Implementation of Data Analysis On a Video Game Sales Data using R programming (1991 – 2010) – Report + Source Code

1- Introduction:

This report conducts with a dataset coming from the video game industry which captures the sales of video games throughout 40 years, from the year 1976 until 2017, and across different gaming platforms, and by different publishers, and in the 3 main regions in the video games' market (United States, Europe, and Japan).

The purpose of this report is to observe the following patterns; how the sales are distributed among the different regions, which are the main gaming publishers leading the gaming industry in terms of sales, and what game genres are selling the most among, which could be linked to the culture spread amongst the video game players to better understand the behavioral patterns of these players and provide help when necessary.

The data shall be preprocessed, a simple linear regression model is built and tested for its fitness in order to predict future global sales, and Decision Tree and Support Vector Machines models will be trained and tested on the data to classify game titles and acquire the rating of these games which indicates the fitness for different ages based on the game's content.

2-Data used:

The video game sales data has been acquired from the data website "Kaggle" which shows video game sales between the years 1976 and 2017 on different gaming platforms and by different publishers. For the sake of this experimental analysis and to lessen the records, data between the years 1991 and 2010 (20 years) will be extracted and used.

Upon loading in the data using RStudio software and observing its structure, the following variable types will be preprocessed and used for analysis:

1- Nominal: Publisher, Genre, and Platform; which are character data.

2- Ordinal: Year_of_Release, and Rating which are initially character data.

3- Ratio/Interval: NA_Sales, EU_Sales, JP_Sales, and Global_Sales which are numeric data.

The preprocessing for the following variables will fall into the **data entry issues** category as follows:

1- Year_of_Release variable: which is initially a character data type shall be transformed into an integer data type for the sake of statistical analysis (correlation requires numeric values). It also has missing data which shall be randomly imputed to avoid bias.

2- Rating Variable: This variable suffers from missing data which is more than %35. When the missing data is more than %10, it would create bias and disturb the statistical analysis of data.

therefore, the rows with missing values in the "Rating" variable shall be dropped entirely which leaves approximately 7000 records that will be used for machine learning purposes.

3-Preprocessing of data

3.1- Loading in the data: Video game sales data (vgsales.csv) comes in “comma-separated value” file format and was read in using the *read.csv()* function, and displayed using the head() function:

```
#Phase One: Data Preprocessing:
#Loading in the "vgsales.csv" data:
game_sales <- read.csv("vgsales.csv", header = T, stringsAsFactors = F)
head(game_sales,10)
```

3.2- Wrong data type preprocessing step (1): Year_of_Release: This variable came with character data type and values of "N/A" to represent real NA values, I changed the missing "chr" data type into real "NA" data:

```
#Replacing the "N/A" character values in Year_of_Release with real NA values:
game_sales %>% filter(game_sales$Year_of_Release == "N/A")
game_sales <- game_sales %>% mutate( Year_of_Release = gsub("N/A","", Year_of_Release))
```

3.3- Missing data imputation: Random imputation is used to avoid bias using with() function, and impute() function from the Hmisc library to impute missing data in the "Year_of_Release" variable which makes it a "chr" data type variable:

```
#Imputing Year_of_Release variable and inserting the imputed values:
imputeyear <- with(game_sales,Hmisc::impute(game_sales$Year_of_Release, 'mean'))
game_sales <- game_sales %>% mutate (Year_of_Release = imputeyear)
```

3.4- Wrong data type preprocessing step (2): changing the "chr" data type in the "Year_of_Release" variable into "int" data type:

```
#Changing the data type of column Year_of_release from "chr" to "int":
game_sales$Year_of_Release <- as.integer(game_sales$Year_of_Release)
```

4- R Programming content:

