STATS 506 HW 4

ZHAO Shengchun

Github URL:

https://github.com/ZHAOShengchun67734538/STAT-506-HW-4

Question 1

(a)

```r
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 becom
```

```r
library(nycflights13)
### (a) ###
# Departure Delay Table
dep_delay = flights %>%
              select(origin, dep_delay) %>%
              group_by(origin) %>%
              summarise(mean_dep = mean(dep_delay,na.rm = TRUE),
                        median_dep = median(dep_delay, na.rm=TRUE),
                        num = n()) %>%
              ungroup() %>%
              filter(num >= 10) %>%
              rename(faa = origin) %>%
              left_join(airports,by = "faa") %>%
              select(name, mean_dep, median_dep) %>%
              arrange(desc(mean_dep))

dep_delay %>% print(n = count(.))
```

```
# A tibble: 3 x 3
  name                mean_dep median_dep
  <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 Delay Table
arr_delay = flights %>%
              select(dest, arr_delay) %>%
              group_by(dest) %>%
              summarise(mean_arr = mean(arr_delay,na.rm = TRUE),
                        median_arr = median(arr_delay, na.rm=TRUE),
                        num = n()) %>%
              ungroup() %>%
              filter(num >= 10) %>%
              rename(faa = dest) %>%
              left_join(airports,by = "faa") %>%
              mutate(name = coalesce(name, faa)) %>%
              select(name, mean_arr, median_arr) %>%
              filter(!is.na(mean_arr)) %>%
              filter(!is.na(median_arr)) %>%
              arrange(desc(mean_arr))

arr_delay %>% print(n = count(.))
```

```
# A tibble: 102 x 3
    name                                   mean_arr median_arr
    <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
```

(b)

```
### (b) ###
model = flights %>%
          left_join(planes, by = "tailnum") %>%
          select(c(model, distance, air_time)) %>%
          mutate(speed = distance/(air_time/60)) %>%
          group_by(model) %>%
          summarise(ave_mph = mean(speed, na.rm = TRUE),
                    num_flights = n()) %>%
          ungroup() %>%
          arrange(desc(ave_mph)) %>%
          head(n = 1L)
model
```

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

Question 2

```
library(tidyverse)
nnmaps = read_csv("C:/Users/z1883/Desktop/chicago-nmmaps.csv")
```

```
Rows: 1461 Columns: 11
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (3): city, season, month
dbl  (7): temp, o3, dewpoint, pm10, yday, month_numeric, year
date (1): date

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
nnmaps %>% head()
```

```
# A tibble: 6 x 11
  city  date        temp    o3 dewpoint  pm10 season  yday month month_numeric
  <chr> <date>     <dbl> <dbl>    <dbl> <dbl> <chr>  <dbl> <chr>         <dbl>
1 chic  1997-01-01  36    5.66    37.5  13.1  Winter     1 Jan               1
2 chic  1997-01-02  45    5.53    47.2  41.9  Winter     2 Jan               1
3 chic  1997-01-03  40    6.29    38    27.0  Winter     3 Jan               1
4 chic  1997-01-04  51.5  7.54    45.5  25.1  Winter     4 Jan               1
5 chic  1997-01-05  27   20.8     11.2  15.3  Winter     5 Jan               1
6 chic  1997-01-06  17   14.9      5.75  9.36 Winter     6 Jan               1
# i 1 more variable: year <dbl>
```

```
# find the range of the year in nnmaps
year.range <- nnmaps$year %>% unique
year.range
```

```
[1] 1997 1998 1999 2000
```

```
#' Title:
#' Calculate average temperature for a given month and year
#' @param month numeric or string, such as 5 or May
#' @param year numeric
#' @param data data set have the numeric month, tempreture, year
#' @param average_fn function to calculate mean tempreture,
#' the default is mean()
#' @param celsius logical, TRUE/FALSE, if TRUE, the tempreture
#'               will be converted to  celsius, the default is
#'               FALSE, which is fahrenheit
#' @return the average tempreture
```

```
#####################################################################

get_temp=function(month, year, data, average_fn=mean, celsius=FALSE)
{
  # Check the validity of month
  if(month %>% is.numeric())
  {
    if(month != (month %>% as.integer()))
    {
      stop("This month is not an integer, please try again.")
    }
    if((month < 1) | (month > 12))
    {
      stop("This month is out of range, please try again.")
    }
    month.input = month

  }else if(month %>% is.character())
  {
    month.set = c("January", "February", "March", "April",
                  "May", "June", "July","August",
                  "September", "October", "November", "December")
    month.input <- grep(month, month.set, ignore.case=TRUE)
    if(month.input %>% identical(integer(0)))
    {
      stop("This month is not valid character, please try again.")
    }

  }else
  {
    stop("Input month must be numeric or character.")
  }


  # Check the validity of year
  if(!(year %>% is.numeric()))
  {
    stop("This year is not a numeric, please try again.")
  }
  if(year != (year %>% as.integer()) )
  {
    stop("This year is not an integer, please try again.")
  }
  if((year<year.range[1])|(year>year.range[year.range %>% length]))
  {
    stop("This year is out of range, please try again.")
  }


  # Check the validity of function
  if (!(average_fn %>% is.function))
  {
    stop("average_fn must be a function")
  }


  # Check the validity of celsius
  if (!(celsius %>% is.logical())) {
    stop("celsius must be a logical")
  }

  data %>%
    select(temp, month_numeric, year) %>%
    rename(nnmaps.year = year) %>%
    filter(month_numeric==month.input, nnmaps.year==year) %>%
    summarize(ave.temp = average_fn(temp)) %>%
    as.numeric -> result

  if(celsius == TRUE)
  {
    cel.result = (result - 32)*(5/9)
    return(cel.result)
  }else{
    return(result)
  }

}
```

