Exploratory Analysis of Video Game Sales Data

Nane Mambreyan

```
library(dplyr)
library(ggplot2)
library(gridExtra)
library(grid)
library(reshape2)
library(reticulate)
library(scales)
library(tidyr)
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
```

Part 1

```
# reading the dataset
games <- read.csv('vgsales.csv')

# finding character columns where there might be na
character_columns <- sapply(games, is.character)

# applying na_if to character columns
# setting na if the vlaue is 'NA'
games[character_columns] <- games[character_columns] %>%
  mutate_all(~na_if(., "N/A"))

# removing duplicate and na rows
games <- games %>%
  distinct() %>%
  na.omit()

str(games)
```

```
## 'data.frame':    16291 obs. of  11 variables:
##  $ Rank        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Name        : chr  "Wii Sports" "Super Mario Bros." "Mario Kart Wii" "Wii Sports Resort" ...
##  $ Platform    : chr  "Wii" "NES" "Wii" "Wii" ...
##  $ Year        : chr  "2006" "1985" "2008" "2009" ...
##  $ Genre       : chr  "Sports" "Platform" "Racing" "Sports" ...
##  $ Publisher   : chr  "Nintendo" "Nintendo" "Nintendo" "Nintendo" ...
```

```
##  $ NA_Sales    : num  41.5 29.1 15.8 15.8 11.3 ...
##  $ EU_Sales    : num  29.02 3.58 12.88 11.01 8.89 ...
##  $ JP_Sales    : num  3.77 6.81 3.79 3.28 10.22 ...
##  $ Other_Sales : num  8.46 0.77 3.31 2.96 1 0.58 2.9 2.85 2.26 0.47 ...
##  $ Global_Sales: num  82.7 40.2 35.8 33 31.4 ...
##  - attr(*, "na.action")= 'omit' Named int [1:307] 180 378 432 471 608 625 650 653 712 783 ...
##   ..- attr(*, "names")= chr [1:307] "180" "378" "432" "471" ...
```

• Let's filter the top 5 publishers of PC games. Let's assume the publisher is better, if it publishes more games.

```
# filter by platforms then group by publishers
# create a new df which contains the count of
# PC games for each publisher
# put them in descending order and
# show top 5 Publishers
# by pull i convert the result into a vector

top_5_publishers <- games %>%
  filter(Platform == 'PC') %>%
  group_by(Publisher) %>%
  summarise(GameCount = n()) %>%
  arrange(desc(GameCount)) %>%
  head(5) %>%
  select(Publisher) %>%
  pull()
# print(top_5_publishers)

# from the original data I get only the rows
# where the publisher is one of the top 5
filtered_data <- games %>%
  filter(Platform == 'PC' & (Publisher %in% top_5_publishers))
# head(filtered_data)

# recheck
print(unique(filtered_data$Publisher))
```

```
## [1] "Electronic Arts"      "Activision"           "Take-Two Interactive"
## [4] "Sega"                 "Ubisoft"
```

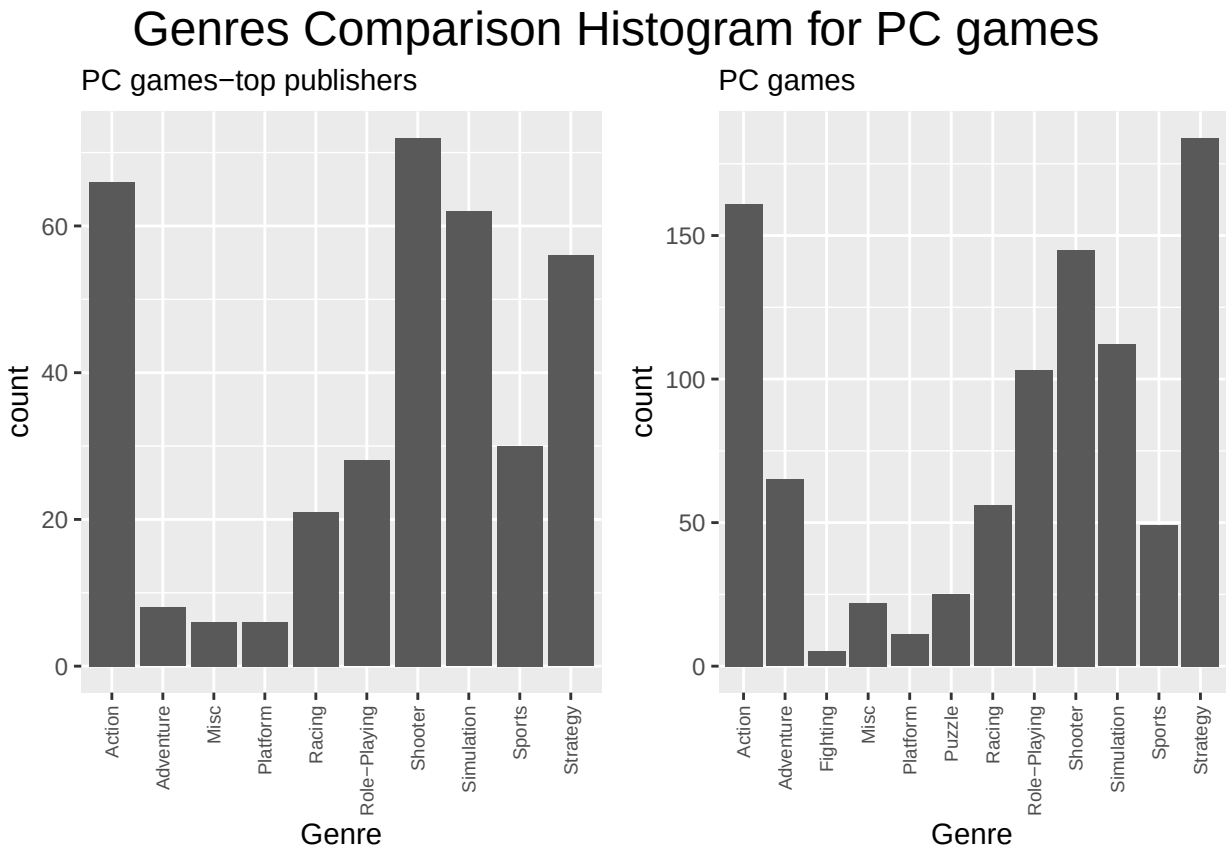
```
print(unique(filtered_data$Platform))
```

```
## [1] "PC"
```

• Creating a histogram that represents both filtered and not filtered dataset Genres. Comparing differences between filtered and not filtered samples. Plot next to each other.

