Rafael Hernandez
IST 687 Final Project
3/39/2021
Pages: 46

INTRODUCTION

Sanskrit is a startup video game company. *Sanskrit's* business owners are debating over what type of video games the company should make. They agree that they will develop games that will be played on all video game console platforms. One of the business owners wants to focus on sports while the other business owner wants to focus on story and drama driven video games. *Sanskrit* wants an evaluation of video game markets and video game sales since the 1980's to help it determine where its focus should be.

BUSINESS QUESTIONS

Which video games sold the most per unit and what genre where they in?
Which video games sold the least per unit and what genre where they in?
Should we focus on a specific genre in a specific market?
What do consumers in North America, Japan, and Europe prefer?

DATA ACQUISITION, CLEANSING, TRANFORMATION AND MUNGING

Describe your data acquisition process

Data was acquired from Kaggle.com. The dataset covers video game sales from 1980 to 2010's. It encompasses multiple game systems and has over 16000 observations with 11 variables. It's an expansive data set that has the ability to provide answers to all of the business questions.

What data did you select, all, subset, why?

Video game sales is broken down by total sales and various world markets. All of the variables were retained with the exception of the first column which ranks video games. All of the other vectors are useful for descriptive statistics and for actionable insights.

What was your initial quality assessment?

Initial quality assessment is that data is broken up into various markets and it allows us to look at possible trends associated to markets and also variable dependencies. The sales are greater than 100,000 which means that sales less than 100,000 or 0.1 it comes up at 0 on the data set. That could be an issue in terms of the quality of the data because we are not getting accurate figures on games that sold less than 100,000. Representing it as zero when the rage is 0 to 99,999 seems as if the quality of the data is skewed.

What Field/Variables did you finally decide on, why?

The following fields will be utilized, Name, Year, Genre, North America Sales, Europe Sales, Japan Sales, Other Sales and Global Sales. Additional analysis might include publisher and platform but at the moment the focus of the guiding question by *Sanskrit* does not lend itself to diving into those fields.

Data Dictionary

- Name – The games name
- Platform – Platform of the game's released
- Year – Year of the game's release
- Genre – Genre of the game
- Publisher – Publisher of the game
- NA_Sales – Sales in North America (in millions)
- EU_Sales – Sales in Europe (in millions)
- JP_Sales – Sales in Japan (in millions)
- Other_Sales – Sales in the rest of the world (in millions)
- Global_Sales – Total worldwide sales.

Cleansing of Data

- Got rid of Rank Column.
- Changed Year column from char to factor.
- Removed records with NA's in the year column and records from 2017-2020. Study focuses on records from 1980-2016.
- Lower case column headings in VideoGame data base.

Structure of Data

```
'data.frame':   16323 obs. of  10 variables:
 $ name        : chr  "Wii Sports" "Super Mario Bros." "Mario Kart Wii" "Wii Sports Resort" ...
 $ platform    : chr  "Wii" "NES" "Wii" "Wii" ...
 $ year        : Factor w/ 37 levels "1980","1981",..: 27 6 29 30 17 10 27 27 30 5 ...
 $ 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 ...
> 
```

Summary of Data

```
     name           platform          year         genre          publisher        na_sales        eu_sales
 Length:16323       Length:16323      2009   :1431   Length:16323       Length:16323       Min.   : 0.0000   Min.   : 0.0000
 Class :character   Class :character  2008   :1428   Class :character   Class :character   1st Qu.: 0.0000   1st Qu.: 0.0000
 Mode  :character   Mode  :character  2010   :1259   Mode  :character   Mode  :character   Median : 0.0800   Median : 0.0200
                                      2007   :1202                                         Mean   : 0.2655   Mean   : 0.1476
                                      2011   :1139                                         3rd Qu.: 0.2400   3rd Qu.: 0.1100
                                      2006   :1008                                         Max.   :41.4900   Max.   :29.0200
                                      (Other):8856
    jp_sales         other_sales        global_sales
 Min.   : 0.00000   Min.   : 0.00000   Min.   : 0.0100
 1st Qu.: 0.00000   1st Qu.: 0.00000   1st Qu.: 0.0600
 Median : 0.00000   Median : 0.01000   Median : 0.1700
 Mean   : 0.07868   Mean   : 0.04834   Mean   : 0.5403
 3rd Qu.: 0.04000   3rd Qu.: 0.04000   3rd Qu.: 0.4800
 Max.   :10.22000   Max.   :10.57000   Max.   :82.7400
```

Observations

Sales tables are numeric. Sales below 100,000 are categorized as zero and as we look at the North American Sales, Europe Sales, Japan sales and Other Sales the minimum in all the categories is zero but when we look at Global Sales the minimum is .0100. Out of all the markets the game that made the most money per market did so in North America. The highest grossing game in the entire world sold 83 million copies (rounded from 82.74)

