delay arr_time arr_de~1 carrier tailnum flight
##    <int> <int> <int>    <int>     <dbl>    <int>    <dbl> <chr>   <chr>    <int>
##  1  2013     2    18     1527        57     1903       48 DL      N711ZX    1322
##  2  2013     2     3      613        14     1008       38 UA      N502UA     691
##  3  2013     2    15      955        -5     1313      -28 DL      N717TW    1765
##  4  2013     2    18     1928        15     2239       -6 UA      N24212    1214
##  5  2013     2    24     1340         2     1644      -21 UA      N76269    1111
##  6  2013     2    25     1415       -10     1737      -13 UA      N532UA     394
##  7  2013     2     7     1032         1     1352      -10 B6      N627JB     641
##  8  2013     2    15     1805        20     2122        2 AA      N335AA     177
##  9  2013     2    13     1056        -4     1412      -13 UA      N532UA     642
```

```
## 10  2013     2     8      656        -4     1039       -6 DL      N710TW    1865
## # ... with 58 more rows, 6 more variables: origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, and abbreviated
## #   variable name 1: arr_delay
```

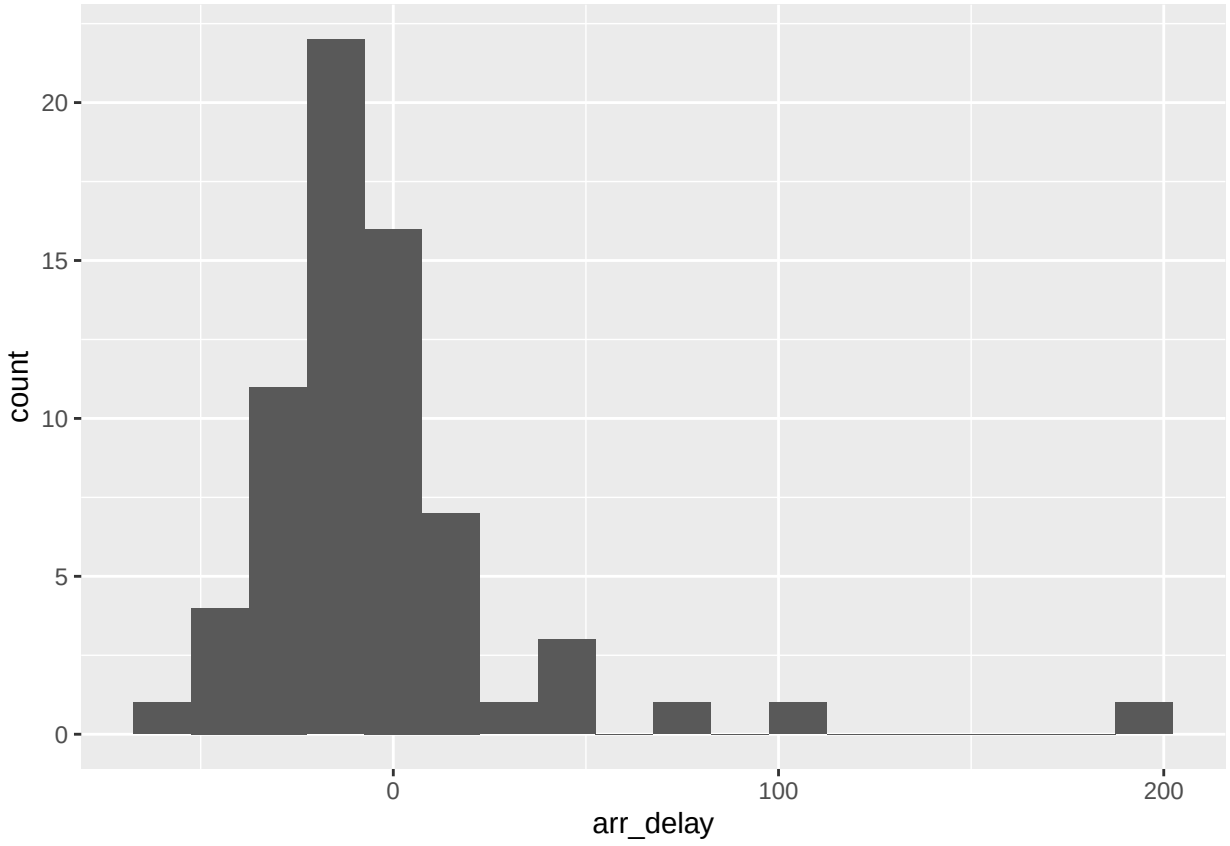
68 Flights meet this criteria

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.

Describe the Distribution in a histogram & summary statistics

