Analysis_VideoGames

Quentin Cartier

7/27/2021

Analysis: Video Game Critic Score

Executive summary

This report aims to explore a dataset containing multiple information of video games, and try to predict a critic score. The dataset used is available at https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings The zip file is provided with the project.

We'll first unzip and load the csv file. The csv uses "," as delimiter.

```
knitr::opts_chunk$set(echo = TRUE, fig.align = 'center', cache=FALSE, cache.lazy = FALSE)
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(lubridate)) install.packages("lubridate", repos = "http://cran.us.r-project.org")
if(!require(readr)) install.packages("readr", repos = "http://cran.us.r-project.org")
if(!require(kableExtra)) install.packages("kableExtra", repos = "http://cran.us.r-project.org")
library(tidyverse)
library(caret)
library(data.table)
library(lubridate)
library(readr)
library(kableExtra)
# Video Game Sales with Ratings:
# https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings
data <- fread(unzip("Video_Games_Sales_as_at_22_Dec_2016.csv.zip","Video_Games_Sales_as_at_22_Dec_2016.
data <- read.csv(unzip("Video_Games_Sales_as_at_22_Dec_2016.csv.zip","Video_Games_Sales_as_at_22_Dec_20
header = TRUE)
data <- tibble(data)
head(data)
```

```
## # A tibble: 6 x 16
##   Name  Platform Year_of_Release Genre Publisher NA_Sales EU_Sales JP_Sales
##   <fct> <fct>    <fct>           <fct> <fct>        <dbl>    <dbl>    <dbl>
## 1 Wii ~ Wii      2006            Spor~ Nintendo      41.4    29.0      3.77
## 2 Supe~ NES      1985            Plat~ Nintendo      29.1     3.58     6.81
## 3 Mari~ Wii      2008            Raci~ Nintendo      15.7    12.8      3.79
## 4 Wii ~ Wii      2009            Spor~ Nintendo      15.6    10.9      3.28
## 5 Poke~ GB       1996            Role~ Nintendo      11.3     8.89    10.2
## 6 Tetr~ GB       1989            Puzz~ Nintendo      23.2     2.26     4.22
## # ... with 8 more variables: Other_Sales <dbl>, Global_Sales <dbl>,
```

```
## #   Critic_Score <int>, Critic_Count <int>, User_Score <fct>, User_Count <int>,
## #   Developer <fct>, Rating <fct>
```

As we see, the dataset contains 16 variables.

Basic information

Basic information includes the name of the game, the platform, the publisher, the year of release, and the genre. A column "Developer" also exists, but is often empty.There is also a "Rating" column, that indicates for who is the games (E: Everyone, T: Teen, M: Adult Only (Mature audience))

Sales

Concerning the sales, there is a variable for each region (North America, Europe, Japan, and "Other"), and a variable "Global_Sales" that regroup the sales of all these regions. The sales are in million units. (For example, 40,240,000 units of Super Mario Bros. on NES have been sold worldwide)

Critics

Two scores are binded to a game, a score given by critics (Critic_Score), and a score given by users. For each kind of score, a counter is associated (Critic_Count / User_Count).

##Analysis

Data Cleaning

We first need to convert the columns to appropriate types. The "Critic_Score" column is converted to numeric type, as weel as the "User_Score". We give a default value if a row does not a a user score. Also we harmonize the user score format with the critic score. (Eg. 8/10 becomes 80% for user score)

```
data$Critic_Score <- as.numeric(data$Critic_Score)
#data <- data %>% filter(!is.na(User_Score))
data <- data %>% mutate(User_Score = replace_na(User_Score, 5))
data <- data %>% mutate(User_Score = as.numeric(as.character(User_Score)) * 10)
```

```
## Warning: Problem with `mutate()` input `User_Score`.
## i NAs introduced by coercion
## i Input `User_Score` is `as.numeric(as.character(User_Score)) * 10`.
```

```
## Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion
```

```
data <- data %>% filter(!is.na(Year_of_Release)) %>% mutate(Year_of_Release = as.numeric(as.character(Y
```

```
## Warning: Problem with `mutate()` input `Year_of_Release`.
## i NAs introduced by coercion
## i Input `Year_of_Release` is `as.numeric(as.character(Year_of_Release))`.
```

```
## Warning: NAs introduced by coercion
```

Data Exploration

```
data %>% select(Name, Platform, Critic_Score, User_Score) %>% head(.)
```

How many Video Games are in the dataset ?

