Stat 650 Final Project

Asees Kaur,net id: rh7442

```
library(pacman)
p_load(tidyverse, nycflights13,knitr)
library(textreadr)
library(nycflights13)
```

##Reading the names and the descriptions of the variables from the FAA from website using the read_html() function.

```
names <- read_html("readme.html")

head(names)
```

```
## [1] "BACKGROUND"
## [2] "The data contained in the compressed file has been extracted from the \n\tReporting Carrier On-
## [3] "RECORD LAYOUT"
## [4] "Below are fields in the order that they appear on the records:"
## [5] "Year"
## [6] "Year"
```

```
# Remove the top four lines.

names <- names[-c(1,2,3,4)]

head(names)
```

```
## [1] "Year"          "Year"          "Quarter"       "Quarter (1-4)"
## [5] "Month"         "Month"
```

```
#Creating a matrix that contains variable names in column 1 and description in column 2.

faa_names <- matrix(names[1:218], ncol = 2, byrow = TRUE)

head(faa_names)
```

```
##      [,1]          [,2]
## [1,] "Year"        "Year"
## [2,] "Quarter"     "Quarter (1-4)"
## [3,] "Month"       "Month"
## [4,] "DayofMonth"  "Day of Month"
## [5,] "DayOfWeek"   "Day of Week"
## [6,] "FlightDate"  "Flight Date (yyyymmdd)"
```

```
sfo_names <- tibble(variable_names = faa_names[,1], description = faa_names[,2])
kable(head(sfo_names))
```

variable_names	description
Year	Year
Quarter	Quarter (1-4)
Month	Month
DayofMonth	Day of Month
DayOfWeek	Day of Week

variable_names	description
FlightDate	Flight Date (yyyymmdd)

```
sfo_names <- sfo_names %>% mutate(variable_names = tolower(variable_names))

kable(head(sfo_names))
```

variable_names	description
year	Year
quarter	Quarter (1-4)
month	Month
dayofmonth	Day of Month
dayofweek	Day of Week
flightdate	Flight Date (yyyymmdd)

#The given code to get the names and descriptions of the variables in the nycflights13 dataset.

```
nycflights13_names <- tribble(
  ~variable_names, ~Description,
  'year', 'Year of departure. - year',
  'month', 'Month of departure. - month',
  'day', 'Day of departure.',
  'dep_time', 'Actual departure time (format HHMM or HMM), local tz.',
  'arr_time', 'Actual  arrival time (format HHMM or HMM), local tz.',
  'sched_dep_time', 'Scheduled departure time (format HHMM or HMM), local tz.',
  'sched_arr_time', 'Scheduled  arrival time (format HHMM or HMM), local tz.',
  'dep_delay', 'Departure delays, in minutes.Negative times represent early departures.',
  'arr_delay', 'arrival delays, in minutes.Negative times represent early arrivals.',
  'carrier', 'Two letter carrier abbreviation. See airlines to get name.',
  'flight', 'Flight number.',
  'tailnum', 'Plane tail number. See planes for additional metadata.',
  'origin', 'Origin and destination. See airports for additional metadata.',
  'dest', 'Origin and destination. See airports for additional metadata.',
  'air_time', 'Amount of time spent in the air, in minutes.',
  'distance', 'Distance between airports, in miles.',
  'hour', 'Time of scheduled departure broken into hour and minutes. - hour',
  'minute', 'Time of scheduled departure broken into hour and minutes. minute',
  'time_hour', 'Scheduled date and hour of the flight as a POSIXct date. Along with origin, can be used
)

kable(nycflights13_names)
```

variable_names	Description
year	Year of departure. - year
month	Month of departure. - month
day	Day of departure.
dep_time	Actual departure time (format HHMM or HMM), local tz.
arr_time	Actual arrival time (format HHMM or HMM), local tz.
sched_dep_time	Scheduled departure time (format HHMM or HMM), local tz.
sched_arr_time	Scheduled arrival time (format HHMM or HMM), local tz.
dep_delay	Departure delays, in minutes.Negative times represent early departures.