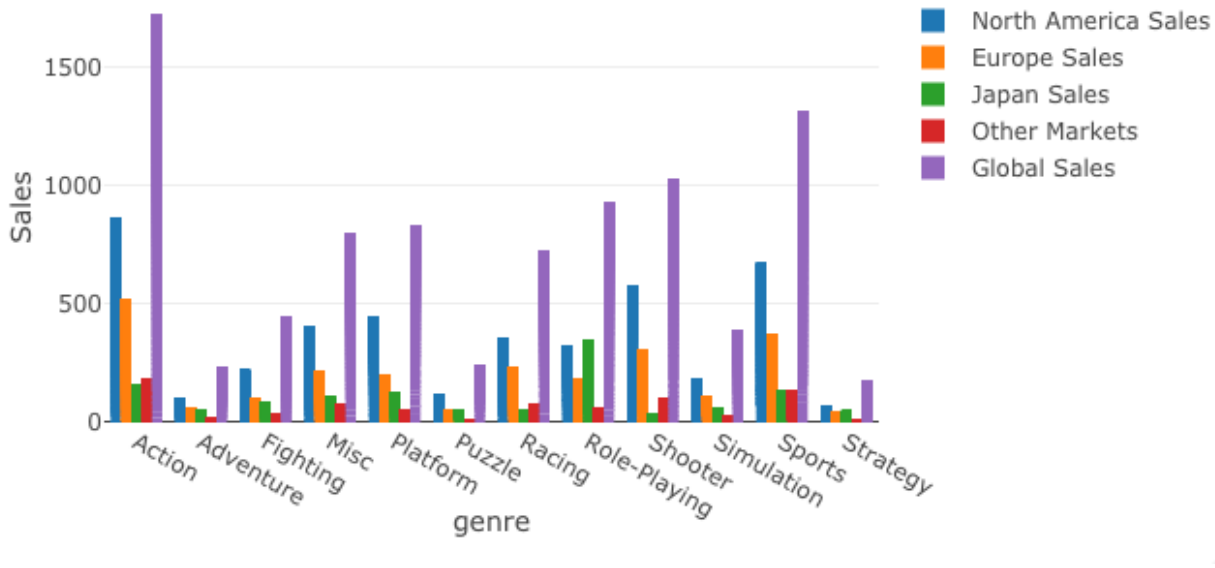
Cleansing Data Code in appendix

```
VideoGame$Year<-as.Date(as.character(VideoGame$Year), format = "%Y")
VideoGame$Year<-year(VideoGame$Year)

#Removing the Rank Column
VideoGame$Rank <- NULL
summary(VideoGame)

#Filtering only the records of interest for this study, removing the records with Year=nan and
records with the  year above 2016
#Code to remove records provided by Murilao via Kaggle.com

VideoGame<- VideoGame[VideoGame$Year != "N/A" & VideoGame$Year != "2017" &
VideoGame$Year != "2020", ]

VideoGame$Year <- factor(VideoGame$Year)
head(VideoGame, 6)

#Renaming columns and lower case headings.
```

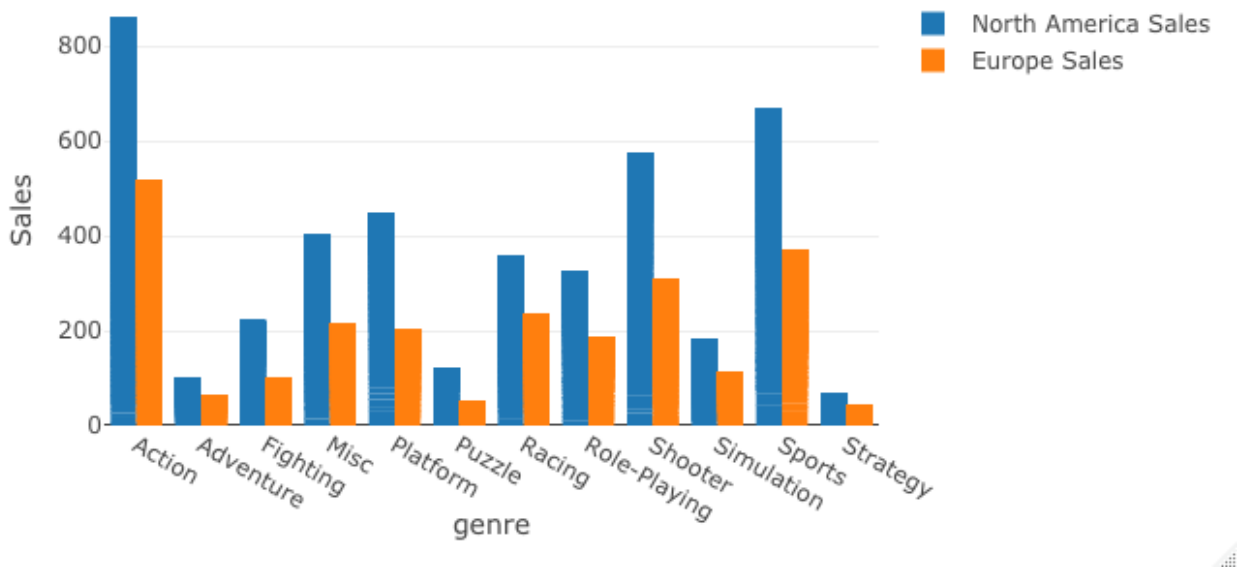
```
setnames(VideoGame, old = c("Name", "Platform", "Year", "Genre", "Publisher", "NA_Sales", "EU_Sales", "JP_Sales", "Other_Sales", "Global_Sales"), new = c("name", "platform", "year", "genre", "publisher", "na_sales", "eu_sales", "jp_sales", "other_sales", "global_sales"))

#Structure
str(VideoGame

#Summary
summary(VideoGame)
```

