Problem Set 4

Yifan Li

Link to the GitHub

The link to my GitHub repository is https://github.com/FYlee39/Stats-506/tree/main/PS4.

Problem 1

```
library(nycflights13)
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

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

    intersect, setdiff, setequal, union
```

a.

```r
# read the flight data
flight_data <- flights %>%
  select(dep_delay, origin, dest, arr_delay)

#
# For departure data
depature_data <- flight_data %>%
  tibble %>%
  left_join(airports, by=c("origin" = "faa")) %>% # get the airports name
  select(dep_delay, name) %>%
  rename(delay=dep_delay,
         airport_name=name)

# data grouping and aggregate
depature_data_mean_median <- depature_data %>%
  filter(!is.na(delay)) %>% # remove the rows with NA
  group_by(airport_name) %>%
  summarise(delay_mean=mean(delay, na.rm=TRUE),
         delay_median=median(delay, na.rm=TRUE),
         n = n()) %>% # get the number of flights
  filter(n >= 10) %>% # filter to exclude any destination with under 10 flights
  arrange(desc(delay_mean)) %>% # arrange in descending mean delay.
  ungroup()

# Print all rows
print(depature_data_mean_median, n=Inf)
```

```
# A tibble: 3 x 4
  airport_name        delay_mean delay_median      n
  <chr>                    <dbl>        <dbl>  <int>
1 Newark Liberty Intl       15.1           -1 117596
2 John F Kennedy Intl       12.1           -1 109416
3 La Guardia                10.3           -3 101509
```

```r
#
# For arrival data
arrival_data <- flight_data %>%
  tibble %>%
  left_join(airports, by=c("dest" = "faa")) %>% # get the airports name
  select(arr_delay, name) %>%
  rename(delay=arr_delay,
```

```
          airport_name=name)

# data grouping and aggregate
arrival_data_mean_median <- arrival_data %>%
  filter(!is.na(delay)) %>% # remove the rows with NA
  group_by(airport_name) %>%
  summarise(delay_mean=mean(delay, na.rm=TRUE),
          delay_median=median(delay, na.rm=TRUE),
          n = n()) %>% # get the number of flights
  filter(n >= 10) %>% # filter to exclude any destination with under 10 flights
  arrange(desc(delay_mean)) %>% # arrange in descending mean delay.
  ungroup()

# Print all rows
print(arrival_data_mean_median, n=Inf)
```

```
# A tibble: 99 x 4
   airport_name                          delay_mean delay_median     n
   <chr>                                      <dbl>        <dbl> <int>
 1 "Columbia Metropolitan"                     41.8           28     106
 2 "Tulsa Intl"                                33.7           14     294
 3 "Will Rogers World"                         30.6           16     315
 4 "Jackson Hole Airport"                      28.1           15      21
 5 "Mc Ghee Tyson"                             24.1            2     578
 6 "Dane Co Rgnl Truax Fld"                    20.2            1     556
 7 "Richmond Intl"                             20.1            1    2346
 8 "Akron Canton Regional Airport"             19.7            3     842
 9 "Des Moines Intl"                           19.0            0     523
10 "Gerald R Ford Intl"                        18.2            1     728
11 "Birmingham Intl"                           16.9           -2     269
12 "Theodore Francis Green State"              16.2            1     358
13 "Greenville-Spartanburg International"      15.9           -0.5   790
14 "Cincinnati Northern Kentucky Intl"         15.4           -3    3725
15 "Savannah Hilton Head Intl"                 15.1           -1     749
16 "Manchester Regional Airport"               14.8           -3     932
17 "Eppley Afld"                               14.7           -2     817
18 "Yeager"                                    14.7           -1.5   134
19 "Kansas City Intl"                          14.5            0    1885
20 "Albany Intl"                               14.4           -4     418
21 "General Mitchell Intl"                     14.2            0    2709
22 "Piedmont Triad"                            14.1           -2    1492
23 "Washington Dulles Intl"                    13.9           -3    5383
```

