Project 2

Yahya Gahbiche, Dohoon Kim, Quynh Nguyen

11/11/2020

```
# Clear environment of variables and functions
rm(list = ls(all = TRUE))
```

```
# Clear environment of packages
if(is.null(sessionInfo()$otherPkgs) == FALSE)lapply(paste("package:", names(sessionInfo()$otherPkgs), s
```

Libraries

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.0.2
```

```
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------------------------------------
```

```
## v ggplot2 3.3.1     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ---------------------------------------------------------------------------------------
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

```
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

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

```
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.0.2
```

```r
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.0.2
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(gridExtra)
```

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

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

```r
library(tinytex)
```

```
## Warning: package 'tinytex' was built under R version 4.0.2
```

Project Goal

- This project aims to help Peter Parker predict sales in different regions using the given video game dataset. The dataset contains 27 different variables with 55792 observations. We used a machine learning technique to accomplish the goal, specifically, the regression decision tree to predict the outcome variable, global sales. This project is necessary for Peter Parker because it helps him save time, human resources, and money to be allocated effectively and efficiently.

Objective

- The formal objective is to predict future sales based on historical data using the predictive analysis technique.

Load data

```r
# Importing data
game <- read.csv("vgsales-3.csv", header = TRUE)
```

EDA : Step 1

Uni-variate non-graphical Analysis

```r
# Looking at 10 rows of the data
head(game, 10)
```

```
##    Rank                               Name                     basename
## 1     1                         Wii Sports                   wii-sports
## 2     2                  Super Mario Bros.             super-mario-bros
## 3     3                     Mario Kart Wii               mario-kart-wii
## 4     4      PlayerUnknown's Battlegrounds playerunknowns-battlegrounds
## 5     5                  Wii Sports Resort            wii-sports-resort
## 6     6 Pokemon Red / Green / Blue Version                   pokmon-red
## 7     7              New Super Mario Bros.         new-super-mario-bros
## 8     8                             Tetris                       tetris
## 9     9          New Super Mario Bros. Wii     new-super-mario-bros-wii
## 10   10                          Minecraft                    minecraft
##           Genre ESRB_Rating Platform        Publisher             Developer
## 1        Sports           E      Wii         Nintendo          Nintendo EAD
## 2      Platform                  NES         Nintendo          Nintendo EAD
## 3        Racing           E      Wii         Nintendo          Nintendo EAD
## 4       Shooter                   PC PUBG Corporation      PUBG Corporation
## 5        Sports           E      Wii         Nintendo          Nintendo EAD
## 6  Role-Playing           E       GB         Nintendo            Game Freak
## 7      Platform           E       DS         Nintendo          Nintendo EAD
## 8        Puzzle           E       GB         Nintendo Bullet Proof Software
## 9      Platform           E      Wii         Nintendo          Nintendo EAD
## 10         Misc                   PC           Mojang             Mojang AB
##    VGChartz_Score Critic_Score User_Score Total_Shipped Global_Sales NA_Sales
## 1              NA          7.7         NA         82.86           NA       NA
## 2              NA         10.0         NA         40.24           NA       NA
## 3              NA          8.2        9.1         37.14           NA       NA
## 4              NA           NA         NA         36.60           NA       NA
## 5              NA          8.0        8.8         33.09           NA       NA
## 6              NA          9.4         NA         31.38           NA       NA
## 7              NA          9.1        8.1         30.80           NA       NA
## 8              NA           NA         NA         30.26           NA       NA
## 9              NA          8.6        9.2         30.22           NA       NA
## 10             NA         10.0         NA         30.01           NA       NA
##    PAL_Sales JP_Sales Other_Sales Year Last_Update
## 1         NA       NA          NA 2006
## 2         NA       NA          NA 1985
## 3         NA       NA          NA 2008 11th Apr 18
## 4         NA       NA          NA 2017 13th Nov 18
## 5         NA       NA          NA 2009
## 6         NA       NA          NA 1998
## 7         NA       NA          NA 2006
## 8         NA       NA          NA 1989
## 9         NA       NA          NA 2009
## 10        NA       NA          NA 2010 05th Aug 18
##                                                                              url
## 1                       http://www.vgchartz.com/game/2667/wii-sports/?region=All
## 2                 http://www.vgchartz.com/game/6455/super-mario-bros/?region=All
## 3                   http://www.vgchartz.com/game/6968/mario-kart-wii/?region=All
## 4  http://www.vgchartz.com/game/215988/playerunknowns-battlegrounds/?region=All
## 5               http://www.vgchartz.com/game/24656/wii-sports-resort/?region=All
## 6  http://www.vgchartz.com/game/4030/pokemon-red-green-blue-version/?region=All
## 7             http://www.vgchartz.com/game/1582/new-super-mario-bros/?region=All
## 8                           http://www.vgchartz.com/game/4534/tetris/?region=All
## 9      http://www.vgchartz.com/game/35076/new-super-mario-bros-wii/?region=All
## 10                     http://www.vgchartz.com/game/47724/minecraft/?region=All
##    status Vgchartzscore                                        img_url
## 1       1            NA  /games/boxart/full_2258645AmericaFrontccc.jpg
## 2       1            NA                   /games/boxart/8972270ccc.jpg
## 3       1           8.7  /games/boxart/full_8932480AmericaFrontccc.jpg
## 4       1            NA  /games/boxart/full_8052843AmericaFrontccc.jpg
## 5       1           8.8  /games/boxart/full_7295041AmericaFrontccc.jpg
## 6       1            NA  /games/boxart/full_6442337AmericaFrontccc.png
## 7       1            NA  /games/boxart/full_2916260AmericaFrontccc.jpg
## 8       1            NA                   /games/boxart/3740960ccc.jpg
## 9       1           9.1  /games/boxart/full_1410872AmericaFrontccc.jpg
## 10      1            NA /games/boxart/full_minecraft_1AmericaFront.png
```

Comments:

```
- There are 23 indipendent variables and there are no duplicate columns
- Data appears tidy
- There are some independent varialbes that we will not be using, so we will remove them in the data
- There are some NA values, we will ignore those values for now
```

```r
# Data Structure
str(game)
```

```
## 'data.frame':    55792 obs. of  23 variables:
##  $ Rank          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Name          : chr  "Wii Sports" "Super Mario Bros." "Mario Kart Wii" "PlayerUnknown's Battlegrou
##  $ basename      : chr  "wii-sports" "super-mario-bros" "mario-kart-wii" "playerunknowns-battlegroun
##  $ Genre         : chr  "Sports" "Platform" "Racing" "Shooter" ...
##  $ ESRB_Rating   : chr  "E" "" "E" "" ...
##  $ Platform      : chr  "Wii" "NES" "Wii" "PC" ...
##  $ Publisher     : chr  "Nintendo" "Nintendo" "Nintendo" "PUBG Corporation" ...
##  $ Developer     : chr  "Nintendo EAD" "Nintendo EAD" "Nintendo EAD" "PUBG Corporation" ...
##  $ VGChartz_Score: logi  NA NA NA NA NA NA ...
##  $ Critic_Score  : num  7.7 10 8.2 NA 8 9.4 9.1 NA 8.6 10 ...
##  $ User_Score    : num  NA NA 9.1 NA 8.8 NA 8.1 NA 9.2 NA ...
##  $ Total_Shipped : num  82.9 40.2 37.1 36.6 33.1 ...
##  $ Global_Sales  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ NA_Sales      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PAL_Sales     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ JP_Sales      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Other_Sales   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Year          : num  2006 1985 2008 2017 2009 ...
##  $ Last_Update   : chr  "" "" "11th Apr 18" "13th Nov 18" ...
##  $ url           : chr  "http://www.vgchartz.com/game/2667/wii-sports/?region=All" "http://www.vgcha
##  $ status        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Vgchartzscore : num  NA NA 8.7 NA 8.8 NA NA NA 9.1 NA ...
##  $ img_url       : chr  "/games/boxart/full_2258645AmericaFrontccc.jpg" "/games/boxart/8972270ccc.jp
```

Comments:

```
- Some of the variables that are characters need to be converted to factor variables

  - Genre, ESRB_Rating, Platform, Publisher, Developer
```

```r
# Summary
summary(game)
```

```
##       Rank            Name             basename            Genre
##  Min.   :    1   Length:55792       Length:55792       Length:55792
##  1st Qu.:13949   Class :character   Class :character   Class :character
##  Median :27896   Mode  :character   Mode  :character   Mode  :character
##  Mean   :27896
##  3rd Qu.:41844
##  Max.   :55792
##
##  ESRB_Rating          Platform          Publisher          Developer
##  Length:55792       Length:55792       Length:55792       Length:55792
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  VGChartz_Score  Critic_Score     User_Score    Total_Shipped    Global_Sales
##  Mode:logical   Min.   : 1.00   Min.   : 2.00   Min.   : 0.03   Min.   : 0.00
##  NA's:55792     1st Qu.: 6.40   1st Qu.: 7.80   1st Qu.: 0.20   1st Qu.: 0.03
##                 Median : 7.50   Median : 8.50   Median : 0.59   Median : 0.12
##                 Mean   : 7.21   Mean   : 8.25   Mean   : 1.89   Mean   : 0.37
##                 3rd Qu.: 8.30   3rd Qu.: 9.10   3rd Qu.: 1.80   3rd Qu.: 0.36
##                 Max.   :10.00   Max.   :10.00   Max.   :82.86   Max.   :20.32
##                 NA's   :49256   NA's   :55457   NA's   :53965   NA's   :36377
##     NA_Sales       PAL_Sales        JP_Sales      Other_Sales         Year
##  Min.   :0.00    Min.   :0.00    Min.   :0.00    Min.   :0.00    Min.   :1970
##  1st Qu.:0.05    1st Qu.:0.01    1st Qu.:0.02    1st Qu.:0.00    1st Qu.:2000
##  Median :0.12    Median :0.04    Median :0.05    Median :0.01    Median :2008
##  Mean   :0.28    Mean   :0.16    Mean   :0.11    Mean   :0.04    Mean   :2006
##  3rd Qu.:0.29    3rd Qu.:0.14    3rd Qu.:0.12    3rd Qu.:0.04    3rd Qu.:2011
##  Max.   :9.76    Max.   :9.85    Max.   :2.69    Max.   :3.12    Max.   :2020
##  NA's   :42828   NA's   :42603   NA's   :48749   NA's   :40270   NA's   :979
##  Last_Update           url                status  Vgchartzscore
##  Length:55792       Length:55792       Min.   :1   Min.   :2.60
##  Class :character   Class :character   1st Qu.:1   1st Qu.:6.80
##  Mode  :character   Mode  :character   Median :1   Median :7.80
##                                        Mean   :1   Mean   :7.43
##                                        3rd Qu.:1   3rd Qu.:8.50
##                                        Max.   :1   Max.   :9.60
##                                                    NA's   :54993
##    img_url
##  Length:55792
##  Class :character
##  Mode  :character
##
##
##
##
```

Comments:

```
    - Critic_Score: The mean is less the median therefore the data is positively skewed (skewed to the

    - There are many NAs in the following variables:

      - Critic_Score, User_Score, Total_Shipped

      - The reason why there so many NA values in Global_Sales, NA_Sales, PAL_Sales, JP_Sales, and Othe
```

Data wrangling

```r
t(t(names(game)))
```

```
##       [,1]
##  [1,] "Rank"
##  [2,] "Name"
##  [3,] "basename"
##  [4,] "Genre"
##  [5,] "ESRB_Rating"
##  [6,] "Platform"
##  [7,] "Publisher"
##  [8,] "Developer"
##  [9,] "VGChartz_Score"
## [10,] "Critic_Score"
## [11,] "User_Score"
## [12,] "Total_Shipped"
## [13,] "Global_Sales"
## [14,] "NA_Sales"
## [15,] "PAL_Sales"
## [16,] "JP_Sales"
## [17,] "Other_Sales"
## [18,] "Year"
## [19,] "Last_Update"
## [20,] "url"
## [21,] "status"
## [22,] "Vgchartzscore"
## [23,] "img_url"
```

```r
#Select variables
game_newdata <- game %>%
  select(c(4:8, 13, 18))

#Veriying changes
str(game_newdata)
```

```
## 'data.frame':    55792 obs. of  7 variables:
##  $ Genre       : chr  "Sports" "Platform" "Racing" "Shooter" ...
##  $ ESRB_Rating : chr  "E" "" "E" "" ...
##  $ Platform    : chr  "Wii" "NES" "Wii" "PC" ...
##  $ Publisher   : chr  "Nintendo" "Nintendo" "Nintendo" "PUBG Corporation" ...
##  $ Developer   : chr  "Nintendo EAD" "Nintendo EAD" "Nintendo EAD" "PUBG Corporation" ...
##  $ Global_Sales: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Year        : num  2006 1985 2008 2017 2009 ...
```

```r
#Changing data type
game_newdata$Genre <- as.factor(game_newdata$Genre)
game_newdata$ESRB_Rating <- as.factor(game_newdata$ESRB_Rating)
game_newdata$Platform <- as.factor(game_newdata$Platform)
game_newdata$Publisher <- as.factor(game_newdata$Publisher)
game_newdata$Developer <- as.factor(game_newdata$Developer)
game_newdata$Year <- as.Date(as.character(game_newdata$Year), format = "%Y")
```

```
str(game_newdata)
```

```
## 'data.frame':    55792 obs. of  7 variables:
##  $ Genre       : Factor w/ 20 levels "Action","Action-Adventure",..: 18 11 13 16 18 14 11 12 11 7 ..
##  $ ESRB_Rating : Factor w/ 9 levels "","AO","E","E10",..: 3 1 3 1 3 3 3 3 3 1 ...
##  $ Platform    : Factor w/ 74 levels "2600","3DO","3DS",..: 65 42 65 48 65 25 21 25 65 48 ...
##  $ Publisher   : Factor w/ 3069 levels "][ Games","@unepic_fran",..: 1883 1883 1883 2151 1883 1883 18
##  $ Developer   : Factor w/ 8065 levels "",".theprodukkt",..: 4984 4984 4984 5635 4984 2726 4984 1159
##  $ Global_Sales: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Year        : Date, format: "2006-11-15" "1985-11-15" ...
```

EDA: Step 2

Univariate Graphical EDA

