Lab_4

Part I: Understanding the Context of the Data

1. **What is the unit of observation in the data frame on the handout?**

The unit of observation in the data frame is one single flight

2. **Which variables are categorical?**

year, month, day, dep_time, sched_dep_time, arr_time, sched_arr_time, carrier, flight, tailnum, origin, dest, time_hour

3. **Which variables are numerical?**

Dep_delay, arr_delay, air_time, distance

4. **Do any of the variable have ambiguous data types to you?**

I would say that hour and minute are ambiguous this is because we don't know what those numbers stand for.

5. **Is there any discernible pattern to the manner in which the rows are ordered?**

The rows are arranged depending on the year, month, day and dep_time. Its a chronological order of which one left first.

6. **What is your guess for the units/format used to record the departure time? Said another way, what would a value of 1517 represent?**

The units/format used to record departure time is write in military time, based on a 24 hour clock.

A value of 15:17 represents a plane that left at 3:17pm

7. **What filter would you use to extract the flights that left in the springtime?**

After googling what dates are for springtime, I just made just to filter the data, so that I would only get data inside the time frame that I decided (spring time),

```
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   1.0.10
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(stat20data)
data(flights)

flights%>%
  filter(month ==3& day>=20 | month %in% c(4,5) | month==6 & day <=21)
```

```
# A tibble: 22,499 x 19
    year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
   <dbl> <dbl> <dbl>    <dbl>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
 1  2020     3    20        5          5       0     600     545      15 UA
 2  2020     3    20       25         47     -22     833     908     -35 AA
 3  2020     3    20       33         45     -12     900     912     -12 AA
 4  2020     3    20       34       2300      94     819     719      60 UA
 5  2020     3    20       40         40       0     636     644      -8 UA
 6  2020     3    20      507        510      -3    1022    1030      -8 WN
 7  2020     3    20      509        515      -6     845     847      -2 UA
 8  2020     3    20      517        520      -3     622     645     -23 WN
 9  2020     3    20      520        525      -5     755     805     -10 WN
10  2020     3    20      535        540      -5     650     710     -20 WN
# ... with 22,489 more rows, 9 more variables: flight <dbl>, tailnum <chr>,
#   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>, and abbreviated variable names
#   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
#   5: arr_delay
```

Part II: Computing on the Data

8. **filter(): Filter the data set to contain only the flights that went to Portland, Oregon and print the first few rows of the data frame. How many were there in 2020?**

There were 3,882 flights that went to Portland, Oregon, as seen in the number of rows in the Data frame.

```
flights%>%
  filter(dest=="PDX")
```

```
# A tibble: 3,882 x 19
    year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
   <dbl> <dbl> <dbl>    <dbl>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
 1  2020     1     1      613        600      13     801     754       7 AS
 2  2020     1     1      656        700      -4     842     854     -12 AS
 3  2020     1     1      657        700      -3     836     845      -9 WN
 4  2020     1     1      825        830      -5    1024    1024       0 UA
 5  2020     1     1      900        900       0    1116    1050      26 OO
 6  2020     1     1     1055       1055       0    1240    1240       0 WN
 7  2020     1     1     1110       1115      -5    1327    1319       8 UA
 8  2020     1     1     1129       1135      -6    1354    1329      25 AS
 9  2020     1     1     1246       1248      -2    1443    1443       0 UA
10  2020     1     1     1304       1305      -1    1518    1459      19 AS
# ... with 3,872 more rows, 9 more variables: flight <dbl>, tailnum <chr>,
#   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>, and abbreviated variable names
#   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
#   5: arr_delay
```

9. **mutate(): Mutate a new variable called avg_speed that is the average speed of the plane during the flight, measured in miles per hour. (Look through the column names or the help file to find variables that can be used to calculate this.)**

```
flights%>%
  mutate(avg_speed=(distance/(air_time/60)))
```

```
# A tibble: 120,605 x 20
    year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
```

```
   <dbl> <dbl> <dbl>    <dbl>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
 1  2020     1     1        8       2359       9     528     532      -4 UA
 2  2020     1     1       29         39     -10     356     420     -24 F9
 3  2020     1     1       37         40      -3     846     856     -10 UA
 4  2020     1     1       41         45      -4     908     913      -5 AA
 5  2020     1     1       44       2300     104     834     709      85 AA
 6  2020     1     1       48         56      -8     641     658     -17 UA
 7  2020     1     1       49         56      -7     614     634     -20 UA
 8  2020     1     1      506        515      -9    1050    1101     -11 UA
 9  2020     1     1      528        530      -2     812     820      -8 WN
10  2020     1     1      540        536       4    1303    1332     -29 AA
# ... with 120,595 more rows, 10 more variables: flight <dbl>, tailnum <chr>,
#   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>, avg_speed <dbl>, and abbreviated variable
#   names 1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
#   5: arr_delay
```

10. **arrange(): Arrange the data set to figure out: which flight holds the record for longest departure delay (in hrs) and what was its destination? What was the destination and delay time (in hrs) for the flight that was least delayed, i.e. that left the most ahead of schedule?**

The flight with the longest departure delay in hours was flight 576 leaving from SFO and heading to PHX. It got delayed 29 hours.

```
flights%>%
  mutate(longest_delay=(dep_delay/60)) %>%
  arrange(desc(longest_delay))
```

```
# A tibble: 120,605 x 20
    year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
   <dbl> <dbl> <dbl>    <dbl>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
 1  2020     3     6     1407        907    1740    1722    1213    1749 AA
 2  2020     2    20     1604       1245    1639    1900    1538    1642 AA
 3  2020     3     2     1247       1140    1507    2049    1958    1491 AA
 4  2020     2    12      955        907    1488    1246    1215    1471 AA
 5  2020     1    24     1005        952    1453    1303    1245    1458 AA
 6  2020     3     6      816       1111    1265    1611    1914    1257 AA
 7  2020     2    29     1046       1403    1243    1221    1529    1252 OO
 8  2020     2    12      828       1253    1175    1647    2124    1163 AA
 9  2020     2    14      655       1125    1170    1015    1435    1180 AA
10  2020     1    13     1238       1800    1118    1423    1933    1130 OO
```

```
# ... with 120,595 more rows, 10 more variables: flight <dbl>, tailnum <chr>,
#   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>, longest_delay <dbl>, and abbreviated
#   variable names 1: sched_dep_time, 2: dep_delay, 3: arr_time,
#   4: sched_arr_time, 5: arr_delay
```