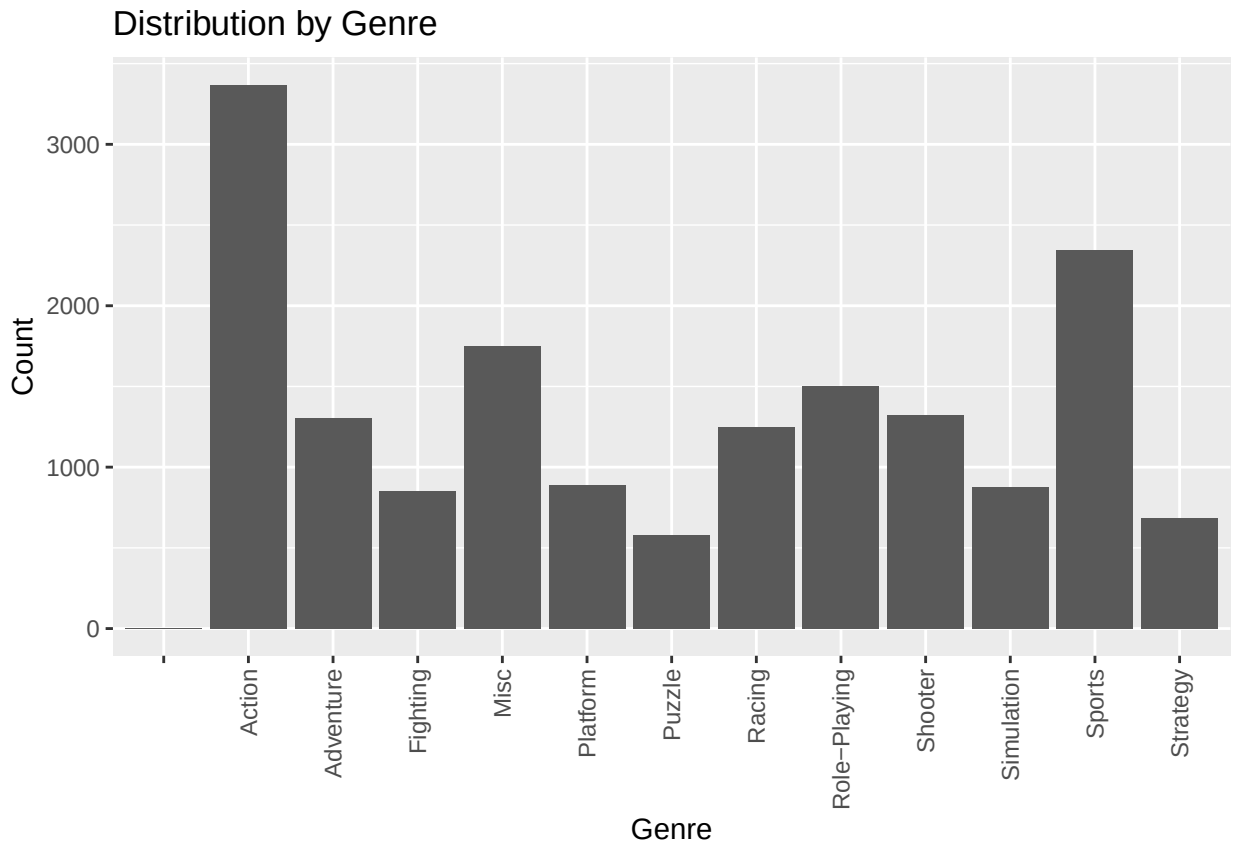
```
## # A tibble: 6 x 4
##   Name                      Platform Critic_Score User_Score
##   <fct>                     <fct>           <dbl>      <dbl>
## 1 Wii Sports                Wii                76         80
## 2 Super Mario Bros.         NES                NA         NA
## 3 Mario Kart Wii            Wii                82         83
## 4 Wii Sports Resort         Wii                80         80
## 5 Pokemon Red/Pokemon Blue GB                  NA         NA
## 6 Tetris                    GB                 NA         NA
```

```
data %>% summarize(n = n(), Critic_Score = mean(Critic_Score, na.rm = TRUE), User_Score = mean(User_Sco
kable() %>%
kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"),
position = "center",
font_size = 10,
full_width = FALSE)
```

n	Critic_Score	User_Score
16719	68.96768	71.25046

The User Score is originally graded on a 0-10 scale, where as the Critic Score is grade on a 100 scale. So we scaled the User score to Critic score scale. #### Genre Distribution As we can see below, Action games are the most often released, followed by sports games.

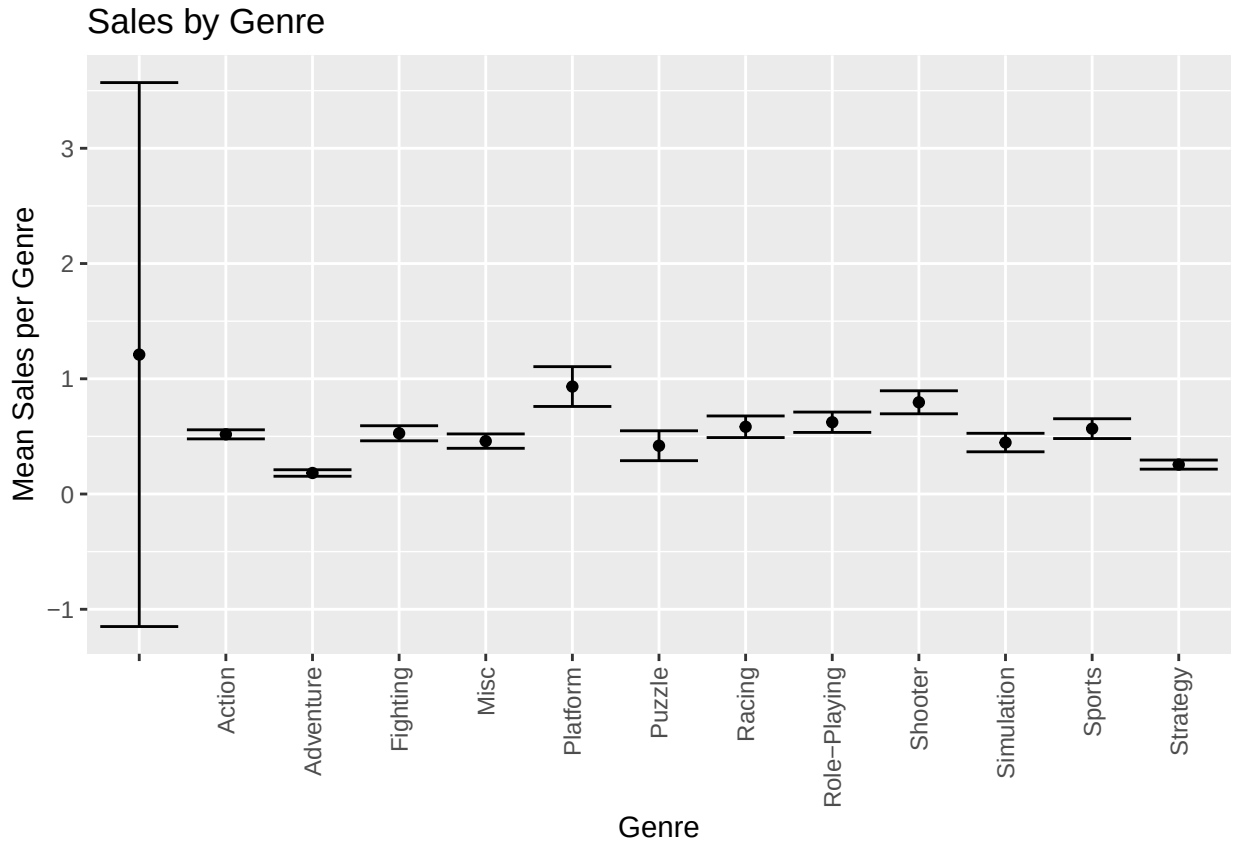
```
data %>%
ggplot(aes(Genre)) +
geom_bar() +
labs(title = "Distribution by Genre",
x = "Genre",
y = "Count") +
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



But the question is, is it the most sold genre?