```
#Genre
graph1 <- game_newdata %>%
  count(Genre) %>%
  top_n(10) %>%
  ggplot(mapping = aes(x = reorder(Genre,n), y = n)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme_classic() +
  labs(title = "Popular Genre",
       x = "Types of Genre",
       y = "Genre Count")
```

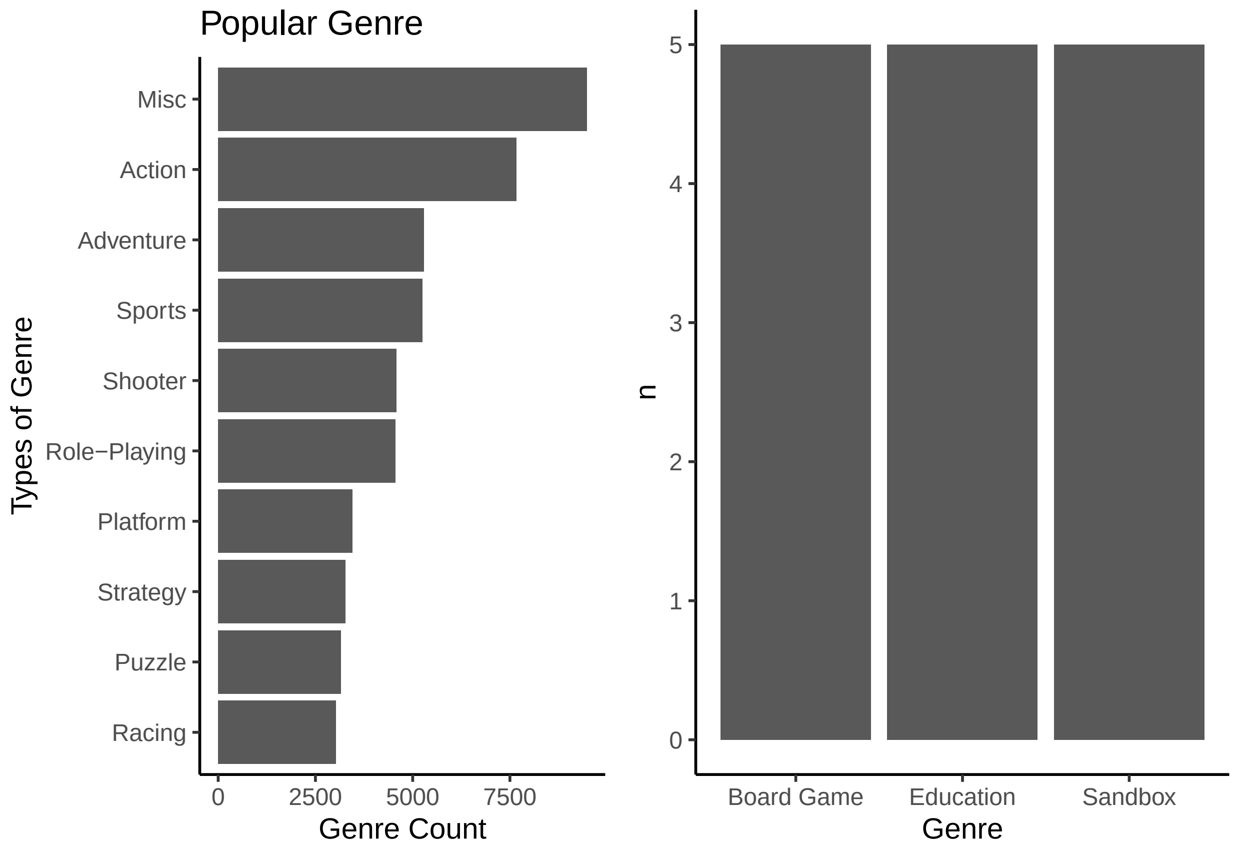
```
## Selecting by n
```

```
#Closer look at the low count of Genre level
graph2 <- game_newdata %>%
  filter(Genre == c("Board Game", "Education", "Sandbox")) %>%
  count(Genre) %>%
  ggplot(mapping = aes(x = Genre, y = n)) +
  geom_bar(stat = "identity") +
  theme_classic()
```

```
## Warning in `==.default`(Genre, c("Board Game", "Education", "Sandbox")): longer
## object length is not a multiple of shorter object length
```

```
## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length
```

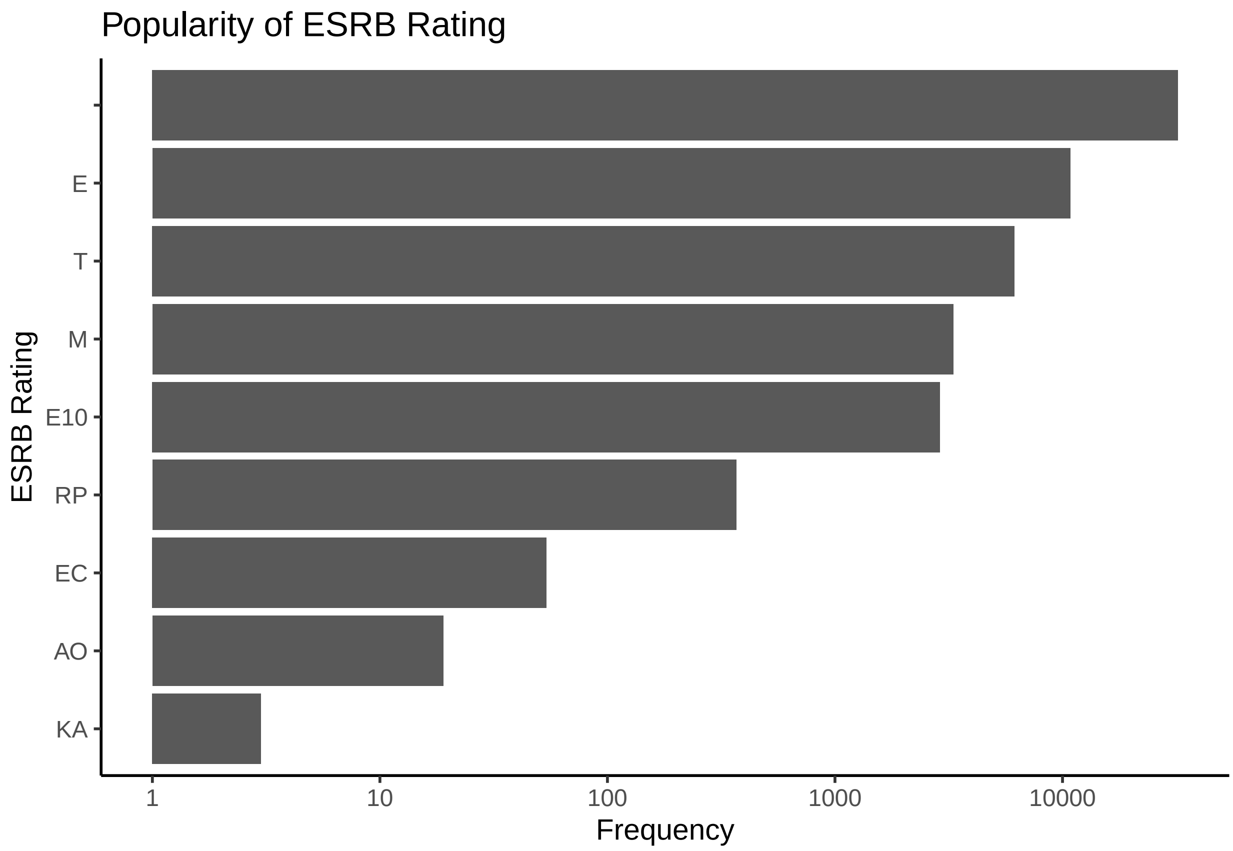
```
grid.arrange(graph1, graph2, ncol = 2)
```



Comments

- Based on bar graph illustrating popularity of the Genre, Misc is the most popular and followed by Action, Adventure, Sports, Shooter, Role-Playing, Platform, Strategy, Puzzle, and Racing.
    - Board Game, Education, and Sandbox have a very low observation in the dataset;therefore, we decided to exclude them from further analysis