24	"Cherry Capital Airport"	13.0	-10	95
25	"James M Cox Dayton Intl"	12.7	-3	1399
26	"Louisville International Airport"	12.7	-2	1104
27	"Chicago Midway Intl"	12.4	-1	4025
28	"Sacramento Intl"	12.1	4	282
29	"Jacksonville Intl"	11.8	-2	2623
30	"Nashville Intl"	11.8	-2	6084
31	"Portland Intl Jetport"	11.7	-4	2288
32	"Greater Rochester Intl"	11.6	-5	2358
33	"Hartsfield Jackson Atlanta Intl"	11.3	-1	16837
34	"Lambert St Louis Intl"	11.1	-3	4142
35	"Norfolk Intl"	10.9	-4	1434
36	"Baltimore Washington Intl"	10.7	-5	1687
37	"Memphis Intl"	10.6	-2.5	1686
38	"Port Columbus Intl"	10.6	-3	3326
39	"Charleston Afb Intl"	10.6	-4	2759
40	"Philadelphia Intl"	10.1	-3	1541
41	"Raleigh Durham Intl"	10.1	-3	7770
42	"Indianapolis Intl"	9.94	-3	1981
43	"Charlottesville-Albemarle"	9.5	-5	46
44	"Cleveland Hopkins Intl"	9.18	-5	4394
45	"Ronald Reagan Washington Natl"	9.07	-2	9111
46	"Burlington Intl"	8.95	-4	2510
47	"Buffalo Niagara Intl"	8.95	-5	4570
48	"Syracuse Hancock Intl"	8.90	-5	1707
49	"Denver Intl"	8.61	-2	7169
50	"Palm Beach Intl"	8.56	-3	6487
51	"Bob Hope"	8.18	-3	370
52	"Fort Lauderdale Hollywood Intl"	8.08	-3	11897
53	"Bangor Intl"	8.03	-9	358
54	"Asheville Regional Airport"	8.00	-1	261
55	"Pittsburgh Intl"	7.68	-5	2746
56	"Gallatin Field"	7.6	-2	35
57	"NW Arkansas Regional"	7.47	-2	992
58	"Tampa Intl"	7.41	-4	7390
59	"Charlotte Douglas Intl"	7.36	-3	13674
60	"Minneapolis St Paul Intl"	7.27	-5	6929
61	"William P Hobby"	7.18	-4	2083
62	"Bradley Intl"	7.05	-10	412
63	"San Antonio Intl"	6.95	-9	659
64	"South Bend Rgnl"	6.5	-3.5	10
65	"Louis Armstrong New Orleans Intl"	6.49	-6	3715
66	"Key West Intl"	6.35	7	17

```
67 "Eagle Co Rgnl"                          6.30          -4     207
68 "Austin Bergstrom Intl"                  6.02          -5    2411
69 "Chicago Ohare Intl"                     5.88          -8   16566
70 "Orlando Intl"                           5.45          -5   13967
71 "Detroit Metro Wayne Co"                 5.43          -7    9031
72 "Portland Intl"                          5.14          -5    1342
73 "Nantucket Mem"                          4.85          -3     264
74 "Wilmington Intl"                        4.64          -7     107
75 "Myrtle Beach Intl"                      4.60         -13      58
76 "Albuquerque International Sunport"      4.38        -5.5     254
77 "George Bush Intercontinental"           4.24          -5    7085
78 "Norman Y Mineta San Jose Intl"          3.45          -7     328
79 "Southwest Florida Intl"                 3.24          -5    3502
80 "San Diego Intl"                         3.14          -5    2709
81 "Sarasota Bradenton Intl"                3.08          -5    1201
82 "Metropolitan Oakland Intl"              3.08          -9     309
83  <NA>                                    3.01          -5    7537
84 "General Edward Lawrence Logan Intl"     2.91          -9   15022
85 "San Francisco Intl"                     2.67          -8   13173
86 "Yampa Valley"                           2.14           2      14
87 "Phoenix Sky Harbor Intl"                2.10          -6    4606
88 "Montrose Regional Airport"              1.79        -10.5      14
89 "Los Angeles Intl"                       0.547         -7   16026
90 "Dallas Fort Worth Intl"                 0.322         -9    8388
91 "Miami Intl"                             0.299         -9   11593
92 "Mc Carran Intl"                         0.258         -8    5952
93 "Salt Lake City Intl"                    0.176         -8    2451
94 "Long Beach"                            -0.0620       -10     661
95 "Martha\\\\'s Vineyard"                 -0.286        -11     210
96 "Seattle Tacoma Intl"                   -1.10         -11    3885
97 "Honolulu Intl"                         -1.37          -7     701
98 "John Wayne Arpt Orange Co"             -7.87         -11     812
99 "Palm Springs Intl"                    -12.7          -13.5    18
```