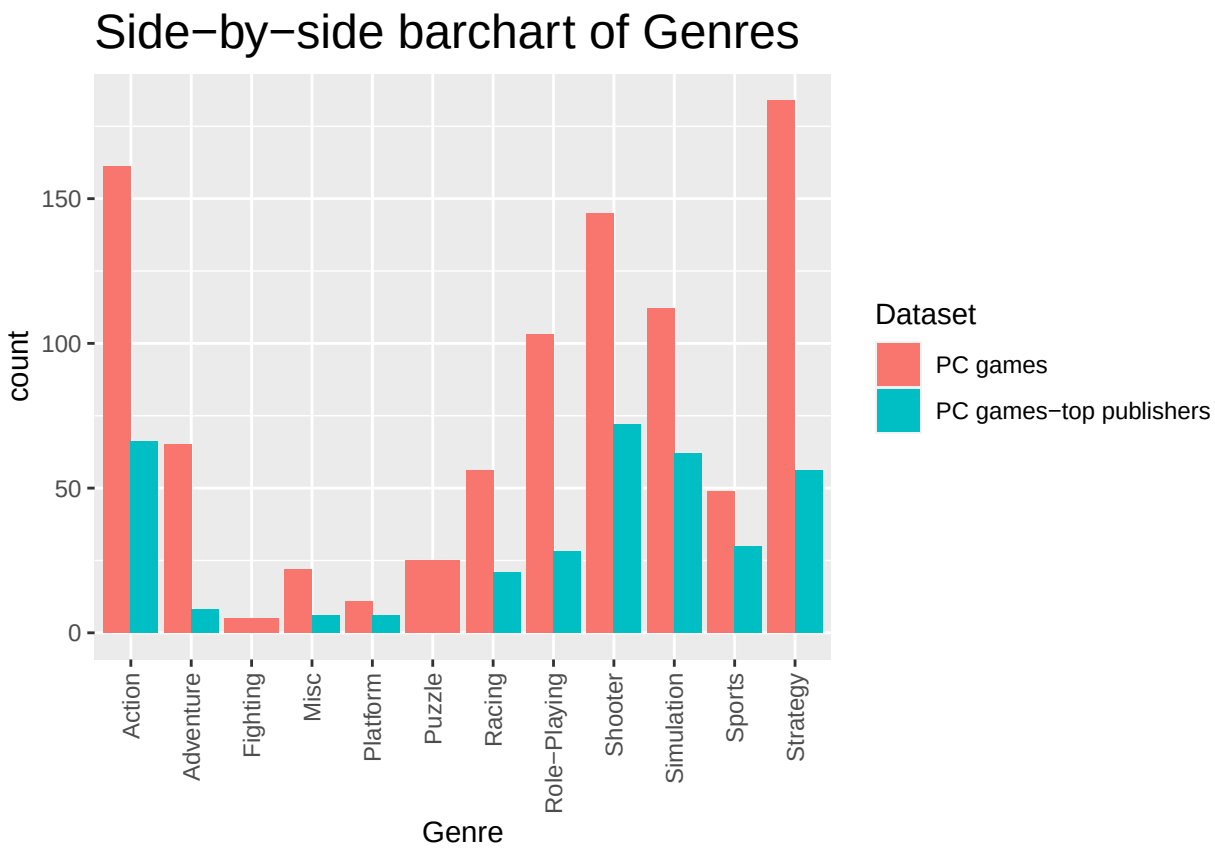
```
# create a bar chart with Genre as x and count as y
# some visual modifications
p1 <- ggplot(data = filtered_data) +
  geom_bar(aes(x = Genre)) +
  ggtitle('PC games-top publishers') +
```

```
  theme(plot.title = element_text(size = 11),
        axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1, size = 7))

# first filter pc games then
# create a bar chart with Genre as x and count as y
# some visual modifications
p2 <- games %>% filter(Platform == 'PC') %>% ggplot() +
  geom_bar(aes(x = Genre)) +
  ggtitle('PC games') +
  theme(plot.title = element_text(size = 11),
        axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1, size = 7))

# Putting side-by-side plots and giving overall title
grid.arrange(p1, p2, ncol = 2,
top = textGrob("Genres Comparison Histogram for PC games", gp = gpar(fontsize = 18)))
```




```
# Or to make it in one graph side-by-side
# binding two datasets with rows
# before binding adding new column in each dataset
# in games for pcs the value of Dataset column will be 'PC games' everywhere
# in filtered_data it will be 'PC games-top publishers'
combined_data <- rbind(
  games  %>% filter(Platform == 'PC') %>% mutate(Dataset = 'PC games'),
  filtered_data %>% mutate(Dataset = 'PC games-top publishers'))
```

```r
# plotting in one graph, side by side
ggplot(combined_data, aes(x = Genre, fill = Dataset)) +
  geom_bar(position = 'dodge') +
  ggtitle('Side-by-side barchart of Genres') +
  theme(plot.title = element_text(size = 18),
        axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```

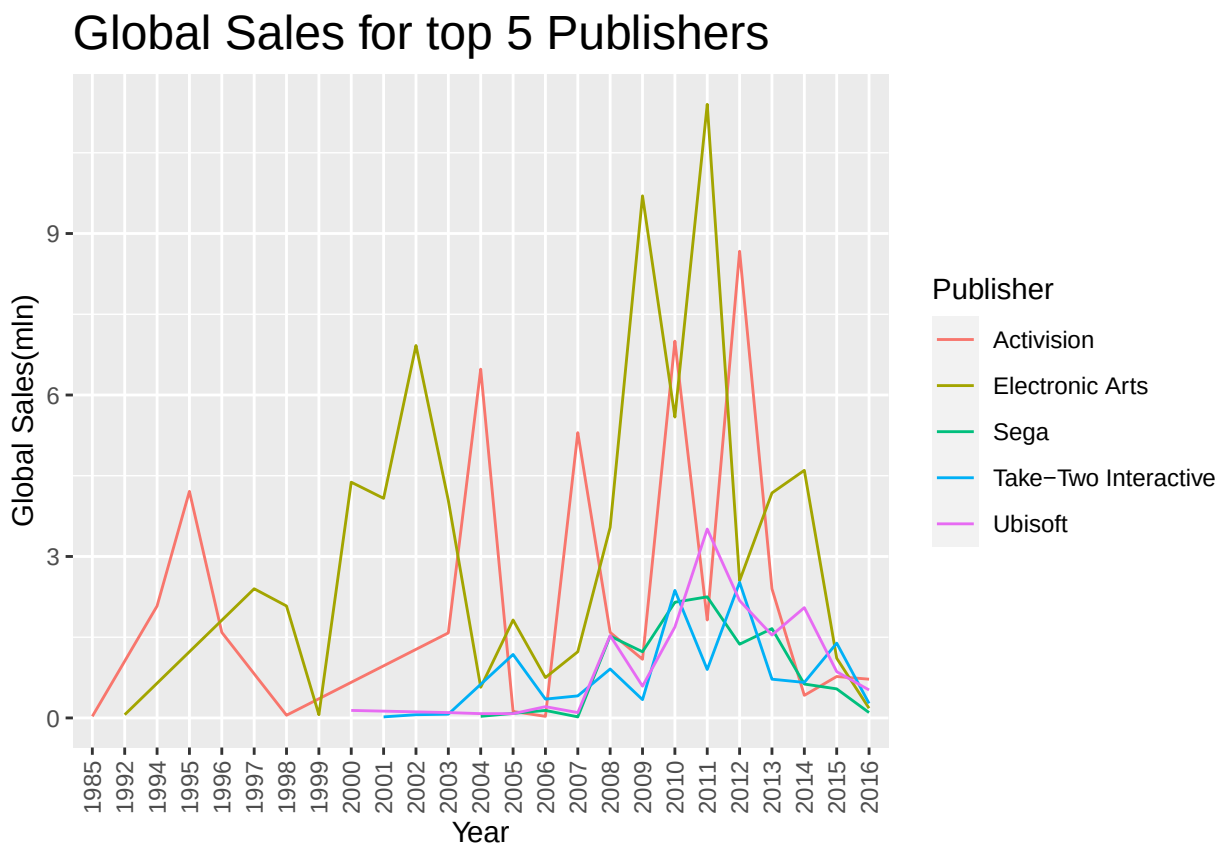


In the first case we have two separate graphs In the second case the same info is conveyed with side by side barchart, which is better for comparison between two categories. For PC games the count of games is always more than for PC games published by top publishers only, which was quite expected. Then, there are some genres of games that haven't been published by top publishers, like puzzle and fighting, hence we only see the bar for PC games(red). As for the same graph, we can pay attention to the scale of y axis to make reasonable comparison.

- Creating a line chart for the top 5 publishers by accumulating global sales for the same year.