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

```
[1] 49.8
```

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

```
[1] 9.888889
```

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

```
[1] 55
```

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

```
Error in get_temp(13, 1998, data = nnmaps): This month is out of range, please try again.
```

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

```
Error in get_temp(2, 2005, data = nnmaps): This year is out of range, please try again.
```

```r
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
```

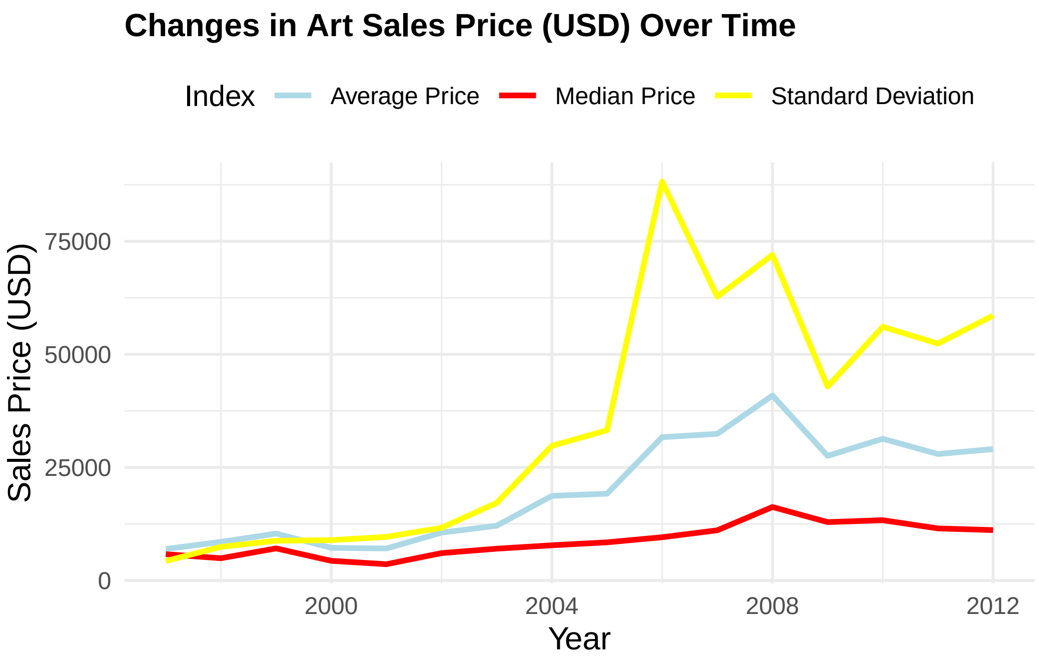
Question 3

(a)

```r
art = read.csv("C:/Users/z1883/Desktop/df_for_ml_improved_new_market.csv")
library(dplyr)
library(ggplot2)
library(tidyverse)

yearly.summary = art %>%
  group_by(year) %>%
  summarize(
    ave.price = mean(price_usd, na.rm = TRUE),
    median.price = median(price_usd, na.rm = TRUE),
    sd.price = sd(price_usd, na.rm = TRUE)
  ) %>%
  ungroup()
```

```
# Plotting the average median, and sd sales price over time
ggplot(yearly.summary, aes(x = year)) +
  geom_line(aes(y = ave.price, color="Average Price"),linewidth = 1) +
  geom_line(aes(y = median.price, color="Median Price"),linewidth = 1) +
  geom_line(aes(y = sd.price, color="Standard Deviation"),linewidth = 1)+
  labs(
    title = "Changes in Art Sales Price (USD) Over Time",
    x = "Year",
    y = "Sales Price (USD)",
    color = "Index"
  ) +
  theme_minimal() +
  scale_color_manual(values = c("Average Price" = "lightblue",
                                "Median Price" = "red",
                                "Standard Deviation" = "yellow"))+
  theme(
    plot.title = element_text(size = 12, face = "bold"),
    axis.title = element_text(size = 12),
    legend.position = "top"
  )
```