variable_names	Description
arr_delay	arrival delays, in minutes.Negative times represent early arrivals.
carrier	Two letter carrier abbreviation. See airlines to get name.
flight	Flight number.
tailnum	Plane tail number. See planes for additional metadata.
origin	Origin and destination. See airports for additional metadata.
dest	Origin and destination. See airports for additional metadata.
air_time	Amount of time spent in the air, in minutes.
distance	Distance between airports, in miles.
hour	Time of scheduled departure broken into hour and minutes. - hour
minute	Time of scheduled departure broken into hour and minutes. minute
time_hour	Scheduled date and hour of the flight as a POSIXct date. Along with origin, can be used to join flights

#recoding the names of the nycflights13 data set to match with the FAA variable names.

```
nycflights13_names <- nycflights13_names %>% mutate(variable_names = as_factor(variable_names))


nycflights13_names$variable_names %>% levels()
```

```
##  [1] "year"           "month"          "day"            "dep_time"
##  [5] "arr_time"       "sched_dep_time" "sched_arr_time" "dep_delay"
##  [9] "arr_delay"      "carrier"        "flight"         "tailnum"
## [13] "origin"         "dest"           "air_time"       "distance"
## [17] "hour"           "minute"         "time_hour"
```

```
nycflights13_names$variable_names %>% nlevels()
```

```
## [1] 19
```

```
nycflights13_names$variable_names <- nycflights13_names$variable_names %>%
  fct_recode("dayofmonth" = "day",
             "deptime"    = "dep_time",
             "arrtime"    = "arr_time",
             "crsdeptime" = "sched_dep_time",
             "crsarrtime" = "sched_arr_time",
             "depdelay"   = "dep_delay",
             "arrdelay"   = "arr_delay",
             "reporting_airline" = "carrier",
             "flight_number_reporting_airline" = "flight",
             "tail_number"= "tailnum",
             "airtime"    = "air_time"
             )

kable(nycflights13_names)
```

variable_names	Description
year	Year of departure. - year
month	Month of departure. - month
dayofmonth	Day of departure.
deptime	Actual departure time (format HHMM or HMM), local tz.
arrtime	Actual arrival time (format HHMM or HMM), local tz.
crsdeptime	Scheduled departure time (format HHMM or HMM), local tz.
crsarrtime	Scheduled arrival time (format HHMM or HMM), local tz.

variable_names	Description
depdelay	Departure delays, in minutes.Negative times represent early departures.
arrdelay	arrival delays, in minutes.Negative times represent early arrivals.
reporting_airline	Two letter carrier abbreviation. See airlines to get name.
flight_number_reporting_airline	Flight number.
tail_number	Plane tail number. See planes for additional metadata.
origin	Origin and destination. See airports for additional metadata.
dest	Origin and destination. See airports for additional metadata.
airtime	Amount of time spent in the air, in minutes.
distance	Distance between airports, in miles.
hour	Time of scheduled departure broken into hour and minutes. - hour
minute	Time of scheduled departure broken into hour and minutes. minute
time_hour	Scheduled date and hour of the flight as a POSIXct date. Along with origin, can be use

##Reading the data of flights from FAA in and filtering for the required Bay Area Airports(San Francisco, Oakland and San Jose)

##I only read the data for first 3 months of 2018.

```
faa_jan <- read.csv("data/On_Time_Reporting_Carrier_On_Time_Performance_(1987_present)_2018_1.csv")
sfoflights_jan <- faa_jan %>% filter(Origin == "SFO"| Origin == "SJC" | Origin == "OAK")

faa_feb <- read.csv("data/On_Time_Reporting_Carrier_On_Time_Performance_(1987_present)_2018_2.csv")
sfoflights_feb <- faa_feb %>% filter(Origin == "SFO"| Origin == "SJC" | Origin == "OAK")

faa_mar <- read.csv("data/On_Time_Reporting_Carrier_On_Time_Performance_(1987_present)_2018_3.csv")
sfoflights_mar <- faa_mar %>% filter(Origin == "SFO"| Origin == "SJC" | Origin == "OAK")
```

