TASK

Exploratory Data Analysis on the Video Game Sales Data Set

Visit our website

Introduction

The Gaming industry over the years has perfected its art in the graphics and quality of their games. The Video game sales dataset contains data that tells a story on a number of sales over the years in the gaming industry. We take a look at the highest selling publisher and platform that has the most sales. Our data set structure is as follows:

Rank - Ranking of overall sales

- Name - The Games name
- Platform - Platform of the games release (i.e. PC,PS4, Nintendo etc.)
- Year - Year of the game's release
- Genre - Genre of the game
- Publisher - Publisher of the game
- NA_Sales - Sales in North America
- EU_Sales - Sales in Europe
- JP_Sales - Sales in Japan (in millions)
- Other_Sales - Sales in the rest of the world
- Global_Sales - Total worldwide sales.

DATA CLEANING

First and foremost the data needs to be read and understood before data cleaning and analysis can occur.

We take a look at our dataset and display the first 5 records in the dataset to see the dataset fields and data populate

```
# load dataset into a data frame
df_vgame = pd.read_csv('vgsales.csv')
# display the first 5 records on the dataframe
df_vgame.head()
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37

2. Get information on the dimension and structure of the dataset

```
# get dataframe dimensions

print("Database dimension     :",df_vgame.shape)
print("Database size          :",df_vgame.size)

# get info about the dataframe (columns,entries,datatype etc...)

df_vgame.info()

# describe the dataframe and get statistics

df_vgame.describe()
```

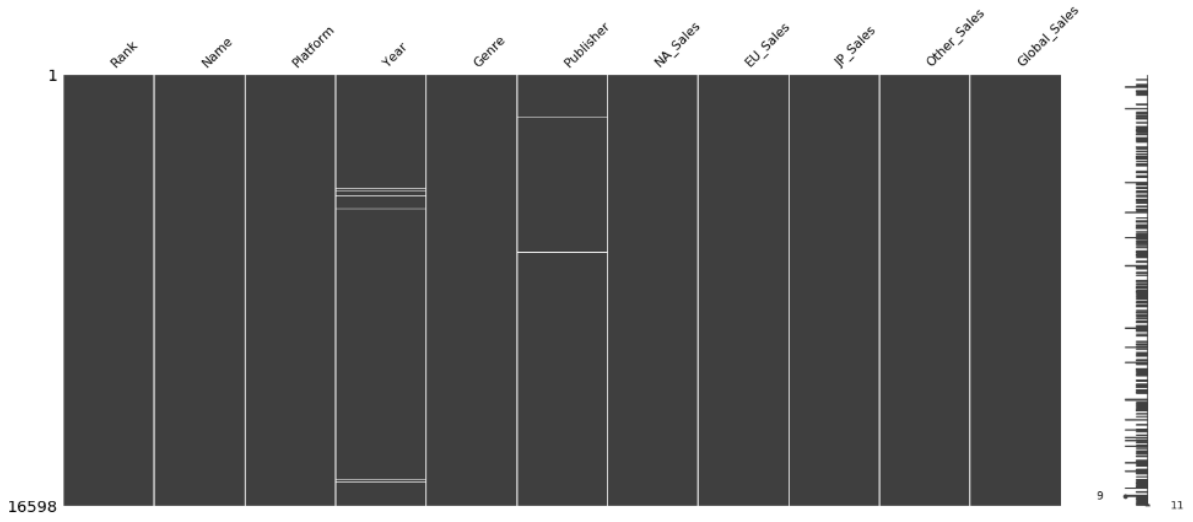
```
Database dimension     : (16598, 11)
Database size          : 182578
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Rank          16598 non-null  int64
 1   Name          16598 non-null  object
 2   Platform      16598 non-null  object
 3   Year          16327 non-null  float64
 4   Genre         16598 non-null  object
 5   Publisher     16540 non-null  object
 6   NA_Sales      16598 non-null  float64
 7   EU_Sales      16598 non-null  float64
 8   JP_Sales      16598 non-null  float64
 9   Other_Sales   16598 non-null  float64
 10  Global_Sales  16598 non-null  float64
dtypes: float64(6), int64(1), object(4)
```

3. Take a look at null values in the dataset

Here we notice that we have two columns with null values that we must handle for better analysis.

```
# Lets take a look at the dataframe to check null values

mn.matrix(df_vgame)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7886111fa0>
```



MISSING DATA

1. Firstly we get the number of missing values found in each column, we notice that the year column has 271 missing values and Publisher has 58 missing values. This poses a problem to our dataset and the missing values must be populated.