Descriptive Statistics

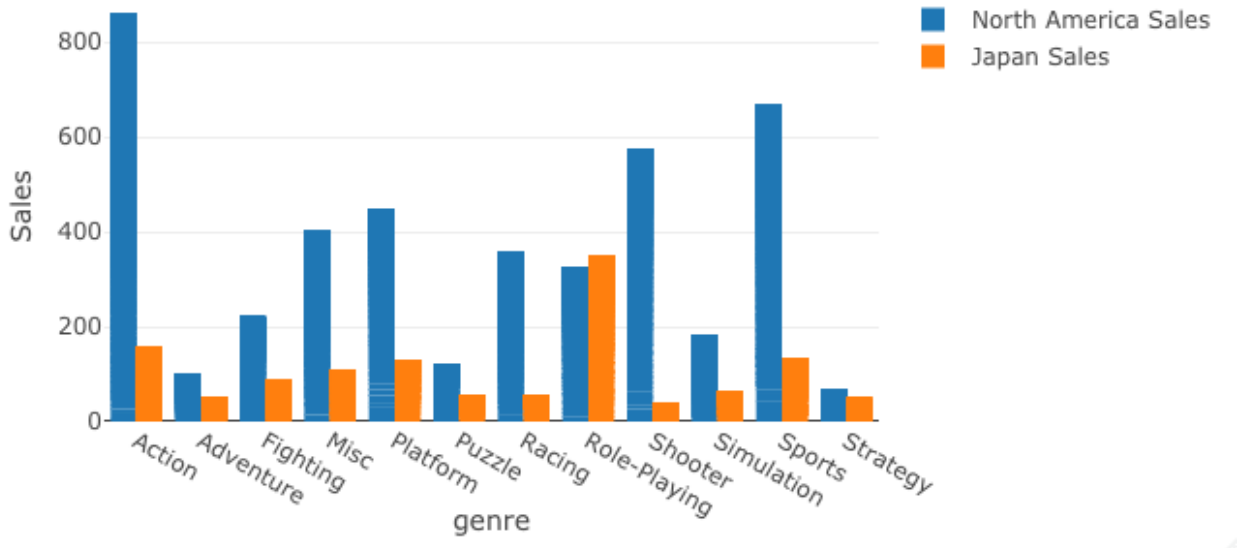
Bar Graph of Genre Sales and Various Markets



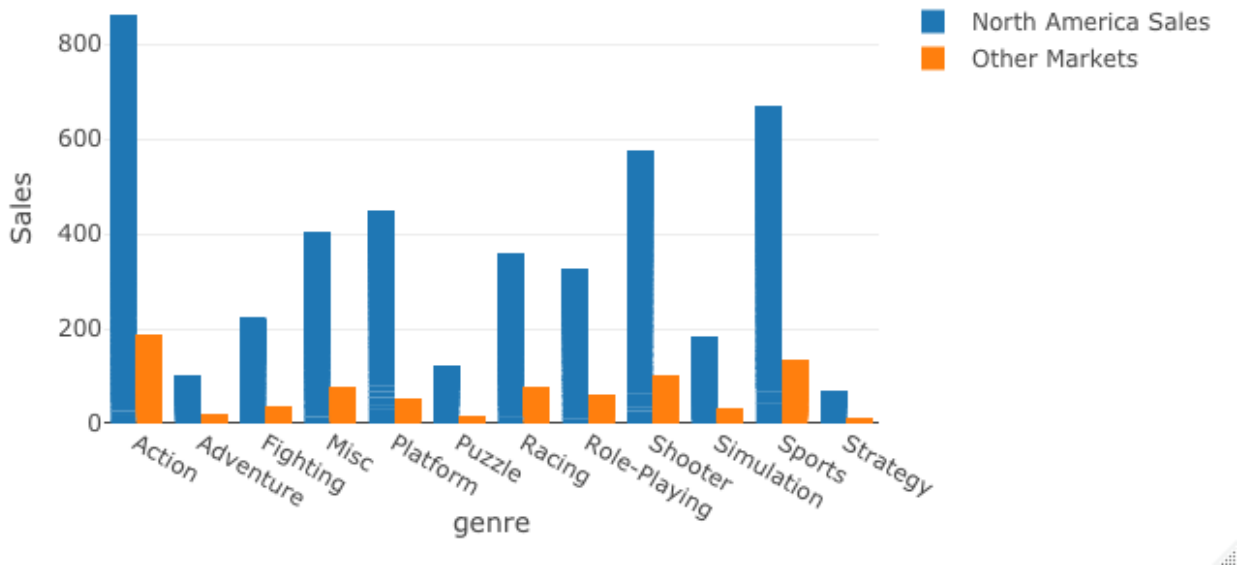
Bar Graph Comparing Genre Sales in North America and Europe