The flights with the least delay time was flight 915, leaving from SFO and heading to GEG, with a delay of -0.666 / left earlier than expected by 0.666 hours.

```
flights%>%
  mutate(longest_delay=(dep_delay/60)) %>%
  arrange(longest_delay)
```

```
# A tibble: 120,605 x 20
    year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
   <dbl> <dbl> <dbl>    <dbl>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
 1  2020     3    31     1930       2010     -40    2119    2222     -63 OO
 2  2020     3    28     1540       1615     -35    2329      48     -79 UA
 3  2020    11    19     2341         16     -35     509     558     -49 UA
 4  2020     3    20     1303       1334     -31    1424    1446     -22 OO
 5  2020     5    24      804        834     -30    1115    1144     -29 G4
 6  2020     3    30     1851       1920     -29    2033    2100     -27 OO
 7  2020     9    10     1834       1903     -29    1959    2030     -31 G4
 8  2020     4     3     2057       2125     -28    2223    2301     -38 AS
 9  2020     6    26     2045       2113     -28    2243    2312     -29 G4
10  2020     3    17     1708       1735     -27    1840    1909     -29 AS
# ... with 120,595 more rows, 10 more variables: flight <dbl>, tailnum <chr>,
#   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>, longest_delay <dbl>, and abbreviated
#   variable names 1: sched_dep_time, 2: dep_delay, 3: arr_time,
#   4: sched_arr_time, 5: arr_delay
```

11. **summarize(): Confirm the records for departure delay from the question above by summarizing that variable by its maximum and its minimum value.**

To confirm the past question, I summarized the max and min, and got the same values.

Longest delay = 29 hours and shortest delay (early departure)=-0.667 (0.666 hours early)

```
flights%>%
  mutate(longest_delay=(dep_delay/60)) %>%
  summarize(max(longest_delay, na.rm=TRUE))
```

```
# A tibble: 1 x 1
  `max(longest_delay, na.rm = TRUE)`
                               <dbl>
1                                 29
```

```
flights%>%
  mutate(longest_delay=(dep_delay/60)) %>%
  summarize(min(longest_delay, na.rm=TRUE))
```

```
# A tibble: 1 x 1
  `min(longest_delay, na.rm = TRUE)`
                               <dbl>
1                             -0.667
```

12. **How many flights left SFO during March 2020?**

As seen in the amount of rows in the data frame, 14,165 flights left from SFO in March 2020

```
flights%>%
 filter(month==3 & origin=="SFO")
```

```
# A tibble: 14,165 x 19
    year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
   <dbl> <dbl> <dbl>    <dbl>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
 1  2020     3     1       12       2359      13     856     830      26 B6
 2  2020     3     1       24         29      -5     808     854     -46 B6
 3  2020     3     1       31         35      -4     539     556     -17 AA
 4  2020     3     1       32       2305      87     624     518      66 UA
 5  2020     3     1       37         40      -3     626     646     -20 UA
 6  2020     3     1       42         45      -3     825     910     -45 AA
 7  2020     3     1       45       2330      75     923     745      98 UA
 8  2020     3     1       58         59      -1     410     440     -30 F9
 9  2020     3     1      122       2355      87     639     535      64 UA
10  2020     3     1      506        513      -7     822     845     -23 UA
# ... with 14,155 more rows, 9 more variables: flight <dbl>, tailnum <chr>,
#   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>, and abbreviated variable names
#   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
#   5: arr_delay
```

13. **How many flights left SFO during April 2020?**

As seen in the amount of rows in the data frame, 4,517 flights left from SFO in April 2020

```
flights%>%
 filter(month==4 & origin=="SFO")
```

