hw4-stats506-lijiabao

JiabaoLi

link of github: https://github.com/lijiabao203/stats506_rwork

Problem 1 - Tidyverse

Use the **tidyverse** for this problem. In particular, use piping and **dplyr** as much as you are able. **Note**: Use of any deprecated functions will result in a point loss.

Install and load the package **nycflights13**.

a. Generate a table (which can just be a nicely printed tibble) reporting the mean and median departure delay per airport. Generate a second table (which again can be a nicely printed tibble) reporting the mean and median arrival delay per airport. Exclude any destination with under 10 flights. Do this exclusion through code, not manually.

```
Additionally,

-   Order both tables in descending mean delay.

-   Both tables should use the airport *names* not the airport *codes*.

-   Both tables should print all rows.
```

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

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
```

```
-- Conflicts --------------------------------------- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
org_and_mm = flights %>%
  group_by(origin) %>%
  filter(n() > 9) %>%
  summarize(mean(dep_delay, na.rm = T), median(dep_delay, na.rm = T)) %>%
  ungroup()
tab1 = org_and_mm %>%
  left_join(airports, by = c("origin" = "faa")) %>%
  select(name,
         "mean(dep_delay, na.rm = T)",
         "median(dep_delay, na.rm = T)") %>%
  rename("airport name" = name,
         "dep-delay time mean" = "mean(dep_delay, na.rm = T)",
         "dep-delay time median" = "median(dep_delay, na.rm = T)") %>%
  arrange(desc(`dep-delay time mean`))
print(tab1)
```

```
# A tibble: 3 x 3
  `airport name`      `dep-delay time mean` `dep-delay time median`
  <chr>                               <dbl>                   <dbl>
1 Newark Liberty Intl                  15.1                      -1
2 John F Kennedy Intl                  12.1                      -1
3 La Guardia                           10.3                      -3
```

```
# here is he first table


dest_and_mm = flights %>%
  group_by(dest) %>%
  filter(n() > 9) %>%
  summarize(mean(arr_delay, na.rm = T), median(arr_delay, na.rm = T)) %>%
  ungroup()
tab2 = dest_and_mm %>%
  left_join(airports, by = c("dest" = "faa")) %>%
  select(name,
         "mean(arr_delay, na.rm = T)",
         "median(arr_delay, na.rm = T)") %>%
  rename("airport name" = name,
```

```
         "arr-delay time mean" = "mean(arr_delay, na.rm = T)",
         "ar-delay time median" = "median(arr_delay, na.rm = T)") %>%
  arrange(desc(`arr-delay time mean`))
print(tab2, n = Inf)
```

```
# A tibble: 102 x 3
    `airport name`                  `arr-delay time mean` `ar-delay time median`
    <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 Intern~                 15.9                    -0.5
 14 "Cincinnati Northern Kentucky ~                 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 Airp~                 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 In~                 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	<NA>	8.25	-1
52	"Bob Hope"	8.18	-3
53	"Fort Lauderdale Hollywood Int~	8.08	-3
54	"Bangor Intl"	8.03	-9
55	"Asheville Regional Airport"	8.00	-1
56	<NA>	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 I~	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 Sun~              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~             2.91                    -9
 86 "San Francisco Intl"                        2.67                    -8
 87  <NA>                                       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  <NA>                                      -3.84                    -9
101 "John Wayne Arpt Orange Co"                -7.87                    -11
102 "Palm Springs Intl"                       -12.7                     -13.5
```

```
# here is the second table, and it's kind of long.
```

Here is a missing airport name, which has dest faa "STT". It may because of a error input of the table or the reason may be that this airport has not been recorded.

```
any(airports$faa %>% match("STT"))
```

```
[1] NA
```

b. How many flights did the aircraft model with the fastest average speed take? Produce a tibble with 1 row, and entires for the model, average speed (in MPH) and number of flights.