```
ggplot(data = sfo_feb_flights, aes(x = arr_delay)) +
  geom_histogram(binwidth = 15)
```




```
sfo_feb_flights %>%
  group_by(dest) %>%
  summarise(mean = mean(arr_delay), median = median(arr_delay), standard_deviation = sd(arr_delay), n_f
```

```
## # A tibble: 1 x 5
##   dest   mean median standard_deviation n_flights
##   <chr> <dbl>  <dbl>              <dbl>     <int>
## 1 SFO    -4.5    -11               36.3        68
```

Practically these are a somewhat bellcurve shaped distribution, with a few outlines (that make sense given the nature of flights and weather)

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_dd = median(arr_delay), iqr_dd = IQR(arr_delay), n_flights = n())
```

```
## # A tibble: 5 x 4
##   carrier median_dd iqr_dd 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 IRQ is a measure of the middle 50% of the data. higher IRQ values would indicate a wider distribution of data. Using the strictest defintion of arrival delays as leafing after a targeted time, the carrier with the highest varability of arrival delay rate would be UA as it not only has the same IRQ value as DL, but a higher median departure delay.

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?

```
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
```

Practically you have two main choices, if you have a delay, do you want it to be most likely be short or are you alright with the chance of a catastrophic issue. The mean is best viewed as the average of the entire set of departure delays, which takes into account catastrophic issues. The median on the other hand more effectively balances extreme cases on both ends, making it feel more akin to what one should normally expect!

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?

```
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
```

If all I cared about was departure rates, I would select LGA as the airport I depart from as it has the highest. The other option would be JFK, as it is remarkably close in terms of on-time departure rates, although I believe it services more destinations. I would avoid EWR as it is significantly lower than the other two options.

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 = (distance/(air_time/60)))
```

```
head(nycflights)
```

```
## # A tibble: 6 x 18
##    year month   day dep_time dep_delay arr_time arr_delay carrier tailnum flight
##   <int> <int> <int>    <int>     <dbl>    <int>     <dbl> <chr>   <chr>    <int>
## 1  2013     6    30      940        15     1216        -4 VX      N626VA     407
## 2  2013     5     7     1657        -3     2104        10 DL      N3760C     329
## 3  2013    12     8      859        -1     1238        11 DL      N712TW     422
## 4  2013     5    14     1841        -4     2122       -34 DL      N914DL    2391
## 5  2013     7    21     1102        -3     1230        -8 9E      N823AY    3652
## 6  2013     1     1     1817        -3     2008         3 AA      N3AXAA     353
## # ... with 8 more variables: origin <chr>, dest <chr>, air_time <dbl>,
## #   distance <dbl>, hour <dbl>, minute <dbl>, dep_type <chr>, avg_speed <dbl>
```