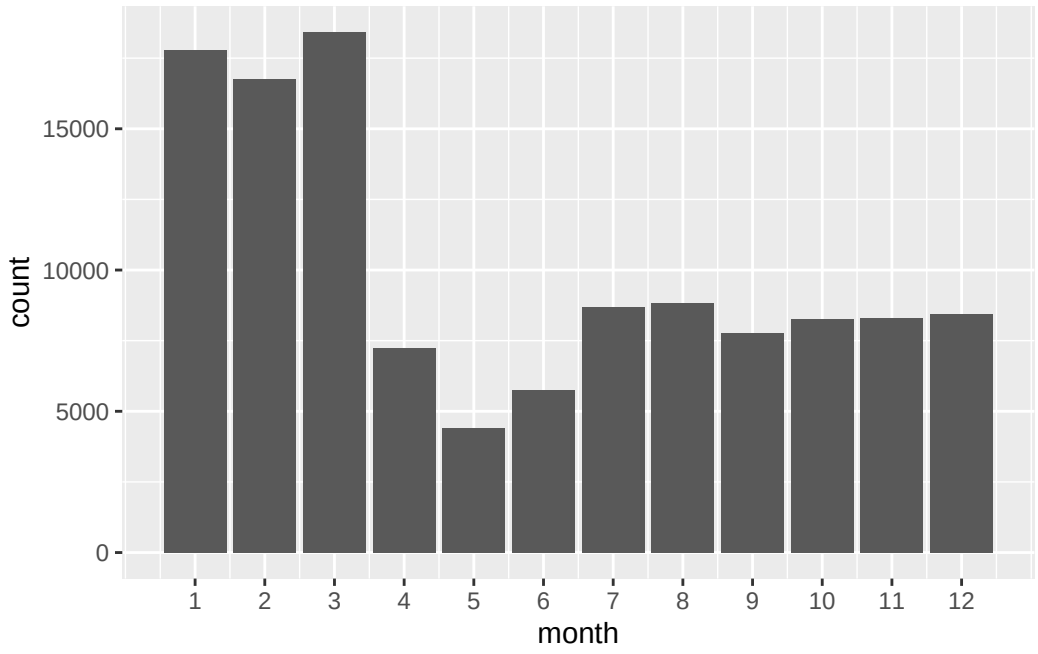
```
# A tibble: 4,517 x 19
    year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
   <dbl> <dbl> <dbl>    <dbl>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
 1  2020     4     1       12       1950     262     138    2124     254 DL
 2  2020     4     1       35         45     -10     837     912     -35 AA
 3  2020     4     1       36         47     -11     824     908     -44 AA
 4  2020     4     1      549        555      -6     901     925     -24 WN
 5  2020     4     1      549        600     -11     656     732     -36 AA
 6  2020     4     1      554        600      -6     752     802     -10 AA
 7  2020     4     1      555        600      -5    1103    1144     -41 DL
 8  2020     4     1      602        600       2    1318    1330     -12 DL
 9  2020     4     1      606        610      -4     733     725       8 WN
10  2020     4     1      615        600      15     850     854      -4 OO
# ... with 4,507 more rows, 9 more variables: flight <dbl>, tailnum <chr>,
#   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>, and abbreviated variable names
#   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
#   5: arr_delay
```

14. **Create a bar chart that shows the distribution by month of all the flights leaving the Bay Area (SFO and OAK). Do you any sign of an effect of the pandemic?**

Because this is 2020, the year when the pandemic started we can see how it affected flights. The pandemic started to take off (pun intended) in March 2020, but at this time people weren't really taking it seriously. The public started taking the pandemic seriously in April, month #4, and started travelling way less, as seen in the graph.

Its clear how as the pandemic grew, people took way less and less flight than they did in months #1-3. It really affected months 4-6 and people have slowly stated to travel more in July (could be because there were a lot of news regarding the vaccine at this point, and people became hopeful). The numbers in 2020 however never came back to how they were at the start of the year 2020.

```
ggplot(flights, aes(x=month))+
  geom_bar() + scale_x_continuous(breaks=seq(1,12,1/1))
```

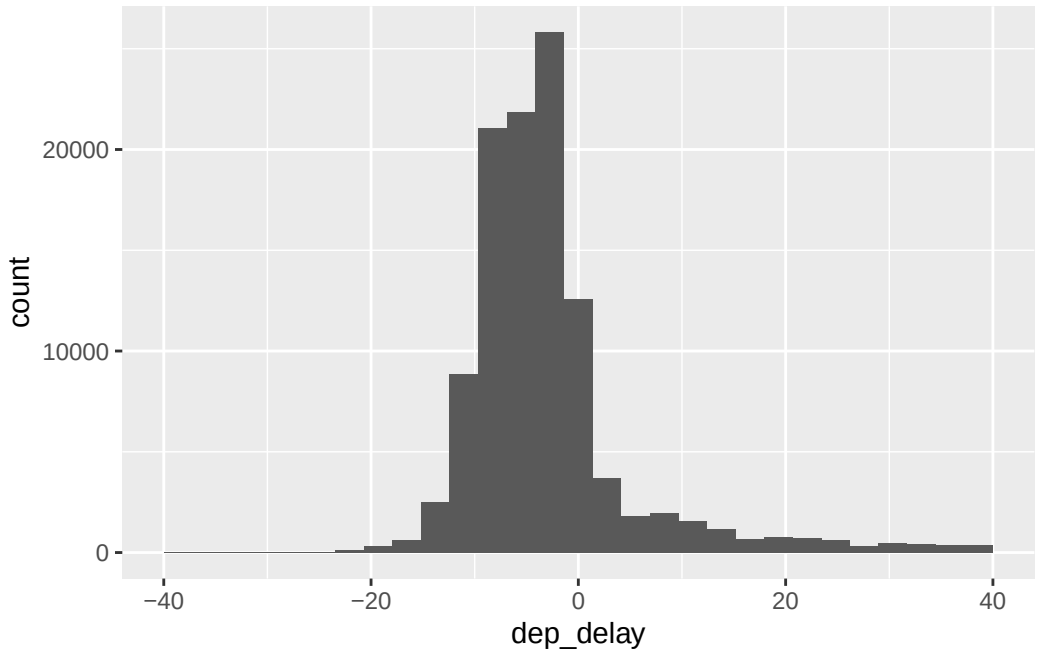


15. **Create a histogram showing the distribution of departure delays for all flights. Describe in words the shape and modality of the distribution and, using numerical summaries, (i.e. summary statistics) its center and spread. Be sure to use measures of center and spread that are most appropriate for this type of distribution. Also set the limits of the x-axis to focus on where most of the data lie.**

The data in the histogram is unimodal and appears to be right skewed.

```
ggplot(flights, aes(x=dep_delay))+
  geom_histogram(bins=30)+
  xlim(-40,40)
```

```
Warning: Removed 11989 rows containing non-finite values (stat_bin).
```



Because the data has a lot of outliers and big range, I decided to calculate the center through the Median (because the mean doesn't do good summaries when numbers have such a big range) and the spread though the Interquartile Range. The median delay of the flights was -4 & and the IQR was 6.

```
summarise(flights, median(dep_delay, na.rm=TRUE))
```

```
# A tibble: 1 x 1
  `median(dep_delay, na.rm = TRUE)`
                              <dbl>
1                                -4
```

```
summarise(flights, IQR(dep_delay, na.rm=TRUE))
```

```
# A tibble: 1 x 1
  `IQR(dep_delay, na.rm = TRUE)`
                           <dbl>
1                              6
```

16. **Add a new column to your data frame called `before_times` that takes values of `TRUE` and `FALSE` indicating whether the flight took place up through the end of March or after April 1st, respectively. Remake the histograms above, but now separated into two subplots: one with the departure delays from the before times, the other with the flights from afterwards.**

Can you visually detect any difference in the distribution of departure delays?

