Homework #4 - Data Wrangling

Raymond Ogunjimi

Change your name above and save the file. Also, install the following packages (that you don't have already). This is the last time I'll remind you of these...

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(nycflights13)
library(mdsr)

summary(flights)
```

```
##       year          month             day           dep_time    sched_dep_time
##  Min.   :2013   Min.   : 1.000   Min.   : 1.00   Min.   :   1   Min.   : 106
##  1st Qu.:2013   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.: 907   1st Qu.: 906
##  Median :2013   Median : 7.000   Median :16.00   Median :1401   Median :1359
##  Mean   :2013   Mean   : 6.549   Mean   :15.71   Mean   :1349   Mean   :1344
##  3rd Qu.:2013   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:1744   3rd Qu.:1729
##  Max.   :2013   Max.   :12.000   Max.   :31.00   Max.   :2400   Max.   :2359
##                                                  NA's   :8255
##    dep_delay          arr_time    sched_arr_time   arr_delay
##  Min.   : -43.00   Min.   :   1   Min.   :   1   Min.   : -86.000
##  1st Qu.:  -5.00   1st Qu.:1104   1st Qu.:1124   1st Qu.: -17.000
##  Median :  -2.00   Median :1535   Median :1556   Median :  -5.000
##  Mean   :  12.64   Mean   :1502   Mean   :1536   Mean   :   6.895
##  3rd Qu.:  11.00   3rd Qu.:1940   3rd Qu.:1945   3rd Qu.:  14.000
##  Max.   :1301.00   Max.   :2400   Max.   :2359   Max.   :1272.000
##  NA's   :8255      NA's   :8713                  NA's   :9430
##    carrier              flight       tailnum             origin
##  Length:336776      Min.   :   1   Length:336776      Length:336776
##  Class :character   1st Qu.: 553   Class :character   Class :character
##  Mode  :character   Median :1496   Mode  :character   Mode  :character
##                     Mean   :1972
##                     3rd Qu.:3465
##                     Max.   :8500
##
##      dest              air_time        distance         hour
##  Length:336776      Min.   : 20.0   Min.   :  17   Min.   : 1.00
##  Class :character   1st Qu.: 82.0   1st Qu.: 502   1st Qu.: 9.00
##  Mode  :character   Median :129.0   Median : 872   Median :13.00
##                     Mean   :150.7   Mean   :1040   Mean   :13.18
##                     3rd Qu.:192.0   3rd Qu.:1389   3rd Qu.:17.00
##                     Max.   :695.0   Max.   :4983   Max.   :23.00
##                     NA's   :9430
##      minute        time_hour
##  Min.   : 0.00   Min.   :2013-01-01 05:00:00
##  1st Qu.: 8.00   1st Qu.:2013-04-04 13:00:00
##  Median :29.00   Median :2013-07-03 10:00:00
##  Mean   :26.23   Mean   :2013-07-03 05:22:54
##  3rd Qu.:44.00   3rd Qu.:2013-10-01 07:00:00
##  Max.   :59.00   Max.   :2013-12-31 23:00:00
##
```

```
head(flights)
```

```
## # A tibble: 6 x 19
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
## 1  2013     1     1      517            515         2      830            819
## 2  2013     1     1      533            529         4      850            830
## 3  2013     1     1      542            540         2      923            850
## 4  2013     1     1      544            545        -1     1004           1022
## 5  2013     1     1      554            600        -6      812            837
## 6  2013     1     1      554            558        -4      740            728
## # ... with 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

Note, there is a *lubridate* package that has some useful date functions, like `month()` and `week()`. They are particularly useful with the `label = TRUE` option. Feel free to play around with it, but this package is not required to complete this assignment.

Problem 1) The `nycflights` package contains data on all flights from the New York City area in 2013. Use the `flights` data frame to answer the following...

a) What month had the highest proportion of canceled flights? (as recorded by a missing departure or arrival time)

February

```
flights %>%
    group_by(month) %>%
        summarise(flight_count = length(flight),
                  cancel_count = sum(is.na(dep_time) | is.na(arr_time))) %>%
        mutate(cancel_percentage = (cancel_count/flight_count)*100) %>%
        arrange(desc(cancel_percentage)) %>%
        head(1)
```

```
## # A tibble: 1 x 4
##   month flight_count cancel_count cancel_percentage
##   <int>        <int>        <int>             <dbl>
## 1     2        24951         1291              5.17
```

b) What month had the lowest proportion of canceled flights?

October

```
flights %>%
    group_by(month) %>%
        summarise(flight_count = length(flight),
                  cancel_count = sum(is.na(dep_time) | is.na(arr_time))) %>%
        mutate(cancel_percentage = (cancel_count/flight_count)*100) %>%
        arrange(desc(cancel_percentage)) %>%
        tail(1)
```

```
## # A tibble: 1 x 4
##   month flight_count cancel_count cancel_percentage
##   <int>        <int>        <int>             <dbl>
## 1    10        28889          247             0.855
```

c) Interpret seasonal patterns of canceled flights.