```
data %>% group_by(Genre) %>%
summarize(n = n(), avg = mean(Global_Sales), se = sd(Global_Sales)/sqrt(n())) %>%
ggplot(aes(x=Genre, y=avg, ymin=avg -2*se, ymax =avg + 2*se)) +
geom_point() +
geom_errorbar() +
labs(title = "Sales by Genre",
x = "Genre",
y = "Mean Sales per Genre") +
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

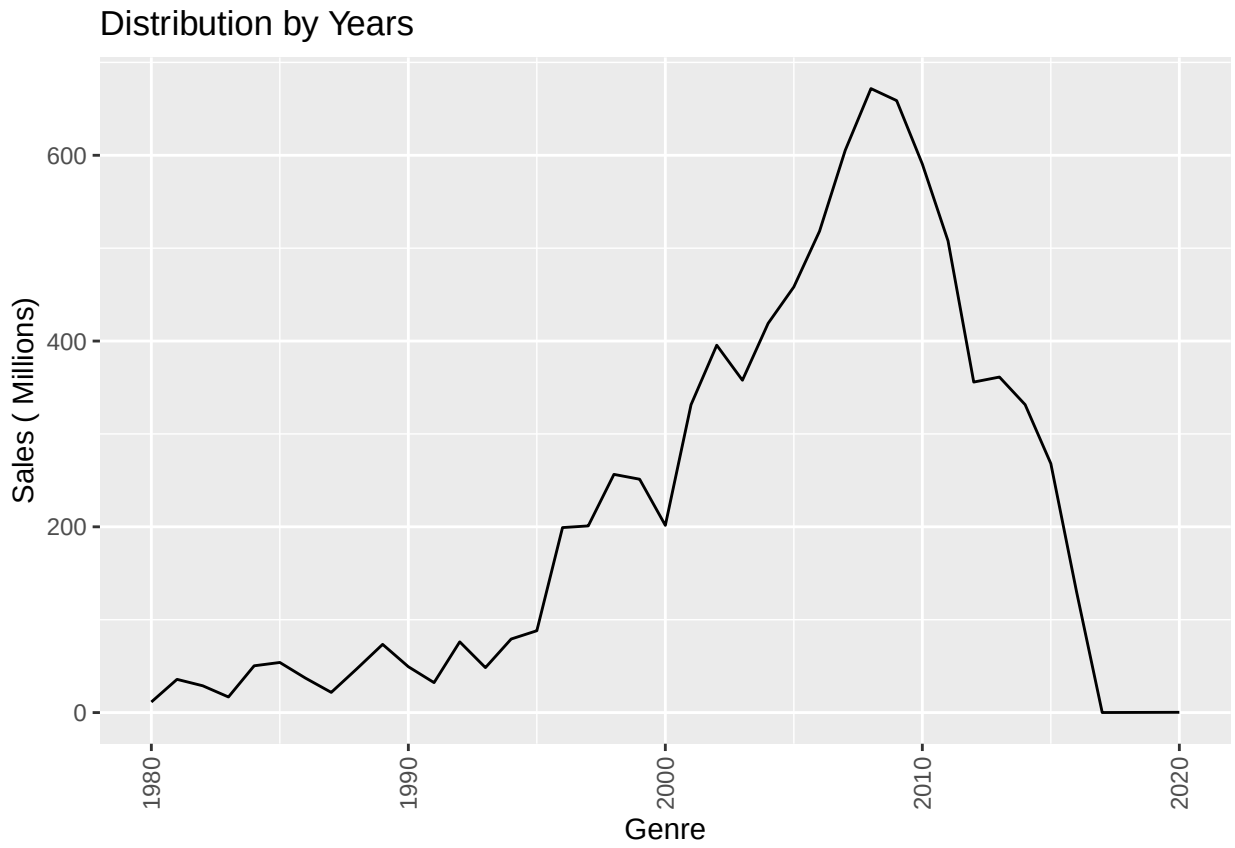


This diagram show that action games are not the more popular in term of sales. It may hide another reality. Lets analyse the evolution of sales on a time pediod

```
data %>% group_by(Year_of_Release) %>%  summarize(sales = sum(Global_Sales, na.rm = TRUE)) %>%
ggplot(aes(x = Year_of_Release, y = sales)) +
geom_line() +
labs(title = "Distribution by Years",
x = "Genre",
y = "Sales ( Millions)") +
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