From the plot, we can see there exist a change of sales price in USD overtime. The average price shows a significant increase starting from around 2000, peaking in the year 2008, and then declining afterward. The median price also rose from around 2000 to a peak in 2008, but it remains much lower than the average price throughout the period. Compared with the mean price, median growth has been more modest. We can also find the variation of price becoming huge from 2004, which means the majority of art sales were at lower prices, with a few high-priced outliers driving up the average price and variation at that period.

(b)

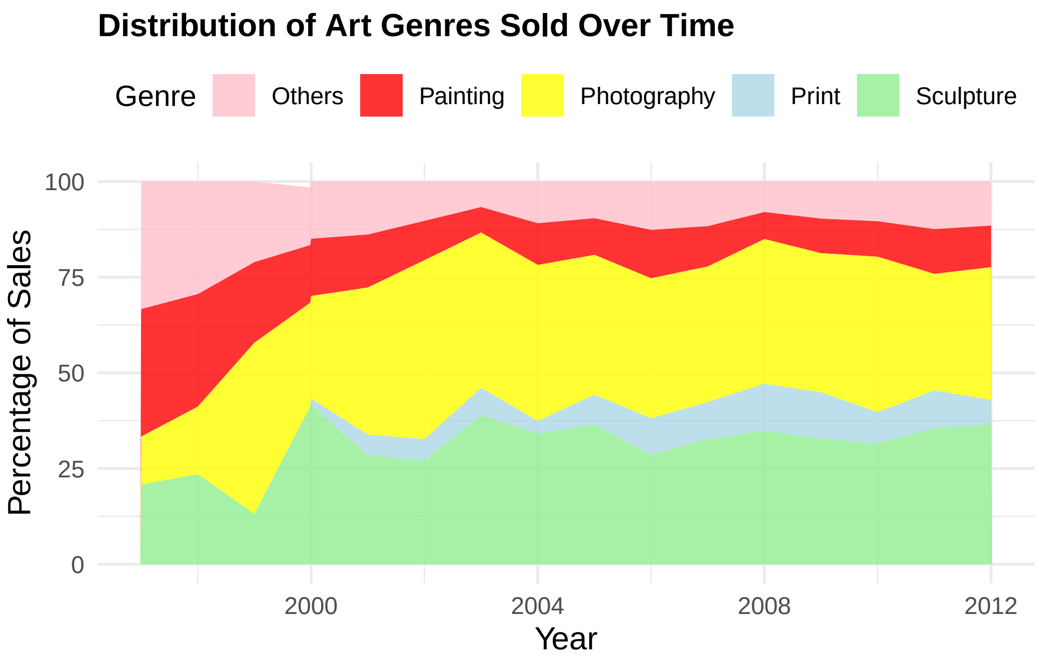
```
# Change the column names
colnames(art)[102] = "Photography"
colnames(art)[103] = "Print"
colnames(art)[104] = "Sculpture"
colnames(art)[105] = "Painting"
colnames(art)[106] = "Others"

# We combine the five binary variables into one column
art.long = art %>%
  pivot_longer(
    cols = c("Photography","Print",
             "Sculpture","Painting",
             "Others"),
    names_to = "Genre",
```

```r
    values_to = "Is_Genre"
  ) %>%
  # Keep only rows where the genre is present
  filter(Is_Genre == 1) %>%
  # Remove the binary indicator column
  select(-Is_Genre)

# Summarize the count of each genre per year
yearly.genre = art.long %>%
  group_by(year, Genre) %>%
  summarize(count = n(), .groups = "drop") %>%
  group_by(year) %>%
  mutate(percentage = count / sum(count) * 100) %>%
  ungroup()
```

```r
# Plot the distribution of genre across years
ggplot(yearly.genre, aes(x = year, y = percentage, fill = Genre)) +
  geom_area(alpha = 0.8) +
  labs(
    title = "Distribution of Art Genres Sold Over Time",
    x = "Year",
    y = "Percentage of Sales",
    fill = "Genre"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 12, face = "bold"),
    axis.title = element_text(size = 12),
    legend.position = "top"
  ) +
  scale_fill_manual(values = c(
    "Photography" = "yellow",
    "Print" = "lightblue",
    "Sculpture" = "lightgreen",
    "Painting" = "red",
    "Others" = "pink"
  ))
```