```r
# calculate the avg speed of each flight first
avgmph_of_flights = flights %>%
  group_by(tailnum) %>%
  summarize(sum(distance) / (sum(hour) + sum(minute) / 60), n()) %>%
  ungroup() %>%
  rename("numf" = "n()",
         "avgsp" = `sum(distance)/(sum(hour) + sum(minute)/60)`)

# calculate divided by flight model
ans = avgmph_of_flights %>%
  inner_join(planes, by = "tailnum") %>%
  group_by(model) %>%
  summarize(sum(avgsp * numf)/sum(numf), sum(numf)) %>%
  rename("number of flights" = `sum(numf)`,
         "average speed in MPH" = `sum(avgsp * numf)/sum(numf)`) %>%
  select("model", "number of flights", "average speed in MPH") %>%
  ungroup() %>%
  arrange(desc(`average speed in MPH`)) %>%
  head(1)
ans
```

```
# A tibble: 1 x 3
  model    `number of flights` `average speed in MPH`
  <chr>                  <int>                  <dbl>
1 A330-243                 342                   511.
```

Problem 2 - `get_temp()`

Use the **tidyverse** for this problem. In particular, use piping and **dplyr** as much as you are able. **Note**: Use of any deprecated functions will result in a point loss.

Load the Chicago NNMAPS data we used in the visualization lectures. Write a function `get_temp()` that allows a user to request the average temperature for a given month. The arguments should be:

- `month`: Month, either a numeric 1-12 or a string.
- `year`: A numeric year.
- `data`: The data set to obtain data from.
- `celsius`: Logically indicating whther the results should be in celsius. Default `FALSE`.

- `average_fn`: A function with which to compute the mean. Default is `mean`.

The output should be a numeric vector of length 1. The code inside the function should, as with the rest of this problem, use the **tidyverse**. Be sure to sanitize the input.

Prove your code works by evaluating the following. Your code should produce the result, or a reasonable error message.

```
nnmaps = read.csv("chicago-nmmaps.csv")
nnmaps$date = as.Date(nnmaps$date)
# the inputs and output are listed in the problem.
get_temp <- function(mon, yea, data, celsius = FALSE, average_fn = mean){
  tab = data %>%
    filter(year == yea, mon == month | mon == month_numeric)
  # error check, to avoid the situation that can't get any elements.
  if(nrow(tab) == 0){
    return("extract null message, please check the year and month message.")
  }
  table_all = tab %>%
    summarize(average_fn(temp)) %>%
    mutate(not_temp = `average_fn(temp)`*9/5+32)
  return(ifelse(celsius, table_all %>% select(`average_fn(temp)`), table_all %>% select(not_t
}
```

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

```
[[1]]
[1] 121.64
```

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

```
[[1]]
[1] 49.8
```

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

```
[[1]]
[1] 131
```

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

```
[1] "extract null message, please check the year and month message."
```

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

```
[1] "extract null message, please check the year and month message."
```

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