#Combining the filtered data into one data set using the rbind() function.

```
sfoflights18 <- rbind(sfoflights_jan,sfoflights_feb,sfoflights_mar)
kable(head(sfoflights18))
```

Year	Quarter	Month	DayofMonth	DayOfWeek	FlightDate	Reporting_Airline	DOT_ID_Reporting_Airline
2018	1	1	27	6	2018-01-27	UA	19977
2018	1	1	27	6	2018-01-27	UA	19977
2018	1	1	27	6	2018-01-27	UA	19977
2018	1	1	27	6	2018-01-27	UA	19977
2018	1	1	27	6	2018-01-27	UA	19977
2018	1	1	27	6	2018-01-27	UA	19977

```
dim(sfoflights18)
```

```
## [1] 65803   110
```

Question 1:

Find the US Government website where Airline On-Time Performance Data can be downloaded. What website is this and how can you download the data? Download the data for the available months in 2018 for the Bay Area. Can you do this? If not, what can you download?

Answer 1:

The US government website to get to the data is Airline On-Time Performance. We have to select the variables that we would like to have in our data set and then download the data file, but you can only do this month by month. I was not able to download the data so I used it from the data provided through Blackboard, and due to the limited space in my machine I only used the data for first 3 months of 2018.

Question 2:

Once you have your data downloaded, develop your code for the first month of data. The last step will be to merge the data and perform an overall analysis for 2018. Extract the flights that departed from the Bay Area. Include all flights departing from San Francisco, Oakland, and San Jose. How many flight were there in January 2018?

Answer 2:

There were 22274 flights in January 2018 that departed from the Bay Area Airports.

```
sfoflights18 %>%
  filter(Month == 1) %>%
  filter(Cancelled == 0) %>% count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1 22274
```

Question 3:

Compare the variables that are available in the full dataset with the variables in the nycflgihts13 data set. Make a table of the variables that are in both datasets, with a description of each variable. Hint: In RStudio see Help > RMarkdown Quick Reference > Tables. Report the intersection of the variables.

Answer 3:

Using the inner_join() function to find out the common variable names in both the datasets.

```
common_names <- inner_join(nycflights13_names, sfo_names,"variable_names") %>%
  select(variable_names, Description)
```

```
## Warning: Column `variable_names` joining factor and character vector,
## coercing into character vector
```

```
kable(common_names)
```

variable_names	Description
year	Year of departure. - year
month	Month of departure. - month
dayofmonth	Day of departure.
deptime	Actual departure time (format HHMM or HMM), local tz.
arrtime	Actual arrival time (format HHMM or HMM), local tz.
crsdeptime	Scheduled departure time (format HHMM or HMM), local tz.
crsarrtime	Scheduled arrival time (format HHMM or HMM), local tz.
depdelay	Departure delays, in minutes.Negative times represent early departures.
arrdelay	arrival delays, in minutes.Negative times represent early arrivals.
reporting_airline	Two letter carrier abbreviation. See airlines to get name.
flight_number_reporting_airline	Flight number.
tail_number	Plane tail number. See planes for additional metadata.
origin	Origin and destination. See airports for additional metadata.
dest	Origin and destination. See airports for additional metadata.
airtime	Amount of time spent in the air, in minutes.
distance	Distance between airports, in miles.

Question 4:

What new variables do you now also have? Make a table of the variables that are in the new dataset (that could also be downloaded from the website), with a description of each variable. (Ok, this is kind of long. Make a table for 10 other variables you consider important.)