4.1 Function used for data manipulation:

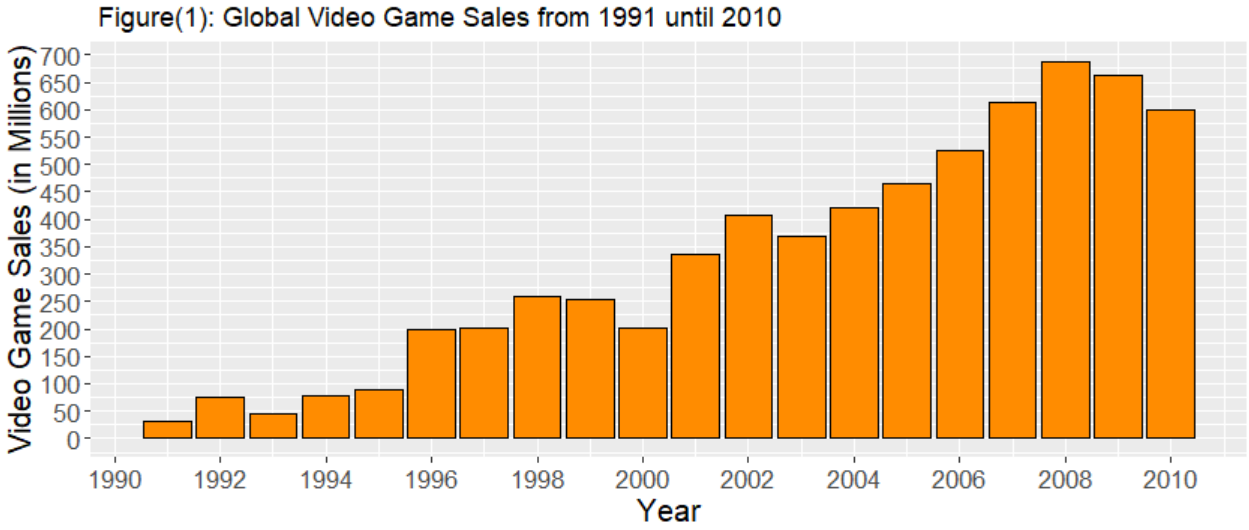
```
#Function: grouping sales in (na,eu,and jp)by genre and region using a function:
genresalefunc <- function (data, Sales, Genre) {
  x <- aggregate(Sales ~ Genre, data, sum)
  return(x)
}
nasalesgenre <- genresalefunc(game_sales, game_sales$NA_Sales, game_sales$Genre)
eusalesgenre <- genresalefunc(game_sales, game_sales$EU_Sales, game_sales$Genre)
jpsalesgenre <- genresalefunc(game_sales, game_sales$JP_Sales, game_sales$Genre)
```

The R function "genreslefunc" has the purpose of obtaining the video game sales based on the genre for the (North American, Europe, and Japan) regions. It takes as input the parameters (a dataset, a numeric Sales variable, and a character Genre variable), the sales are then aggregated using the built in aggregate() function and grouped for each of the 12 genres in the dataset. Then the sales by genre are obtained for each one of the regions.

5- Display of data/results:

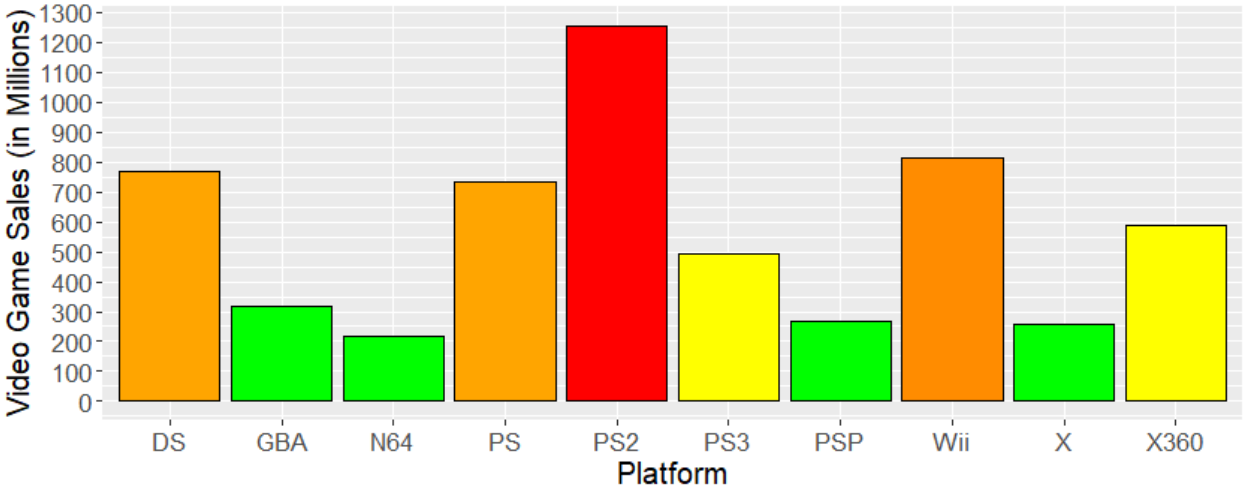
Data Visualization outputs:

1- **Figure (1):** The video game sales have been increasing in a linear pattern throughout the years between 1991 and 2010, reaching their peak in 2008 with 688 million game sales.



