Stats 506 PS4

Alyssa Yang

GitHub repo link: https://github.com/alyssawyang/stats506ps4

Problem 1: Tidyverse

```
library(nycflights13)
suppressMessages(library(dplyr))
library(tidyr)
library(stringr)
library(ggplot2)

# View(flights)
# View(planes)
```

1a

```
# Departure delays
flights %>%
  group_by(origin) %>%
  summarize(mean_dep_delay = mean(dep_delay, na.rm = TRUE),
            median_dep_delay = median(dep_delay, na.rm = TRUE),
            num_flights = n()) %>%
  ungroup() %>%
  filter(num_flights >= 10) %>%
  rename(faa = origin) %>%
  left_join(airports, by = "faa") %>%
  arrange(desc(mean_dep_delay)) %>%
  select(name, mean_dep_delay, median_dep_delay)
```

```
# A tibble: 3 x 3
  name                mean_dep_delay median_dep_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
```

```r
# Arrival delays
flights %>%
  group_by(dest) %>%
  summarize(mean_arr_delay = mean(arr_delay, na.rm = TRUE),
            median_arr_delay = median(arr_delay, na.rm = TRUE),
            num_flights = n()) %>%
  ungroup() %>%
  filter(num_flights >= 10) %>%
  rename(faa = dest) %>%
  left_join(airports, by = "faa") %>%
  mutate(name = coalesce(name, faa)) %>%
  arrange(desc(mean_arr_delay)) %>%
  select(name, mean_arr_delay, median_arr_delay) %>%
  print(n = count(.))
```

```
# A tibble: 102 x 3
    name                                    mean_arr_delay median_arr_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"	8.25	-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"	6.95	-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"	5.45	-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"	-0.0620	-10
97	"Martha\\\\'s Vineyard"	-0.286	-11
98	"Seattle Tacoma Intl"	-1.10	-11
99	"Honolulu Intl"	-1.37	-7
100	"STT"	-3.84	-9
101	"John Wayne Arpt Orange Co"	-7.87	-11
102	"Palm Springs Intl"	-12.7	-13.5

1b

```
# Find number of flights the aircraft model with the fastest average speed has
flights %>%
  left_join(planes, by = "tailnum") %>%
  mutate(mph = distance / (air_time / 60)) %>%
  group_by(model) %>%
  summarize(avg_mph = mean(mph),
            num_flights = n()) %>%
  ungroup() %>%
  arrange(desc(avg_mph)) %>%
  slice(1)
```

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

Problem 2: get_temp()

```
# Load in Chicago NNMAPS data
nnmaps <- read.csv("chicago-nmmaps.csv")
# View(nnmaps)
```

```
#' Function that finds average monthly temperature
#'
#' @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 is FALSE
#' @param average_fn A function with which to compute the mean. Default is mean
#'
#' @return Average temperature as a numeric vector of length 1
get_temp <- function(month, year, data, celsius = FALSE, average_fn = mean) {
  # Check if month is a valid month
  if (month %>% is.numeric) {
    if (month < 1 | month > 12) {
```

```
    warning("Month is not valid")
    return(NA)
  }
}
else if (month %>% is.character) {
  months <- c("January", "February", "March", "April", "May", "June", "July",
              "August", "September", "October", "November", "December")
  month <- month %>% match.arg(months)
  month <- month %>% match(months)
}
else {
  warning("Month must be a valid numeric or character")
  return(NA)
}

# Check if year is a valid year
if (!(year %>% is.numeric)) {
  warning("Year must be numeric")
  return(NA)
}
if (year < 1997 | year > 2000) {
  warning("Year is not valid")
  return(NA)
}

# Check if function is a valid function
 if (!(average_fn %>% is.function)) {
  warning("Average_fn must be a function")
   return(NA)
 }

# If celsius is true, convert from F to C
if (celsius == TRUE) {
  data <- data %>%
    mutate(temp = (temp - 32) * (5/9))
}