```r
# Accumulating by years, hence grouping based on
# Publisher and Year value and summing up the Global Sales values in each group
filtered_data_ <- filtered_data %>%
  ungroup() %>%
  group_by(Publisher, Year) %>%
  summarize(Global_Sales = sum(Global_Sales), .groups = 'drop') %>%
  ungroup()


# specifying x and y, giving each publisher unique color,
# setting up separate independent lines for each publisher with group param
```

```
# specifying geom to be line, changing labels
# changing the visuals of x axis
ggplot(filtered_data_, aes(x = Year, y = Global_Sales, color = Publisher, group = Publisher)) +
  geom_line() + labs(y = 'Global Sales(mln)', title = 'Global Sales for top 5 Publishers') +
  theme(plot.title = element_text(size = 18),
        axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```

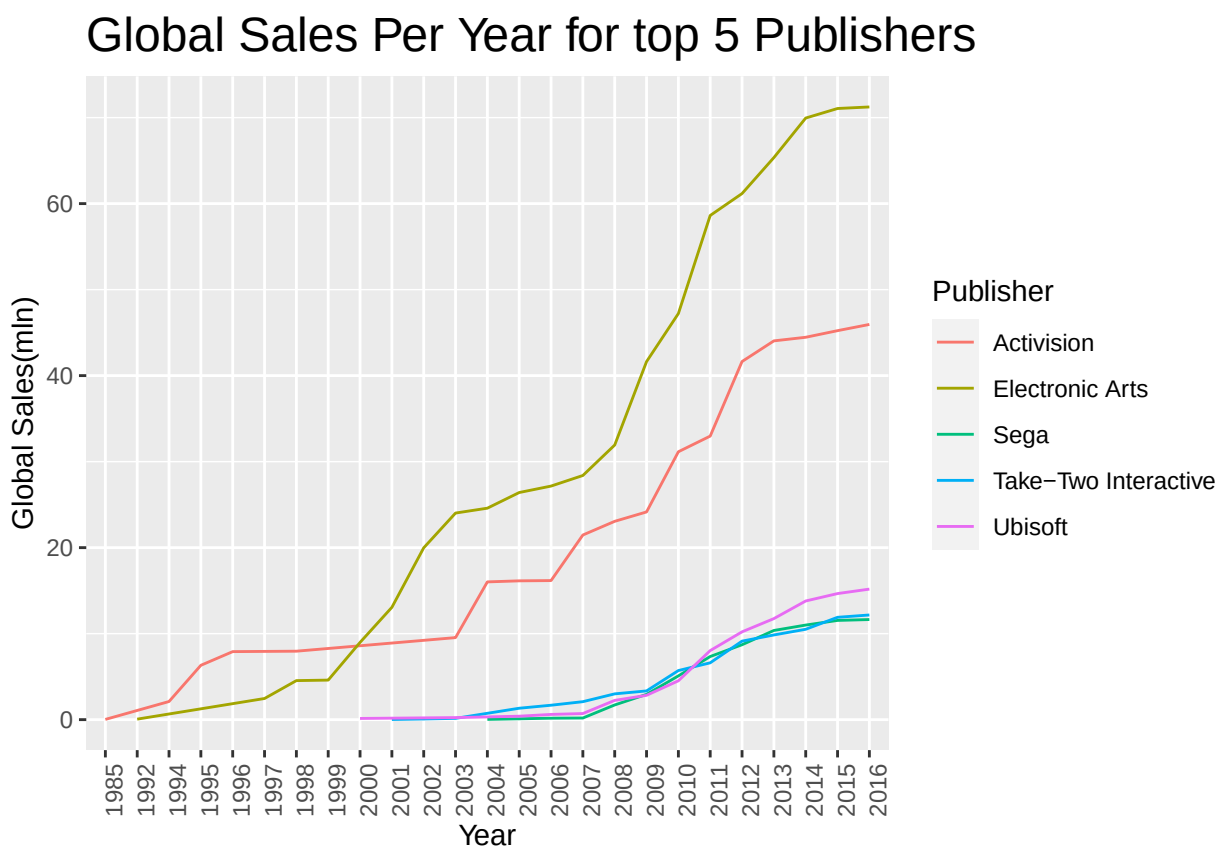


Based on this graph we can get info about when(Year) was the pick for global sales for each Publisher(color) We can extract some patterns of increase and decrease in global sales based on Year and Publisher We get the overall picture of global sales from 1985 througout 2016 for each Publisher

