Problem Set 4

Rajvi Jasani

GitHub Repository

This is the link to my GitHub repository https://github.com/rajvijasani/STATS506-Problem-Set-4.git

Problem 1 - Tidyverse

Documentation reference: https://cran.r-project.org/web/packages/nycflights13/nycflights13.pdf

```
library(tidyverse)
library(nycflights13)
```

a.

Departure

```
flights %>%
  group_by(origin) %>%
  summarise(mean_delay = round(mean(dep_delay, na.rm = TRUE), 3),
            median_delay = median(dep_delay, na.rm = TRUE)) %>%
  ungroup() %>%
  left_join(airports, by = c("origin" = "faa")) %>%
  select(name, mean_delay, median_delay) %>%
  arrange(desc(mean_delay))
```

```
# A tibble: 3 x 3
  name                mean_delay median_delay
  <chr>                    <dbl>        <dbl>
1 Newark Liberty Intl       15.1           -1
2 John F Kennedy Intl       12.1           -1
3 La Guardia                10.3           -3
```

Arrival

```
flights %>%
  group_by(dest) %>%
  summarise(
    mean_delay = round(mean(arr_delay, na.rm = TRUE), 3),
    median_delay = median(arr_delay, na.rm = TRUE),
    num_flights = n()
  ) %>%
  ungroup() %>%
  filter(num_flights >= 10) %>%
  left_join(airports, by = c("dest" = "faa")) %>%
  # to rename airport name with
  # faa code where name is NA
  mutate(name = if_else(is.na(name), dest, name)) %>%
  select(name, mean_delay, median_delay) %>%
  arrange(desc(mean_delay)) %>%
  print(n = count(.))
```

```
# A tibble: 102 x 3
    name                                mean_delay median_delay
    <chr>                                    <dbl>        <dbl>
  1 "Columbia Metropolitan"                   41.8           28
  2 "Tulsa Intl"                              33.7           14
  3 "Will Rogers World"                       30.6           16
  4 "Jackson Hole Airport"                    28.1           15
  5 "Mc Ghee Tyson"                           24.1            2
  6 "Dane Co Rgnl Truax Fld"                  20.2            1
  7 "Richmond Intl"                           20.1            1
  8 "Akron Canton Regional Airport"           19.7            3
  9 "Des Moines Intl"                         19.0            0
 10 "Gerald R Ford Intl"                      18.2            1
 11 "Birmingham Intl"                         16.9           -2
 12 "Theodore Francis Green State"            16.2            1
 13 "Greenville-Spartanburg International"    15.9           -0.5
 14 "Cincinnati Northern Kentucky Intl"       15.4           -3
 15 "Savannah Hilton Head Intl"               15.1           -1
 16 "Manchester Regional Airport"             14.8           -3
 17 "Eppley Afld"                             14.7           -2
 18 "Yeager"                                  14.7           -1.5
 19 "Kansas City Intl"                        14.5            0
 20 "Albany Intl"                             14.4           -4
```