# Find and return average monthly temperature
avg <- data %>%
  select(year, month_numeric, temp) %>%
  rename(year_data = year) %>%
  filter(year_data == year,
```

```
            month_numeric == month) %>%
    summarize(average_fn(temp))

  return(avg)
}
```

```
# Evaluate code
get_temp("Apr", 1999, data = nnmaps)
```

```
  average_fn(temp)
1             49.8
```

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

```
  average_fn(temp)
1         9.888889
```

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

```
  average_fn(temp)
1               55
```

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

```
Warning in get_temp(13, 1998, data = nnmaps): Month is not valid
```

```
[1] NA
```

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

```
Warning in get_temp(2, 2005, data = nnmaps): Year is not valid
```

```
[1] NA
```

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

```
  average_fn(temp)
1         7.301587
```

Problem 3: Visualization

```
# Read in art sales dataset
art_sales <- read.csv("df_for_ml_improved_new_market.csv")

# View(art_sales)
```

3a: Is there a change in the sales price in USD over time?

```
# Look at summary to help determine axis bounds for sales prices
summary(art_sales$price_usd)
```

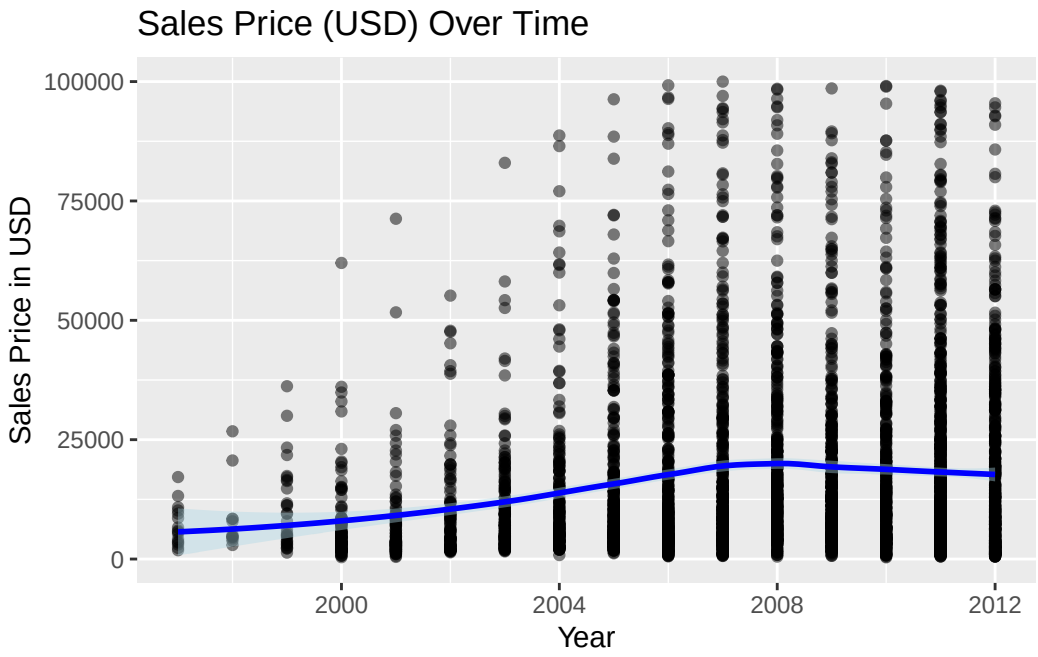
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    361    4757   10387   27028   25774 1395790 
```

```
# Plot sales price over the years
ggplot(art_sales, aes(x = year, y = price_usd)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "loess", color = "blue", fill = "lightblue") +
  labs(title = "Sales Price (USD) Over Time",
       x = "Year",
       y = "Sales Price in USD") +
  ylim(0, 100000)
```

```
`geom_smooth()` using formula = 'y ~ x'
```

```
Warning: Removed 241 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 241 rows containing missing values or values outside the scale range
(`geom_point()`).
```



From the graph, we can see that there has been a change in Sales Price (USD) over time which may be due to factors such as a growing demand in the market or other broader economic factors. The range of sales prices has increased over time with more art pieces being a lot more expensive than in earlier years. I added a LOESS smoothing line in order to capture the overall trend of sales prices, and we can see that the prices gradually increased until around 2008 when it started slightly falling again.

