AppliedStatistics_FinalProject

Shradha Balasaheb Godse

2023-12-02

Factors Influencing Video Game Sales

Problem Statement

In this project, by employing the statistical methodologies, we aim to extract meaningful insights from the dataset, uncover patterns, and contribute to a deeper understanding of the factors influencing video game sales in the market.

The primary goal of this exploration is to delve into the correlations between various factors such as gaming platforms, genres, and publishers, and the resultant impact on video game sales. Furthermore, the dataset lends itself to hypothesis testing, allowing for the formulation and validation of hypotheses related to specific variables influencing global video game sales. Regression models will be developed to predict video game sales based on selected features, providing insights into the factors contributing significantly to a game's success.

The analytical methods applied include descriptive statistics and exploratory data analysis (EDA) to understand the distribution of data, hypothesis testing to validate or reject hypotheses, and regression analysis to model the relationships between independent variables and the dependent variable (Global_Sales).

Contents

1. Introduction

The dataset under consideration encompasses information on video games with sales exceeding 100,000 copies, providing a valuable repository for comprehensive statistical exploration. The dataset can be accessed here: https://www.kaggle.com/datasets/gregorut/videogamesales. The dataset includes a range of pertinent variables, such as the ranking of games, their titles, the platforms they are available on, release years, genres, and sales figures across different regions including North America (NA), Europe (EU), Japan (JP), and other territories. Additionally, the dataset features a cumulative "Global_Sales" variable, providing a holistic measure of a game's success on a global level.

This dataset not only facilitates a deep understanding of the video game market but also allows for the extraction of actionable insights that can be invaluable for industry professionals, researchers, and enthusiasts alike. As we embark on this analytical journey, we aim to uncover hidden trends, identify influential factors, and contribute to a nuanced understanding of the dynamics within the global video game sales landscape.

```r
# Importing required libraries
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
vgsales = read.csv('vgsales.csv')
cat("Read the first 6 rows of dataset\n"); head(vgsales)
```

```
## Read the first 6 rows of dataset

##   Rank                     Name Platform Year        Genre Publisher NA_Sales
## 1    1               Wii Sports      Wii 2006       Sports  Nintendo    41.49
## 2    2        Super Mario Bros.      NES 1985     Platform  Nintendo    29.08
## 3    3           Mario Kart Wii      Wii 2008       Racing  Nintendo    15.85
## 4    4        Wii Sports Resort      Wii 2009       Sports  Nintendo    15.75
## 5    5 Pokemon Red/Pokemon Blue       GB 1996 Role-Playing  Nintendo    11.27
## 6    6                   Tetris       GB 1989       Puzzle  Nintendo    23.20
##   EU_Sales JP_Sales Other_Sales Global_Sales
## 1    29.02     3.77        8.46        82.74
## 2     3.58     6.81        0.77        40.24
## 3    12.88     3.79        3.31        35.82
## 4    11.01     3.28        2.96        33.00
## 5     8.89    10.22        1.00        31.37
## 6     2.26     4.22        0.58        30.26
```

```r
num_rows = nrow(vgsales)
num_cols = ncol(vgsales)
cat("Total number of rows in the dataset =", num_rows, "\n")
```

```
## Total number of rows in the dataset = 16598
```

```
cat("Total number of columns in the dataset =", num_cols, "\n")
```

```
## Total number of columns in the dataset = 11
```

As a first step in the analysis, we should take a look at the variables in the dataset. This can be done using the str function.

```
str(vgsales)
```

```
## 'data.frame':    16598 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 ...
```

Exploratory Data Analysis

Outlier Detection

We will identify and analyze outliers in the dataset to understand if there are any exceptional cases or anomalies. We will also compute summary statistics (mean, median, range, standard deviation) for sales figures (NA_Sales, EU_Sales, JP_Sales, Other_Sales, Global_Sales) to understand the distribution of sales.

```
summary(vgsales$NA_Sales)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.0000  0.0800  0.2647  0.2400 41.4900
```

```
summary(vgsales$EU_Sales)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.0000  0.0200  0.1467  0.1100 29.0200
```

```
summary(vgsales$JP_Sales)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##  0.00000  0.00000  0.00000  0.07778  0.04000 10.22000
```

```
summary(vgsales$Other_Sales)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.00000  0.00000  0.01000  0.04806  0.04000 10.57000
```

```
summary(vgsales$Global_Sales)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0100  0.0600  0.1700  0.5374  0.4700 82.7400
```