```
#ESRB_Rating
game_newdata %>%
  count(ESRB_Rating) %>%
  ggplot(mapping = aes(x = reorder(ESRB_Rating,n), y = n)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  scale_y_log10() +
  theme_classic() +
  labs(title = "Popularity of ESRB Rating",
       x = "ESRB Rating",
       y = "Frequency")
```

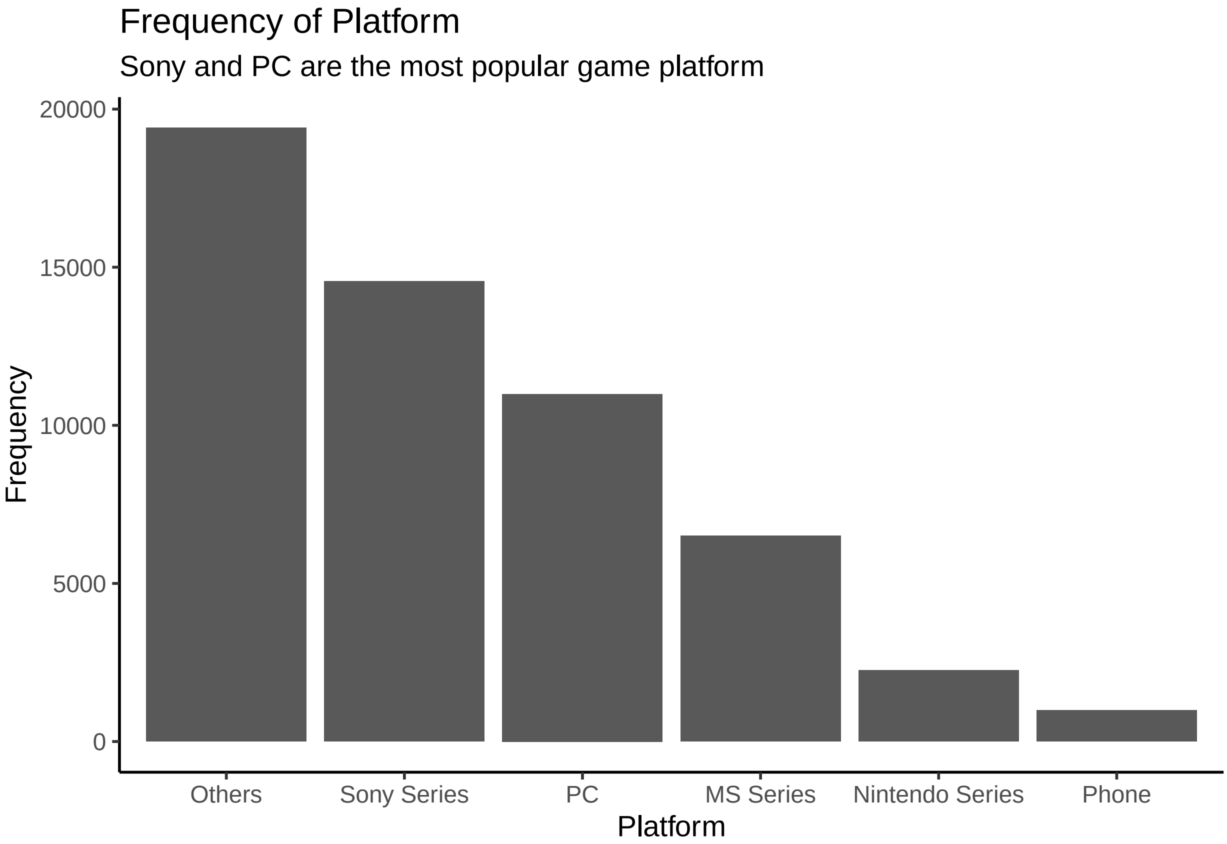


Comments

- The large number of ERSB ratings is missing from the dataset.
- E is the most popular as it includes all ages and followed by T, M, and E10.

```r
#platform
game_newdata %>%
  select(Platform, Year) %>%
  filter(Year >= 2010) %>%
  group_by(Platform) %>%
  summarize(count_plat = n()) %>%
  arrange(desc(count_plat)) %>%
  mutate(Platform = case_when(Platform %like% 'PS' ~ "Sony Series",
                              Platform %in% '3DS' ~ "Nintendo Series",
                              Platform %like% 'NS' ~ "Nintendo Series",
                              Platform %like% 'X' ~ "MS Series",
                              Platform %like% 'And|ios' ~ "Phone",
                              Platform %like% 'PC' ~ "PC",
                              TRUE ~ 'Others')) %>%
  group_by(Platform) %>%
  summarise(count_plat = sum(count_plat)) %>%
  ggplot(mapping = aes(x = reorder(Platform, -count_plat), y = count_plat)) +
  geom_bar(stat = "identity") +
  theme_classic() +
  labs(x = "Platform",
       y = "Frequency",
       title = "Frequency of Platform",
       subtitle = "Sony and PC are the most popular game platform")
```

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



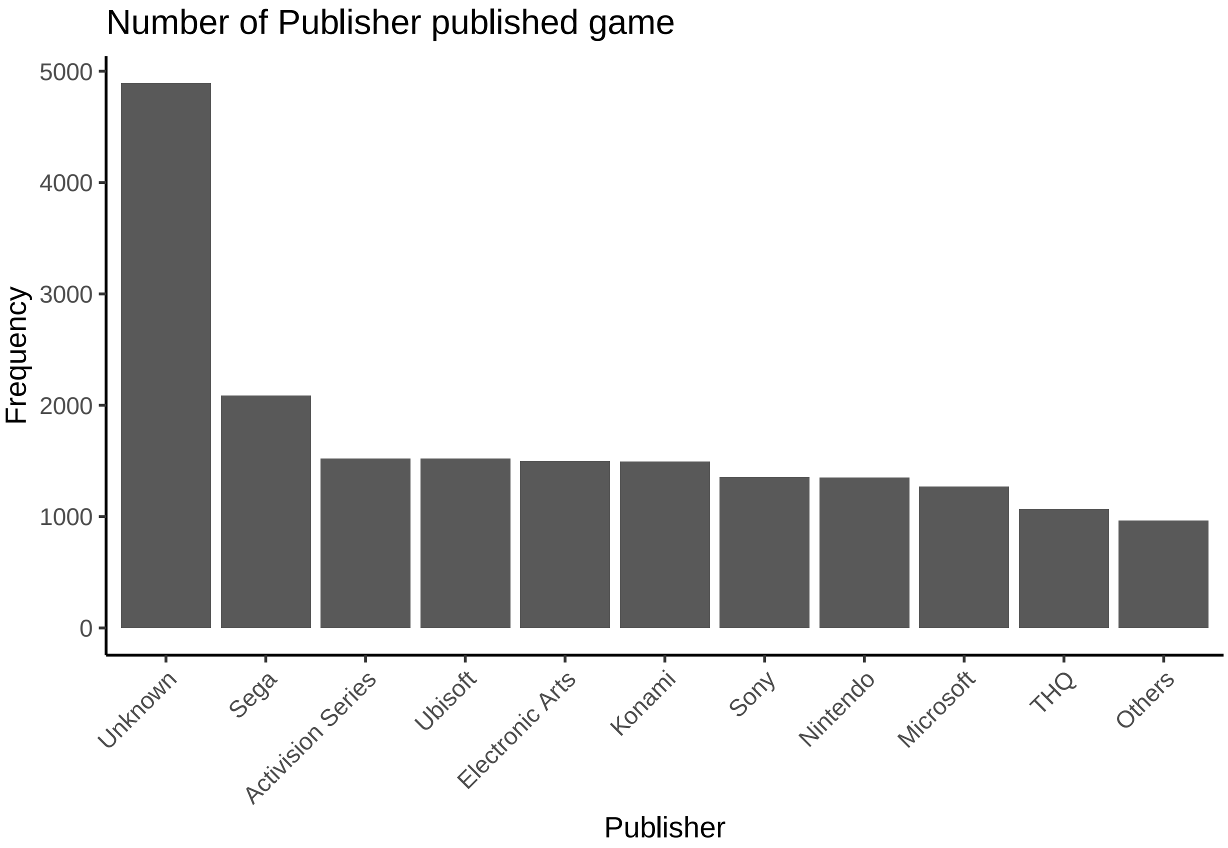
Comments

- "Others" represent the combinations of multiple small platforms that are not very popular, hence grouping them in one category will help us with our analysis.
- Sony is the leader in the sales by volume, followed by PC, Microsoft and Nintendo. However, Phones (android and ios) have the lowest count which can be due to lack of compatibility in the gaming industry