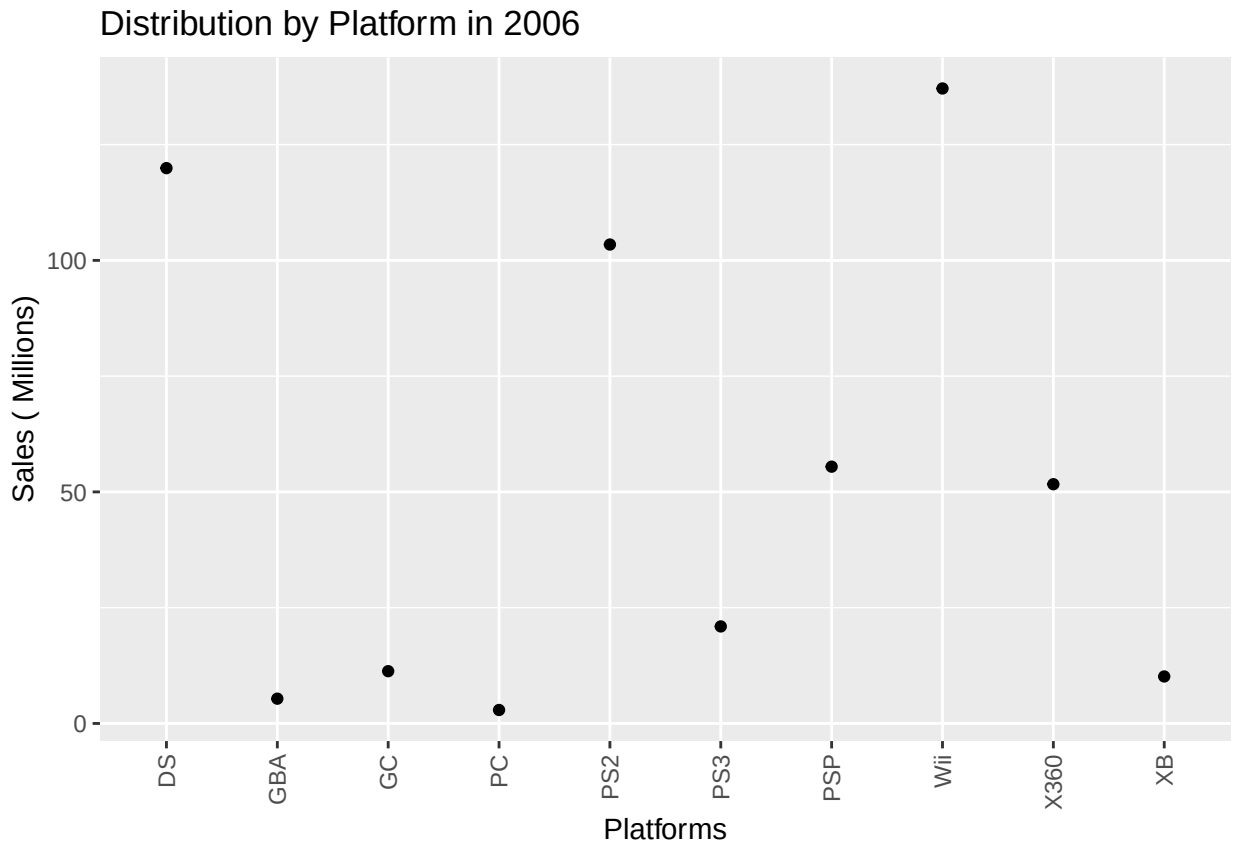
```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



2006 is a special year in the game industry. This is the year of release of the new console generations (Xbox360, Wii, and PS3). We can follow here the commercial success of the Wii.

```
data %>% filter(Year_of_Release == "2006") %>% group_by(Platform) %>% summarize(n = n(), sales = sum(Gl
ggplot(aes(x = Platform, y = sales)) +
geom_point() +
labs(title = "Distribution by Platform in 2006",
x = "Platforms",
y = "Sales ( Millions)") +
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

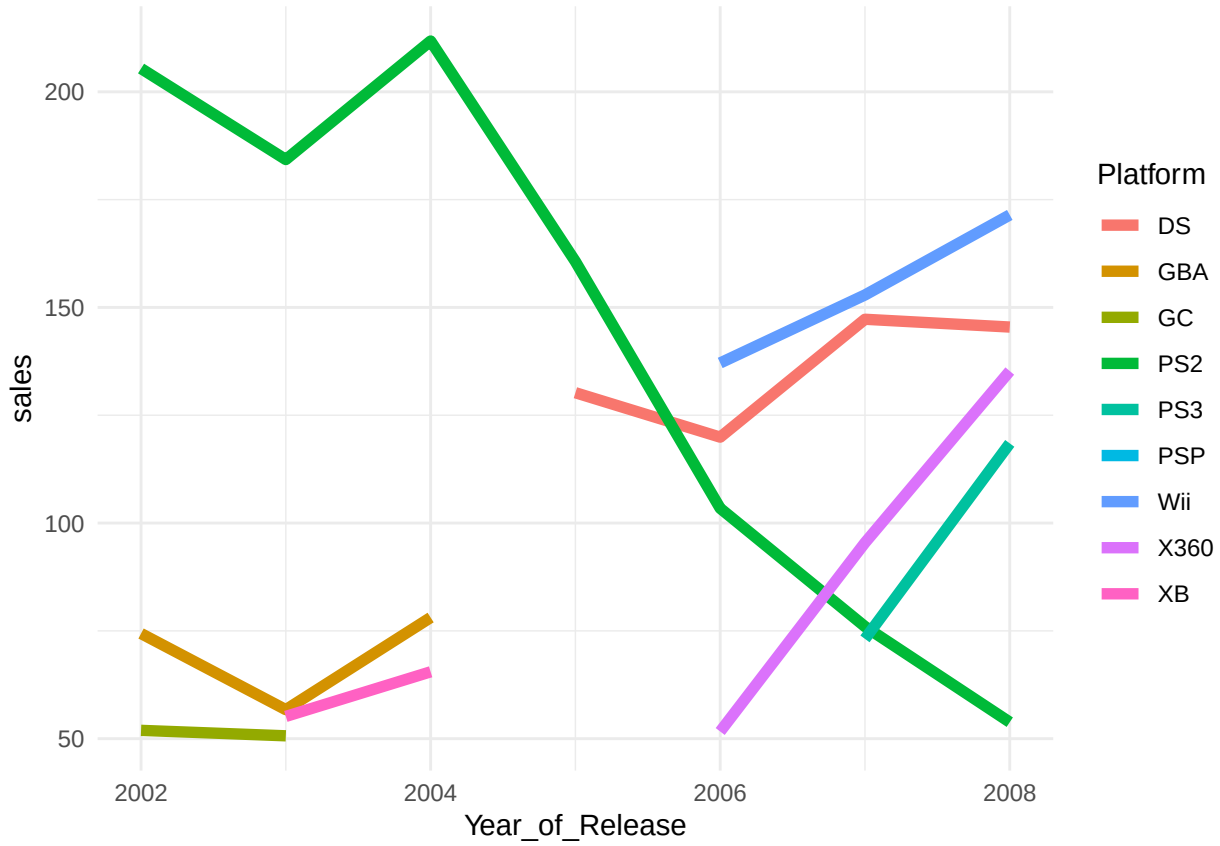
```
## `summarise()` ungrouping output (override with `.groups` argument)
```



Below, we can observe the progression of the Wii from 2006, as well as the progressive retirement of the PS2.