```
# Data Cleaning
# count number null values in each column

null_value_perc = pd.DataFrame((df_vgame.isnull().sum())*100/df_vgame.shape[0]).reset_index()
null_value_perc.columns = ['Column Name', 'Null Values Percentage']
null_value = pd.DataFrame(df_vgame.isnull().sum()).reset_index()
null_value.columns = ['Column Name', 'No. of Null Values']
null_vgd = pd.merge(null_value, null_value_perc, on='Column Name')
null_vgd
```

	Column Name	No. of Null Values	Null Values Percentage
0	Rank	0	0.000000
1	Name	0	0.000000
2	Platform	0	0.000000
3	Year	271	1.632727
4	Genre	0	0.000000
5	Publisher	58	0.349440
6	NA_Sales	0	0.000000
7	EU_Sales	0	0.000000
8	JP_Sales	0	0.000000
9	Other_Sales	0	0.000000
10	Global_Sales	0	0.000000

2. Year and Publisher column need to be replaced with appropriate data. Imputation of correct values. The year 2009 was used since most games were published in that year. It is therefore the average year. The publisher column with missing Values was dropped since they sold one copy; this affects our sales by a very small fraction and can be excluded/dropped from our dataset.

```
# get number of games with missing publisher

print("Total Publisher missing values ", df_vgame['Publisher'].isnull().sum(),' rows')

# drop thesse values as they only sold once and sell value is insignificant

df_vgame = df_vgame.dropna()

# check if records were dropped

df_vgame.isnull().sum()
```

```
Total Publisher missing values  58  rows

Rank            0
Name            0
Platform        0
Year            0
Genre           0
Publisher       0
NA_Sales        0
EU_Sales        0
JP_Sales        0
Other_Sales     0
Global_Sales    0
dtype: int64
```

```
 1   Name          16598 non-null  object
 2   Platform      16598 non-null  object
 3   Year          16598 non-null  int64
```

```
# get number of games with missing publisher

print("Total Publisher missing values ", df_vgame['Publisher'].isnull().sum(),' rows')

# drop thesse values as they only sold once and sell value is insignificant

df_vgame = df_vgame.dropna()

# check if records were dropped

df_vgame.isnull().sum()
```

```
Total Publisher missing values  58  rows
```

```
Rank            0
Name            0
Platform        0
Year            0
Genre           0
Publisher       0
NA_Sales        0
EU_Sales        0
JP_Sales        0
Other_Sales     0
Global_Sales    0
dtype: int64
```

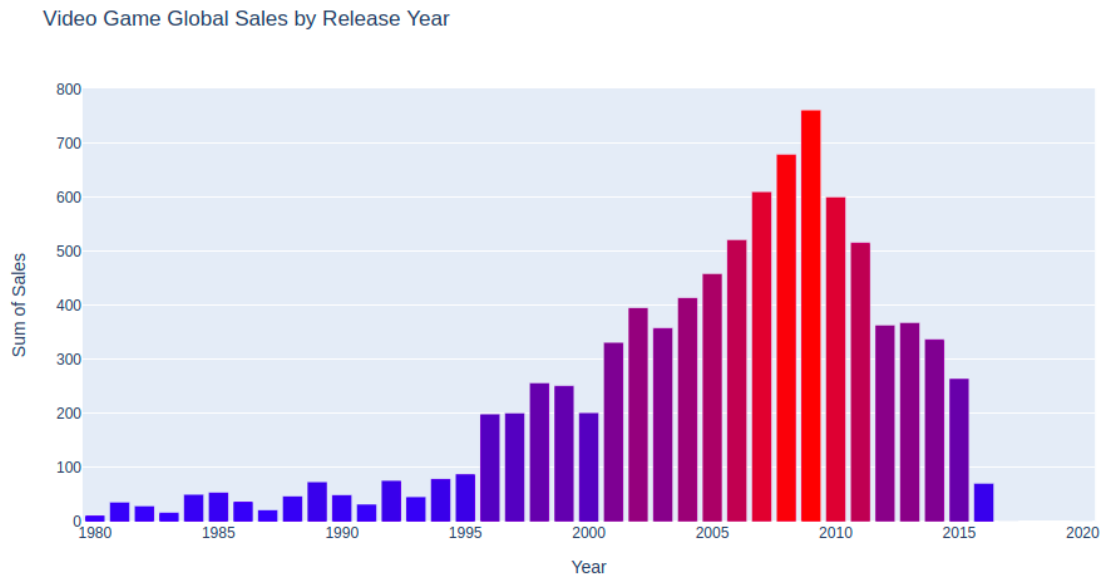
DATA STORIES AND VISUALIZATIONS

Visualization is an important part of data analysis as it can be used to tell a story about your data and visualize data findings. Let us see our dataset story as follows:
Firstly we get the dataset in-depth statistics.