```
game_newdata %>%
  select(Publisher) %>%
  group_by(Publisher) %>%
  summarize(count_pub = n()) %>%
  top_n(11) %>%
  mutate(Publisher = case_when(Publisher %like% 'Unknown' ~ "Unknown",
                              Publisher %in% 'Sega' ~ "Sega",
                              Publisher %like% 'Activision' ~ "Activision Series",
                              Publisher %like% 'Ubisoft' ~ "Ubisoft",
                              Publisher %like% 'Electronic Arts' ~ "Electronic Arts",
                              Publisher %like% 'Konami' ~ "Konami",
                              Publisher %like% 'Sony Computer Entertainment' ~ "Sony",
                              Publisher %like% 'Nintendo' ~ "Nintendo",
                              Publisher %like% 'Microsoft' ~ "Microsoft",
                              Publisher %like% 'THQ' ~ "THQ",
                              TRUE ~ "Others")) %>%
  ggplot(mapping = aes(reorder(x = Publisher, -count_pub), y = count_pub)) +
  geom_bar(stat = "identity") +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 45,
                                   hjust = 1)) +
  labs(title = "Number of Publisher published game",
       x = "Publisher",
       y = "Frequency")
```

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

## Selecting by count_pub
```

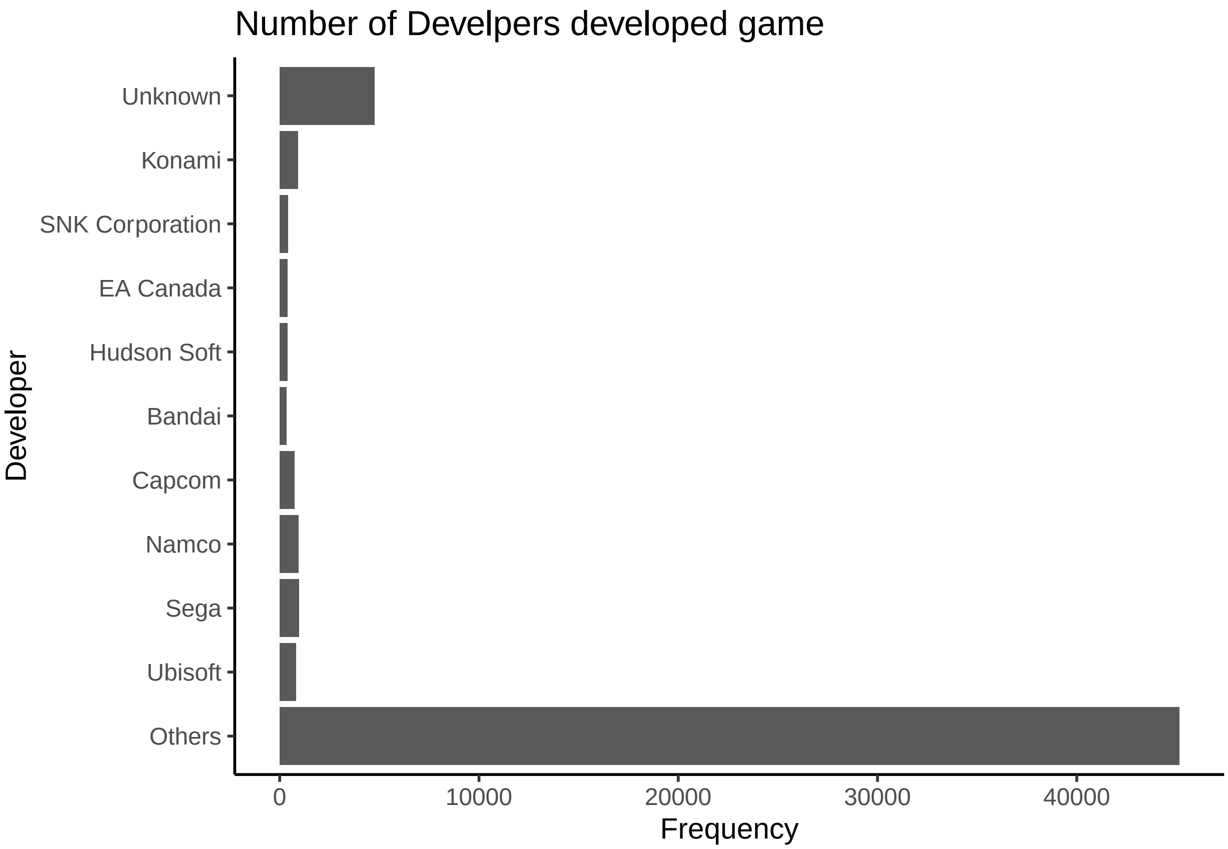


Comments

- Many dataset is marked as unknown.
- Sega outperforms the rest of the Publisher (over 2000). The remaining publishers have a relatively similar count.

```
game_newdata %>%
  select(Developer) %>%
  group_by(Developer) %>%
  summarize(count_dev = n()) %>%
  arrange(desc(count_dev)) %>%
  mutate(Developer = case_when(Developer %like% 'Unknown' ~ "Unknown",
                               Developer %in% 'Konami' ~ "Konami",
                               Developer %like% 'Sega' ~ "Sega",
                               Developer %like% 'Capcom' ~ "Capcom",
                              Developer %like% 'Namco' ~ "Namco",
                              Developer %like% 'SNK Corporation' ~ "SNK Corporation",
                              Developer %like% 'Hudson Soft' ~ "Hudson Soft",
                              Developer %like% 'EA Canada' ~ "EA Canada",
                              Developer %like% 'Bandai' ~ "Bandai",
                              Developer %like% 'Ubisoft' ~ "Ubisoft",
                              TRUE ~ "Others")) %>%
  ggplot(mapping = aes(x = reorder(Developer, count_dev), y = count_dev)) +
  geom_bar(stat = "identity") +
  theme_classic() +
  coord_flip() +
  labs(title = "Number of Develpers developed game",
       x = "Developer",
       y = "Frequency")
```

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



Comments

- To reduce the number of levels in Developer, we aggregated all data which is not listed in the top 10 as “Others”.
- Konami, Sega, Namco, Ubisoft, and Capcom outperfrom other developers. They have released the most popular games