Bar Graph Comparing Genre Sales in North America and Japan

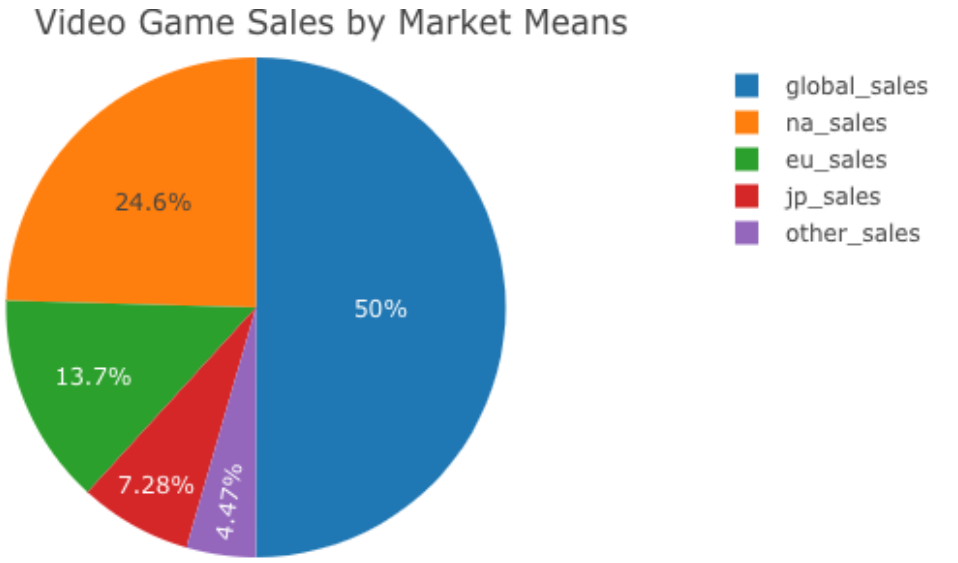


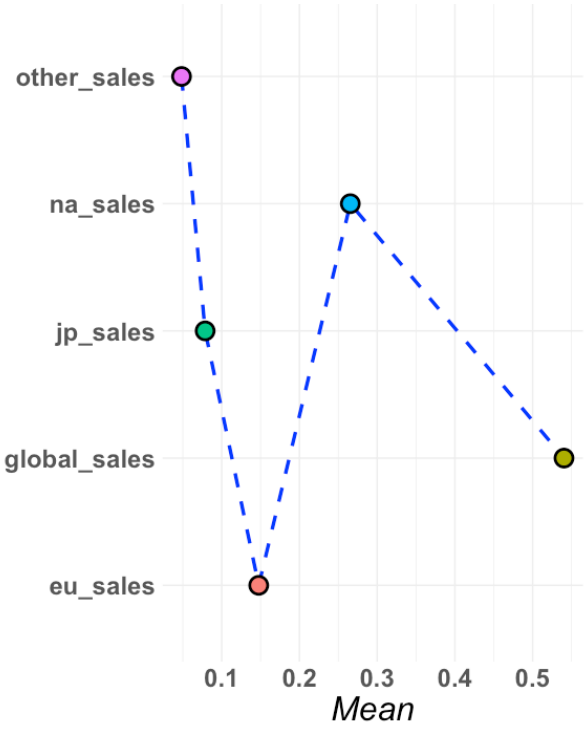
Bar Graph Comparing Genre Sales in North America and Other Markets