b

```
fastest_aircraft <- flights %>%
  # Exclude rows with NA air_time or distance
  filter(!is.na(air_time) & !is.na(distance)) %>%
  mutate(mph=distance / (air_time / 60)) %>% # Calculate speed in MPH
  group_by(tailnum) %>%
```

```
  summarise(
    average_speed=mean(mph, na.rm=TRUE),
    number_flights=n()
  ) %>%
  arrange(desc(average_speed)) %>%
  slice(1) %>% # Select the row with the fastest average speed
  left_join(planes, by="tailnum") %>% # get the aircraft model
  select(model, average_speed, number_flights)

print(fastest_aircraft)
```

```
# A tibble: 1 x 3
  model   average_speed number_flights
  <chr>           <dbl>          <int>
1 777-222          501.              1
```

Problem 2

Load the data.

```
nnmaps <- read.csv("chicago-nmmaps.csv")
```

```
#' Function to compute the average temperature for a given month
#'
#' @param month Month, either a numeric 1-12 or a string.
#' @param year a numeric year
#' @param data The data set to obtain data from.
#' @param celsius Logically indicating whether the results should be in celsius.
#' Default FALSE.
#' @param average_fn A function with which to compute the mean. Default is mean.
#'
#' @return average_temperature average temperature for a given month
get_temp <- function(month, year, data, celsius=FALSE, average_fn=mean){
  # Sanitize the input
  # month
  month_short <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep",
                   "Oct", "Nov", "Dec")
  month_full <- month.name
  all_months <- c(month_full, month_short)
```

```r
if (is.character(month)){
  month <- month %>%
    tolower() %>%
    match(tolower(all_months))

  if (is.na(month)){
    # not a single match
    message("Not a valiad month input")
    return()
  }

  if (month > 12){
    month <- month - 12
  }
}

if (is.na(month) || !(month %in% 1:12)) {
  message("Not a valiad month input")
  return()
}

given_month <- month

# year
if (!is.numeric(year) || year < min(nnmaps$year) || year > max(nnmaps$year)){
  message("Not a valiad year input")
  return()
}

given_year <- year

# Filter the data for the given month and year
filtered_data <- data %>%
  filter(month_numeric == given_month, year == given_year)

if (nrow(filtered_data) == 0) {
  message("No data available for the given month and year.")
  return()
}

# Calculate the average temperature using the provided average function
average_temperature <- filtered_data %>%
```

```
    summarise(average_temperature = average_fn(temp)) %>%
    pull(average_temperature)

  # Convert temperature to Celsius if requested
  if (celsius) {
    average_temperature <- (average_temperature - 32) * 5 / 9
  }

  return(average_temperature)
}
```

Test:

```
get_temp("Apr", 1999, data = nnmaps)
```

```
[1] 49.8
```

```
get_temp("Apr", 1999, data = nnmaps, celsius = TRUE)
```

```
[1] 9.888889
```

```
get_temp(10, 1998, data = nnmaps, average_fn = median)
```

```
[1] 55
```

```
get_temp(13, 1998, data = nnmaps)
```

```
Not a valiad month input
```

```
NULL
```

```
get_temp(2, 2005, data = nnmaps)
```

```
Not a valiad year input
```

```
NULL
```

```
get_temp("November", 1999, data =nnmaps, celsius = TRUE,
         average_fn = function(x) {
           x %>% sort -> x
           x[2:(length(x) - 1)] %>% mean %>% return
         })
```

```
[1] 7.301587
```