• In the last step, Global Sales refers to the year it produces. The below plot is over all time Global sales for 5 publishers. This means, for each year I also consider its previous years.

```
# I get the top 5 PC games publishers
# group games based on their publisher, then
# arrange the entries inside each Publisher category
# in an ascending order with respect to their Year
# Then for the resulting categories, I get the sum
# of Global_sales for the current and preceding years
# lastly, I group the whole thing based on Years

global_sales_per_year <- filtered_data_ %>%
  group_by(Publisher) %>%
  arrange(Year) %>%
  mutate(per_year = cumsum(Global_Sales)) %>%
  group_by(Year)
```

```
# specifying aesthetics to be Year and newly created column
#  setting up color based on Publisher info
# setting up separate independent lines for each publisher with group param
# visual modifications (labs, theme...)
ggplot(global_sales_per_year, aes(x = Year, y = per_year, color = Publisher, group = Publisher)) +
  geom_line() + labs(y = 'Global Sales(mln)', title = 'Global Sales Per Year for top 5 Publishers') +
  theme(plot.title = element_text(size = 18),
        axis.text.x = element_text(angle = 90))
```



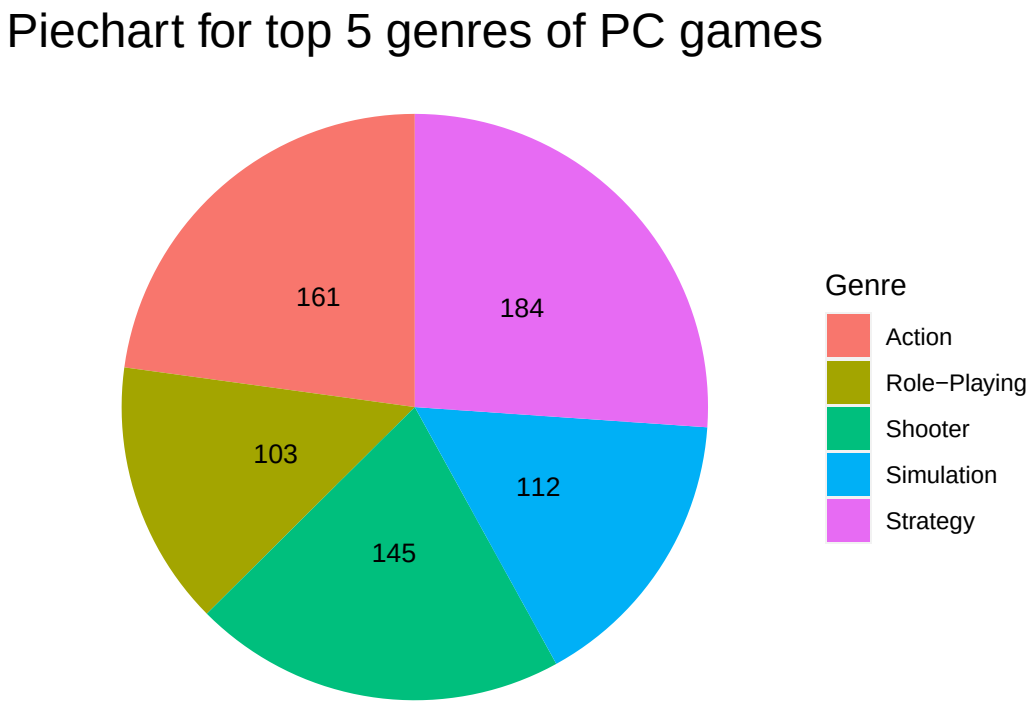
This shows increasing pattern year by year as expected since we have only positive values for global sales The highest sales at the end of the period has Electronic Arts then Activision then the last 3 categories(almost the same)

Part 2

• Creating a pie chart for all genres of PC games. Filtering only the top 5 based on count. Also creating another pie chart for other Platform and comparing the results.

```
# getting the list of genre names for the games that are for PC
# and that have one of top 5 genres
# the same logic as for top 5 Publishers...
top_genres_pc <- games %>%
  filter(Platform == 'PC') %>%
  group_by(Genre) %>%
```

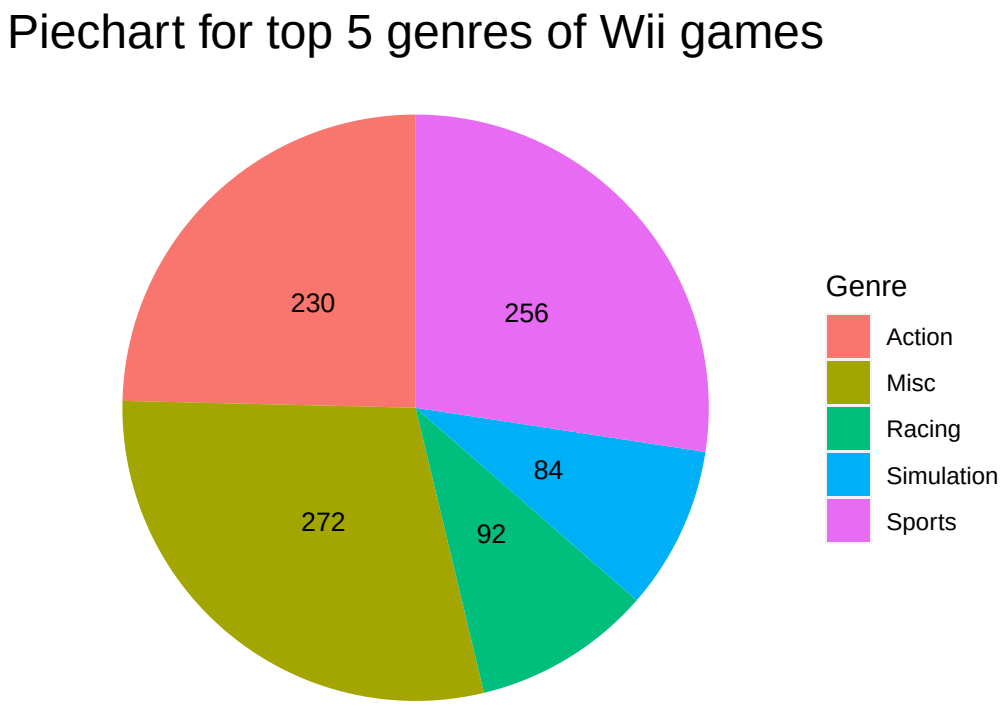
```r
  summarise(GenreCount = n()) %>%
  arrange(desc(GenreCount)) %>%
  head(5) %>%
  select(Genre) %>%
  pull()


# getting whole dataset based on top genres and PC as a platform
top_genres_data <- games %>%
  filter(Platform == 'PC', (Genre %in% top_genres_pc))

# specifying aesthetics, creating barplot then chainging the coordinate
# plane to be polar to get a pie chart , specifying that y represents theta now
# getting the count of each category from base R using ..count..
ggplot(data = top_genres_data, aes(x = "", fill = Genre)) +
  geom_bar() + coord_polar(theta = 'y') +
  labs(title = 'Piechart for top 5 genres of PC games', x = '', y = '')  +
  geom_text(stat = 'count', aes(label = ..count..),
            position = position_stack(vjust = 0.5), size = 3.5) +
  theme(plot.title = element_text(size = 18, vjust = -2, hjust = 0.5),
        panel.background = element_blank(),
        axis.ticks = element_blank(),
        axis.text = element_blank())
```