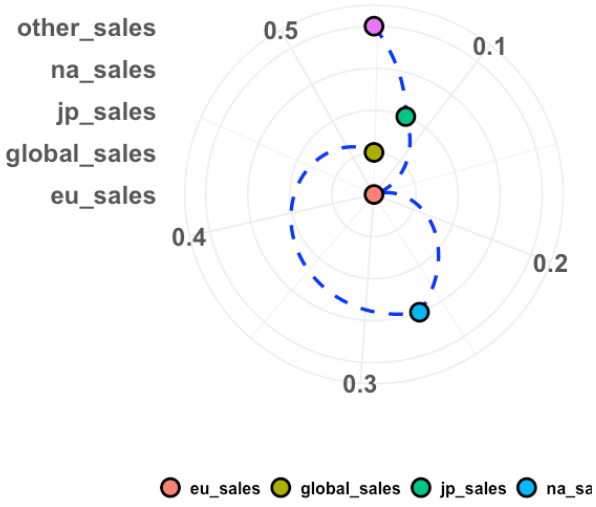


Sales Mean Table and Visualizations

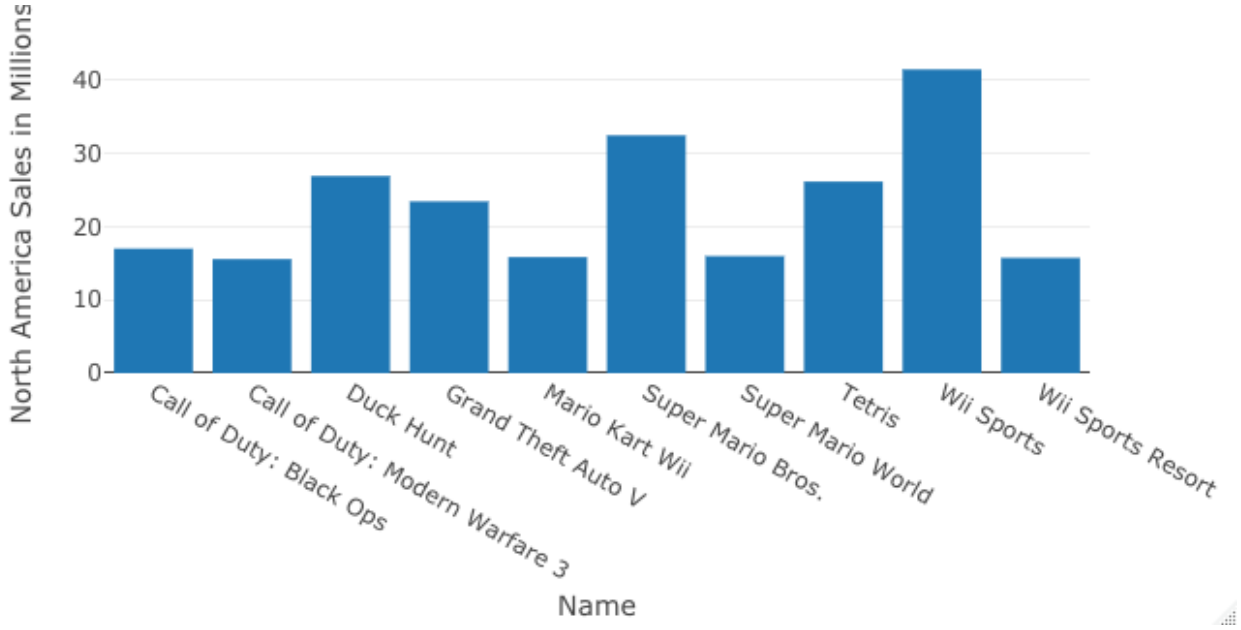
	Mean
na_sales	0.26546346
eu_sales	0.14759052
jp_sales	0.07867733
other_sales	0.04833609
global_sales	0.54034307


Video Game Sales by Market Means
global_sales
na_sales
eu_sales
jp_sales
other_sales
50%
24.6%
13.7%
7.28%
4.47%