After remaking the graphs with different values, either TRUE or FALSE, I didn't really see a big difference in the distribution of departure delays. The graphs maintained the unimodal and right skewed distribution.

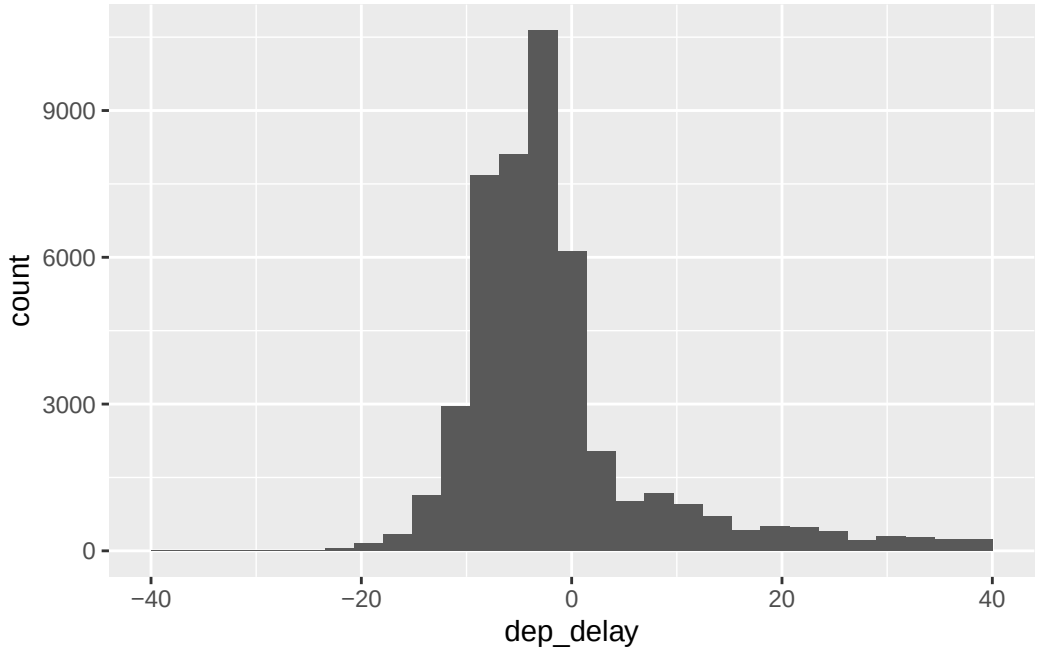
```
flights%>%
  mutate(before_times=(month<=3))
```

```
# A tibble: 120,605 x 20
    year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
   <dbl> <dbl> <dbl>    <dbl>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
 1  2020     1     1        8       2359       9     528     532      -4 UA
 2  2020     1     1       29         39     -10     356     420     -24 F9
 3  2020     1     1       37         40      -3     846     856     -10 UA
 4  2020     1     1       41         45      -4     908     913      -5 AA
 5  2020     1     1       44       2300     104     834     709      85 AA
 6  2020     1     1       48         56      -8     641     658     -17 UA
 7  2020     1     1       49         56      -7     614     634     -20 UA
 8  2020     1     1      506        515      -9    1050    1101     -11 UA
 9  2020     1     1      528        530      -2     812     820      -8 WN
10  2020     1     1      540        536       4    1303    1332     -29 AA
# ... with 120,595 more rows, 10 more variables: flight <dbl>, tailnum <chr>,
#   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>, before_times <lgl>, and abbreviated
#   variable names 1: sched_dep_time, 2: dep_delay, 3: arr_time,
#   4: sched_arr_time, 5: arr_delay
```

```
flights%>%
  mutate(before_times=(month<=3))%>%
  filter(before_times==TRUE)%>%
ggplot(aes(x=dep_delay)) + geom_histogram()+ xlim(-40,40)
```

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

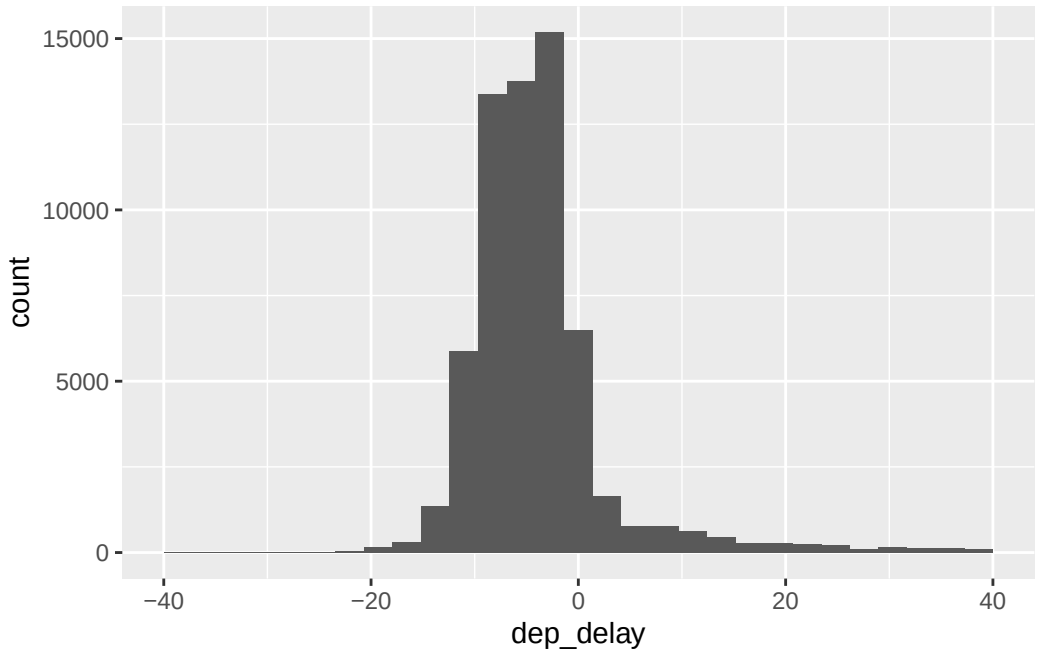
```
Warning: Removed 6843 rows containing non-finite values (stat_bin).
```




```
flights%>%
  mutate(before_times=(month<=3))%>%
  filter(before_times==FALSE)%>%
ggplot(aes(x=dep_delay)) + geom_histogram()+ xlim(-40,40)
```