```
data %>% filter(Year_of_Release >= 2002 & Year_of_Release <= 2008) %>%
group_by(Year_of_Release, Platform) %>% summarize(sales = sum(Global_Sales, na.rm = TRUE)) %>%
filter( sales > 50) %>%
ungroup(Year_of_Release, Platform) %>%
ggplot(aes(x = Year_of_Release, y = sales)) +
geom_line(aes(color = Platform), size = 2) +
theme_minimal()
```

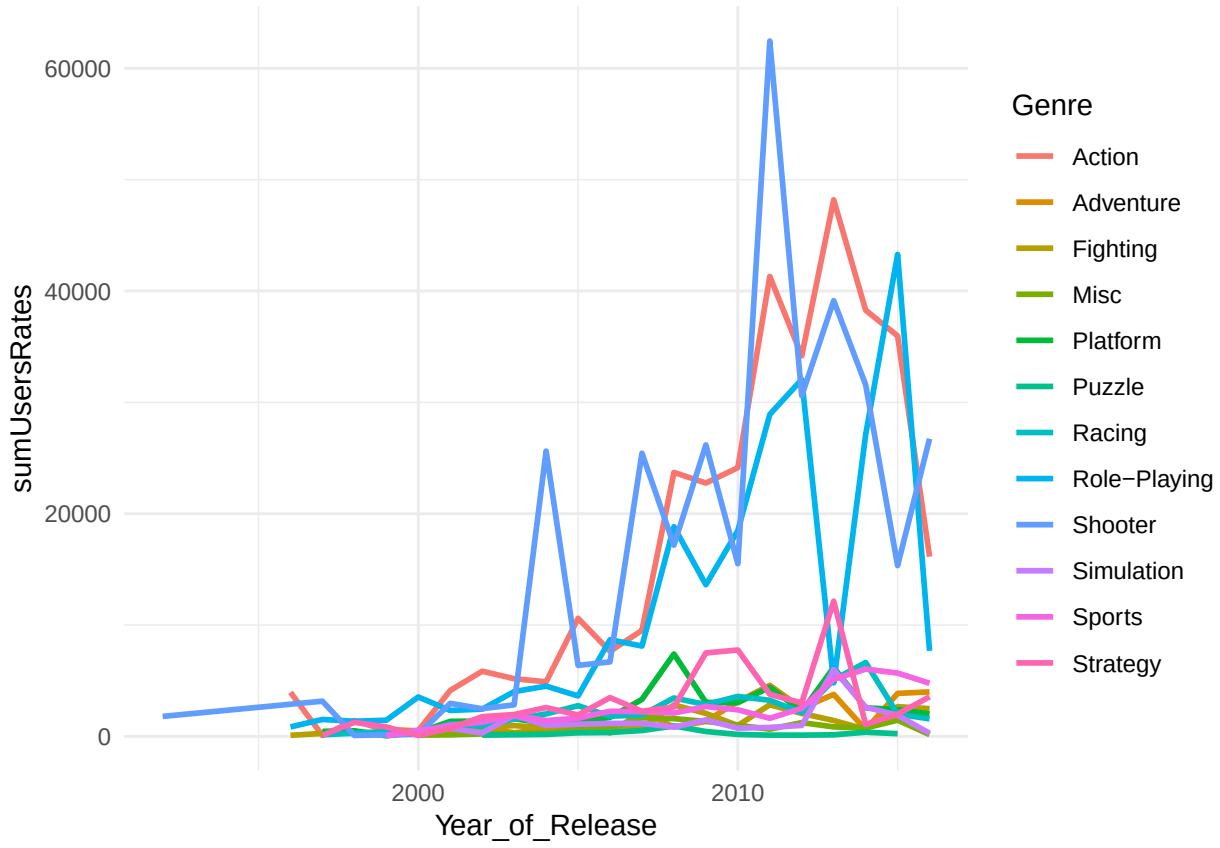
```
## `summarise()` regrouping output by 'Year_of_Release' (override with `.groups` argument)
```



Popular genres through years We can observe here that Shooter, Role-Playing and Action games became very popular since 2000.