I chose to include only sales prices from \$0 to \$100,000 in order to better visualize the trend - by excluding some very high-priced outliers, this range helps to highlight the general patterns of sales prices more.

3b: Does the distribution of genre of sales across years appear to change?

```
# Collapse genre columns into one column
art_sales_genres <- art_sales %>%
  pivot_longer(cols = starts_with("Genre___"),
               names_to = "genre",
               values_to = "count")

# Rename genre values and select relevant columns
art_sales_genres <- art_sales_genres %>%
  mutate(genre = str_remove(genre, "Genre___")) %>%
```

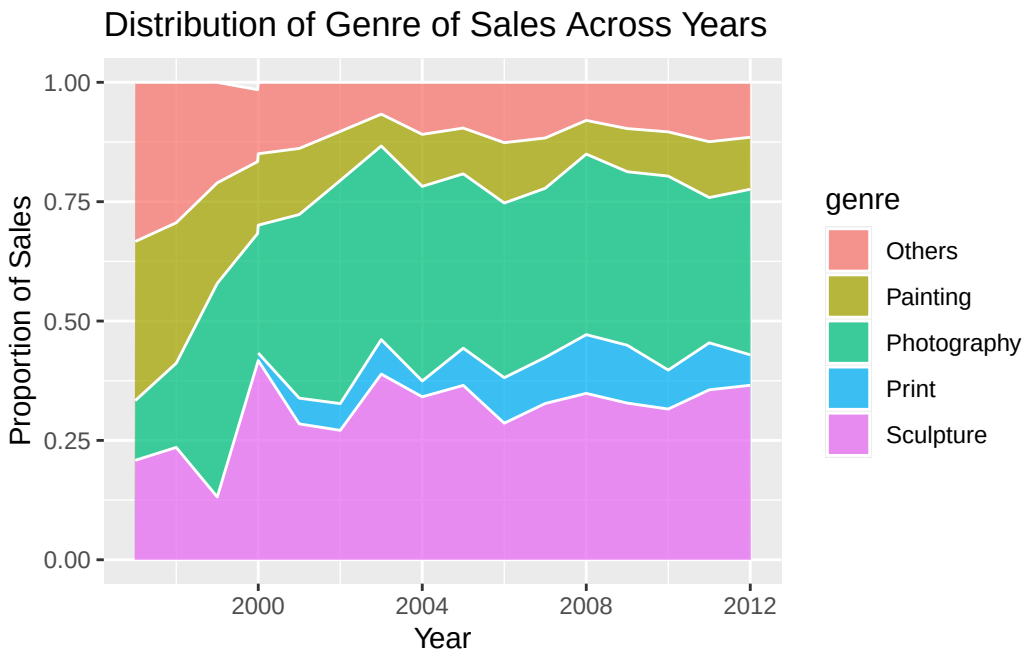
```r
  filter(count > 0) %>%
  select(year, genre)

# Calculate proportions of genre of sales across years
genre_dist <- art_sales_genres %>%
  group_by(year, genre) %>%
  summarize(num_sales = n()) %>%
  mutate(proportion = num_sales / sum(num_sales))
```