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

```
Warning: Removed 5146 rows containing non-finite values (stat_bin).
```



17. **If you flew out of OAK or SFO during this time period, what is the tail number of the plane that you were on? If you did not fly in this period, find the tail number of the plane that flew JetBlue flight 40 to New York's JFK Airport from SFO on May 1st.**

I did not fly during this time period in SFO or OAK, so I calculated the flight to JFK.

The flight number 40 going to JFK had a tail number of N982JB.

```
flights%>%
filter(flight==40,dest=="JFK",month==5,day==1)
```

```
# A tibble: 1 x 19
   year month   day dep_time sched_dep~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
  <dbl> <dbl> <dbl>    <dbl>       <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
1  2020     5     1     1511        1520      -9    2304    2358     -54 B6
# ... with 9 more variables: flight <dbl>, tailnum <chr>, origin <chr>,
#   dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
#   time_hour <dttm>, and abbreviated variable names 1: sched_dep_time,
#   2: dep_delay, 3: arr_time, 4: sched_arr_time, 5: arr_delay
```

18. **What proportion of the flights left on or ahead of schedule?**

The proportion of flights that left ahead of schedule is calculated by knowing how many flights left on or ahead of schedule, which we can see by the number of rows is 91873. Then I divided that number by the total amount of flights that left, which is the amount of rows in the big flights data set, 120605.

Dividing these two gives = 91873/120605 = 0.7617678

```
flights%>%
  filter(dep_delay <=0)
```

```
# A tibble: 91,873 x 19
    year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
   <dbl> <dbl> <dbl>    <dbl>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
 1  2020     1     1       29         39     -10     356     420     -24 F9
 2  2020     1     1       37         40      -3     846     856     -10 UA
 3  2020     1     1       41         45      -4     908     913      -5 AA
 4  2020     1     1       48         56      -8     641     658     -17 UA
 5  2020     1     1       49         56      -7     614     634     -20 UA
 6  2020     1     1      506        515      -9    1050    1101     -11 UA
 7  2020     1     1      528        530      -2     812     820      -8 WN
 8  2020     1     1      550        600     -10     803     810      -7 AS
 9  2020     1     1      551        555      -4     909     935     -26 WN
10  2020     1     1      555        604      -9    1412    1429     -17 B6
# ... with 91,863 more rows, 9 more variables: flight <dbl>, tailnum <chr>,
#   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>, and abbreviated variable names
#   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
#   5: arr_delay
```

```
91873/120605
```

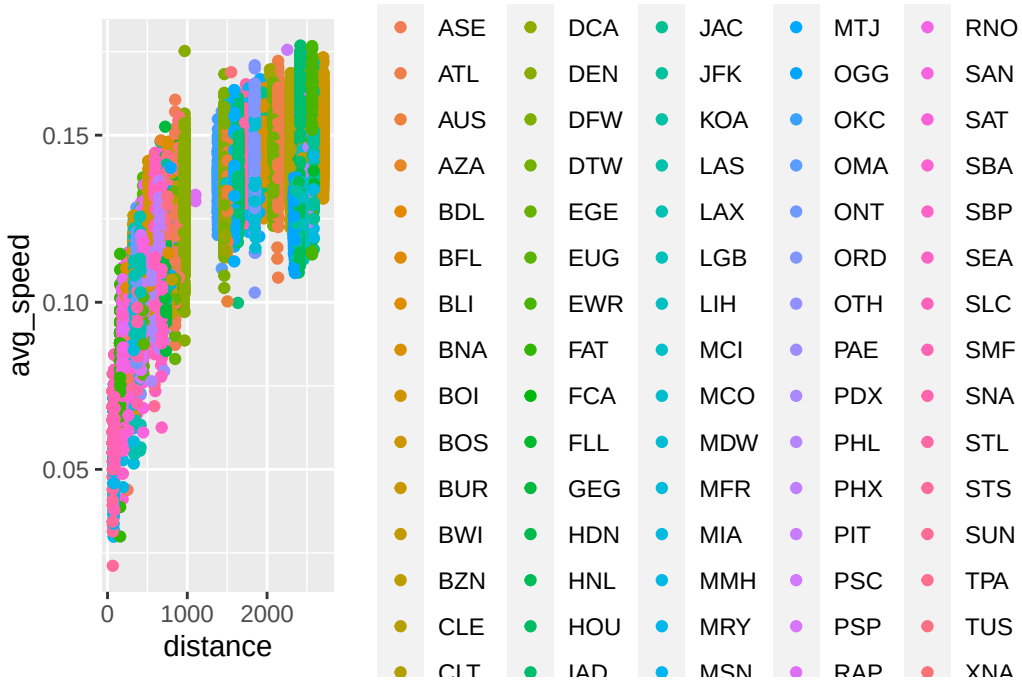
```
[1] 0.7617678
```

19. **Create a plot that captures the relationship of average speed vs. distance and describe the shape and structure that you see. What phenomena related to taking flights from the Bay Area might explain this structure?**