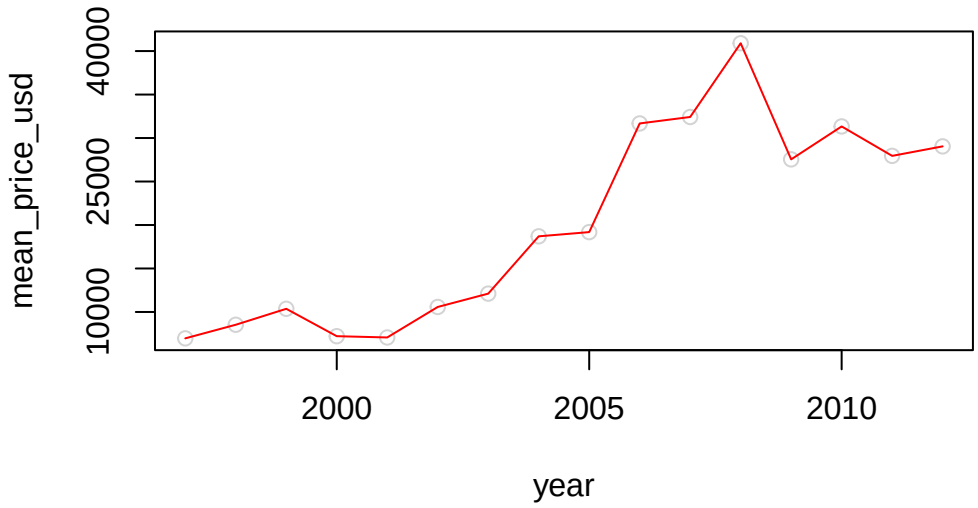
Problem 3

```
library(ggplot2)
art_df <- read.csv("df_for_ml_improved_new_market.csv")
```

a.

```
a_art_df <- art_df %>%
  select(year,
         price_usd) %>%
  group_by(year) %>%
  summarise(mean_price_usd=mean(price_usd)) %>%
  ungroup()

with(a_art_df, plot(mean_price_usd ~ year, col="lightgrey"))
with(a_art_df, lines(mean_price_usd ~ year, col='red'))
```

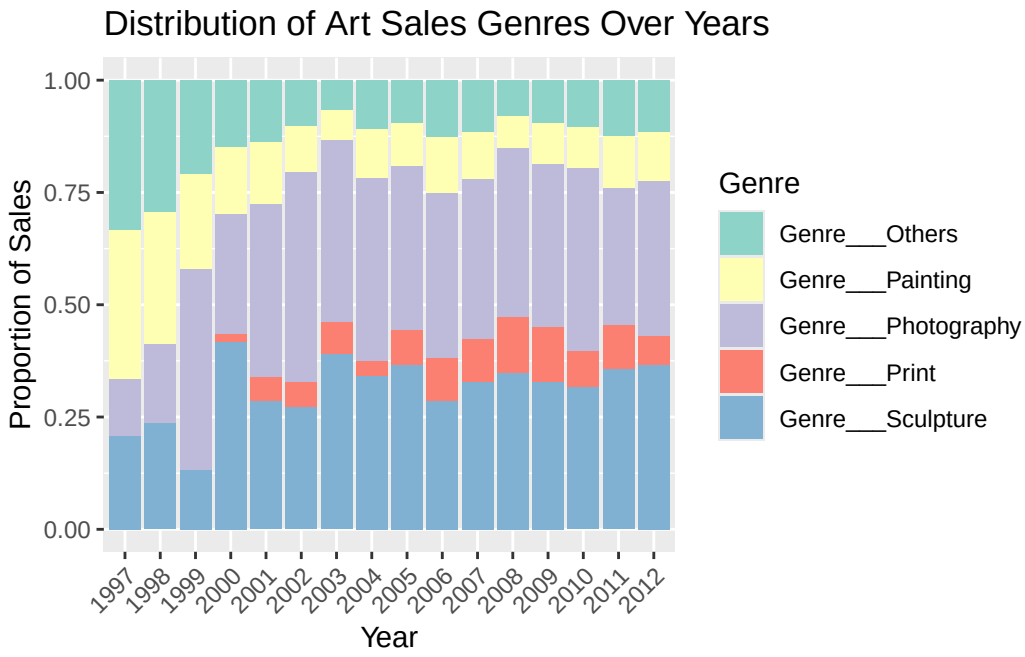


As shown in the plot, there is some changes in the sales price in USD over time.

b.

```
b_art_df <- art_df %>%
  tidyr::gather(key="genre", value="is_genre", starts_with("Genre___")) %>%
  filter(is_genre == 1) %>%
  select(year, genre)

# Create a stacked bar plot showing the distribution of genres over years
ggplot(b_art_df, aes(x=factor(year), fill=genre)) +
  geom_bar(position="fill") +
  scale_fill_brewer(palette="Set3") +
  labs(title="Distribution of Art Sales Genres Over Years",
       x="Year",
       y="Proportion of Sales",
       fill="Genre") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



As shown in the plot, the distribution of five genres varies according to the year.

c.