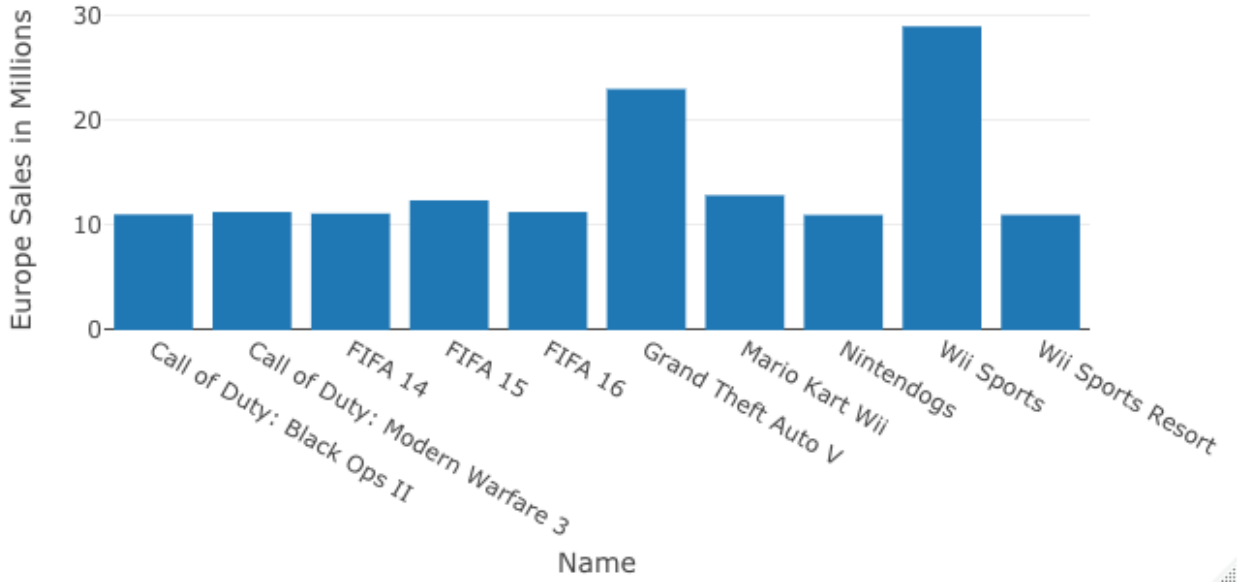

other_sales
na_sales
jp_sales
global_sales
eu_sales
0.1
0.2
0.3
0.4
0.5
Mean


other_sales
na_sales
jp_sales
global_sales
eu_sales
0.5
0.1
0.4
0.2
0.3
eu_sales
global_sales
jp_sales
na_sa

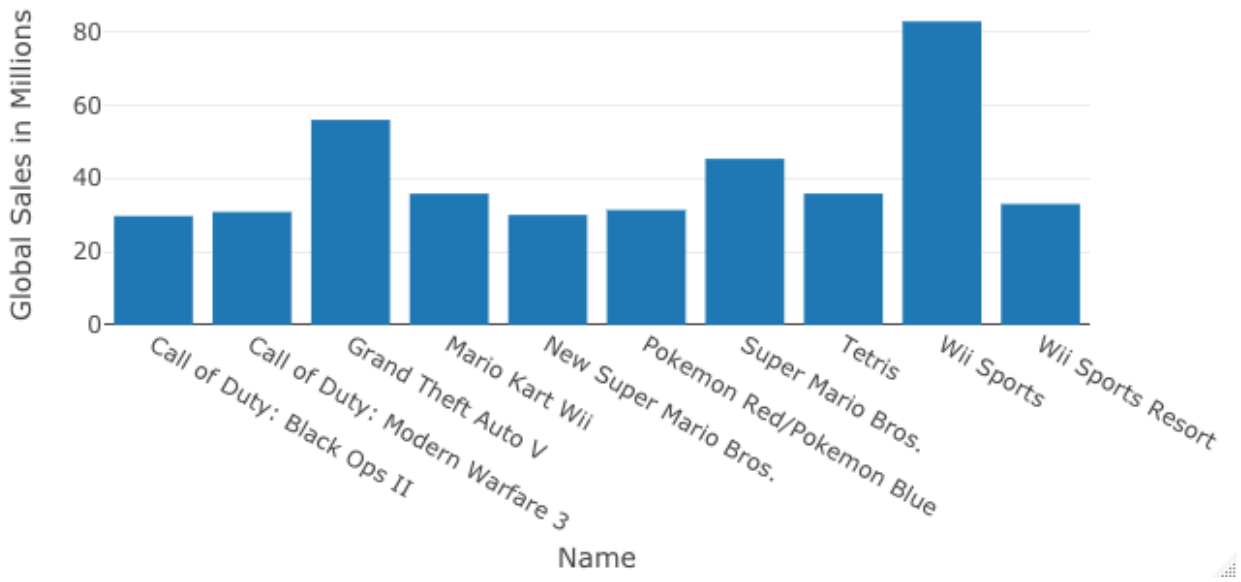
Top 10 Selling Games in North America



Top 10 Selling Games in Europe



Top 10 Selling Games in the World



Mean and Median

	Mean	Median
na_sales	0.26546346	0.08
eu_sales	0.14759052	0.02
jp_sales	0.07867733	0.00
other_sales	0.04833609	0.01
global_sales	0.54034307	0.17

Mean, Variance and Standard Deviation of Market Sales

	DM	Variance	std
na_sales	0.26546346	0.67516400	0.8216836
eu_sales	0.14759052	0.25890063	0.5088228
jp_sales	0.07867733	0.09709044	0.3115934
other_sales	0.04833609	0.03606483	0.1899074
global_sales	0.54034307	2.45206289	1.5659064

OBSERVATIONS

There appears to be a similar pattern when we compare video game sales in North America and Europe. The sales are higher in North America, but the peaks and valleys are the same. The same cannot be said for North America and both Japan and Other markets. Role Playing video games are big in Japan. The role-playing genre is the best-selling genre in Japan whereas in North America the best-selling genre are Action and Sports. The main takeaway from North American sales and Other Market sales is the quantity of sales is highest in North America and lowest in other markets.