EDA: Step:3

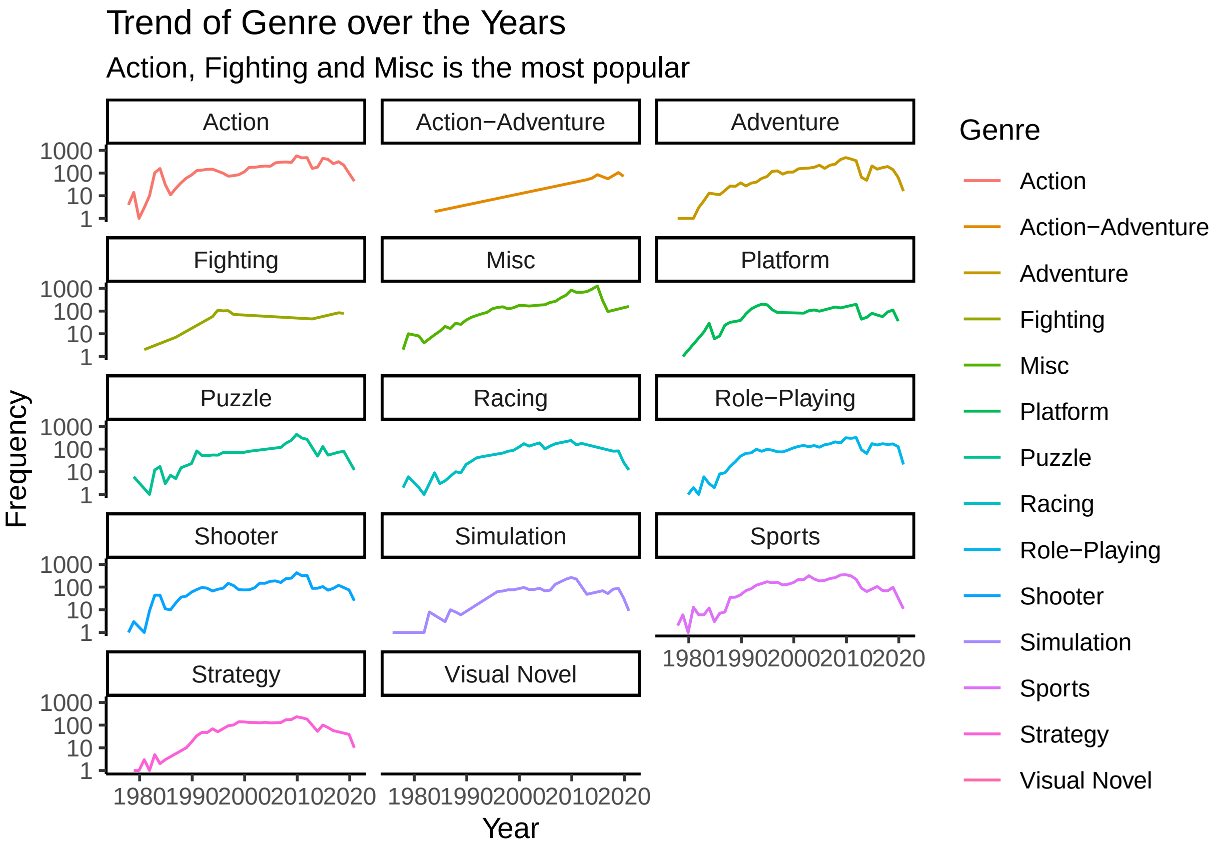
Multivariate Graphical EDA

```
#Trend of Genre over the years
game_newdata %>%
  select(Genre, Year) %>%
  filter(Year >= 2010) %>%
  group_by(Year, Genre) %>%
  summarize(count_genre = n()) %>%
  top_n(10) %>%
  arrange(desc(count_genre)) %>%
  ggplot(mapping = aes(x = Year, y = count_genre, color = Genre)) +
  scale_y_log10() +
  geom_line() +
  facet_wrap(~Genre, nrow = 5) +
  theme_classic() +
  labs(title = "Trend of Genre over the Years",
       subtitle = "Action, Fighting and Misc is the most popular",
       y = "Frequency")
```

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

```
## Selecting by count_genre
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```



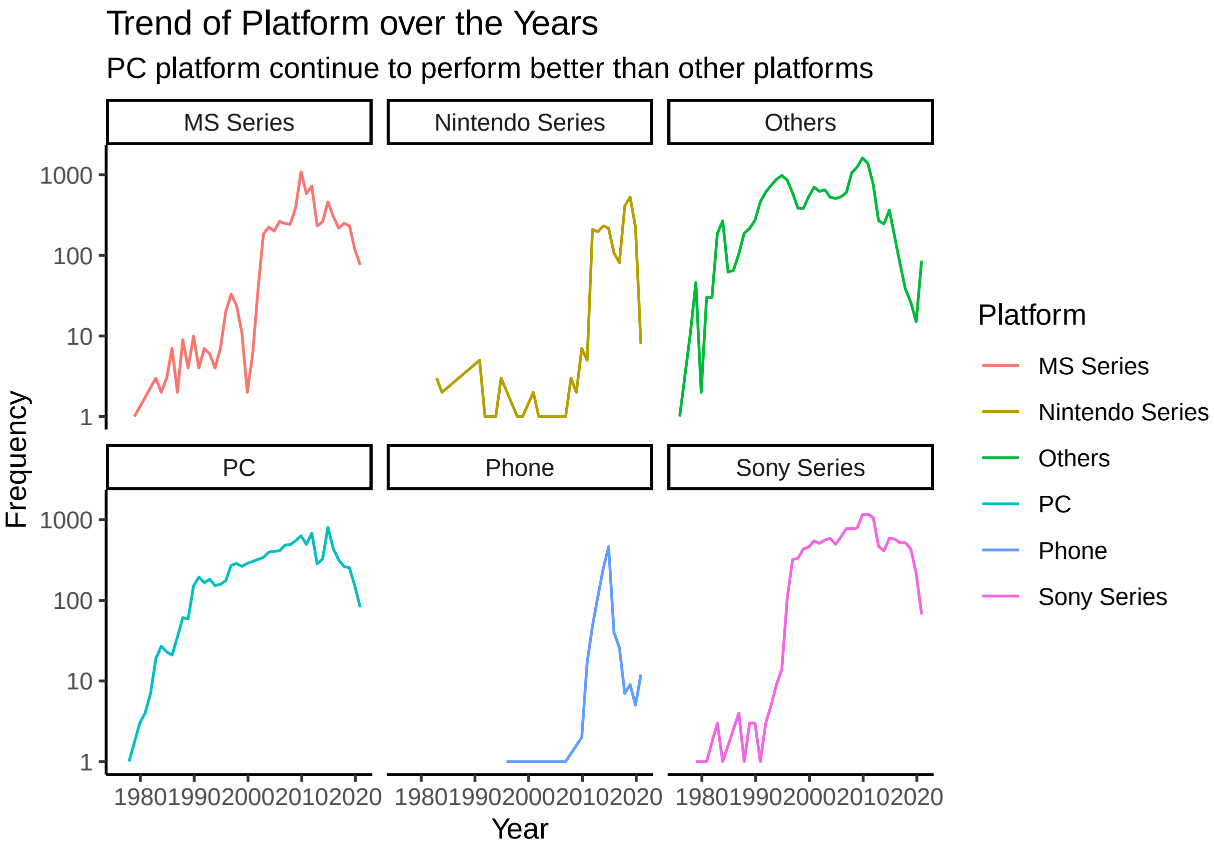
Comments

- Except for Action-Adeventure genre, others experience fluctuation afer 2010.
  - This seems that the demand for the gaming industry is slowly decreasing as other technologies bring entertainment for people such as mobile streaming services. (Twitch and Youtube)
  - However, the action-adventure genre is steadily beloved by people even though the gaming industry is not performing well.

```r
#Trend of Platform over the years
game_newdata %>%
  mutate(Platform = case_when(Platform %like% 'PS' ~ "Sony Series",
                              Platform %in% '3DS' ~ "Nintendo Series",
                              Platform %like% 'NS' ~ "Nintendo Series",
                              Platform %like% 'X' ~ "MS Series",
                              Platform %like% 'And|ios' ~ "Phone",
                              Platform %like% 'PC' ~ "PC",
                              TRUE ~ 'Others')) %>%
  select(Platform, Year) %>%
  filter(Year >= 2010) %>%
  group_by(Year, Platform) %>%
  summarize(count_plat = n()) %>%
  top_n(10) %>%
  arrange(desc(count_plat)) %>%
  ggplot(mapping = aes(x = Year, y = count_plat, color = Platform)) +
  scale_y_log10() +
  geom_line() +
  facet_wrap(~Platform) +
  theme_classic() +
  labs(title = "Trend of Platform over the Years",
       subtitle = "PC platform continue to perform better than other platforms",
       y = "Frequency")
```

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

```
## Selecting by count_plat
```



Comments

- As we observed a sudden decrease in Genre, this phenomenon happens with the platform over the years.
- Nintendo comparing with other platforms such as Sony, MS, and PC experiences more sudden decrease after 2010.
  - However, the decrease in PC is not as drastic as other platforms.

Reshaping the data

```
#transform platform variable
game_newdata<- game_newdata %>%
    filter(Year >= "2010-01-01") %>%
    mutate(Platform_new = case_when(Platform %like% 'PS' ~ "Sony Series",
                              Platform %in% '3DS' ~ "Nintendo Series",
                              Platform %like% 'NS' ~ "Nintendo Series",
                              Platform %like% 'X' ~ "MS Series",
                              Platform %like% 'And|ios' ~ "Phone",
                              Platform %like% 'PC' ~ "PC",
                              TRUE ~ 'Others')) %>%
    mutate(Publisher_new = case_when(Publisher %like% 'Unknown' ~ "Unknown",
                              Publisher %in% 'Sega' ~ "Sega",
                              Publisher %like% 'Activision' ~ "Activision Series",
                              Publisher %like% 'Ubisoft' ~ "Ubisoft",
                              Publisher %like% 'Electronic Arts' ~ "Electronic Arts",
                              Publisher %like% 'Konami' ~ "Konami",
                              Publisher %like% 'Sony Computer Entertainment' ~ "Sony",
```