Answer 4:

```
not_common <- anti_join(sfo_names, nycflights13_names,  "variable_names")
```

```
## Warning: Column `variable_names` joining character vector and factor,
## coercing into character vector
```

```
kable(not_common %>%
  filter(variable_names == "quarter" |
           variable_names == "dayofweek"|
           variable_names=="originstatename"|
           variable_names=="deststatename"|
           variable_names == "cancelled"|
           variable_names == "diverted"|
           variable_names == "flights"|
           variable_names == "weatherdelay"|
           variable_names=="lateaircraftdelay" |
           variable_names == "divairportlandings"
         )
  )
```

variable_names	description
quarter	Quarter (1-4)
dayofweek	Day of Week
originstatename	Origin Airport, State Name
deststatename	Destination Airport, State Name
cancelled	Cancelled Flight Indicator (1=Yes)
diverted	Diverted Flight Indicator (1=Yes)
flights	Number of Flights
weatherdelay	Weather Delay, in Minutes
lateaircraftdelay	Late Aircraft Delay, in Minutes
divairportlandings	Number of Diverted Airport Landings

Question 5:

Answer Exercises 4.2, 4.3, 4.4 (you may only be able to answer part of 4.4) on page 89 of the book, changing nycflights13 to sfoflights18. Answer all of the questions for the SF Bay Area in 2018.

Answer 5:

##Exercise 4.2:

What month had the highest proportion of cancelled flights? What month had the lowest? Interpret any seasonal patterns.

##Answer 4.2:

Out of the first 3 months March(Month == 3) has the highest proportion of the cancelled flights and Feburary(Month == 2) had the least proportion of the cancelled flights.

```
kable(sfoflights18 %>%
  select(Month,Cancelled) %>%
  group_by(Month) %>%
  summarise(mean_cancelled = mean(Cancelled == 1)) )
```

Month	mean_cancelled
1	0.0146864
2	0.0071492
3	0.0225114

##Exercise 4.3

What plane (specified by the tail_number variable) traveled the most times from Bay Area airports in 2018? Plot the number of trips per week over the year.

##Answer 4.3:

Plane N633VA travelled the most times from Bay Area airports(SFO,OAK,SJC) in 2018.

```
kable(sfoflights18 %>%
  filter(Cancelled == 0)  %>%
  select(Tail_Number) %>%
  group_by(Tail_Number) %>%
  count() %>%
  arrange(desc(n)) %>%
  head(1))
```