The sector with the largest area is purple one which stands for Strategy, hence it is the dominating Genre out of top 5 genres based on their count. The sector in green-yellowish color, which stands for Role Playing

appears to be the least.

```
# getting the list genre names for the games that are for wii
# and that have one of top 5 genres
# the same logic as for top 5 Publishers...
top_genres_wii <- games %>%
  filter(Platform == 'Wii') %>%
  group_by(Genre) %>%
  summarise(GenreCount = n()) %>%
  arrange(desc(GenreCount)) %>%
  head(5) %>%
  select(Genre) %>%
  pull()


# from the original data I get only the rows
# where the genre is one of the top 5
top_wii <- games %>%
  filter(Platform == 'Wii', (Genre %in% top_genres_wii))
# head(top_wii)


# specifying aesthetics, creating barplot then chainging the coordinate
# plane to be polar to get a pie chart , specifying that y represents theta now
ggplot(data = top_wii, aes(x = "", fill = Genre)) +
  geom_bar() + coord_polar(theta = 'y') +
  labs(title = 'Piechart for top 5 genres of Wii games', x = '', y = '') +
  geom_text(stat = 'count', aes(label = ..count..),
            position = position_stack(vjust = 0.5), size = 3.5) +
  theme(plot.title = element_text(size = 18, vjust = -2, hjust = 0.5),
        panel.background = element_blank(),
        axis.ticks = element_blank(),
        axis.text = element_blank())
```



As compared to the above piechart for PC, here(for wii) the dominating Genre is Misc which was not present as one of top 5 genres for pc games. Here the least one is Racing, which wasn't present previously, too.

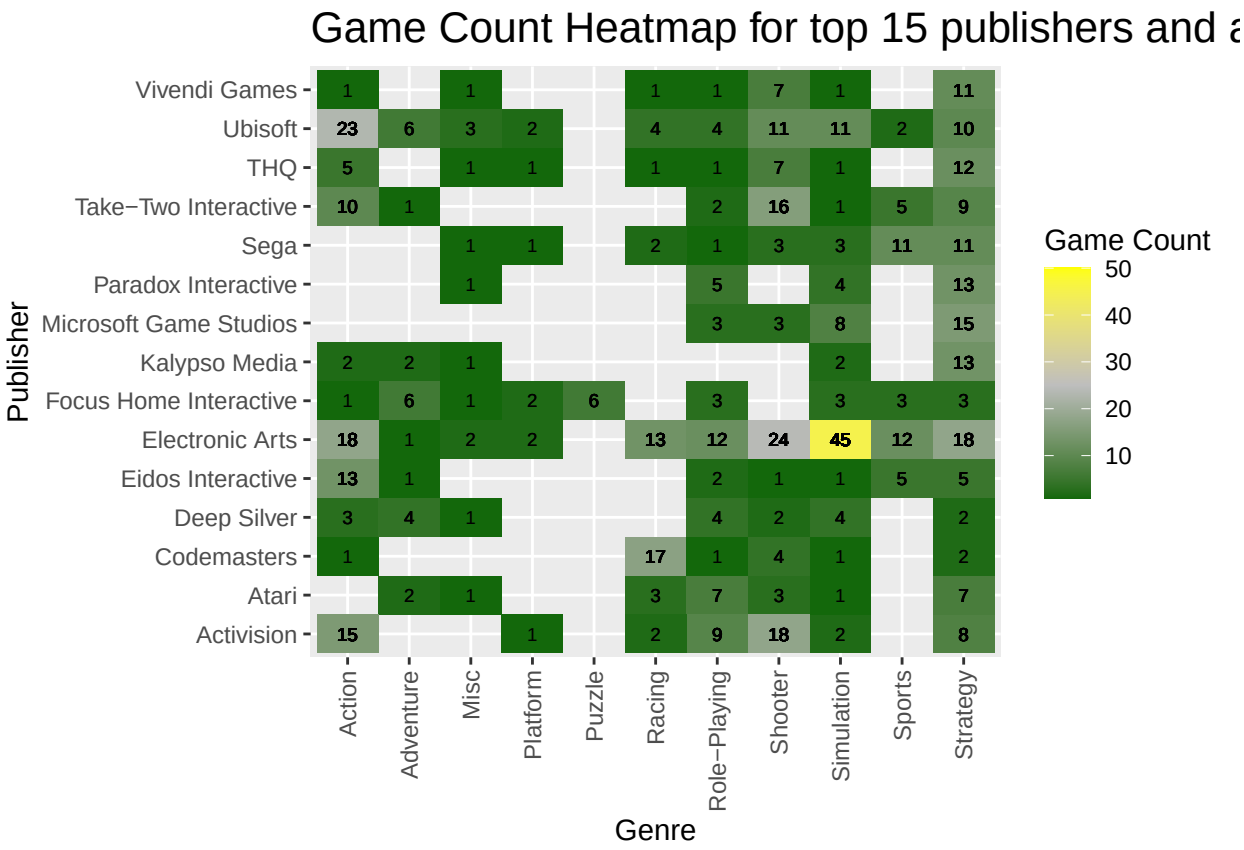
One of the conclusions we can make is that from platform to platform the top 5 genres change.

For this part, I am going to work on the top 15 publishers of PC games. Let's assume the publisher is better, if it published more games.