```r
                               Publisher %like% 'Nintendo' ~ "Nintendo",
                               Publisher %like% 'Microsoft' ~ "Microsoft",
                               Publisher %like% 'THQ' ~ "THQ",
                               TRUE ~ "Others")) %>%
    mutate(Genre_new = case_when(Genre %like% 'Misc' ~ "Misc",
                               Genre %in% 'Action' ~ "Action",
                               Genre %like% 'Adventure' ~ "Adventure",
                               Genre %like% 'Sports' ~ "Sports",
                               Genre %like% 'Shooter' ~ "Shooter",
                               Genre %like% 'Role-playing' ~ "Role-playing",
                               Genre %like% 'Platform' ~ "Platform",
                               Genre %like% 'Strategy' ~ "Strategy",
                               Genre %like% 'Puzzle' ~ "Puzzle",
                               Genre %like% 'Racing' ~ "Racing",
                               TRUE ~ "Others")) %>%
    mutate(Developer_new = case_when(Developer %like% 'Unknown' ~ "Unknown",
                               Developer %in% 'Konami' ~ "Konami",
                               Developer %like% 'Sega' ~ "Sega",
                               Developer %like% 'Capcom' ~ "Capcom",
                               Developer %like% 'Namco' ~ "Namco",
                               Developer %like% 'SNK Corporation' ~ "SNK Corporation",
                               Developer %like% 'Hudson Soft' ~ "Hudson Soft",
                               Developer %like% 'EA Canada' ~ "EA Canada",
                               Developer %like% 'Bandai' ~ "Bandai",
                               Developer %like% 'Ubisoft' ~ "Ubisoft",
                               TRUE ~ "Others"))

#drop old variables
game_newdata <- game_newdata %>%
  select(-c(1,3:5))
#verifying changes
str(game_newdata)
```

```
## 'data.frame':    20237 obs. of  7 variables:
##  $ ESRB_Rating  : Factor w/ 9 levels "","AO","E","E10",..: 1 1 3 7 7 3 3 3 7 3 ...
##  $ Global_Sales : num  NA NA NA 20.3 19.4 ...
##  $ Year         : Date, format: "2017-11-15" "2010-11-15" ...
##  $ Platform_new : chr  "PC" "PC" "MS Series" "Sony Series" ...
##  $ Publisher_new: chr  "Others" "Others" "Microsoft" "Others" ...
##  $ Genre_new    : chr  "Shooter" "Misc" "Others" "Action" ...
##  $ Developer_new: chr  "Others" "Others" "Others" "Others" ...
```

Predictive Analysis

Training validation split

```r
#setting Seed
set.seed(666)
```

```r
#Training validation split
train_index <- sample(1:nrow(game_newdata), 0.6 * nrow(game_newdata))
valid_index <- setdiff(1:nrow(game_newdata), train_index)
```

```
train_df <- game_newdata[train_index,]
valid_df <- game_newdata[valid_index,]
```

```
#Data transformation
game_newdata$Year <- as.factor(game_newdata$Year)
game_newdata$Platform_new <- as.factor(game_newdata$Platform_new)
game_newdata$Publisher_new <- as.factor(game_newdata$Publisher_new)
game_newdata$Developer_new <- as.factor(game_newdata$Developer_new)
game_newdata$Genre_new <- as.factor(game_newdata$Genre_new)
```

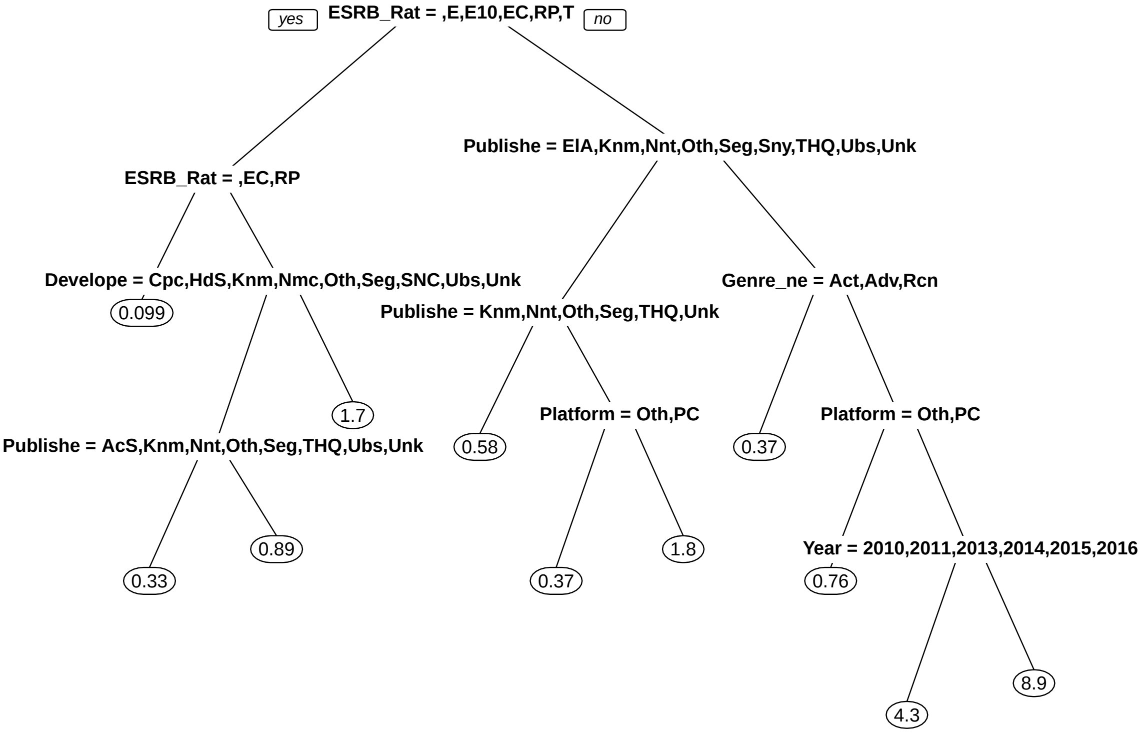
```
train_df$Year <- as.factor(train_df$Year)
train_df$Platform_new <- as.factor(train_df$Platform_new)
train_df$Publisher_new <- as.factor(train_df$Publisher_new)
train_df$Developer_new <- as.factor(train_df$Developer_new)
train_df$Genre_new <- as.factor(train_df$Genre_new)
```

```
valid_df$Year <- as.factor(valid_df$Year)
valid_df$Platform_new <- as.factor(valid_df$Platform_new)
valid_df$Publisher_new <- as.factor(valid_df$Publisher_new)
valid_df$Developer_new <- as.factor(valid_df$Developer_new)
valid_df$Genre_new <- as.factor(valid_df$Genre_new)
```

Regression Tree 1

```
#Building regression tree
regress_tr <- rpart(Global_Sales ~ ESRB_Rating + Year + Publisher_new + Developer_new + Genre_new + Pla
                    data = train_df, method = "anova", maxdepth = 10)
```

```
prp(regress_tr)
```


yes
ESRB_Rat = ,E,E10,EC,RP,T
no
ESRB_Rat = ,EC,RP
Publishe = ElA,Knm,Nnt,Oth,Seg,Sny,THQ,Ubs,Unk
Develope = Cpc,HdS,Knm,Nmc,Oth,Seg,SNC,Ubs,Unk
0.099
Publishe = Knm,Nnt,Oth,Seg,THQ,Unk
Genre_ne = Act,Adv,Rcn
1.7
Publishe = AcS,Knm,Nnt,Oth,Seg,THQ,Ubs,Unk
0.58
Platform = Oth,PC
0.37
Platform = Oth,PC
0.33
0.89
0.37
1.8
Year = 2010,2011,2013,2014,2015,2016
0.76
4.3
8.9

Training validation split