We see there is a gap when it comes to the 1000 - 1500 distance. This could be because there are no airports in those miles, the land is not built for planes to land or to take off from, hence there is no data at those distances and it "restarts" at about 2000 miles, when there are more airports. This is a scatter plot and its non linear and we see an average increases in avg speed as miles increase.

```
avg_speed_data <- mutate(flights,avg_speed = (distance/air_time/60))
  ggplot(avg_speed_data, aes(x=distance,
                             y= avg_speed,
                             color =dest))+
  geom_point()
```

```
Warning: Removed 7592 rows containing missing values (geom_point).
```



20. **For OAK and SFO separately, what proportion of the flights left on or ahead of schedule?**

For OAK, calculating the proportion of flights that left early or on schedule, I did not do it based off of the total flights in the data frame (120,605), I did based off the amount of flights that left OAK, which was 312,111, and based the proportion of that number. For OAK, 22,746 flights left early or on schedule. For OAK, 22,746/31,211 = 0.7287815 is the proportion of flights that left early or on schedule.

For SFO, calculating the proportion of flights that left early or on schedule, I did not do it based off of the total flights in the data frame (120,605), I did based off the amount of flights that left SFO, which was 89,394, and based the proportion of that number. For SFO, 69,127 flights left early or on schedule.

For SFO, 69,127/89,394 = 0.7732 is the proportion of flights that left early or on schedule.

```
flights%>%
filter(origin=="OAK")
```

```
# A tibble: 31,211 x 19
    year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
   <dbl> <dbl> <dbl>    <dbl>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
 1  2020     1     1      528        530      -2     812     820      -8 WN
 2  2020     1     1      550        600     -10     803     810      -7 AS
 3  2020     1     1      555        600      -5     839     854     -15 AA
 4  2020     1     1      557        600      -3     837     858     -21 DL
 5  2020     1     1      557        600      -3     712     725     -13 NK
 6  2020     1     1      604        605      -1     925     935     -10 WN
 7  2020     1     1      621        620       1     918     959     -41 AS
 8  2020     1     1      624        630      -6     755     805     -10 WN
 9  2020     1     1      633        650     -17     953    1035     -42 AS
10  2020     1     1      633        635      -2     907     920     -13 WN
# ... with 31,201 more rows, 9 more variables: flight <dbl>, tailnum <chr>,
#   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>, and abbreviated variable names
#   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
#   5: arr_delay
```

```
flights%>%
  filter(dep_delay <=0 & origin == "OAK")
```

```
# A tibble: 22,746 x 19
    year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
   <dbl> <dbl> <dbl>    <dbl>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
 1  2020     1     1      528        530      -2     812     820      -8 WN
 2  2020     1     1      550        600     -10     803     810      -7 AS
 3  2020     1     1      555        600      -5     839     854     -15 AA
 4  2020     1     1      557        600      -3     837     858     -21 DL
 5  2020     1     1      557        600      -3     712     725     -13 NK
 6  2020     1     1      604        605      -1     925     935     -10 WN
 7  2020     1     1      624        630      -6     755     805     -10 WN
 8  2020     1     1      633        650     -17     953    1035     -42 AS
 9  2020     1     1      633        635      -2     907     920     -13 WN
10  2020     1     1      637        640      -3    1243    1245      -2 WN
```

```
# ... with 22,736 more rows, 9 more variables: flight <dbl>, tailnum <chr>,
#   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>, and abbreviated variable names
#   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
#   5: arr_delay
```

```
22746/31211
```

```
[1] 0.7287815
```

```
flights%>%
filter(origin=="SFO")
```

```
# A tibble: 89,394 x 19
    year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
   <dbl> <dbl> <dbl>    <dbl>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
 1  2020     1     1        8       2359       9     528     532      -4 UA
 2  2020     1     1       29         39     -10     356     420     -24 F9
 3  2020     1     1       37         40      -3     846     856     -10 UA
 4  2020     1     1       41         45      -4     908     913      -5 AA
 5  2020     1     1       44       2300     104     834     709      85 AA
 6  2020     1     1       48         56      -8     641     658     -17 UA
 7  2020     1     1       49         56      -7     614     634     -20 UA
 8  2020     1     1      506        515      -9    1050    1101     -11 UA
 9  2020     1     1      540        536       4    1303    1332     -29 AA
10  2020     1     1      551        555      -4     909     935     -26 WN
# ... with 89,384 more rows, 9 more variables: flight <dbl>, tailnum <chr>,
#   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>, and abbreviated variable names
#   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
#   5: arr_delay
```

```
flights%>%
  filter(dep_delay <=0 & origin == "SFO")
```

```
# A tibble: 69,127 x 19
    year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
```

```
   <dbl> <dbl> <dbl>    <dbl>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
 1  2020     1     1       29         39     -10     356     420     -24 F9
 2  2020     1     1       37         40      -3     846     856     -10 UA
 3  2020     1     1       41         45      -4     908     913      -5 AA
 4  2020     1     1       48         56      -8     641     658     -17 UA
 5  2020     1     1       49         56      -7     614     634     -20 UA
 6  2020     1     1      506        515      -9    1050    1101     -11 UA
 7  2020     1     1      551        555      -4     909     935     -26 WN
 8  2020     1     1      555        604      -9    1412    1429     -17 B6
 9  2020     1     1      555        559      -4     846     900     -14 OO
10  2020     1     1      556        600      -4    1108    1136     -28 AA
# ... with 69,117 more rows, 9 more variables: flight <dbl>, tailnum <chr>,
#   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>, and abbreviated variable names
#   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
#   5: arr_delay
```

```
69127/89394
```

```
[1] 0.7732846
```

21. **Create a data frame that contains the median and interquartile range for departure delays, grouped by carrier. Which carrier has the lowest typical departure delay? Which one has the least variable departure delays?**

The carrier with the lowest typical departure delay (median) was B6, with -8.