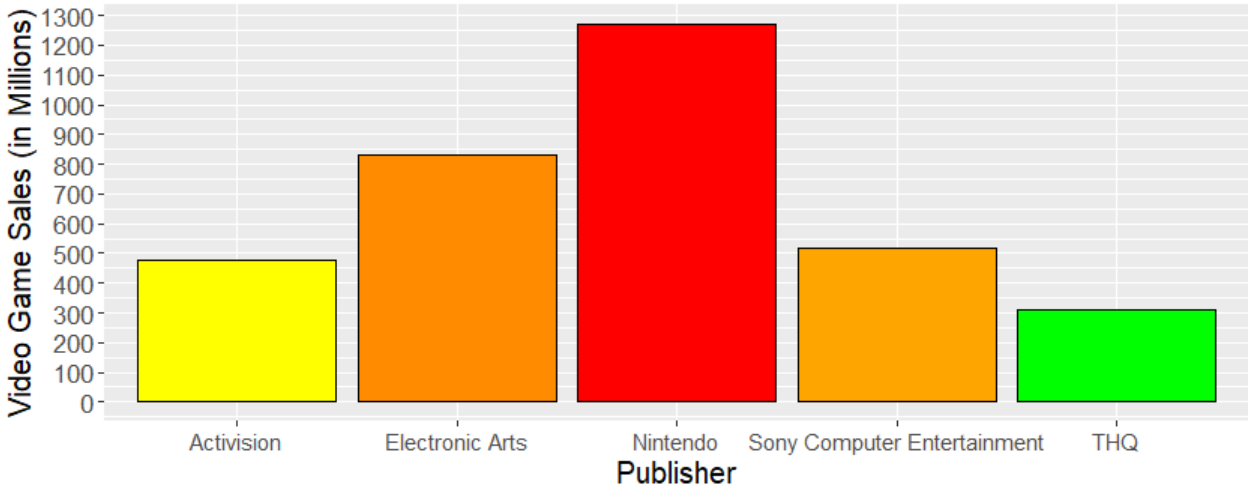
2- **Figure (2):** Sony PlayStation 2 is leading the board of the most successful video game platforms of all time with over 1.25 billion game sales. Nintendo's DS is the most successful handheld gaming device of all time with more than 765 million game sales, almost tripling the success of its competitor Sony's PSP (265 million game sales).

Figure(2): Most Successful Video Game Platforms from 1991 until 2010

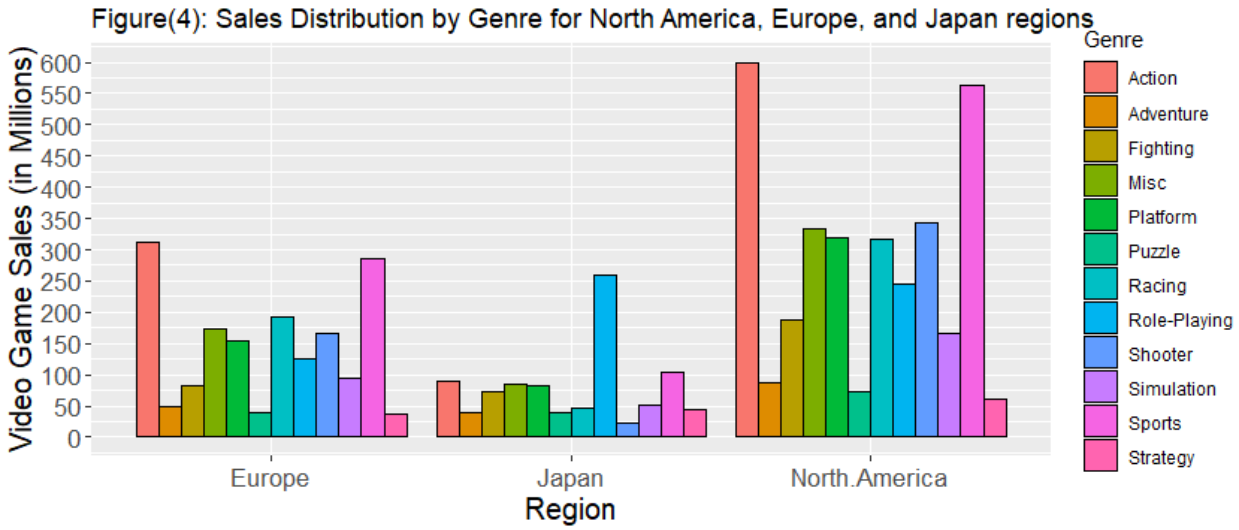


3- **Figure (3):** Nintendo is the most successful gaming publisher with over 1,25 billion game sales between the years (1991-2010), followed by Electronic Games (EA) with more than 830 million game sales, Sony Computer Entertainment comes 3rd with 519 million game sales.