The North American market is the most lucrative market in the video game industry. If we examine the mean of sales between North America, Europe, Japan and other markets the sales means are highest in North America and Europe. The standard deviation is highest in global sales which makes sense since there is a significance variance. If we examine individual markets standard deviation is highest in North America followed by Europe, Japan and the smallest standard deviation takes place in other markets. Overall sales are higher in North America and Europe and there seems to be a similarity in taste when we look at the chart comparing genre sales, however when we examine the top 10 video game sales there is a similarity between North America and Japan. There is a difference between North America and Europe as it relates to FIFA Futbol video games. FIFA Futbol video games sell well in Europe but they do not crack the top ten in North America.

Descriptive Statistics Appendix and Code

```
install.packages("plotly")
library(plotly)

fig<-plot_ly(
  x=c(VideoGame$genre),
  y=c(VideoGame$na_sales),
  name = "Genre Sales",
```

```
  type = "bar"
)
fig

#Code provided by plotly.com
fig1<-plot_ly(VideoGame, x =~genre, y=~na_sales, type = 'bar', name = 'North America Sales')
fig1 <-fig1 %>% add_trace(y = ~eu_sales, name = 'Europe Sales')
fig1 <-fig1 %>% add_trace(y = ~jp_sales, name = 'Japan Sales')
fig1 <-fig1 %>% add_trace(y = ~other_sales, name = 'Other Markets')
fig1 <-fig1 %>% add_trace(y = ~global_sales, name = 'Global Sales')
fig1 <- fig1 %>% layout(yaxis =list(title = 'Sales'), barmode = 'group')
fig1

fig2<-plot_ly(VideoGame, x =~genre, y=~na_sales, type = 'bar', name = 'North America Sales')
fig2 <-fig2 %>% add_trace(y = ~eu_sales, name = 'Europe Sales')
fig2 <- fig2 %>% layout(yaxis =list(title = 'Sales'), barmode = 'group')
fig2

fig3<-plot_ly(VideoGame, x =~genre, y=~na_sales, type = 'bar', name = 'North America Sales')
fig3 <-fig3 %>% add_trace(y = ~jp_sales, name = 'Japan Sales')
fig3 <- fig3 %>% layout(yaxis =list(title = 'Sales'), barmode = 'group')
fig3


fig4<-plot_ly(VideoGame, x =~genre, y=~na_sales, type = 'bar', name = 'North America Sales')
fig4 <-fig4 %>% add_trace(y = ~other_sales, name = 'Other Markets')
fig4 <- fig4 %>% layout(yaxis =list(title = 'Sales'), barmode = 'group')
fig4

#Mean Median and Mode
#Median code provided by Murilao via Kaggle.com
median_Df<-data.frame(Median = c(median(VideoGame$na_sales),
median(VideoGame$eu_sales),
                         median(VideoGame$jp_sales), median(VideoGame$other_sales),
                         median(VideoGame$global_sales)))

row.names(median_Df)<- c("na_sales", "eu_sales", "jp_sales", "other_sales", "global_sales")
median_Df

#Mode code provided by Mrilao via Kaggle.com
mode_df<-data.frame(Mode = c(mode(VideoGame$na_sales), mode(VideoGame$eu_sales),
                        mode(VideoGame$jp_sales), mode(VideoGame$other_sales),
                        mode(VideoGame$global_sales)))
```

```
row.names(mode_df)<- c("na_sales", "eu_sales", "jp_sales", "other_sales", "global_sales")
mode_df

#Central Tendencies
central_tend<-data.frame(means_df, median_Df, mode_df)
central_tend
row.names(central_tend)<-c("Mean", "Median", "Mode")

central_tend<-central_tend[,-3]
central_tend

options(repr.plot.width = 14, repr.plot.height = 6)
a <- ggplot(data = means_df, mapping = aes(x = Mean, y = row.names(means_df))) +
  geom_line(group = 1, size = 1.2, linetype = "dashed", color = "blue") +
  geom_point(size = 5, shape = 21, stroke = 1.5, mapping = aes(fill = row.names(means_df))) +
  theme_minimal() +
  ylab("") +
  theme(plot.title = element_text(size = 24, hjust = .5, face = "bold"),
        axis.title.x = element_text(size = 24, hjust = .5, face = "italic"),
        axis.title.y = element_text(size = 24, hjust = .5, face = "italic"),
        axis.text.x = element_text(size = 18, face = "bold"),
        axis.text.y = element_text(size = 18, face = "bold"),
        legend.position = "none")

b <- ggplot(data = means_df, mapping = aes(x = Mean, y = row.names(means_df))) +
  geom_line(group = 1, size = 1.2, linetype = "dashed", color = "blue") +
  geom_point(size = 5, stroke = 1.5, shape = 21, mapping = aes(fill = row.names(means_df))) +
  theme_minimal() +
  ylab("") +
  xlab("") +
  theme(plot.title = element_text(size = 24, hjust = .5, face = "bold"),
        axis.title.x = element_text(size = 24, hjust = .5, face = "italic"),
        axis.title.y = element_text(size = 24, hjust = .5, face = "italic"),
        axis.text.x = element_text(size = 18, face = "bold"),
        axis.text.y = element_text(size = 18, face = "bold"),
        legend.position = "bottom",
        legend.title = element_text(color = "white"),
        legend.text = element_text(size = 12, face = "bold"))

plot_grid(a, b + coord_polar(), nrow = 1, ncol = 2)
```