21	"General Mitchell Intl"	14.2	0
22	"Piedmont Triad"	14.1	-2
23	"Washington Dulles Intl"	13.9	-3
24	"Cherry Capital Airport"	13.0	-10
25	"James M Cox Dayton Intl"	12.7	-3
26	"Louisville International Airport"	12.7	-2
27	"Chicago Midway Intl"	12.4	-1
28	"Sacramento Intl"	12.1	4
29	"Jacksonville Intl"	11.8	-2
30	"Nashville Intl"	11.8	-2
31	"Portland Intl Jetport"	11.7	-4
32	"Greater Rochester Intl"	11.6	-5
33	"Hartsfield Jackson Atlanta Intl"	11.3	-1
34	"Lambert St Louis Intl"	11.1	-3
35	"Norfolk Intl"	10.9	-4
36	"Baltimore Washington Intl"	10.7	-5
37	"Memphis Intl"	10.6	-2.5
38	"Port Columbus Intl"	10.6	-3
39	"Charleston Afb Intl"	10.6	-4
40	"Philadelphia Intl"	10.1	-3
41	"Raleigh Durham Intl"	10.1	-3
42	"Indianapolis Intl"	9.94	-3
43	"Charlottesville-Albemarle"	9.5	-5
44	"Cleveland Hopkins Intl"	9.18	-5
45	"Ronald Reagan Washington Natl"	9.07	-2
46	"Burlington Intl"	8.95	-4
47	"Buffalo Niagara Intl"	8.95	-5
48	"Syracuse Hancock Intl"	8.90	-5
49	"Denver Intl"	8.61	-2
50	"Palm Beach Intl"	8.56	-3
51	"BQN"	8.24	-1
52	"Bob Hope"	8.18	-3
53	"Fort Lauderdale Hollywood Intl"	8.08	-3
54	"Bangor Intl"	8.03	-9
55	"Asheville Regional Airport"	8.00	-1
56	"PSE"	7.87	0
57	"Pittsburgh Intl"	7.68	-5
58	"Gallatin Field"	7.6	-2
59	"NW Arkansas Regional"	7.47	-2
60	"Tampa Intl"	7.41	-4
61	"Charlotte Douglas Intl"	7.36	-3
62	"Minneapolis St Paul Intl"	7.27	-5
63	"William P Hobby"	7.18	-4

```
 64 "Bradley Intl"                              7.05         -10
 65 "San Antonio Intl"                          6.94          -9
 66 "South Bend Rgnl"                           6.5           -3.5
 67 "Louis Armstrong New Orleans Intl"          6.49          -6
 68 "Key West Intl"                             6.35           7
 69 "Eagle Co Rgnl"                             6.30          -4
 70 "Austin Bergstrom Intl"                     6.02          -5
 71 "Chicago Ohare Intl"                        5.88          -8
 72 "Orlando Intl"                              5.46          -5
 73 "Detroit Metro Wayne Co"                    5.43          -7
 74 "Portland Intl"                             5.14          -5
 75 "Nantucket Mem"                             4.85          -3
 76 "Wilmington Intl"                           4.64          -7
 77 "Myrtle Beach Intl"                         4.60         -13
 78 "Albuquerque International Sunport"         4.38          -5.5
 79 "George Bush Intercontinental"              4.24          -5
 80 "Norman Y Mineta San Jose Intl"             3.45          -7
 81 "Southwest Florida Intl"                    3.24          -5
 82 "San Diego Intl"                            3.14          -5
 83 "Sarasota Bradenton Intl"                   3.08          -5
 84 "Metropolitan Oakland Intl"                 3.08          -9
 85 "General Edward Lawrence Logan Intl"        2.91          -9
 86 "San Francisco Intl"                        2.67          -8
 87 "SJU"                                       2.52          -6
 88 "Yampa Valley"                              2.14           2
 89 "Phoenix Sky Harbor Intl"                   2.10          -6
 90 "Montrose Regional Airport"                 1.79         -10.5
 91 "Los Angeles Intl"                          0.547         -7
 92 "Dallas Fort Worth Intl"                    0.322         -9
 93 "Miami Intl"                                0.299         -9
 94 "Mc Carran Intl"                            0.258         -8
 95 "Salt Lake City Intl"                       0.176         -8
 96 "Long Beach"                               -0.062        -10
 97 "Martha\\\\'s Vineyard"                    -0.286        -11
 98 "Seattle Tacoma Intl"                      -1.10         -11
 99 "Honolulu Intl"                            -1.36          -7
100 "STT"                                      -3.84          -9
101 "John Wayne Arpt Orange Co"                -7.87         -11
102 "Palm Springs Intl"                       -12.7          -13.5
```

Attribution of source: Used ChatGPT for help with join and renaming airports with faa codes that have NA values in names.

b.

There is a speed column in the planes table, but the values haven't been populated so we calculate speed on our own.

```
flights %>%
  left_join(planes, by = "tailnum") %>%
  mutate(time = air_time / 60, # converting from minutes to hours
         flight_speed = distance / time) %>%
  group_by(model) %>% # different tail number planes can be of same model
  summarise(avg_speed = mean(flight_speed, na.rm = TRUE),
            num_flights = n()) %>%
  filter(avg_speed == max(avg_speed, na.rm = TRUE))
```

```
# A tibble: 1 x 3
  model   avg_speed num_flights
  <chr>       <dbl>       <int>
1 777-222      483.           4
```