The carrier with the least variable departure delay (IQR) was DL, with 5.

```
flights%>%
  group_by(carrier)%>%
  summarize(median_delay=(median(dep_delay, na.rm=TRUE)),
            iqr_delay=(IQR(dep_delay, na.rm=TRUE))) %>%
arrange(median_delay)
```

```
# A tibble: 12 x 3
   carrier median_delay iqr_delay
   <chr>          <dbl>     <dbl>
 1 B6                -8       7
 2 AS                -7       9
 3 G4                -7      13.8
```

```
 4 YV                -7      11
 5 AA                -6       6
 6 DL                -5       5
 7 F9                -5       9
 8 HA                -5       9
 9 OO                -5       7
10 UA                -5       6
11 NK                -4       6
12 WN                -3       5
```

```
flights%>%
  group_by(carrier)%>%
  summarize(median_delay=(median(dep_delay, na.rm=TRUE)),
            iqr_delay=(IQR(dep_delay, na.rm=TRUE))) %>%
arrange(iqr_delay)
```

```
# A tibble: 12 x 3
   carrier median_delay iqr_delay
   <chr>          <dbl>     <dbl>
 1 DL                -5       5
 2 WN                -3       5
 3 AA                -6       6
 4 NK                -4       6
 5 UA                -5       6
 6 B6                -8       7
 7 OO                -5       7
 8 AS                -7       9
 9 F9                -5       9
10 HA                -5       9
11 YV                -7      11
12 G4                -7      13.8
```

Part III: Extensions

22. **For flights leaving SFO, which month has the highest average departure delay? What about the highest median departure delay? Which of these measures is more useful to know when deciding which month(s) to avoid flying if you particularly dislike flights that are severely delayed?**

According to the data frame, month 1 (January) has the highest average (mean) departure delay, with 9.0488

According to the data frame, month 1 (January) has the highest median departure delay, with -3

If you really dislike severe delays, then you might want tot take the mean (average) as it takes into account these severe flight delays into the calculation. Not the median as it is not sensitive to big outliers, but the mean is.

```
flights%>%
 filter(origin=="SFO")%>%
group_by(month)%>%
summarize(avg_dep_delay=mean(dep_delay,na.rm=TRUE))%>%
  arrange(desc(avg_dep_delay))
```

```
# A tibble: 12 x 2
   month avg_dep_delay
   <dbl>         <dbl>
 1     1         9.05
 2     2         8.06
 3     3         1.15
 4    10        -0.155
 5    12        -0.970
 6     7        -1.56
 7     9        -1.75
 8     8        -1.80
 9     6        -1.86
10    11        -1.91
11     5        -2.68
12     4        -3.81
```

```
flights%>%
 filter(origin=="SFO")%>%
group_by(month)%>%
summarize(median_dep_delay=median(dep_delay,na.rm=TRUE))%>%
  arrange(desc(median_dep_delay))
```

```
# A tibble: 12 x 2
   month median_dep_delay
   <dbl>            <dbl>
 1     1               -3
 2     2               -3
```

```
 3     3               -5
 4     6               -5
 5     8               -5
 6     9               -5
 7    10               -5
 8    12               -5
 9     5               -6
10     7               -6
11    11               -6
12     4               -7
```

23. **Each individual airplane can be uniquely identified by its tailnumber in the same way that US citizens can be by their social security numbers. Which airplane flew the farthest during this year for which we have data? How many times around the planet does that translate to?**

The airplane with the most air distance traveled in the whole year had a tail number of N705TW, with 245670 miles. This is based off of the data that we have.

According to NASA, the earth is approximately 24,901 miles round, which means that the plane went around the planet approximately 9.865 times

```
flights%>%
  group_by(tailnum)%>%
  summarize(total_flights_distance=sum(distance))%>%
arrange(desc(total_flights_distance))
```

```
# A tibble: 3,774 x 2
   tailnum total_flights_distance
   <chr>                    <dbl>
 1 <NA>                   4570452
 2 N705TW                  245670
 3 N980JT                  243490
 4 N969JT                  242297
 5 N986JB                  242229
 6 N984JB                  238697
 7 N983JT                  238624
 8 N968JT                  234069
 9 N989JT                  231144
10 N977JE                  229138
# ... with 3,764 more rows
```

```
245670/24901
```

```
[1] 9.865869
```

24. **What is the tailnumber of the fastest plane in the data set? What type of plane is it (google it!)? Be sure to be clear how you're defining fastest.**

The tail number of the fastest plane in the data set is N77022, going 9.884393 miles per minute.

After googling the tail number, it appears like this is a Boeing 777-224. An "American long-range wide body airliner developed and manufactured by Boeing Commercial Airplanes. It is the world's largest twin jet".