```
data %>% filter(Year_of_Release >= 1990 & Year_of_Release <= 2016) %>%
group_by(Year_of_Release, Genre) %>% summarize(sumUsersRates = sum(User_Count, na.rm = TRUE)) %>%
filter( sumUsersRates > 50) %>%
ungroup(Year_of_Release, Genre) %>%
ggplot(aes(x = Year_of_Release, y = sumUsersRates)) +
geom_line(aes(color = Genre), size = 1) +
theme_minimal()
```

```
## `summarise()` regrouping output by 'Year_of_Release' (override with `.groups` argument)
```



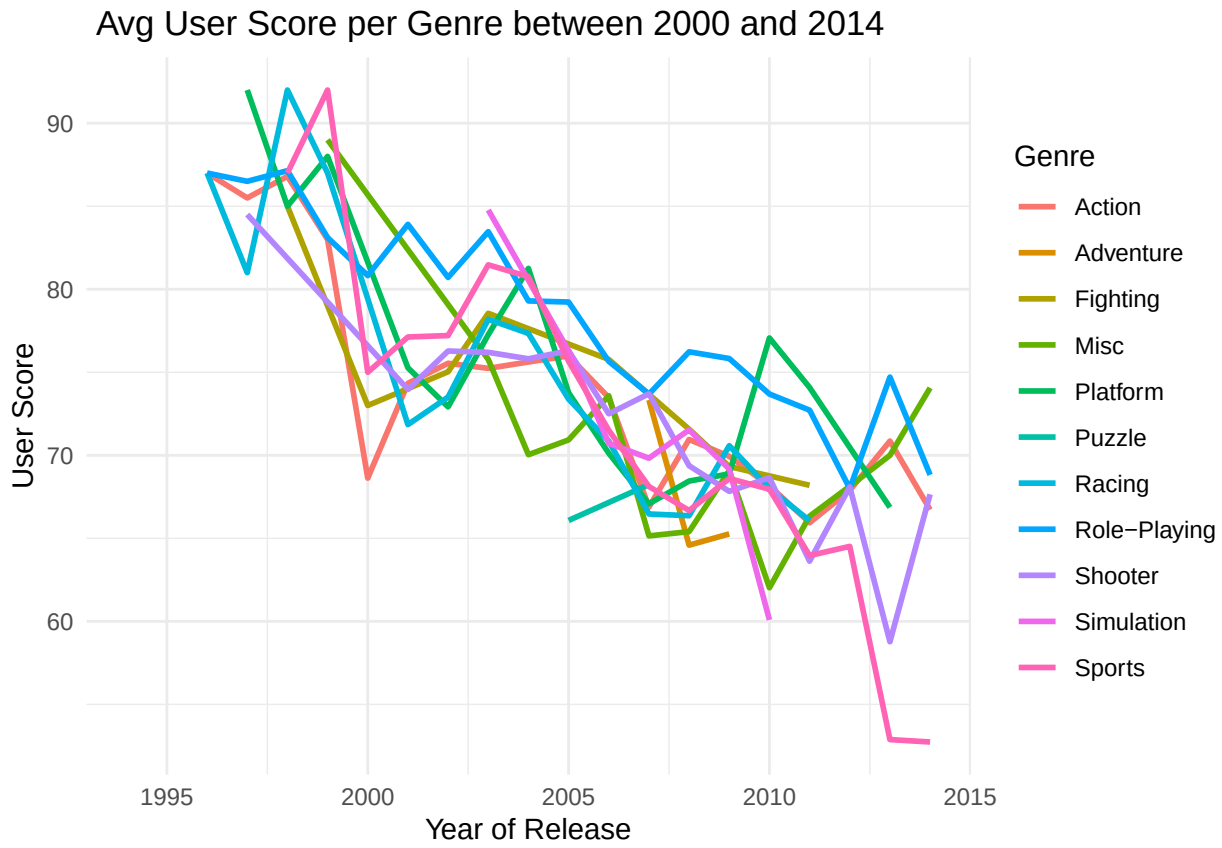
Rating Distribution

Mean Rating per Genre Which genre scores the best amongst game critics and users? We can observe belowe that critics tends to give lower score than users. However, if we look at the trend, critic scores are more constant, where as user scores tends to get lower year after year.

```
data %>% filter(Year_of_Release >= 1994 & Year_of_Release <= 2014) %>%
group_by(Year_of_Release, Genre) %>%
summarize(sales = sum(Global_Sales, na.rm = TRUE),
score_user = mean(User_Score, na.rm = TRUE),
score_critic = mean(Critic_Score, na.rm = TRUE)) %>%
filter( sales > 20) %>%
ungroup(Year_of_Release, Genre) %>%
ggplot(aes(x = Year_of_Release, y = score_user)) +
geom_line(aes(color = Genre), size = 1) +
theme_minimal() +
labs(title = " Avg User Score per Genre between 2000 and 2014" , y = "User Score", x = "Year of Release
```