- Creating a heatmap chart for the same 15 publishers and all Genres.

```
# Already explained above, doing the same for 15 best Publishers
top_15_publishers_PC <- games %>%
  filter(Platform == 'PC') %>%
  group_by(Publisher) %>%
  summarise(GameCount = n()) %>%
  arrange(desc(GameCount)) %>%
  head(15) %>%
  select(Publisher) %>%
  pull()

# getting the rows where platform is pc publisher is one of top 15
# and creating new column for the count of games for each publisher for each genre
filtered_data_15 <- games %>%
  filter(Platform == 'PC', Publisher %in% top_15_publishers_PC) %>%
  group_by(Publisher, Genre) %>%
  mutate(GameCount_per_Genre = n())
```

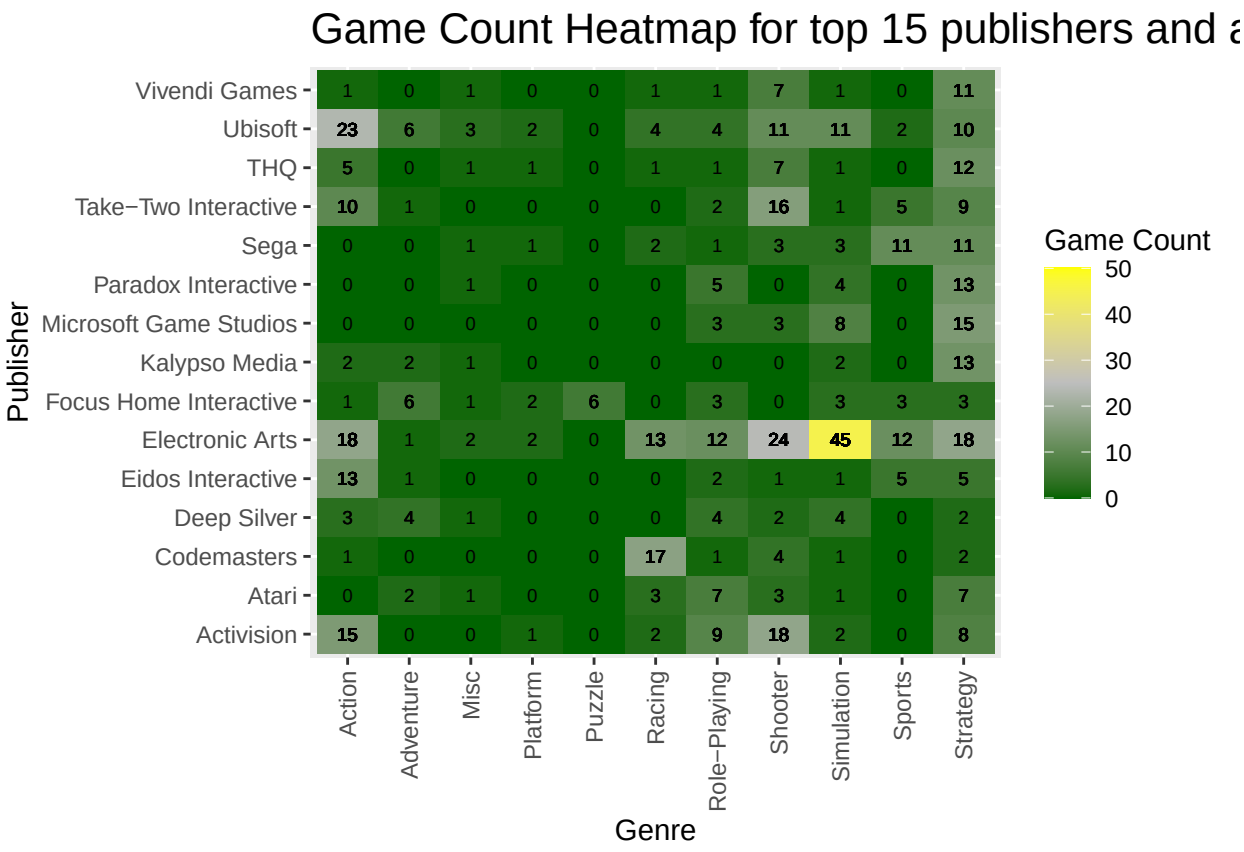
```r
# specifying Genre and Publisher as aesthetics, filling the colors with diverging values
# corresponding GameCount_per_Genre column, specifying title, angle of x labels,
# also specifying the color for lowest, highest and median values
ggplot(filtered_data_15, aes(x = Genre, y = Publisher, fill = GameCount_per_Genre)) +
  geom_tile() + ggtitle('Game Count Heatmap for top 15 publishers and all Genres') +
  geom_text(aes(label = GameCount_per_Genre), color = "black", size = 2.5) +
  scale_fill_gradient2(low = "darkgreen", high = "yellow", mid = "grey",
                       midpoint = 25, limit = c(1,50), name="Game Count") +
  theme(plot.title = element_text(size = 16),
        axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```



This shows that there is only one Publisher Electronic Arts which has published comparably many(45) games of Simulation Genre(hence light yellow color). There is also obvious pattern of dark green color, meaning that most of the publishers have few games per genre(if any). There are less grey tiles, for e.g. Ubisoft has 23 Action games, which is close to the median of our color range, hence it is grey. The empty parts in the heatmap indicate that the given publisher doesn't have any game of the given genre. For example, Activision doesn't have any Adventure games. To solve this problem,

```r
# creating a data frame with all combinations of Genre and Publisher to avoid empty tiles
all_combinations <- expand.grid(Genre = unique(filtered_data_15$Genre),
                                Publisher = unique(filtered_data_15$Publisher))

# merging filtered_data_15 with all_combinations to fill in missing values with 0
new_data <- merge(filtered_data_15, all_combinations, all = TRUE)
new_data[is.na(new_data)] <- 0  # replacing NAs with 0
```

```
# Only changing the color value range
ggplot(new_data, aes(x = Genre, y = Publisher, fill = GameCount_per_Genre)) +
  geom_tile() + ggtitle('Game Count Heatmap for top 15 publishers and all Genres') +
  geom_text(aes(label = GameCount_per_Genre), color = "black", size = 2.5) +
  scale_fill_gradient2(low = "darkgreen", high = "yellow", mid = "grey",
                       midpoint = 25, limit = c(0,50), name="Game Count") +
  theme(plot.title = element_text(size = 16),
        axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```