```
# Visualization

# lets get the dataframe statistics indepth

games =df_vgame['Name'].unique()
publishers = df_vgame['Publisher'].unique()
platforms = df_vgame['Platform'].unique()
game_type = df_vgame['Genre'].unique()

print("Number of Games: ",len(games))
print("Publishers: ",len(publishers))
print("Platforms: ",len(platforms))
print("Game Types: ",len(game_type))
```

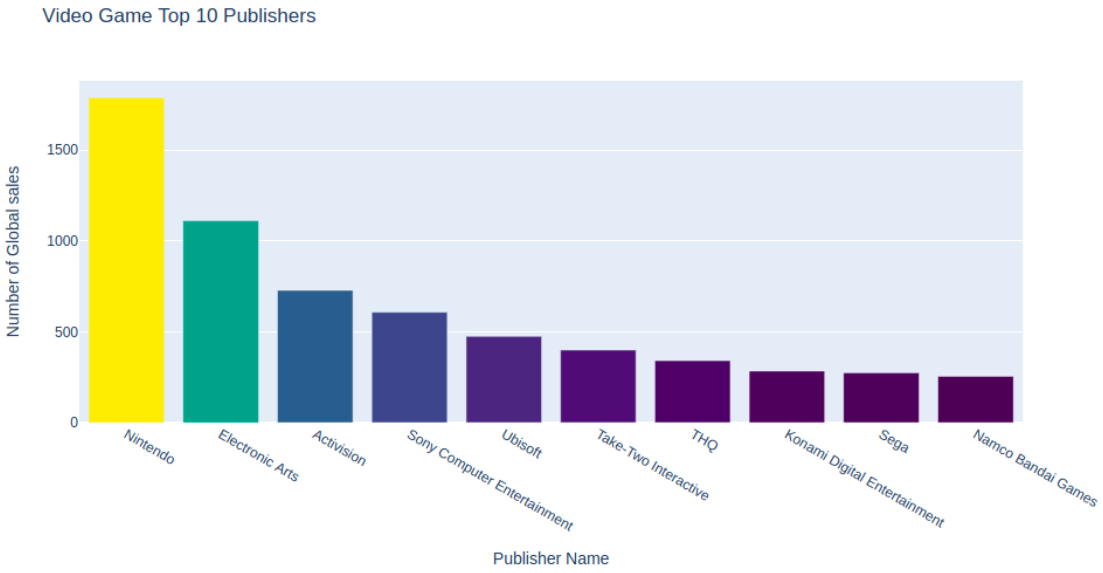
```
Number of Games:  11442
Publishers:  578
Platforms:  31
Game Types:  12
```

1. Video game global sales by release year: here we see that there was an increase in global sales as the years went up.