```
[1] "extract null message, please check the year and month message."
```

Problem 3 - Visualization

Note: This is, intentionally, a very open-ended question. There is no "right" answer. The goal is for you to explore your plotting options, and settle on something reasonable. You can use base R, ggplot, or soemthing else. You'll likely have to look online for resources on plotting beyond what we covered in class.

This dataset lists characteristics of art sales. Download the file named "df_for_ml_improved_new_market" (NOT the "df_for_ml_improved_new_market_1" version!). For each of the following, produce a publication-ready plot which demonstrates the answer to the question. Use your plot to support an argument for your question.

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

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

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

You will be graded on:

i. Is the type of graph & choice of variables appropriate to answer the question?

ii. Is the graph clear and easy to interpret?

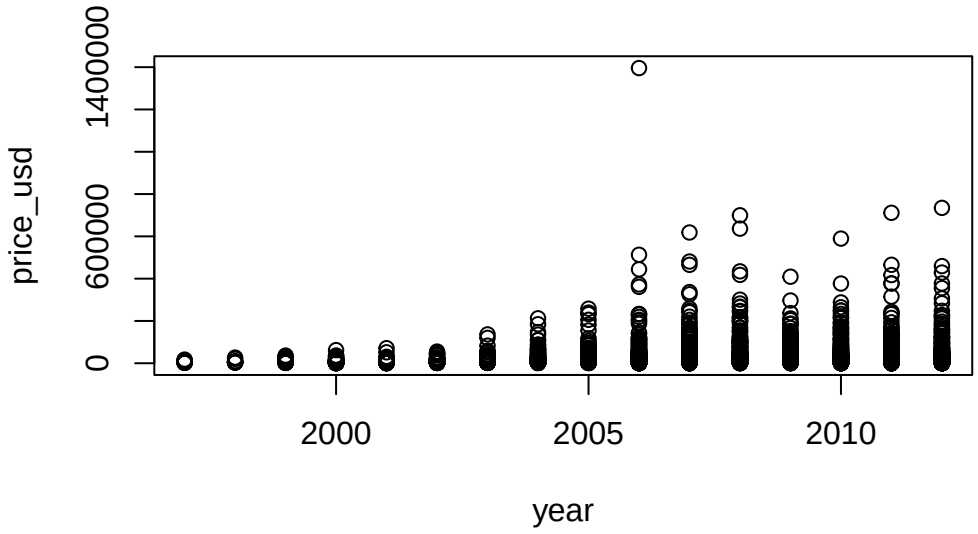
iii. Is the graph publication ready?

Here start the answer

a.

Based on the plot, there is a noticeable trend suggesting that newer items (or items from more recent years) tend to have higher prices.

```
artsales = read.csv("df_for_ml_improved_new_market.csv")
with(artsales, plot(year, price_usd))
```



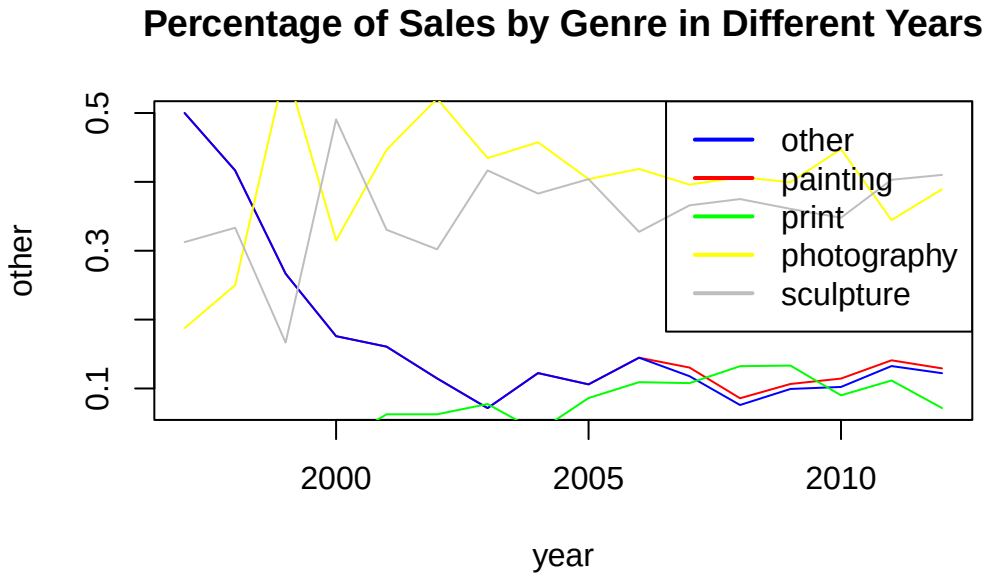
b.

The figure showing the percentage of sales by genre indicates an increase in sales of paintings and "other" in recent years, while sales classified as photography and sculpture have declined. Additionally, the percentage of print sales has experienced a slight increase.

On another note, some sales are classified as both 'other' and additional categories. If we exclude these from the 'other' classification (consider classes beside "other" first), we can analyze the updated plots (two hist plots latter): there is an increase in sales of photography, print, and 'other' categories, while sales of paintings have decreased. Sales of sculpture show slight fluctuations.