Tail_Number	n
N633VA	115

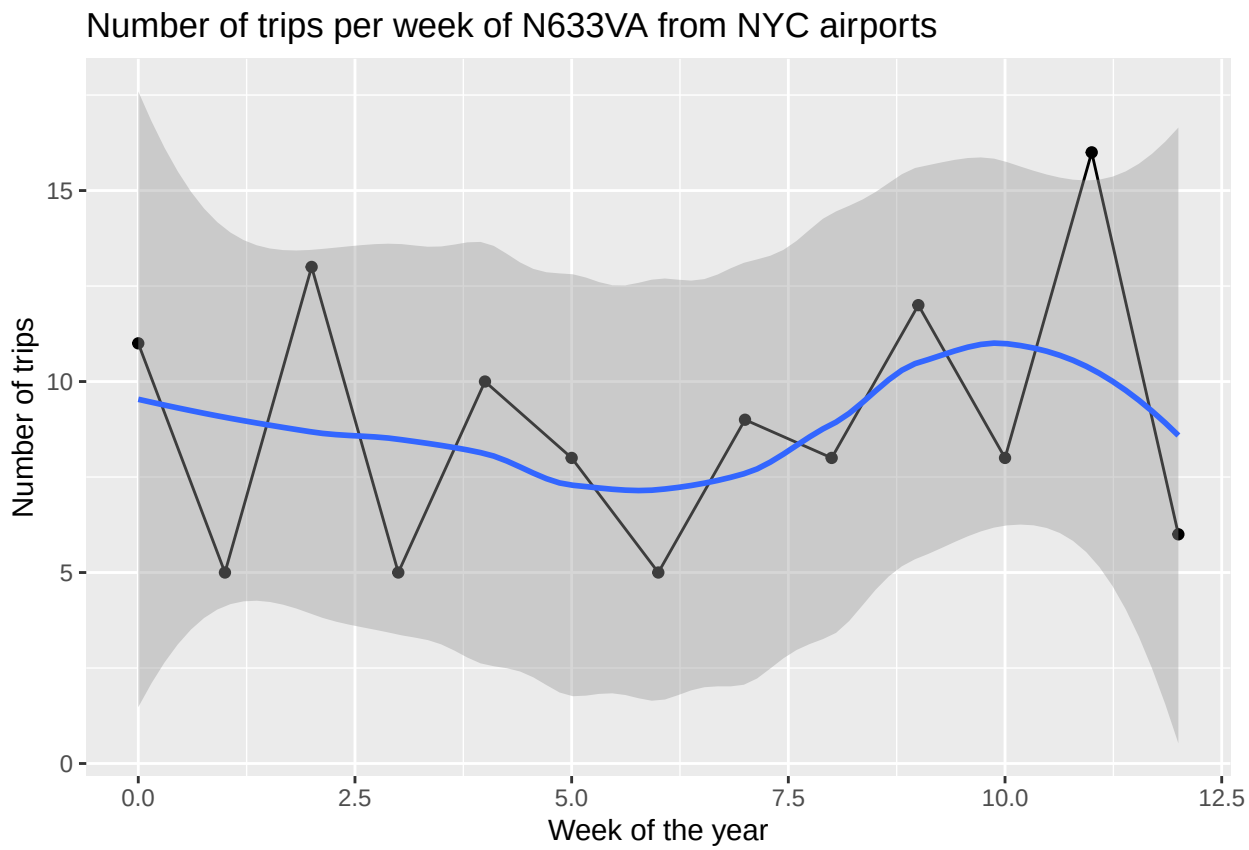
```
kable(sfoflights2 <- sfoflights18 %>%
 mutate( week = as.integer(format(as.Date(FlightDate), "%U"))  ) %>%
  filter(Tail_Number == "N633VA") %>%
 group_by(week) %>% tally())
```

week	n
0	11
1	5
2	13

week	n
3	5
4	10
5	8
6	5
7	9
8	8
9	12
10	8
11	16
12	6

```
ggplot(sfoflights2,aes(x=week, y=n)) +
geom_point() +
geom_line() +
geom_smooth() +
  labs(x="Week of the year", y = "Number of trips",
       title = "Number of trips per week of N633VA from NYC airports")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



##Exercise 4.4

What is the oldest plane (specified by the Tail_Number variable) that flew from Bay Area airports in 2018? How many airplanes that flew from Bay Area are included in the planes table?

##Answer 4.4:

The oldest plane that flew from Bay Area Airports in 2018 was N505UA which was manufactured in 1989.

```
plane_year <- planes %>%
  rename( year_manuf = year )



kable(plane_year %>%
      right_join(sfoflights18, by = c("tailnum"="Tail_Number") )%>%
  filter(!is.na(year_manuf)) %>%
      arrange(year_manuf) %>%
      select(tailnum, year_manuf) %>%
   distinct() %>%
  filter(tailnum == tailnum[1]))
```

```
## Warning: Column `tailnum`/`Tail_Number` joining character vector and
## factor, coercing into character vector
```

tailnum	year_manuf
N505UA	1989

The number of airplanes that flew from the Bay Area Airports and are included in the planes table is 1762.

```
kable(sfoflights18 %>%
  inner_join(plane_year, by = c("Tail_Number" = "tailnum")) %>%
  filter(Cancelled == 0) %>%
  distinct(Tail_Number) %>%
  summarise(n = n()))
```

```
## Warning: Column `Tail_Number`/`tailnum` joining factor and character
## vector, coercing into character vector
```

n
1762