Problem 2 - `get_temp()`

```
nnmaps <- read.csv("data/chicago-nmmaps.csv")
#' Function to get average temperature based on user's query
#'
#' @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 FALS
#' @param average_fn A function to compute the average. Default mean
#'
#' @return average temperature for a given month
get_temp <- function(month, year, data, celsius = FALSE, average_fn = mean) {
  # sanitizing month input
  if (month %>% is.numeric()) {
    if (month < 1 || month > 12) {
      stop("Month should be between 1 and 12")
    }
  }
  else if (month %>% is.character()) {
    # taking into consideration month abbreviations (index 1-12)
```

```
    # and full names (index 13-24)
    all_months <- c(month.abb, month.name) %>% tolower()
    # getting index of match if exists
    month <- match(tolower(month), all_months)
    if (month %>% is.na()) {
      stop("Invalid month string")
    }
    else if (month > 12) {
      # converting month index for full name from 13-24 to 1-12
      month <- month %>% `-` (12)
    }
  }
  else {
    stop("Month should be numeric or string only")
  }

  # sanitizing year input
  if (!year %>% is.numeric()) {
    stop("Year must be numeric")
  }
  if (year < min(data$year) || year > max(data$year)) {
    stop("Year not in range")
  }

  # querying
  result <- data %>%
    select(year, month_numeric, temp) %>%
    rename(year_data = year) %>%
    filter(year_data == year, month_numeric == month) %>%
    summarise(avg_temp = average_fn(temp)) %>%
    mutate(avg_temp = ifelse(celsius, (avg_temp - 32) * 5 / 9, avg_temp)) %>%
    as.numeric()

  return(result)
}
```

Attribution of source: Asked ChatGPT for possible methods to sanitize month input when month is as string that can be a full name or an abbreviation.

```
tryCatch({
  print(get_temp("Apr", 1999, data = nnmaps))
}, error = function(e) {
```

```
  message("Error: ", e$message)
})
```

```
[1] 49.8
```

```
tryCatch({
  print(get_temp("Apr", 1999, data = nnmaps, celsius = TRUE))
}, error = function(e) {
  message("Error: ", e$message)
})
```

```
[1] 9.888889
```

```
tryCatch({
  print(get_temp(10, 1998, data = nnmaps, average_fn = median))
}, error = function(e) {
  message("Error: ", e$message)
})
```

```
[1] 55
```

```
tryCatch({
  print(get_temp(13, 1998, data = nnmaps))
}, error = function(e) {
  message("Error: ", e$message)
})
```

```
Error: Month should be between 1 and 12
```

```
tryCatch({
  print(get_temp(2, 2005, data = nnmaps))
}, error = function(e) {
  message("Error: ", e$message)
})
```

```
Error: Year not in range
```

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

```
[1] 7.301587
```

Attribution of source: Asked ChatGPT for possible solution when quarto document was not rendering due to error when function is given wrong input. ChatGPT suggested using try-catch blocks to handle errors.

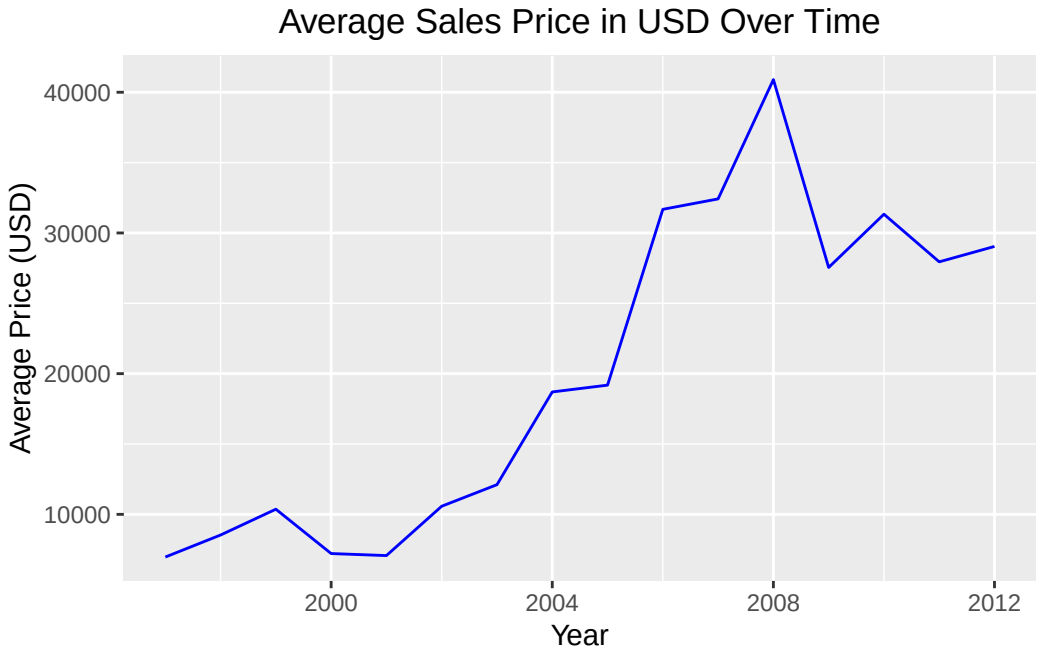
Problem 3 - Visualization

```
library(ggplot2)
library(dplyr)
df <- read.csv("data/df_for_ml_improved_new_market.csv")
```