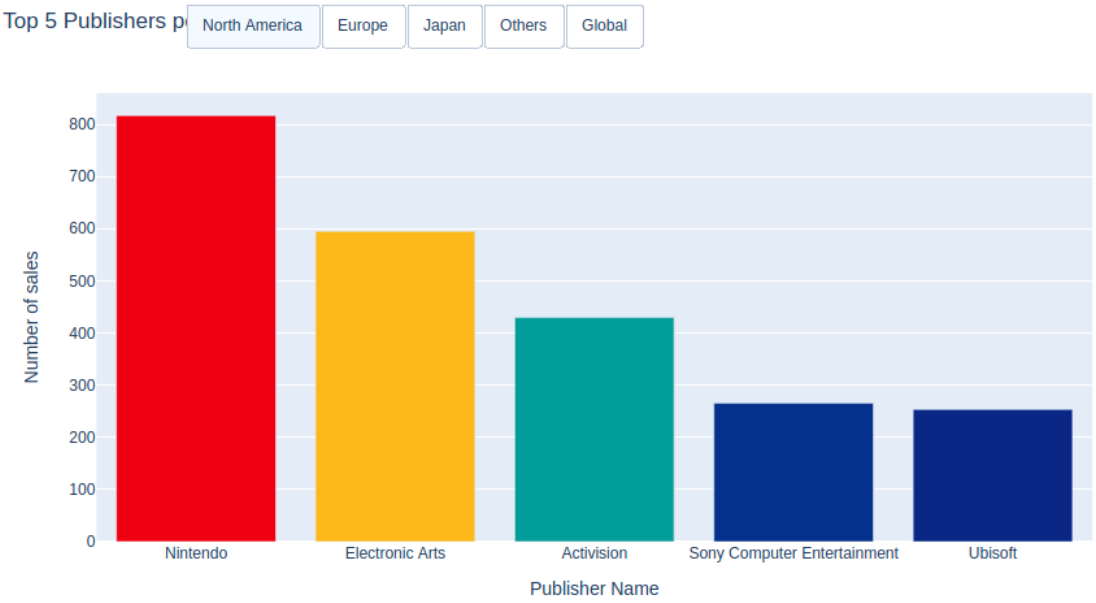


2. Top 10 best video game publishers in the gaming industry

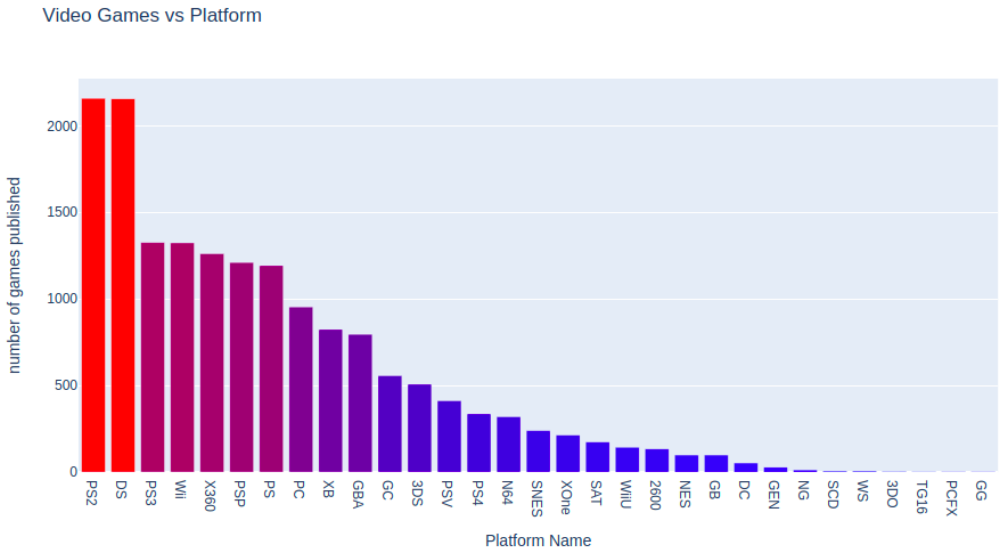
3. Video Game Top 10 Publishers by Global sales



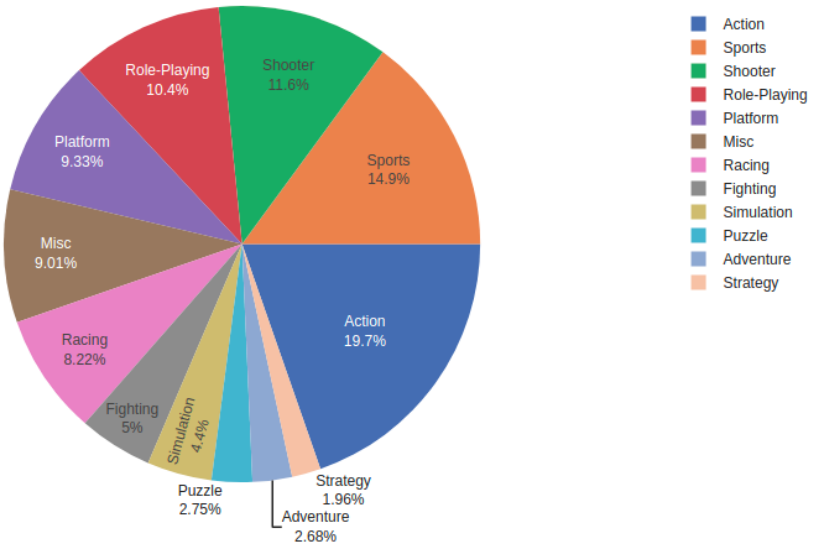
4. Top 5 Publishers in each region and global



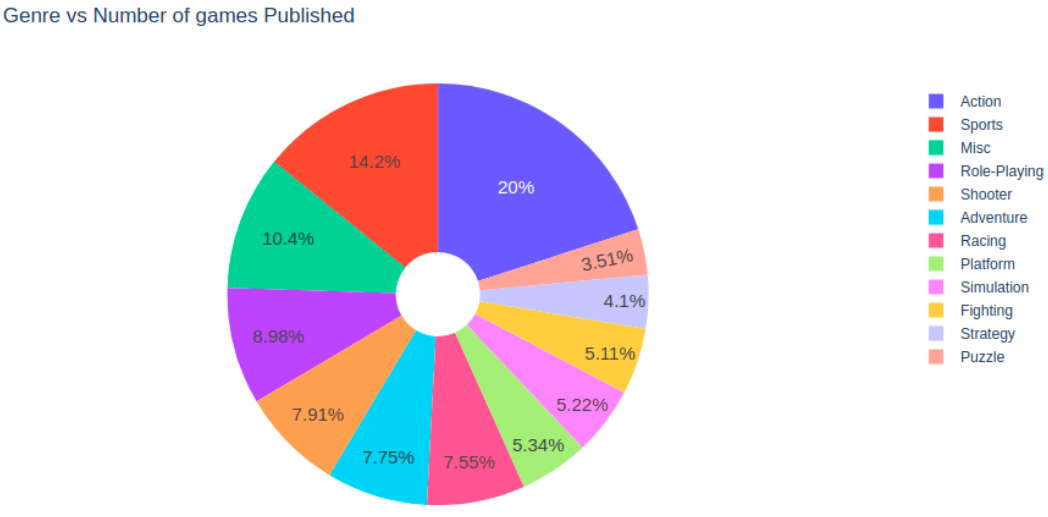
5. Number of games published per platform



6. Sales percentile versus genre

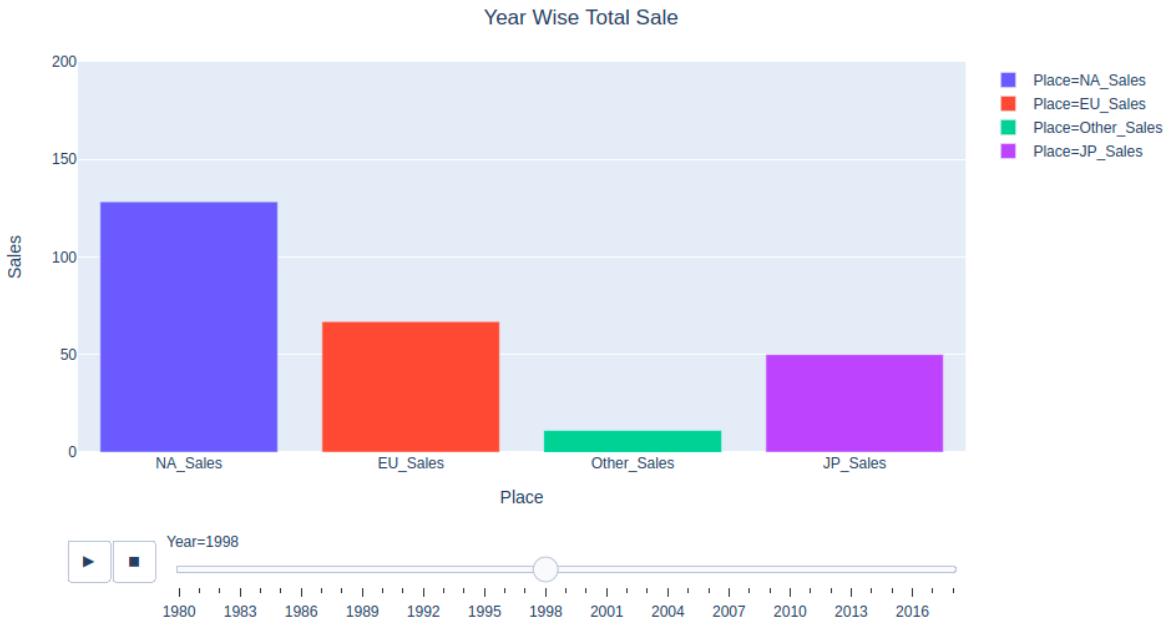


7. Genre versus number of game published percentile



8. Word cloud on the Platform of video games

9. Number of Sales per region over the years



CONCLUSION

In our data analysis we can make the following conclusion based on our finding in the video game sales dataset:

1. The was more Global sales in the year 2009
2. Electronic Arts ranks number 1 in terms of number of games published
3. Nintendo ranks number 1 in global sales.
4. PS2 and DS are the highest platforms with highest number of games published globally
5. Action genre ranks number one in game sales Globally
6. Action genre is published more globally
7. Sales increased each year in each region
8. More gamers prefer PS3,PS2 and Nintendo Wii

THIS REPORT WAS WRITTEN BY: Molato Paul Sekgobela