Figure(3): Top 5 Video Game Publishers



4- **Figure (4):** Action, Sport and shooter games are the most popular genres in North America; The same pattern is seen in Europe where the Shooter genre is replaced by Racing. in Japan, Role-Playing games take the lead which sheds a light on the peaceful nature of the current Japanese generation of gamers.



8- Conclusions :

Video games are a multi-billion industry. Having an understanding of the distribution of video game sales in the discussed regions (North America, Europe, and Japan), and which game genres are more popular is crucial for video game companies and researchers.

Upon analysis of the video game sales data, the linear pattern of the increase in video game sales from 1991 until 2010 is noticeable, with a peak in 2008 (688 million game sales). Given a successfully built simple linear model, the global sales for the year 2022 could be predicted (1089 million game sales.)

Additional interesting patterns are Nintendo being the most successful game publisher between 1991 and 2010 (1225 million game sales), and the Sony Playstation 2 platform is the most successful platform with (1.25 billion game sales).

The Decision Tree algorithm has yielded an accuracy of (%62.2) when used to predict the "Rating" of video games based on (Publisher, Platform, and Genre); it is outperformed by the Support Vector Machines algorithm's (66.5%) accuracy which makes it the algorithm of choice.

7- Source code listing:

#Dataset source: https://www.kaggle.com/gregorut/videogamesales

#Dataset needs to be downloaded and have the session directory set to where it is saved.

#Libraries used:

library(tidyverse)

library(reshape2)