a.

```
df_avg_price <- df %>%
  group_by(year) %>%
  summarise(avg_usd_year = mean(price_usd, na.rm = TRUE)) %>%
  ungroup() %>%
  select(year, avg_usd_year)

print(ggplot(df_avg_price, aes(x = year, y = avg_usd_year)) +
  geom_line(color = "blue") +
  labs(title = "Average Sales Price in USD Over Time", x = "Year", y = "Average Price (USD)"
  theme(plot.title = element_text(hjust = 0.5)))
```



The line plot is the simplest and most informative graph to look at a trend. Looking at the graph here, we see that there is an overall increase in average sales price over the years until 2008 after which we can notice an overall decline. The increase and decrease in the prices is not constant and varies over period of years.

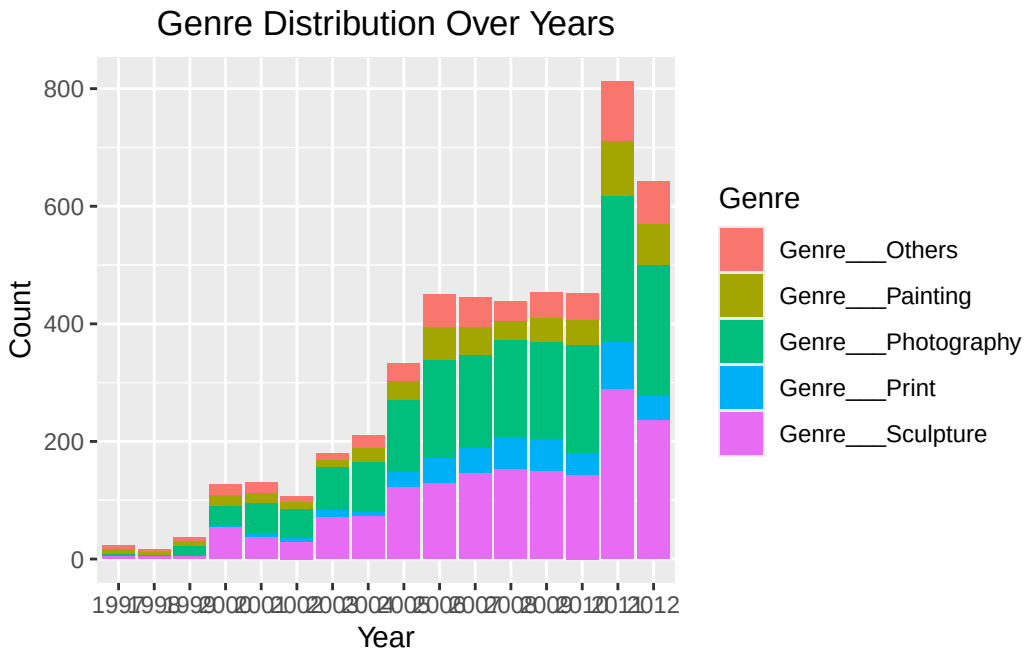
Attribution of source: Used ChatGPT to find functions to make the graph look good such a theme().

b.

```
df_long <- df %>%
  pivot_longer(cols = starts_with("Genre"),
               names_to = "Genre",
               values_to = "Presence") %>%
  filter(Presence == 1)

df_genre_distribution <- df_long %>%
  count(year, Genre)

print(ggplot(df_genre_distribution, aes(x = factor(year), y = n, fill = Genre)) +
  geom_bar(stat = "identity", position = "stack") +
  labs(title = "Genre Distribution Over Years", x = "Year", y = "Count") +
  theme(plot.title = element_text(hjust = 0.5)))
```