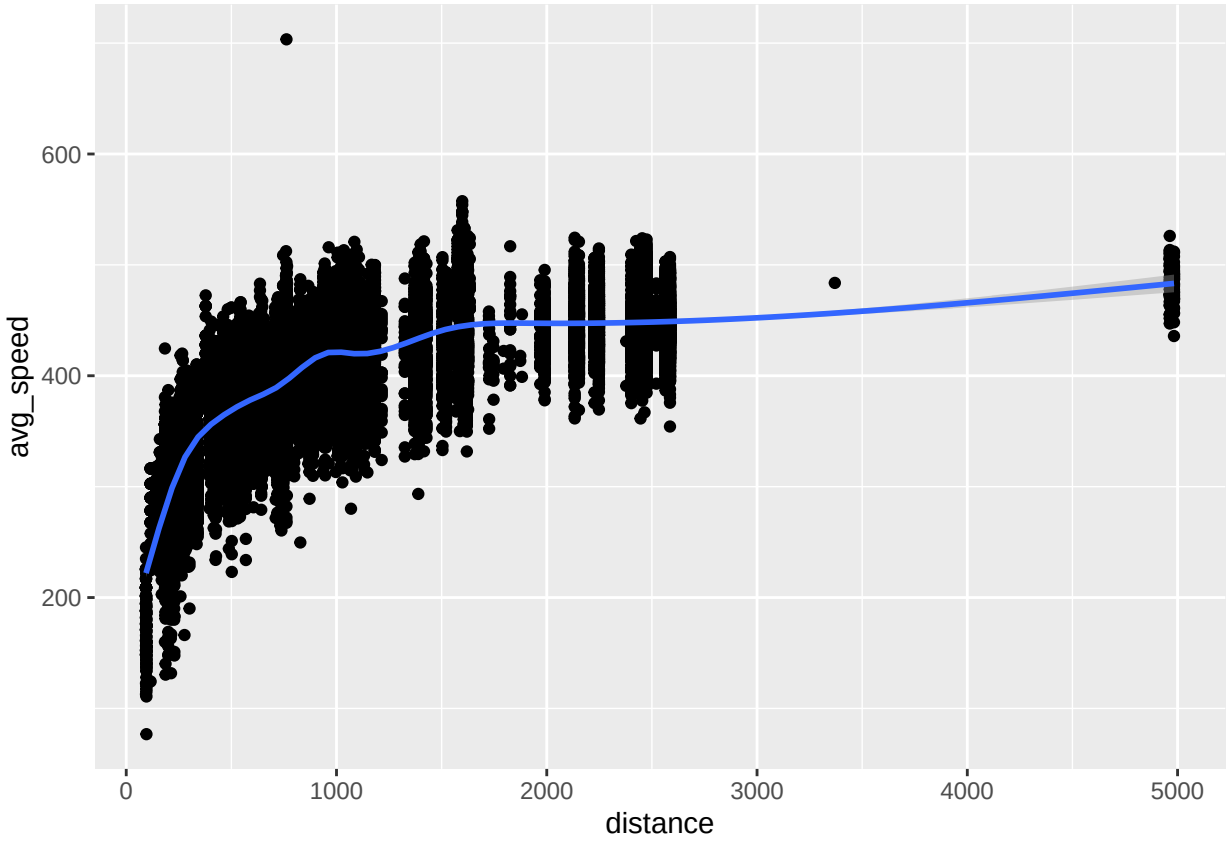
As you can see, you can take distance divide it by air time and divide it by 60, to get miles per hour. These number roughly align with a jet at cruising speeds.

Exercise 8

Make a scatterplot of avg_speed vs. distance. Describe the relationship between average speed and distance.