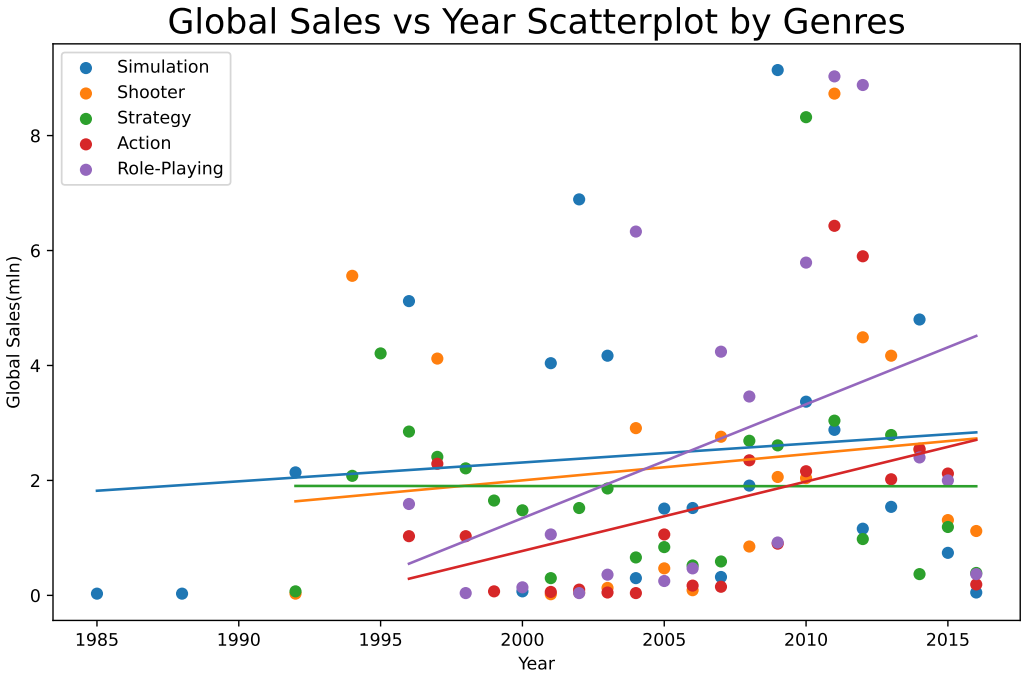


Part 3

• Creating a Scatter plot of Years and Global Sales by summing up all Global Sales over the year (all PC games and the top 5 Genres based on count). Also, drawing a linear regression line for all those Genres.

```
# Ignore FutureWarnings
warnings.simplefilter(action='ignore', category=FutureWarning)

games = pd.read_csv('vgsales.csv')
# games.head()
# games.info()

# Finding character columns where there might be 'NA' values
character_columns = games.select_dtypes(include=['object']).columns
```

```
# Applying 'na_if' to character columns by setting 'NA' to NaN
games[character_columns] = games[character_columns].apply(lambda col: col.str.replace('NA', 'NaN'))

# Removing duplicate rows
games = games.drop_duplicates()

# Removing rows with missing values
games = games.dropna()

# Reset the index
games.reset_index(drop=True, inplace=True)

# filtering PC games
my_filter = games.Platform == 'PC'
pc_games = games[my_filter]

#filtering top 5 genres
grouped_by_genre = pc_games.groupby(['Genre'])
top_5_genres = grouped_by_genre['Genre'].value_counts().nlargest(5).index
my_filter2 = pc_games.Genre.isin(top_5_genres)
filtered_data = pc_games[my_filter2]

filtered_data = filtered_data.groupby(['Year', 'Genre'])
filtered_data = filtered_data['Global_Sales'].sum().reset_index()


# getting unique genres
genres = filtered_data['Genre'].unique()

# creating scatterplot and adding regression lines for each genre separately
_ = plt.figure(figsize=(10, 6))
for genre in genres:
    genre_data = filtered_data[filtered_data['Genre'] == genre]
    x = genre_data['Year']
    y = genre_data['Global_Sales']
    _ = plt.scatter(x, y, label=genre)

    # calculating the linear regression line
    # fitting model
    slope, intercept = np.polyfit(x, y, 1)
    regression_line = slope * x + intercept
    _ = plt.plot(x, regression_line)

# setting labels, title and legend
_ = plt.xlabel('Year', fontsize = 10)
_ = plt.ylabel('Global Sales(mln)', fontsize = 10)
_ = plt.legend()
_ = plt.title('Global Sales vs Year Scatterplot by Genres', fontsize = 20)

_ = plt.show()
```

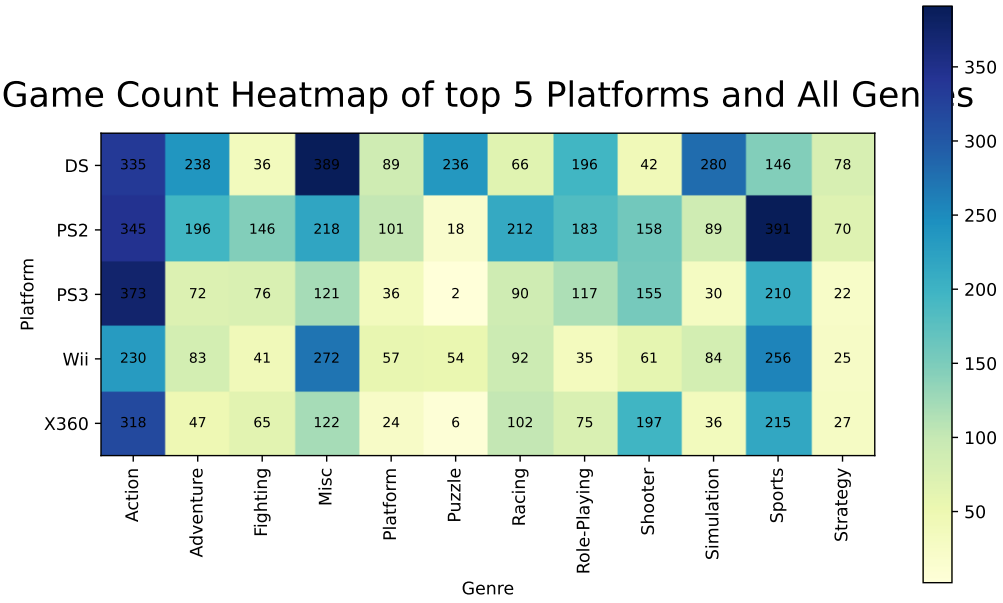


There is weak but positive correlation between Global Sales and Year for all genres of games. Still, the datapoints for each genre are quite scattered which is sign of small pearson correlation coef. This is similar to the idea in our lectures, meaning that depending on the context, having this kind of problem, we may want to group the points by their genre, or we may not. In the last scenario, we would get stronger linear relationship compaired to the grouped case.