Data Cleaning and Preparation:

We will first check for the missing values in the dataset.

```
# Check for "N/A" values in the 'Publisher' column
na_values_in_publisher <- sum(vgsales$Publisher == "N/A")

# Display the count of "N/A" values in the 'Publisher' column
print("N/A Values in 'Publisher' Column:")
```

```
## [1] "N/A Values in ’Publisher’ Column:"
```

```
print(na_values_in_publisher)
```

```
## [1] 58
```

There are 58 Null values in the Publisher column.

```
# Check for "N/A" values in the 'Year' column
na_values_in_year <- sum(vgsales$Year == "N/A")

# Display the count of "N/A" values in the 'Year' column
print("N/A Values in 'Year' Column:")
```

```
## [1] "N/A Values in ’Year’ Column:"
```

```
print(na_values_in_year)
```

```
## [1] 271
```

There are 271 Null values in the Year column.

Hence, we will further check for the null values in all columns and drop these rows with null values.

```
# Specify the columns you want to check for "N/A" values
columns_to_check <- c("Rank", "Name", "Platform", "Year", "Genre", "Publisher", "NA_Sales", "EU_Sales",

# Remove rows with "N/A" values in the specified columns
vgsales <- vgsales %>%
  filter_all(all_vars(!is.na(.))) %>%
  filter_at(vars(columns_to_check), all_vars(. != "N/A"))
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use ‘all_of()‘ or ‘any_of()‘ instead.
##   # Was:
##   data %>% select(columns_to_check)
##
##   # Now:
##   data %>% select(all_of(columns_to_check))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
# Check if there are any "N/A" values now
na_count <- sum(is.na(vgsales) | vgsales == "N/A")
print("Count of 'N/A' Values:")
```

```
## [1] "Count of ’N/A’ Values:"
```

```
print(na_count)
```

```
## [1] 0
```

```
# Check the number of rows after removing "N/A" values
num_rows <- nrow(vgsales)
print("Number of Rows After Removal:")
```

```
## [1] "Number of Rows After Removal:"
```

```
print(num_rows)
```

```
## [1] 16291
```

There are no missing values in the dataset. Further, we can check for the duplicate records. We will drop the duplicates if found.

```
# Check for duplicates
duplicate_rows = vgsales[duplicated(vgsales), ]
print(duplicate_rows)
```

```
##  [1] Rank         Name         Platform     Year         Genre
##  [6] Publisher    NA_Sales     EU_Sales     JP_Sales     Other_Sales
## [11] Global_Sales
## <0 rows> (or 0-length row.names)
```

No duplicate records found.

```
# Set the "Rank" column as row names (index)
rownames(vgsales) <- vgsales$Rank

# Remove the "Rank" column as it's now set as row names
vgsales <- vgsales[, -1]

# Print the modified dataset with "Rank" as the index
print("\nModified Dataset with Rank as Index:")
```

```
## [1] "\nModified Dataset with Rank as Index:"
```

```
head(vgsales)
```

```
##                         Name Platform Year        Genre Publisher NA_Sales
## 1               Wii Sports      Wii 2006       Sports  Nintendo    41.49
## 2        Super Mario Bros.      NES 1985     Platform  Nintendo    29.08
## 3           Mario Kart Wii      Wii 2008       Racing  Nintendo    15.85
## 4        Wii Sports Resort      Wii 2009       Sports  Nintendo    15.75
## 5 Pokemon Red/Pokemon Blue       GB 1996 Role-Playing  Nintendo    11.27
## 6                   Tetris       GB 1989       Puzzle  Nintendo    23.20
##   EU_Sales JP_Sales Other_Sales Global_Sales
## 1    29.02     3.77        8.46        82.74
## 2     3.58     6.81        0.77        40.24
## 3    12.88     3.79        3.31        35.82
## 4    11.01     3.28        2.96        33.00
## 5     8.89    10.22        1.00        31.37
## 6     2.26     4.22        0.58        30.26
```