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

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

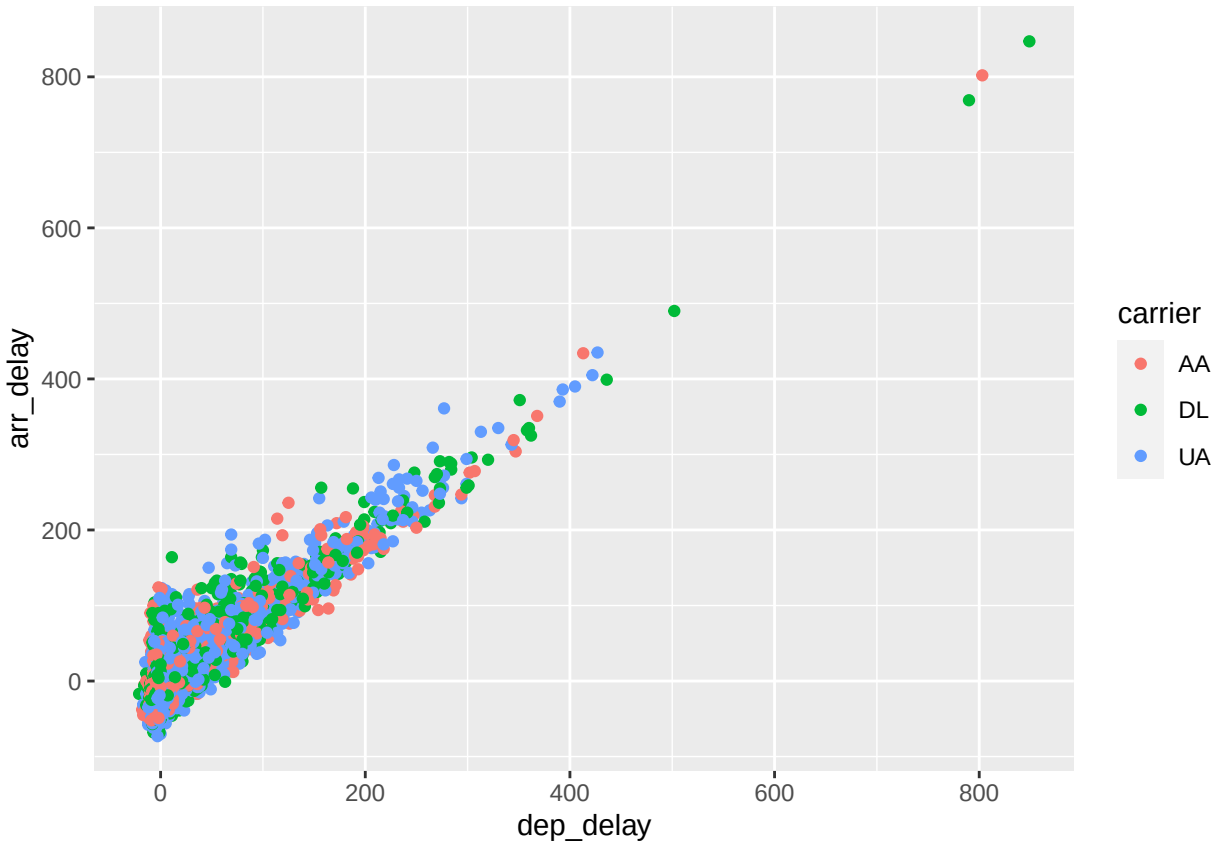


Practically as distance increases, avg_speed increases until it levels off. This actually makes sense as certain flights that are prop planes are used for shorter jumps (whith a lower top speed), vs longer distance flights, that use a jet with higher top speeds.

```
nycflights_short <- nycflights %>%
  filter( carrier == "DL" | carrier == "AA" | carrier == "UA")
head(nycflights_short)
```

```
## # A tibble: 6 x 18
##    year month   day dep_time dep_delay arr_time arr_delay carrier tailnum flight
##   <int> <int> <int>    <int>     <dbl>    <int>     <dbl> <chr>   <chr>    <int>
## 1  2013     5     7     1657        -3     2104        10 DL      N3760C     329
## 2  2013    12     8      859        -1     1238        11 DL      N712TW     422
## 3  2013     5    14     1841        -4     2122       -34 DL      N914DL    2391
## 4  2013     1     1     1817        -3     2008         3 AA      N3AXAA     353
## 5  2013     9    26      725       -10     1027        -8 AA      N3FSAA    2279
## 6  2013     8     5      757        -3     1041       -23 DL      N380DA    1271
## # ... with 8 more variables: origin <chr>, dest <chr>, air_time <dbl>,
## #   distance <dbl>, hour <dbl>, minute <dbl>, dep_type <chr>, avg_speed <dbl>
```

```
ggplot(data = nycflights_short, aes(y = arr_delay, x = dep_delay, color= carrier )) +
  geom_point()
```


800
600
400
200
0
arr_delay
0
200
400
600
800
dep_delay
carrier
AA
DL
UA