A stacked bar plot with different genres colored differently can help easily spot the trend. It is quite clear from this graph that the distribution of genre of sales changes over the years but differs from genre to genre. At the beginning, the sales for each genre were quite low. With time, we see there is an increase in sales of sculptures and photographs. However, they both seem to have a steady amount of sales in the last half, between the years 2005-2010. The other genres also have increase in sales but the count isn't as major as photography and sculptures. One interesting observation I noticed was that the sales of paintings genre looks almost equal to other genre. This could be due to various reasons and should be investigated further.

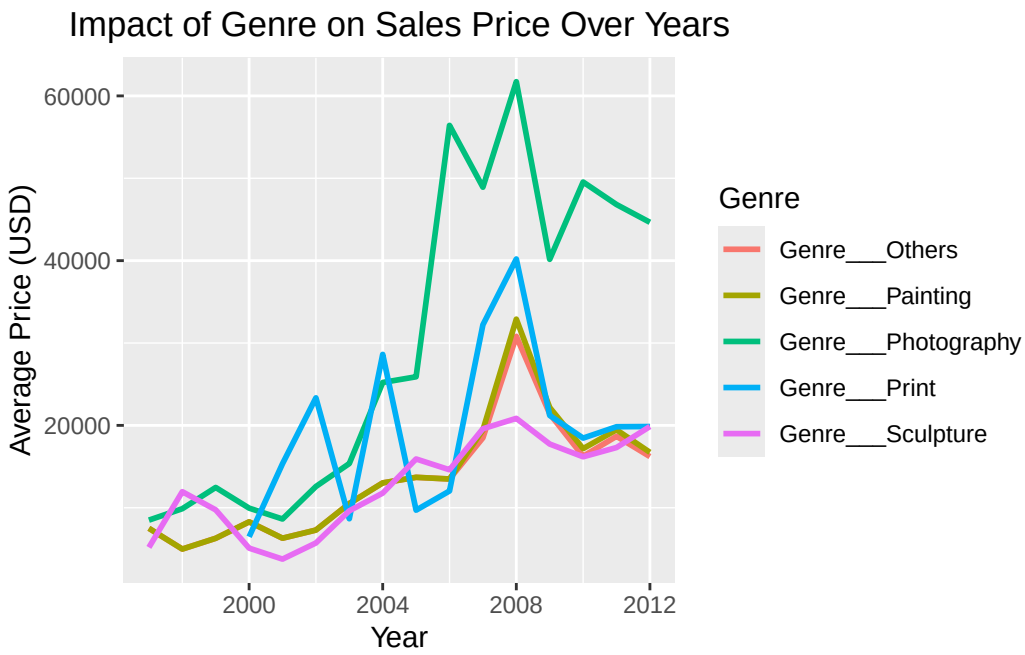
Attribution of source: Asked ChatGPT how to combine the different genre columns into 1 column (opposite of factoring). Looked at ggplot2 documentation https://ggplot2.tidyverse.org/reference/ for available geom functions.

c.

```
df_genre_price <- df_long %>%
  group_by(year, Genre) %>%
  summarise(avg_price = mean(price_usd, na.rm = TRUE)) %>%
  ungroup() %>%
  select(year, avg_price, Genre)
```

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

```
print(ggplot(df_genre_price, aes(x = year, y = avg_price, color = Genre)) +
  geom_line(linewidth = 1) +
  labs(title = "Impact of Genre on Sales Price Over Years", x = "Year", y = "Average Price (
  theme(plot.title = element_text(hjust = 0.5)))
```



These plots show the demand of each genre through their sale prices over years. We can identify that some genres are more susceptible to market trends/popularities, like Photography and Print, while others are more stable. When there is an increase in demand of a particular genre, like Photography peaking in 2008, the total sale prices also increased dramatically, which can be verified from the graph in part a. Similarly, the rise and falls seen in graph in part a can be justified and understood by looking at these plots. We can also verify that the sales of paintings genre looks almost equal to other genre as their line plots are overlapping for up till a point.

Attribution of source: Looked at ggplot2 documentation https://ggplot2.tidyverse.org/reference/ for available geom functions.