library(Hmisc)

```
library(rpart)
library(e1071)
library(caTools)
library(rpart.plot)
library(RColorBrewer)
library(rattle)
library(graphics)
library(VIM)
library(caret)
library(scales)

#Phase One: Data Preprocessing:
#Loading in the "vgsales.csv" data:
game_sales <- read.csv("vgsales.csv", header = T, stringsAsFactors = F)
head(game_sales,10)

#turning the structure of the data to tibble for ease of use:
game_sales <- as_tibble(game_sales)

#Replacing the "N/A" character values in Year_of_Release with real NA values:
game_sales %>% filter(game_sales$Year_of_Release == "N/A")
game_sales <- game_sales %>% mutate( Year_of_Release = gsub("N/A","", Year_of_Release))

#Imputing Year_of_Release variable and inserting the imputed values:
imputeyear <- with(game_sales,Hmisc::impute(game_sales$Year_of_Release, 'mean'))
game_sales <- game_sales %>% mutate (Year_of_Release = imputeyear)

#Changing the data type of column Year_of_release from "chr" to "int":
game_sales$Year_of_Release <- as.integer(game_sales$Year_of_Release)

#filtering data for "year_of_release" >= 2010 then ordering data ascending:
game_sales <- game_sales %>% filter(Year_of_Release >= 1991) %>% filter(Year_of_Release <=2010)

#Checking for NA values in the 5 sales variables:
a <- subset(game_sales, is.na(game_sales$NA_Sales)) #no NA
```

```
b <- subset(game_sales, is.na(game_sales$JP_Sales)) #no NA

c <- subset(game_sales, is.na(game_sales$EU_Sales)) #no NA

d <- subset(game_sales, is.na(game_sales$Other_Sales)) #no NA

e <- subset(game_sales, is.na(game_sales$Global_Sales)) #no NA


#Creating a subset of not NA values in the Rating variable

#Because the missing data is too many and not imputable (35%)

#This subset are for machine learning purposes only:

ml_subset_y <- game_sales %>% filter( Rating == "E"| Rating == "M" | Rating =="T" |

            Rating == "E10+"| Rating == "AO" | Rating =="K-A" | Rating =="RP")


#Selecting the variables required for Classification:

ml_subset_y <- ml_subset_y %>% select("Platform", "Genre", "Publisher", "Rating")


#Reducing variance in the "Publisher" variable by selecing the top 10 Publishers:

ml_subset_y <- ml_subset_y %>% filter (Publisher == "Activision" |Publisher == "Electronic Arts" |

            Publisher == "Konami Digital Entertainment" |Publisher == "Microsoft Game Studios" |

        Publisher == "Nintendo" |Publisher == "Sega" |Publisher == "Sony Computer Entertainment" |Publisher == "Ubisoft" |

        Publisher == "Take-Two Interactive" |Publisher == "THQ" )



#Phase Two: Data Manipulation:

#grouping global sales by year:

globsalesyear <- aggregate(game_sales$Global_Sales ~ game_sales$Year_of_Release, game_sales, sum)

globsalesyear <- rename(globsalesyear, Year = `game_sales$Year_of_Release`, Global.Sales = `game_sales$Global_Sales`)

globsalesyear <- globsalesyear %>% filter(Global.Sales > 0)

str(globsalesyear)


#Function: grouping sales in (na,eu,and jp)by genre and region using a function:

genresalefunc <- function (data, Sales, Genre) {

 x <- aggregate(Sales ~ Genre, data, sum)

 return(x)

}

nasalesgenre <- genresalefunc(game_sales, game_sales$NA_Sales, game_sales$Genre)

eusalesgenre <- genresalefunc(game_sales, game_sales$EU_Sales, game_sales$Genre)
```

```
jpsalesgenre <- genresalefunc(game_sales, game_sales$JP_Sales, game_sales$Genre)



#Merging the sales columns in na,eu, and jp regions based on genre:
merger <- merge(nasalesgenre, eusalesgenre, by = "Genre")
allsalesgenre <- merge(merger,jpsalesgenre, by = "Genre")
allsalesgenre <- rename(allsalesgenre,North.America = Sales.x ,Europe= Sales.y, Japan = Sales)

#Function:grouping top 10 global sales by Platform using a function:
globsalesplat <- aggregate(game_sales$Global_Sales ~ game_sales$Platform, game_sales, sum)
globsalesplat <- rename(globsalesplat, Platform = `game_sales$Platform`, Global.Sales = `game_sales$Global_Sales`)
globsalesplat <- globsalesplat %>% filter(Global.Sales > 210)

#Grouping top 5 global sales by publisher:
globsalespub <- aggregate(game_sales$Global_Sales ~ game_sales$Publisher, game_sales, sum)
globsalespub <- rename(globsalespub, Publisher = `game_sales$Publisher`, Global.Sales = `game_sales$Global_Sales`)
globsalespub <- globsalespub %>% filter(Global.Sales > 300)



#Phase Three: Data visualization:
#Plotting global sales by year:
yeargraph <- ggplot(globsalesyear, aes(Year,Global.Sales)) +
 geom_col(color = "black" , fill = "darkorange") +
 labs(title = " Figure(1): Global Video Game Sales from 1991 until 2010",
      x = "Year", y = "Video Game Sales (in Millions)")  +
 scale_y_continuous(breaks = scales::pretty_breaks(n= 15)) +
 scale_x_continuous(breaks = scales::pretty_breaks(n = 10)) +
  theme(axis.text = element_text(size = 13) , axis.title = element_text(size = 15))
yeargraph

#plotting global sales by top 10 platforms:
platgraph <- ggplot(globsalesplat, aes(Platform,Global.Sales)) +
 geom_col(color = "black" , fill = c("orange", "green", "green","orange", "red","yellow","green","darkorange","green",'yellow')) +
 labs(title = " Figure(2): Most Successful Video Game Platforms from 1991 until 2010",
```

```
    x = "Platform", y = "Video Game Sales (in Millions)")  +
 scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
 theme(axis.text = element_text(size = 13) , axis.title = element_text(size = 15))
platgraph

#plotting global sales by top 5 publishers:
#plotting global/publisher:
pubgraph <- ggplot(globsalespub, aes(Publisher, Global.Sales)) +
 geom_col(color = "black", fill = c("yellow", "darkorange", "red","orange", "green"))+
 labs(title = "Figure(3): Top 5 Video Game Publishers",x = "Publisher",
    y = "Video Game Sales (in Millions)") +
 scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
 theme(axis.text.y = element_text(size = 13),axis.text.x = element_text(size = 11) , axis.title = element_text(size = 15))
pubgraph


#plotting sales by genre for Na,Eu,and Jp regions:
genre_region_graph <- allsalesgenre %>%  pivot_longer(-Genre) %>%
 ggplot(aes(name, value, fill = Genre)) +
 geom_col(position = "dodge", colour = "black") + scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
 labs(title = "Figure(4): Sales Distribution by Genre for North America, Europe, and Japan regions",x = "Region",
    y = "Video Game Sales (in Millions)") +
 theme(axis.text = element_text(size = 13) , axis.title = element_text(size = 15))
genre_region_graph
```