• Filtering the top 5 platforms and for all genres creating a heatmap that will also produce the count on it (the same for all platforms).

```
top_5_platforms = games.groupby('Platform').Platform.value_counts().nlargest(5).index

my_filter3 = games.Platform.isin(top_5_platforms)
filtered_data = games[my_filter3]

# Group the data by 'Genre' and 'Platform' and count the occurrences
heatmap_data = filtered_data.groupby(['Genre', 'Platform']).size().unstack(fill_value=0).transpose()

_ = plt.figure(figsize=(10, 6))

# Create a heatmap using imshow
_ = plt.imshow(heatmap_data, cmap=plt.get_cmap('YlGnBu'))

# Set axis labels and title
_ = plt.xlabel('Genre', fontsize = 10)
_ = plt.ylabel('Platform', fontsize = 10)
_ = plt.title('Game Count Heatmap of top 5 Platforms and All Genres', fontsize = 20, y = 1.05)

# Display the counts in each cell
for i in range(len(heatmap_data.index)):
    for j in range(len(heatmap_data.columns)):
        _ = plt.text(j, i, heatmap_data.iat[i, j], ha='center', va='center', color='black', size = 8)
```

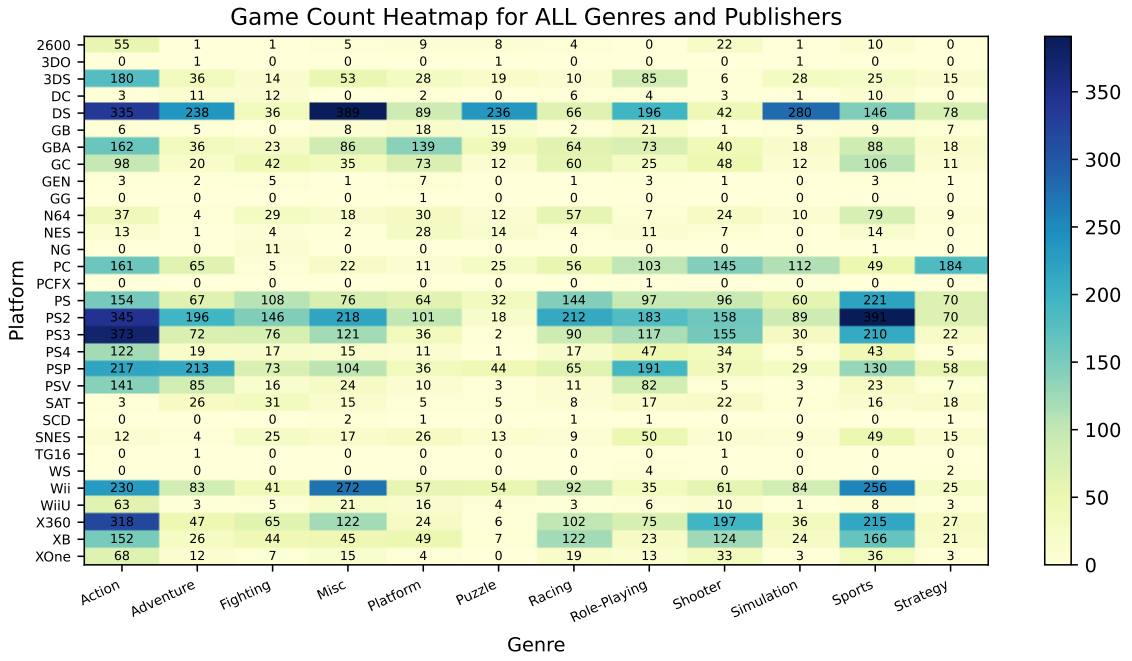
```
# Customize the ticks on both axes
_ = plt.xticks(np.arange(len(heatmap_data.columns)), heatmap_data.columns, rotation=90)
_ = plt.yticks(np.arange(len(heatmap_data.index)), heatmap_data.index)

# Display the colorbar
_ = plt.colorbar()

# Show the heatmap
plt.show()
```



A heatmap showing the number of games that have given platform(as a row) and given genre(as a colum). For e.g., the heatmap shows that in our data 400 games are for PS2 platform and are sport related games. The color of the tile indicates that it is glonal max value, since it has darkest color out of all tiles. As suggested by the legend, the darker the title, the greater the num of games with the tile-specified properties.

```
# Grouping the data by 'Genre' and 'Platform' and count the occurrences
heatmap_data = games.groupby(['Genre', 'Platform']).size().unstack(fill_value=0).transpose()


fig, ax = plt.subplots(figsize=(9, 5))  # Adjust the figure size as needed
# creating heatmap using imshow with YlGnBu color map and aspect set to auto
im = ax.imshow(heatmap_data, cmap='YlGnBu', aspect='auto')

# Set axis labels and title
_ = ax.set_xlabel('Genre', labelpad=6)
_ = ax.set_ylabel('Platform')
_ = ax.set_title('Game Count Heatmap for ALL Genres and Publishers')

# Displaying the counts in each cell in black
```

```
for i in range(len(heatmap_data.index)): # for each row
    for j in range(len(heatmap_data.columns)): # for each col
        count = heatmap_data.iat[i, j] # traverse the matrix, get count
        _ = ax.text(j, i, count, ha='center', va='center', color='black',size=6.5)  # Set count, text c

# Customizing the ticks on both axes
_ = ax.set_xticks(np.arange(len(heatmap_data.columns)))
_ = ax.set_yticks(np.arange(len(heatmap_data.index)))
_ = ax.set_xticklabels(heatmap_data.columns, rotation=25, fontsize = 7, ha='right', va = 'top')
_ = ax.set_yticklabels(heatmap_data.index, fontsize = 7)

# Displaying the colorbar
cbar = ax.figure.colorbar(im, ax=ax)

# optimally adjusting the sizes in the plot
plt.tight_layout()

# Show the heatmap
plt.show()
```



The same as above, but now we have all the platforms. As for me, I would say this is a bad visualization, since it has too much info, which can be misleading. Still, one case in which this could be useful is whenever we are interested in extremas in our data rather than the exact count of games for each platform and genre. So, this is fine if we want to get some insight whether the count of games in each genre and platform are distributed evenly. Or what genre and platform combinations are most popular...

We can use these type of heatmaps to analyze the gaming industry, for e.g. which genres are most popular on different platforms to identify trends in game development. They can help market researchers understand preferences for games on different platforms and how they vary from genre to genre.