```
# Calculate the average sales price per genre per year
c_art_df <- art_df %>%
  tidyr::gather(key="genre", value="is_genre", starts_with("Genre___")) %>%
  filter(is_genre == 1) %>%
  select(year, genre, price_usd)

photography_df <- c_art_df %>%
  filter(genre == "Genre___Photography") %>%
  select(year, price_usd) %>%
  group_by(year) %>%
  summarise(mean_price_usd=mean(price_usd)) %>%
  ungroup()

painting_df <- c_art_df %>%
  filter(genre == "Genre___Painting") %>%
  select(year, price_usd) %>%
  group_by(year) %>%
  summarise(mean_price_usd=mean(price_usd)) %>%
```

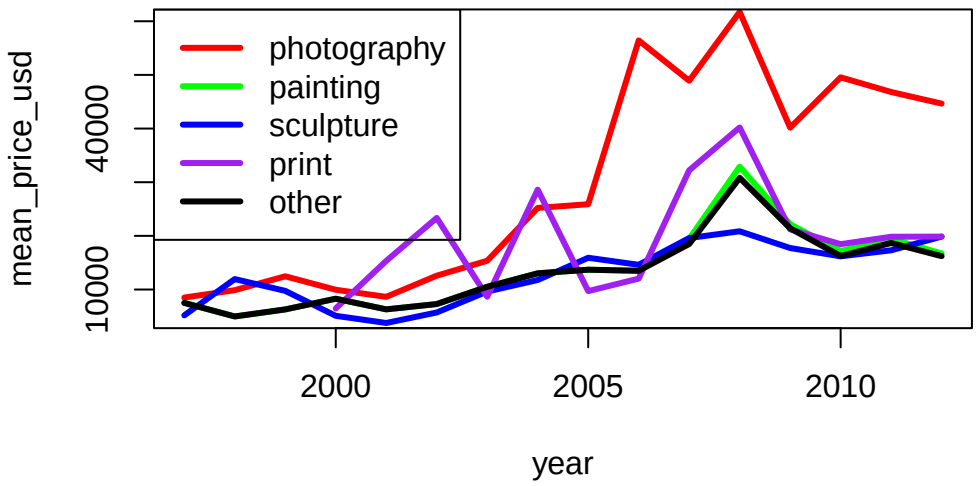
```
  ungroup()

other_df <- c_art_df %>%
  filter(genre == "Genre___Others") %>%
  select(year, price_usd) %>%
  group_by(year) %>%
  summarise(mean_price_usd=mean(price_usd)) %>%
  ungroup()

sculpture_df <- c_art_df %>%
  filter(genre == "Genre___Sculpture") %>%
  select(year, price_usd) %>%
  group_by(year) %>%
  summarise(mean_price_usd=mean(price_usd)) %>%
  ungroup()

print_df <- c_art_df %>%
  filter(genre == "Genre___Print") %>%
  select(year, price_usd) %>%
  group_by(year) %>%
  summarise(mean_price_usd=mean(price_usd)) %>%
  ungroup()

with(photography_df, plot(mean_price_usd ~ year, type = "l",
                          lwd=3, ylim = c(5000, 60000), col = "red"))
lines(painting_df$mean_price_usd ~ painting_df$year, lwd=3, col = "green")
lines(sculpture_df$mean_price_usd ~ sculpture_df$year, lwd=3, col = "blue")
lines(print_df$mean_price_usd ~ print_df$year, lwd=3, col = "purple")
lines(other_df$mean_price_usd ~ other_df$year, lwd=3, col = "black")
legend("topleft", legend = c("photography", "painting", "sculpture", "print", "other"),
       lty = c(1, 1, 1, 1, 1), lwd = c(3, 3, 3, 3, 3),
       col = c("red", "green", "blue", "purple", "black"))
```



From the plot, one can argue that, the price of photography change more rapidly. The price of print is also changing very rapidly. While, for painting and other genres, their prices change in a similar way. The price of sculpture is smoother than other four genres.