I'm defining fastest by dividing the planes total distance flown(adding all their individual distances) by the total time (sum of all the individual air times) it took it to fly all those miles.

```
flights%>%
  group_by(tailnum)%>%
  summarize(total_flights_distance=sum(distance),
            total_flights_air_time=sum(air_time))%>%
  mutate(fastest_plane=(total_flights_distance/total_flights_air_time))%>%
  arrange(desc(fastest_plane))
```

```
# A tibble: 3,774 x 4
   tailnum total_flights_distance total_flights_air_time fastest_plane
   <chr>                    <dbl>                  <dbl>         <dbl>
 1 N77022                    5130                    519          9.88
 2 N226UA                    5805                    591          9.82
 3 N773AN                    2585                    264          9.79
 4 N78002                    5130                    524          9.79
 5 N839AA                    1464                    151          9.70
 6 N188DN                    2586                    267          9.69
 7 N792UA                   18568                   1919          9.68
 8 N78004                    7695                    796          9.67
 9 N2749U                   12825                   1327          9.66
10 N79011                    4984                    516          9.66
# ... with 3,764 more rows
```

25. **Using the airport nearest your hometown, which day of the week and which airline seems best for flying there from San Francisco (if you're from near SFO or OAK or from abroad, use Chicago as your hometown)? Be clear on how you're defining *best*. (note that there is no explicit weekday column in this data set, but there is sufficient information to piece it together. The following line of code can be added to your pipeline to create that new column. It uses functions in the `lubridate`package, so be sure to load it in at the start of this exercise)**

I'm using Chicago data since I am not from the US.

I'm defining best as the day that has the least departure delays. I am also considering the mean and not the median because I want to calculate the best and worst case scenarios (early departure, late departure).

Calculating with the mean, the best day to travel from SFO to ORD in Chicago would be on Monday, flying with AS, as they appear to have an average delay departure of -7.10 / 7.10 early departure.

```
    library(lubridate)
```

```
Attaching package: 'lubridate'
```

```
The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

```
flights%>%
mutate(day_of_week = wday(ymd(paste(year, month, day, set = "-")), label = T))%>%
filter(origin=="SFO", dest=="ORD")%>%
group_by(day_of_week,carrier)%>%
  summarize(mean_dep_delay=(mean(dep_delay, na.rm=TRUE)))%>%
arrange(mean_dep_delay)
```

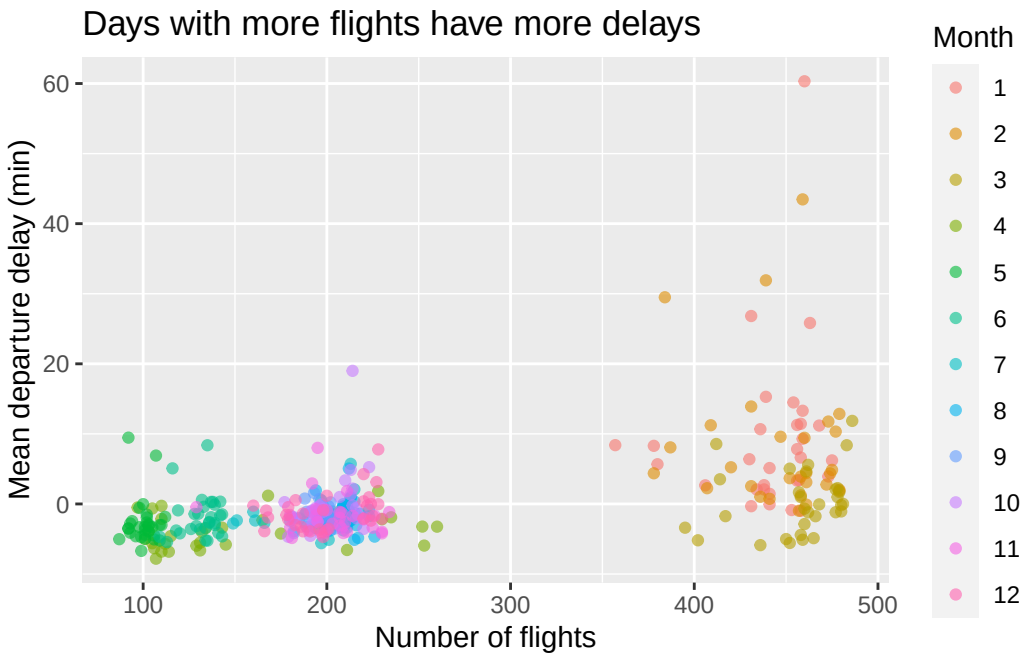
```
`summarise()` has grouped output by 'day_of_week'. You can override using the
`.groups` argument.
```

```
# A tibble: 21 x 3
# Groups:   day_of_week [7]
   day_of_week carrier mean_dep_delay
   <ord>       <chr>            <dbl>
 1 Mon         AS               -7.11
 2 Wed         AS               -6.02
 3 Sat         AS               -5.65
 4 Tue         AS               -5.47
 5 Wed         AA               -3.18
 6 Tue         AA               -2.54
 7 Fri         AS               -1.31
 8 Sun         AS               -0.826
 9 Mon         AA               -0.435
10 Fri         AA                0.122
# ... with 11 more rows
```

26. **The plot below shows a relationship between the number of flights going out of SFO and the average departure delay. It illustrates the hypothesis that more flights on a given day would lead to a more congested airport which would lead to greater delays on average (mean is used here specifically to capture the impact of the outliers - very long delays). Each point represents single day in 2020; there are 366 of them on the plot. Please form a single chain that will create this plot, starting with the raw data set.**

```
flights%>%
  group_by(month,day)%>%
  filter(origin=="SFO")%>%
  summarize(number_flights=n(),mean_dep=(mean(dep_delay,na.rm=TRUE)))%>%
  ggplot(aes(x=number_flights,
         y=mean_dep,
         color=as.factor(month)))+
    geom_point(alpha=0.6) +
  labs(x="Number of flights",
       y= "Mean departure delay (min)",
       color="Month",
       title="Days with more flights have more delays")
```

```
`summarise()` has grouped output by 'month'. You can override using the
`.groups` argument.
```


Days with more flights have more delays
Mean departure delay (min)
60
40
20
0
100
200
300
400
500
Number of flights
Month
1
2
3
4
5
6
7
8
9
10
11
12