```
#setting Seed
set.seed(666)
```

Accuracy

```
#Predict using the train and valid
predict_train <- predict(regress_tr, train_df)
accuracy(predict_train, train_df$Global_Sales)
```

```
##                   ME      RMSE       MAE  MPE MAPE
## Test set 3.17732e-17 0.8647234 0.3440641 -Inf  Inf
```

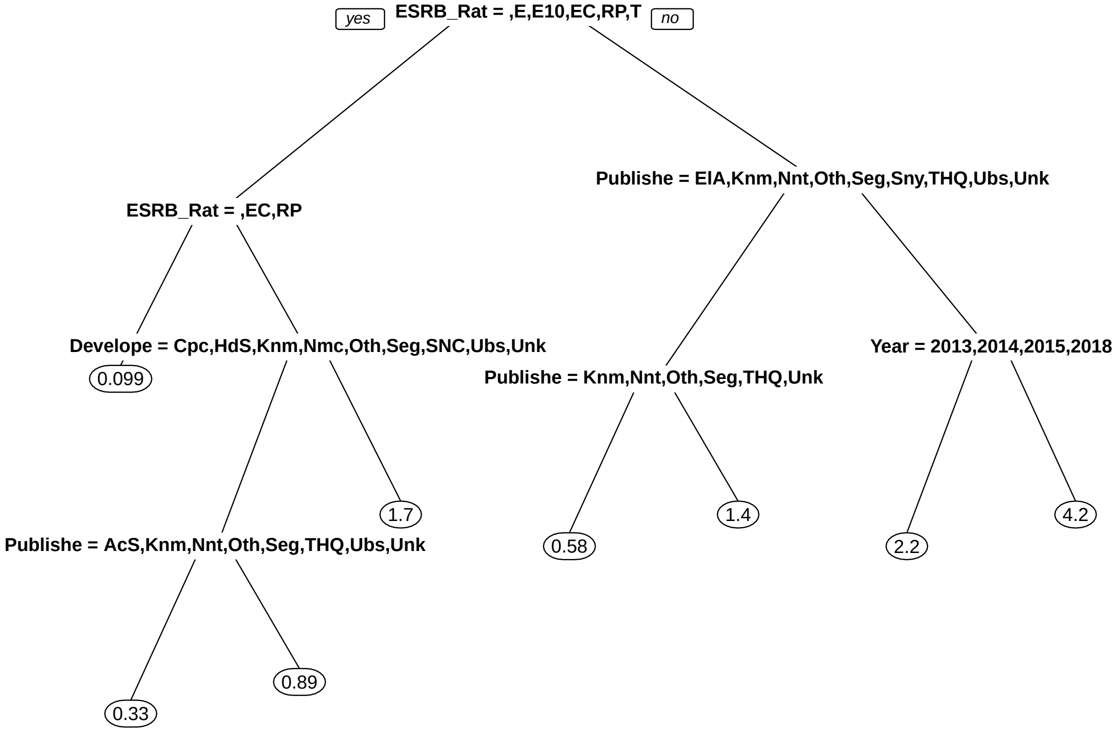
```
predict_valid <- predict(regress_tr, valid_df)
accuracy(predict_valid, valid_df$Global_Sales)
```

```
##                  ME      RMSE       MAE  MPE MAPE
## Test set 0.01483655 0.8844269 0.3541208 -Inf  Inf
```

Regression Tree 2

```
regress_tr_shallow <- rpart(Global_Sales ~ ESRB_Rating + Year + Publisher_new + Developer_new,
                           data = train_df, method = "anova",
                           minbucket = 2, maxdepth = 10)
```

```
prp(regress_tr_shallow)
```


yes
ESRB_Rat = ,E,E10,EC,RP,T
no
ESRB_Rat = ,EC,RP
Publishe = EIA,Knm,Nnt,Oth,Seg,Sny,THQ,Ubs,Unk
Develope = Cpc,HdS,Knm,Nmc,Oth,Seg,SNC,Ubs,Unk
0.099
Publishe = Knm,Nnt,Oth,Seg,THQ,Unk
Year = 2013,2014,2015,2018
1.7
Publishe = AcS,Knm,Nnt,Oth,Seg,THQ,Ubs,Unk
0.58
1.4
2.2
4.2
0.33
0.89

Accuracy

```r
#Predict using the train and valid
predict_train_shallow <- predict(regress_tr_shallow, train_df)
accuracy(predict_train_shallow, train_df$Global_Sales)
```

```
##                    ME    RMSE       MAE  MPE MAPE
## Test set 4.32022e-17 0.92078 0.3655189 -Inf  Inf
```

```r
predict_valid_shallow <- predict(regress_tr_shallow, valid_df)
accuracy(predict_valid_shallow, valid_df$Global_Sales)
```

```
##                  ME      RMSE       MAE  MPE MAPE
## Test set 0.02180537 0.9073625 0.3609558 -Inf  Inf
```

Comments

- Final datamining model
  - The algorithm that we used to predict global sales is regression decision tree. This algorithm is widely used when we want to predict the numerical outcome variables (global sales). The algorithm will choose the variables that are the most significant in predicting the outcome variable.
  - The transformed dataset to build an algorithm contains fewer variables compared to the original dataset. Therefore, we decided to keep all variables to build an algorithm but reduce the number of factors in each variable to avoid problems such as overfitting and complexity of the algorithm.
  - Before building the model predicting the volume of the gloabl sales, we assumed that highest sales is associated with recent years as gaming industry is evolving continously.
  - We also assumped that ESRB rating A and M will bring the most sales because they hold more buying power compared to teenagers.
- Interpretation of findings
  - Since the goal of the project is predicting the highest sales of the target variable, we are going to focous only on highest end node that has highest number in sales (8.9 million dollars).
  - If the ESRB is not equal to NA, E, E10, EC RP, and T then we move to Publisher.
  - If the Publisher is not equal to EIA, Konami, Nintento, Others, Sega, Sony, THQ, Ubisoft, and Unk, then we move to Genre.
  - If the Genre is not equal to ACT, ADV, RCN, then we move to Platform
  - If the platform is not equal to others and PC then we move to Year
  - If the Year is not equal to 2010-2011, 2013-2016, then we predict that the global sales will be 8.9 million dollars
- Quality of the model
  - The RMSE of the first model in the training dataset is 0.8647, whereas the RMSE of the validation dataset is 0.8844. This represents that the model is not overfitting, as the difference in RMSE is small.
  - The RMSE of the second model in the training dataset is 0.92078, whereas the RMSE of the validation dataset is 0.9074. This represents that the model has an overfitting problem and is not good for predicting global sales.
- Recommendations
  - Based on the findings, our recomeendations are as follows:

    - In order to maximize the global sales, we should focus on games targeting towards adults and mature audience. In addition, Peter Parker should focus on games that has been released on the yaer of 2010-2011, 2013-2016 to maximize the profit of the sales. Furthermore, he needs to focous on platform that is not PC and Genre that is not Action, Adventure and RCN to maximize the profit.

- Sustainability of the project.
  - Because of rapid changes in the gaming industry, Peter Parker should update the dataset every quarter to follow the change of trend and build the model accordingly.
- What other data can be used to enhance the model?
  - Adding more variables such as E-Sports related, Twitch, and YouTube can help to enhance the model.
  - Upon the new release of PS5 and Xbox, we should update our dataset to capture the outcome variable more accurately.
  - Furthermore, with the enhancement of technology, VR technology is becoming more popular, and we should consider including these in the future dataset.