```
## `summarise()` regrouping output by 'Year_of_Release' (override with `.groups` argument)
```

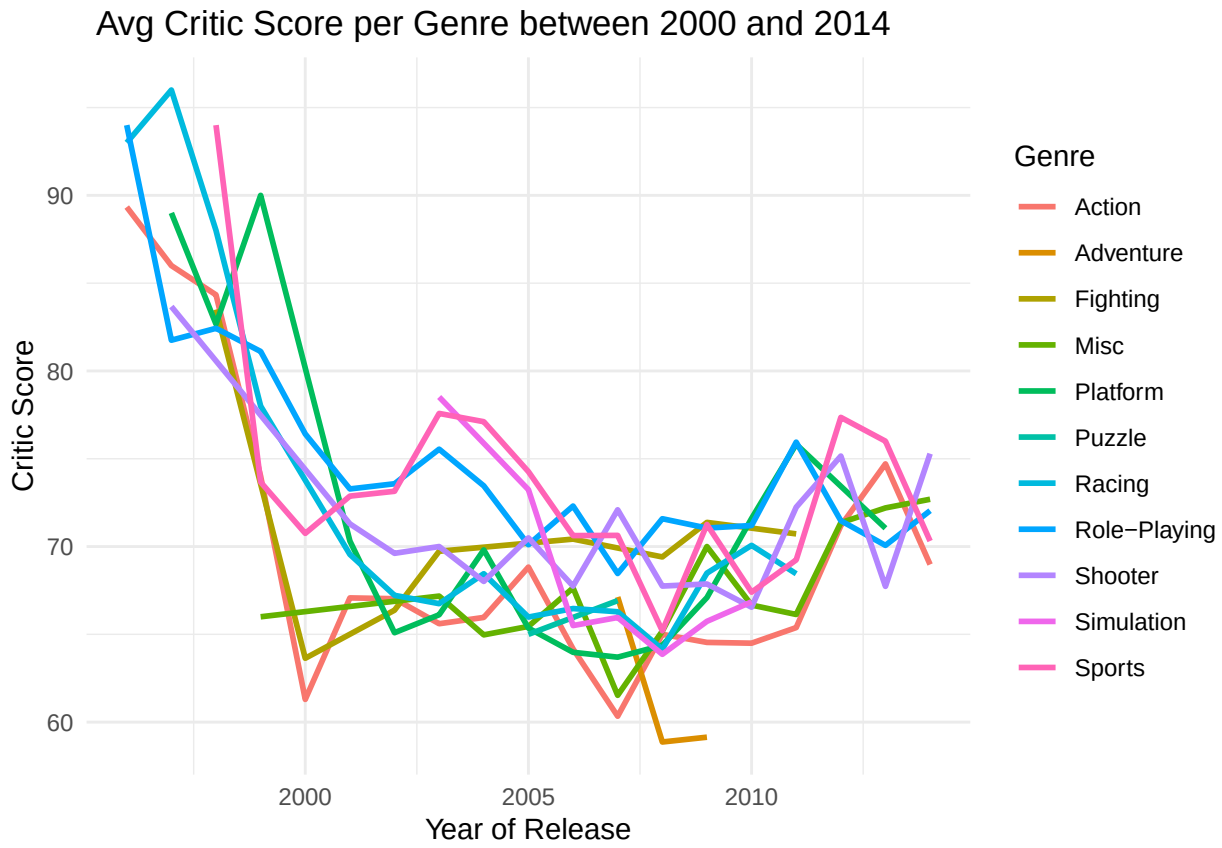
```
## Warning: Removed 3 row(s) containing missing values (geom_path).
```




```
data %>% filter(Year_of_Release >= 1995 & Year_of_Release <= 2014) %>%
group_by(Year_of_Release, Genre) %>%
summarize(sales = sum(Global_Sales, na.rm = TRUE),
score_user = mean(User_Score, na.rm = TRUE),
score_critic = mean(Critic_Score, na.rm = TRUE)) %>%
filter( sales > 20) %>%
ggplot(aes(x = Year_of_Release, y = score_critic)) +
geom_line(aes(color = Genre), size = 1) +
theme_minimal() +
labs(title = " Avg Critic Score per Genre between 2000 and 2014" , y = "Critic Score", x = "Year of Rel
```

```
## `summarise()` regrouping output by 'Year_of_Release' (override with `.groups` argument)
```

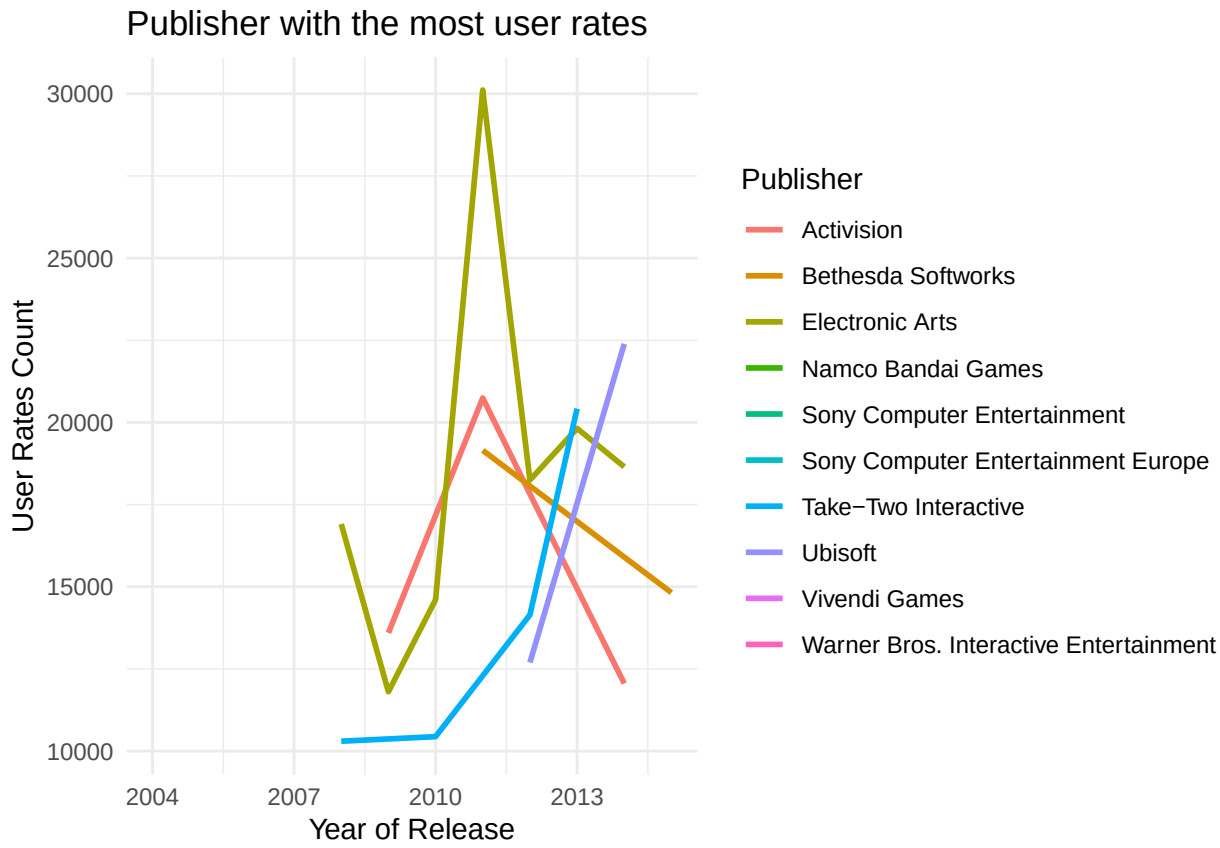
```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```




```
data %>% filter(Year_of_Release >= 1985 & Year_of_Release <= 2016) %>%
group_by(Year_of_Release, Publisher) %>% summarize(sumUsersCount = sum(User_Count, na.rm = TRUE)) %>%
arrange(desc(sumUsersCount)) %>%
filter( sumUsersCount > 10000) %>%
ungroup(Publisher) %>%
ggplot(aes(x = Year_of_Release, y = sumUsersCount)) +
geom_line(aes(color = Publisher), size = 1) +
theme_minimal() +
labs(title = "Publisher with the most user rates", x = "Year of Release", y = "User Rates Count")
```