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

```r
# Plot genre distribution across years
ggplot(genre_dist, aes(x = year, y = proportion, fill = genre)) +
  geom_area(alpha = 0.75, size = 0.5, color = "white") +
  labs(title = "Distribution of Genre of Sales Across Years",
       x = "Year",
       y = "Proportion of Sales")
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```



From the graph, we can see that the distribution of genre of sales did change quite a bit from 1997 to 2004, but after 2004, the distribution has stayed relatively constant. People only started selling prints in 2000, and sculpture and photography gained popularity whereas painting and others decreased in popularity until around 2004. After 2004, photography and sculpture remain the most popular genre of art sales, and print, painting, and others are not as popular.

3c: How does the genre affect the change in sales price over time?

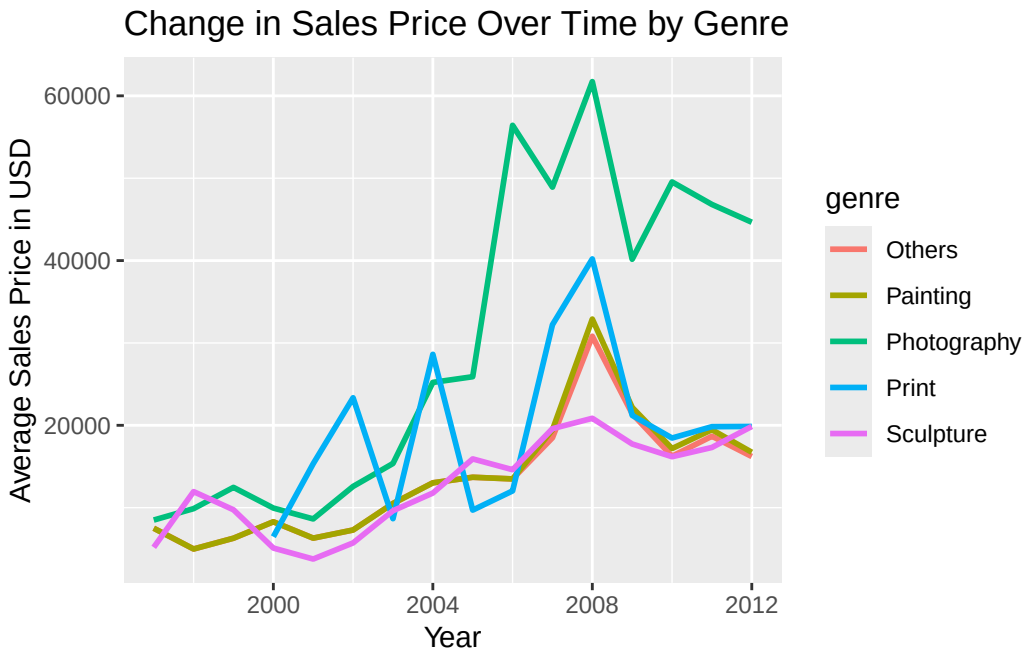
```
# Collapse genre columns into one column
art_sales_genres <- art_sales %>%
  pivot_longer(cols = starts_with("Genre___"),
               names_to = "genre",
               values_to = "count")

# Rename genre values and select relevant columns
art_sales_genres <- art_sales_genres %>%
  mutate(genre = str_remove(genre, "Genre___")) %>%
  filter(count > 0) %>%
  select(year, genre, price_usd)

# Calculate average sales price by year and genre
avg_price_genre <- art_sales_genres %>%
  group_by(year, genre) %>%
  summarize(avg_sales_price = mean(price_usd, na.rm = TRUE))
```

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

```
# Plot price change over time by genre
ggplot(avg_price_genre, aes(x = year, y = avg_sales_price, color = genre)) +
  geom_line(size = 1) +
  labs(title = "Change in Sales Price Over Time by Genre",
       x = "Year",
       y = "Average Sales Price in USD")
```



From the graph, we can see that over time, different genres have had different average sales prices. We can see that the average sales price for painting and others genres have stayed almost identical to each other across the years with sculpture following a similar trend. However, with sculpture, it didn't have as high of a peak of average salary as the other genres in 2008 and has consistently remained quite low. Since it's arrival in 2000, prints' average sales prices have fluctuated drastically, sometimes being the highest average sales price genre and sometimes being the lowest. Photography started out having a similar average sales price as painting, others, and sculpture, but has since increased drastically from around 2004. Since around 2005, it has had the highest average sales price compared to all other genres by a significant margin, and since 2010, all other genres have had around the same average sales price.