```
ann = artsales %>%
  select(year, Genre___Others, Genre___Print, Genre___Photography, Genre___Painting, Genre__
  group_by(year) %>%
  summarize(sum(Genre___Others)/n(), sum(Genre___Print)/n(), sum(Genre___Photography)/n(), su
```

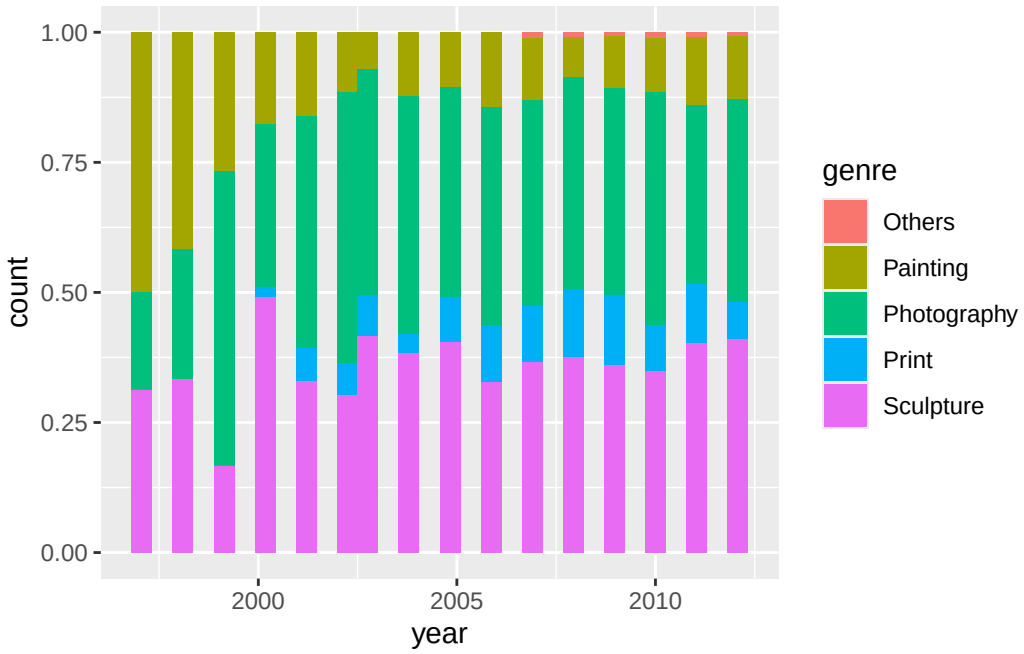
```
  rename(other = `sum(Genre___Others)/n()`, print = `sum(Genre___Print)/n()`, photography =
with(ann, plot(year, other, type = "l", col = "red"))
with(ann, lines(year, painting, type = "l", col = "blue"))
with(ann, lines(year, print, type = "l", col = "green"))
with(ann, lines(year, photography, type = "l", col = "yellow"))
with(ann, lines(year, sculpture, type = "l", col = "grey"))
legend("topright", legend = c("other", "painting", "print", "photography", "sculpture"),
       col = c("blue", "red", "green", "yellow", "grey"), lty = 1, lwd = 2)
title("Percentage of Sales by Genre in Different Years")
```