February and December (winter) and June and July (summer) are peaks in terms of the number of cancellations, perhaps due to extreme weather.

```
flights %>%
    group_by(month) %>%
        summarise(flight_count = length(flight),
                  cancel_count = sum(is.na(dep_time) | is.na(arr_time))) %>%
        mutate(cancel_percentage = (cancel_count/flight_count)*100) %>%
        arrange(month) %>%
        head(12)
```

```
## # A tibble: 12 x 4
##    month flight_count cancel_count cancel_percentage
##    <int>        <int>        <int>             <dbl>
##  1     1        27004          536             1.98
##  2     2        24951         1291             5.17
##  3     3        28834          891             3.09
##  4     4        28330          710             2.51
##  5     5        28796          601             2.09
##  6     6        28243         1072             3.80
##  7     7        29425         1043             3.54
##  8     8        29327          506             1.73
##  9     9        27574          504             1.83
## 10    10        28889          247             0.855
## 11    11        27268          253             0.928
## 12    12        28135         1059             3.76
```

Problem 2) Continuing with the nycflights data...

a) What plane (specified by `tailnum`) traveled the most times from NYC airports in 2013?

N725MQ

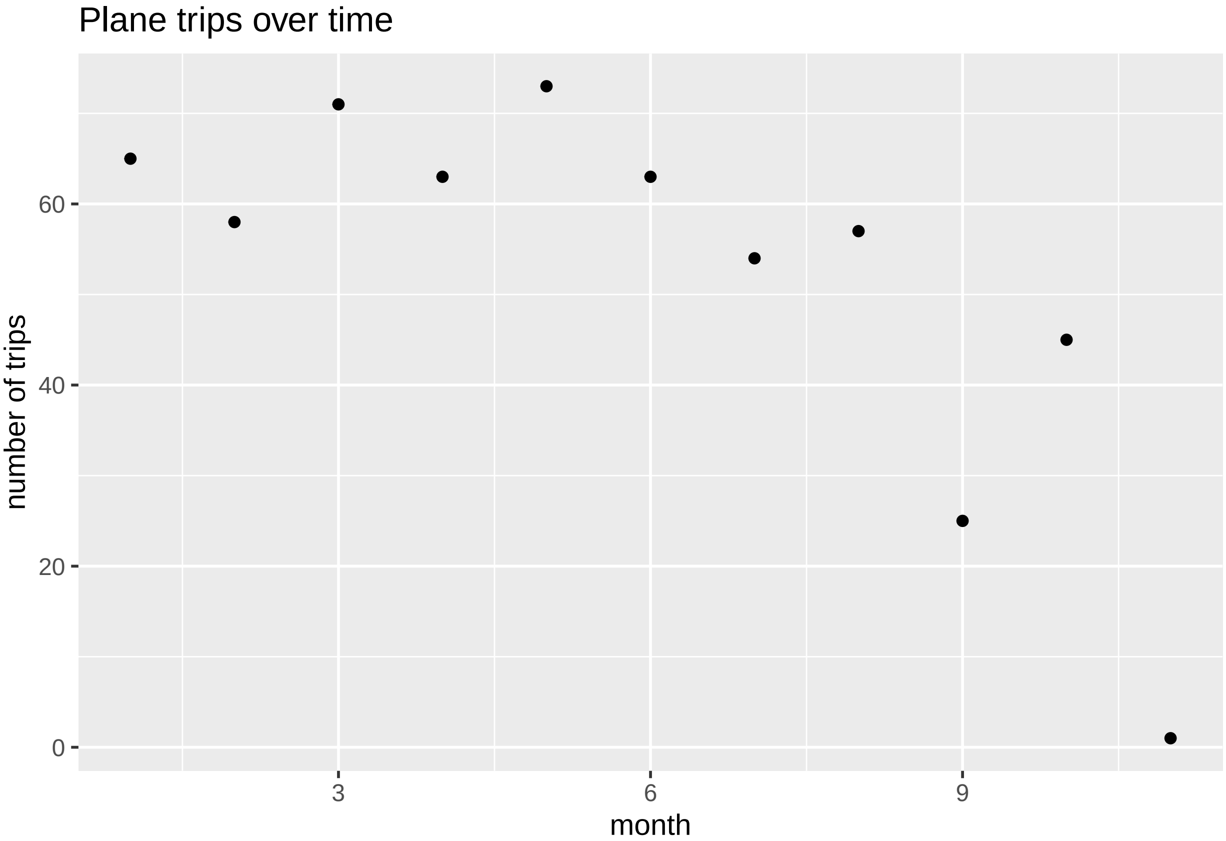
```
flights %>%
    filter(year == "2013") %>%
    filter(origin == "JFK" | origin == "LGA") %>%
    mutate(frequency = 1) %>%
    group_by(tailnum) %>%
    summarise(frequency = sum(frequency)) %>%
    arrange(desc(frequency)) %>%
    filter(!is.na(tailnum)) %>%
    head(1)
```

```
## # A tibble: 1 x 2
##   tailnum frequency
##   <chr>       <dbl>
## 1 N725MQ        575
```

b) Plot the number of trips per week over the year for the plane with the most times traveled. Make sure to label the axes appropriately and add a title to the graph. Comment on what you observe.