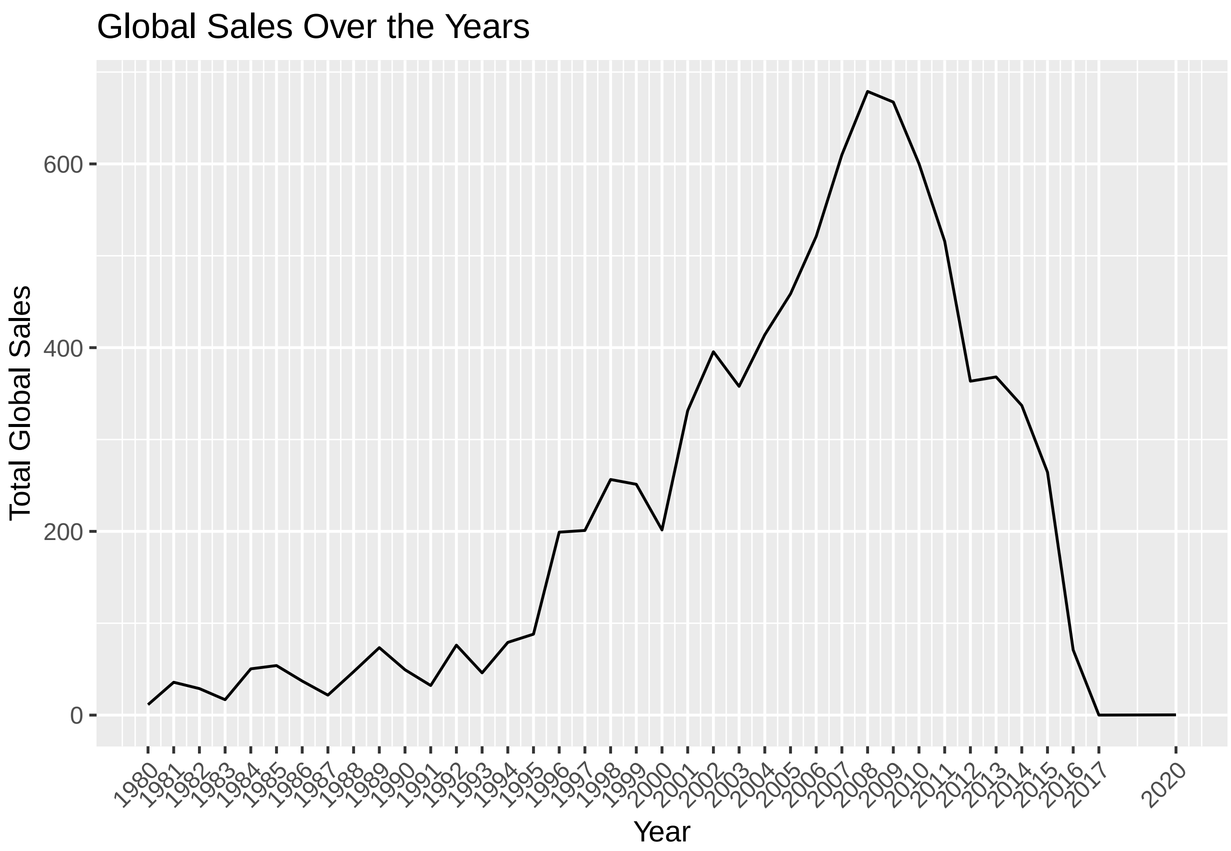
3. Data Analysis

Analyzing trends in global sales over the years. Identify years with significant changes in sales patterns.

```
# Convert 'Year' to numeric
vgsales$Year <- as.numeric(vgsales$Year)

# Time Series Analysis - Trends in Global Sales over the Years
sales_by_year <- vgsales %>%
  group_by(Year) %>%
  summarise(Total_Global_Sales = sum(Global_Sales))

# Visualize Time Series - Global Sales over the Years
ggplot(sales_by_year, aes(x = Year, y = Total_Global_Sales)) +
  geom_line() +
  labs(title = "Global Sales Over the Years",
       x = "Year",
       y = "Total Global Sales") +
  scale_x_continuous(breaks = unique(sales_by_year$Year)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



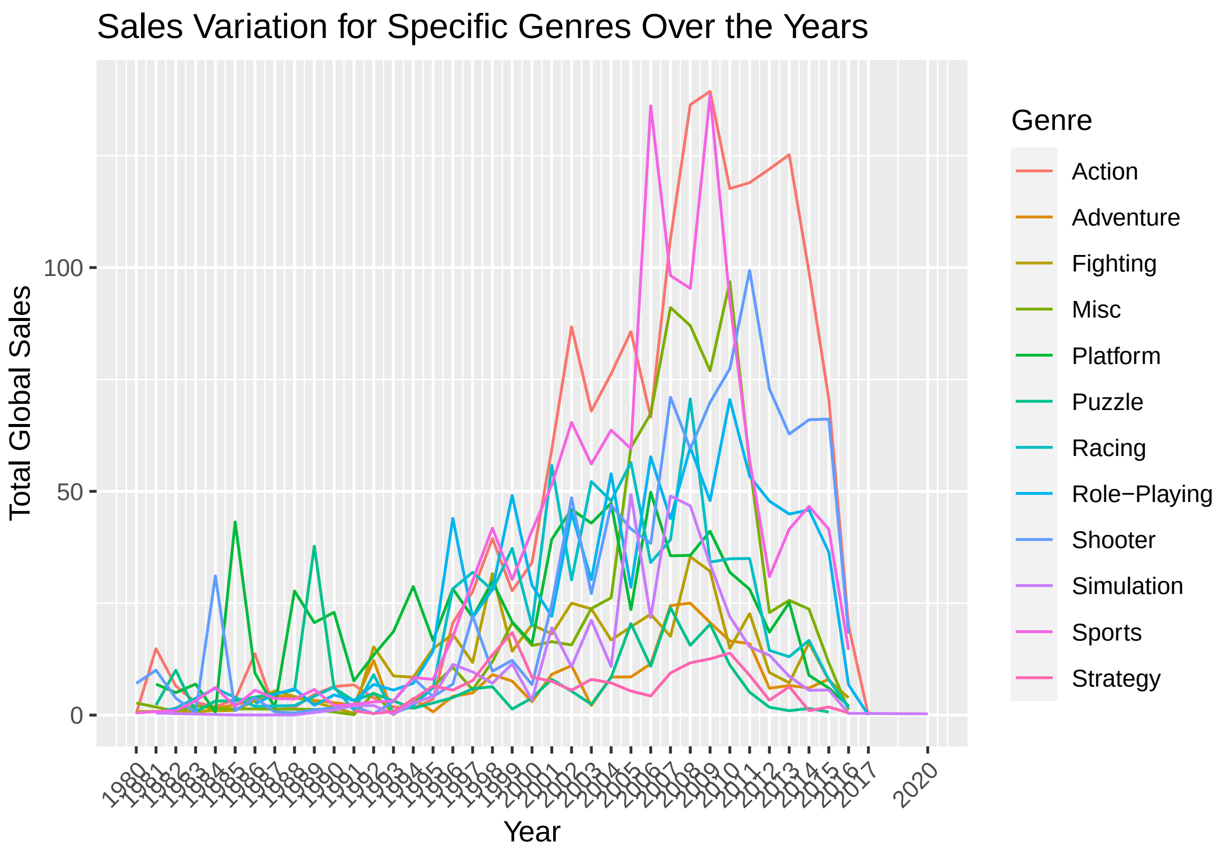
The Global Sales are highest in the year 2008. We will further conduct a more granular analysis by examining the performance of specific genres or individual game releases during the identified years with significant changes. This may reveal which types of games contributed most to the observed trends.

Examine how sales vary for specific genres or platforms over time.

```
# Time Series Analysis - Sales Variation for Specific Genres over the Years
sales_by_genre <- vgsales %>%
  group_by(Year, Genre) %>%
  summarise(Total_Global_Sales = sum(Global_Sales))
```

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

```
# Visualize Time Series - Sales Variation for Specific Genres
ggplot(sales_by_genre, aes(x = as.numeric(Year), y = Total_Global_Sales, color = Genre)) +
  geom_line() +
  labs(title = "Sales Variation for Specific Genres Over the Years",
       x = "Year",
       y = "Total Global Sales",
       color = "Genre")+
       scale_x_continuous(breaks = unique(sales_by_year$Year)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



From the graph, it is clear that the Actions games have highest global sales in that year.