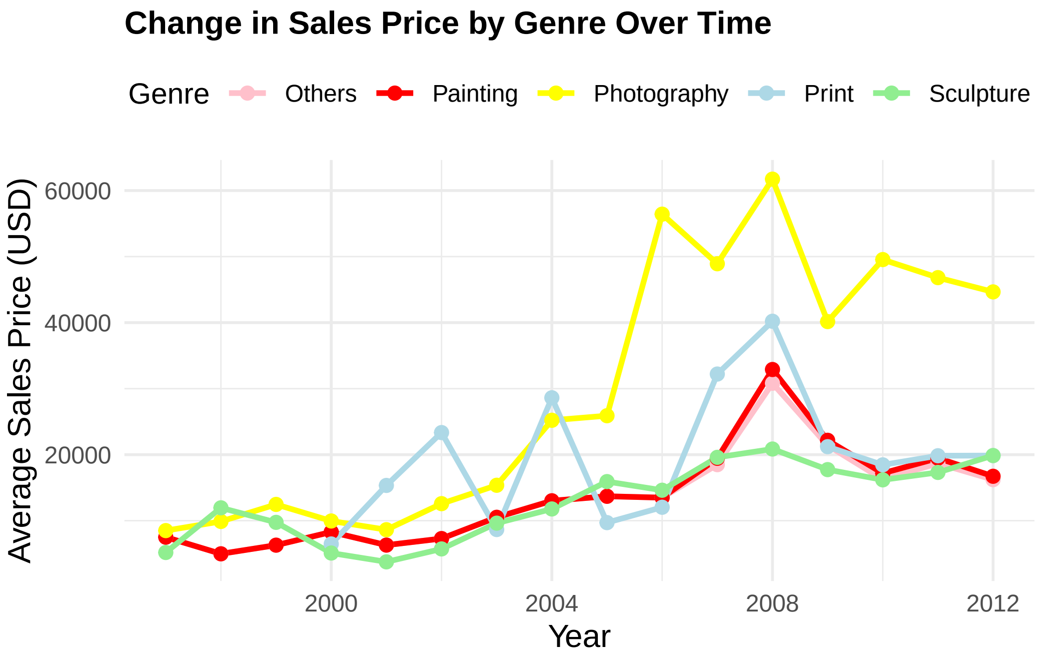


Until about 2000, there was no market share for PRINT art, and the sales share of the other four categories of art was roughly the same. After 2000, PRINT artworks appeared, and their market share remained small; SCULPTURE & PHOTOGRAPHY artworks had a larger demand and occupied the main market share; PAINTING & OTHERS artworks had a relatively smaller market share, and the market share between them was similar. To sum up, after 2000, the market share of all kinds of artworks changed and stabilized without much change.

(c)

```
genre.price = art.long %>%
  group_by(year, Genre) %>%
  summarize(ave.price = mean(price_usd, na.rm = TRUE), .groups = "drop")
```

```
# Plot the change in average sales price over time for each genre
ggplot(genre.price, aes(x = year, y = ave.price, color = Genre)) +
  geom_line(linewidth = 1) +
  geom_point(size = 2) +
  labs(
    title = "Change in Sales Price by Genre Over Time",
    x = "Year",
    y = "Average Sales Price (USD)",
    color = "Genre"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 12, face = "bold"),
    axis.title = element_text(size = 12),
    legend.position = "top"
  ) +
  scale_color_manual(values = c(
    "Photography" = "yellow",
    "Print" = "lightblue",
    "Sculpture" = "lightgreen",
    "Painting" = "red",
    "Others" = "pink"
  ))
```



Photography:

It has the highest variability in average price over time, with significant peaks around 2006-2008, the maximum reaching over $60,000. It reflects a high demand of the market starting in 2004.

Print:

The average price has been fluctuating within a certain range. It may be that the market has been fluctuating, or it may be that the pricing range of Print artworks is relatively large, thus affecting the average value.

Painting & Others:

The two lines largely coincided and grew slowly until 2008, when prices fell back after reaching a small peak.

Sculpture:

The price of sculpture over the years has been lower than other categories. Also, the kind of slow price growth, indicating that the selling price is very stable, slow growth may be the reason for inflation, It is also possible that the buyers for such art are very small and fixed.