There are a lower number of trips later on in the year.

```
flights %>%
    filter(tailnum == "N725MQ") %>%
    filter(year == "2013") %>%
    mutate(numof_trips = 1) %>%
    group_by(month) %>%
    summarise(numof_trips = sum(numof_trips)) %>%
    ggplot() +
        aes(x = month, y = numof_trips) +
        geom_point() +
        xlab("month") +
        ylab("number of trips") +
        ggtitle("Plane trips over time")
```

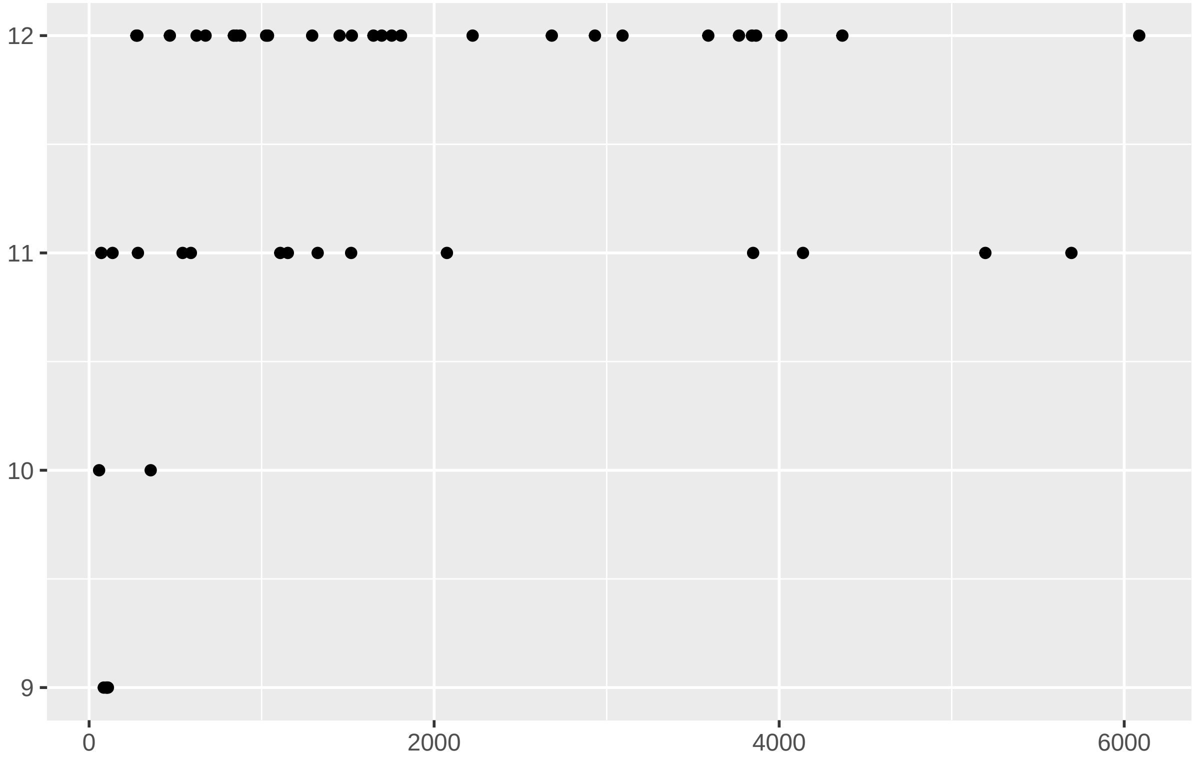


Problem 3) The `Violations` data set in the `mdsr` package contains information regarding the outcome of health inspections of restaurants in NYC. Use these data to calculate the median violation score by zip code for zip codes in Manhattan with 50 or more inspections. What pattern do you see between the number of inspections and the median score?

The restaurants with the highest number of inspections generally achieve higher scores, however there are restaurants with a low number of inspections that still achieve high scores.

```
violations_toplot = Violations %>%
    filter(boro == "MANHATTAN") %>%
    group_by(zipcode) %>%
    na.omit() %>%
    summarise(numof_insp = n(), med_scr = median(score)) %>%
    filter(numof_insp >= 50) %>%
    select(zipcode, numof_insp, med_scr) %>%
    arrange(numof_insp)

ggplot(data = violations_toplot) +
    aes(numof_insp, med_scr) +
    geom_point() +
    ggtitle( "") +
    xlab("") +
    ylab("")
```


12
11
10
9
0
2000
4000
6000