```
library(ggplot2)
ann = artsales %>%
  select(year, Genre___Others, Genre___Print, Genre___Photography, Genre___Painting, Genre__
  mutate(genre = if_else(Genre___Sculpture == 1, "Sculpture", if_else(Genre___Print == 1, "Pr
ann = ann[!is.na(ann$genre), ]
ggplot(ann, aes(x = year, fill = genre)) +
  geom_histogram(position = "fill")
```

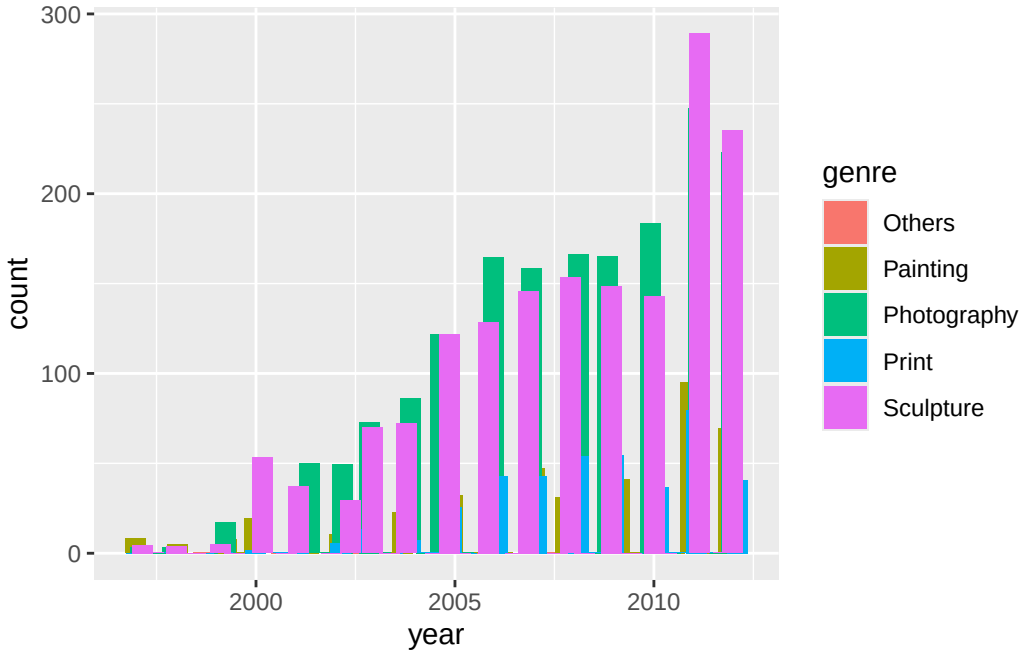
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
Warning: Removed 70 rows containing missing values or values outside the scale range
(`geom_bar()`).
```




```
ggplot(ann, aes(x = year, fill = genre)) +
  geom_histogram(position = "jitter")
```

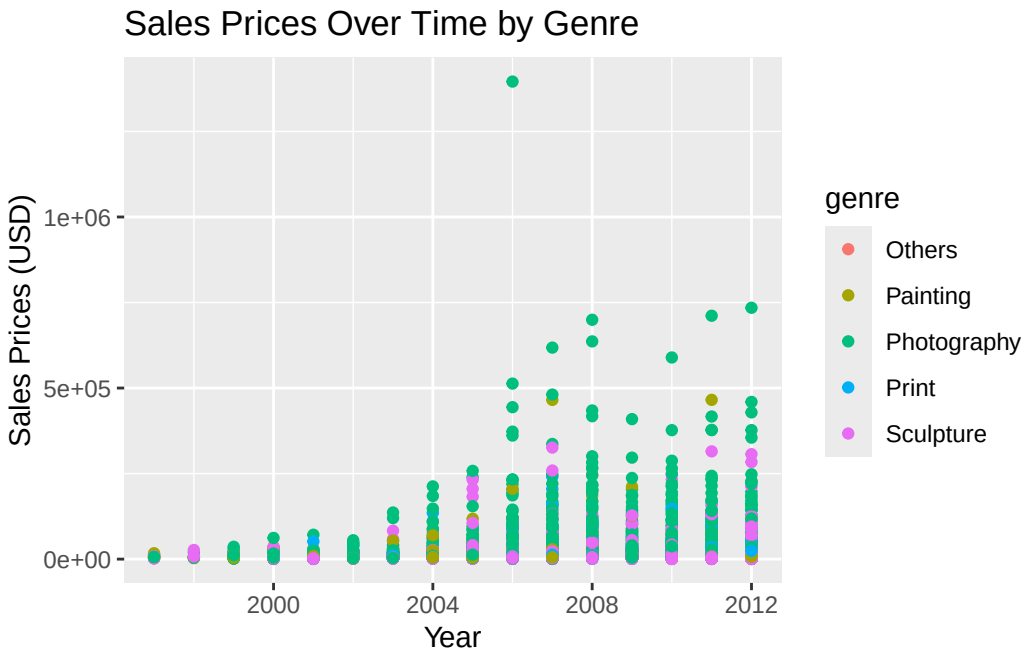
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