Mean Rating per Puslisher

```
## `summarise()` regrouping output by 'Year_of_Release' (override with `.groups` argument)
```




```
data %>%
group_by(Publisher) %>% summarize(sumUsersCount = sum(User_Count, na.rm = TRUE)) %>%
arrange(desc(sumUsersCount)) %>%  head(5) %>%
kable() %>%
kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"),
position = "center",
font_size = 10,
full_width = FALSE)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

Publisher	sumUsersCount
Electronic Arts	169765
Activision	121404
Take-Two Interactive	109772
Sony Computer Entertainment	88341
Ubisoft	85994

Insights gained

Through the exploration of the dataset,...

Prediction Strategy

We'll use random forest as an attempt to predict a Critic Score.

```
## Create partitions for Train and Test
# Not all the game have Critic Scores so we get rid of these that miss scores.
data <- data %>% mutate(User_Score = round( User_Score / 10))
data <- data %>% mutate(Critic_Score = round( Critic_Score / 10))
data <- data %>% mutate(!is.na(Critic_Score))
data <- data %>% filter(!is.na(User_Score))
data <- data %>% mutate(User_Score = replace_na(User_Score, 5))
data <- data %>% mutate(Critic_Score = replace_na(Critic_Score, 5))
data <- data %>% group_by(User_Score) %>% filter(n() > 100) %>% ungroup(User_Score)
dim(data)
```

```
## [1] 7481   17
```

```
data <- data %>% filter(!is.na(Critic_Score))
data <- data %>% filter(!is.na(User_Score))
data <- data %>% filter(!is.na(Year_of_Release))
data <- data %>% filter(!is.na(Genre))
data$Genre <- as.factor(data$Genre)
data <- data %>% group_by(Critic_Score) %>% filter(n() > 50) %>% ungroup(Critic_Score)
data <- data %>% group_by(User_Score) %>% filter(n() > 50) %>% ungroup(User_Score)
data <- data %>% group_by(Year_of_Release) %>% filter(n() > 50) %>% ungroup(Year_of_Release)
data <- data %>% group_by(Genre) %>% filter(n() > 50) %>% ungroup(Genre)
data <- data %>% group_by(Publisher) %>% filter(n() > 50) %>% ungroup(Publisher)
data %>% summarise(user_scor_avg = mean(User_Score), critic_score_avg = mean(Critic_Score))
```

```
## # A tibble: 1 x 2
##   user_scor_avg critic_score_avg
##           <dbl>            <dbl>
## 1          7.20             6.98
```

```
data$User_Score <- as.factor(data$User_Score)
data$Critic_Score <- as.factor(data$Critic_Score)
test_index <- createDataPartition(data$Critic_Score, times = 1, p = 0.2, list = FALSE)
test_set <-  data %>% slice(test_index) %>% select(Critic_Score, User_Score, Year_of_Release, Genre)
train_set <- data  %>% slice(-test_index) %>% select(Critic_Score, User_Score, Year_of_Release, Genre)
train_rf <- train(User_Score ~ . , method = "rf", data = train_set)
y_hat = predict(train_rf, newdata = test_set)
test_set <- test_set %>%
mutate(m_prediction = predict(train_rf, newdata = test_set))
y_hat = predict(train_rf, newdata = test_set)
test_set <- test_set %>%
mutate(m_prediction = predict(train_rf, newdata = test_set))
accuracy <- confusionMatrix(predict(train_rf, test_set),
test_set$User_Score)$overall["Accuracy"]
paste("The accuracy of the prediction is : " , accuracy)
```

```
## [1] "The accuracy of the prediction is :  0.403957131079967"
```

```r
RMSE <- function(m_actual_rating, m_predicted_rating){
sqrt(mean((as.numeric(as.character(m_actual_rating)) - as.numeric(as.character(m_predicted_rating)))^2))
}
rmse <- RMSE(test_set$User_Score, test_set$m_prediction)
rmse2 <- RMSE(train_set$User_Score, test_set$m_prediction)
```

```
## Warning in as.numeric(as.character(m_actual_rating)) -
## as.numeric(as.character(m_predicted_rating)): longer object length is not a
## multiple of shorter object length
```

```r
paste("The RMSE (on the train set) of the model is : " , rmse2)
```

```
## [1] "The RMSE (on the train set) of the model is :  1.74156960319871"
```

```r
paste("The RMSE (on the test set) of the model is : " , rmse)
```

```
## [1] "The RMSE (on the test set) of the model is :  1.32419927118632"
```

Results

We could only predict the User Score with a accuracy of around 0.41, which is not that good. But is not that bad either, considering the discrete distribution of the scores spanning 100 possible outcomes (0.100). The RMSE (test set) is approximately 1.41. The fact that the two RMSE are close indicates that the model is not underfit nor overfit.

Conclusion

In this report, we explored the dataset and tried to predict the critic score for games. It happened that we only obtained around 0.41 of accuracy. However, we need to say that the critic score has 100 possible outcomes (0-100), and the accuracy does not take into account how close are the scores that has been badly predicted. For a further analysis it would be interesting to add this aspect when deciding wether the prediction is correct or not, by giving a weight to the correctness of a prediction.