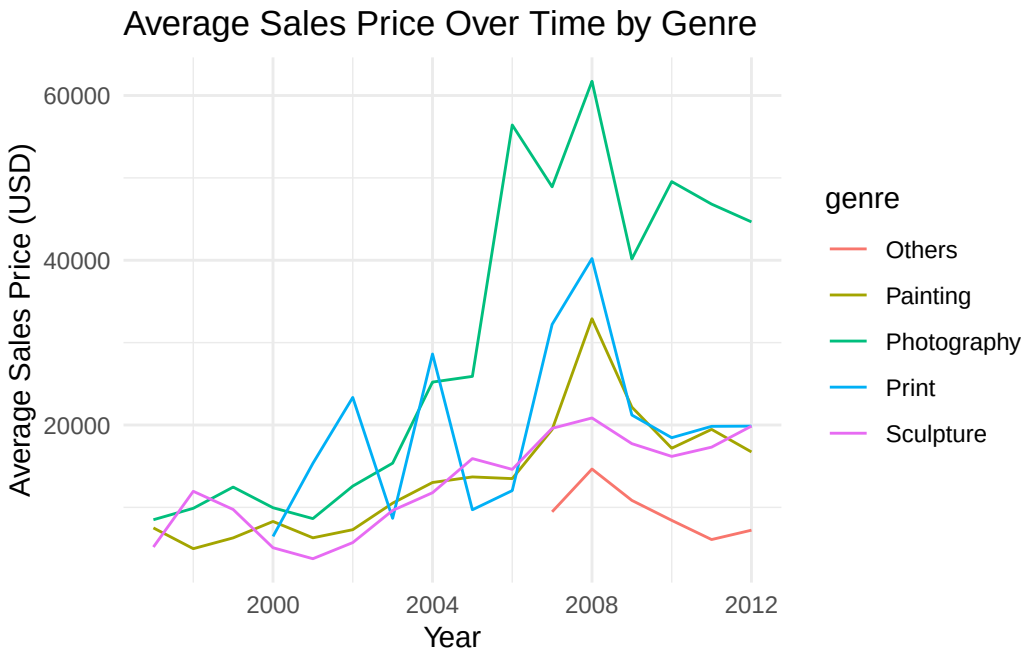
c. Although for all genres of sales, prices all have rising trends in these years, we can see in the first plot, thre are lots of green points with higher prices in latter years. And in the latter figure, the mean value can help us to summarize it better. we can see that genres affect the price by rising the amount of high price genres like “Photography”.

```
ann = artsales %>%
  select(price_usd, year, Genre___Others, Genre___Print, Genre___Photography, Genre___Paintir
  mutate(genre = if_else(Genre___Sculpture == 1, "Sculpture", if_else(Genre___Print == 1, "Pr
  select(year, price_usd, genre)

ggplot(ann, aes(x = year, y = price_usd, color = genre)) +
  geom_point() +
  labs(title = "Sales Prices Over Time by Genre",
       x = "Year",
       y = "Sales Prices (USD)")
```




```
ggplot(ann, aes(x = year, y = price_usd, color = genre)) +
  stat_summary(fun = mean, geom = "line") +
  labs(title = "Average Sales Price Over Time by Genre",
       x = "Year",
       y = "Average Sales Price (USD)") +
  theme_minimal()
```


Average Sales Price Over Time by Genre
Average Sales Price (USD)
60000
40000
20000
2000
2004
2008
2012
Year
genre
Others
Painting
Photography
Print
Sculpture