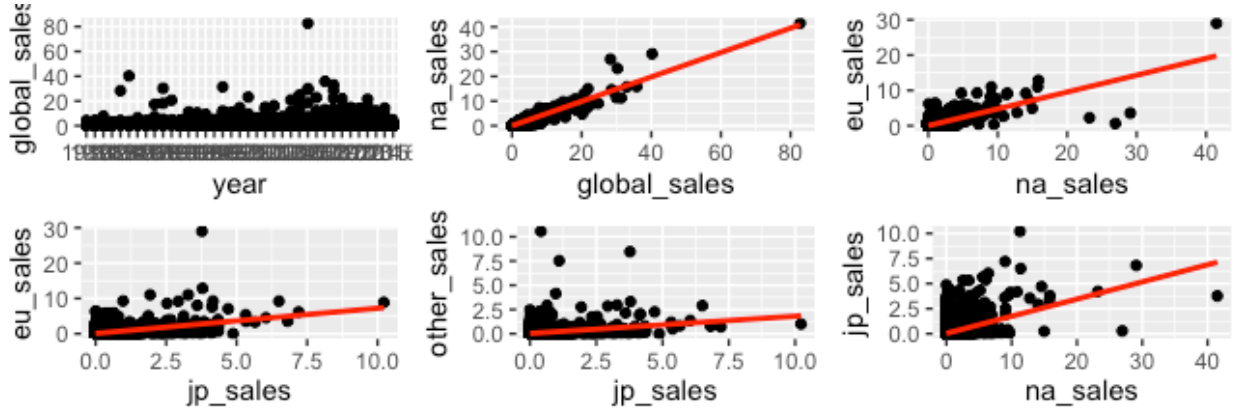
```
#Code provided by plotly
Sales_Means<- data.frame("Categorie"=rownames(means_df), means_df)
data<-Sales_Means[c('Categorie', 'Mean')]
fig5 <- plot_ly(data, labels = ~Categorie, values = ~Mean, type = 'pie')
fig5 <- fig5%>% layout(title = 'Video Game Sales by Market Means',
                       xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
                       yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))

fig5


# NA_Sales #code provided by murilao via Kaggle.com
t_v_name_NA <- aggregate(list(na_sales = VideoGame$na_sales), list(Name =
VideoGame$name), sum)
t_v_name_NA <- t_v_name_NA[order(t_v_name_NA$na_sales, decreasing = T), ]
t_v_name_NA

t_v_name_EU<- aggregate(list(eu_sales= VideoGame$eu_sales), list(Name =
VideoGame$name), sum)
t_v_name_EU<-t_v_name_EU[order(t_v_name_EU$eu_sales, decreasing = T),]
t_v_name_EU


#Code provided by plotly
fig6<- plot_ly(data= head(t_v_name_NA, 10), x=~Name, y =~na_sales, type = 'bar', name =
'North America Top')
fig6 <- fig6 %>% layout(yaxis =list(title = 'North America Sales in Millions'))
fig6

fig9<- plot_ly(data=head(t_v_name_EU, 10), x=~Name, y =~eu_sales, type = 'bar', name =
'Europe Top')
fig9<- fig9 %>% layout(yaxis = list(title = 'Europe Sales in Millions'))
fig9

# Global_Sales code provided by murilao via kaggle.com
t_v_name_Global <- aggregate(list(global_sales = VideoGame$global_sales), list(Name =
VideoGame$name), sum)
t_v_name_Global <- t_v_name_Global[order(t_v_name_Global$global_sales, decreasing = T), ]
t_v_name_Global

fig7<- plot_ly(data= head(t_v_name_Global, 10), x=~Name, y =~global_sales, type = 'bar', name
= 'Global Top 10 Sales')
fig7 <- fig7 %>% layout(yaxis =list(title = 'Global Sales in Millions'))
```

fig7

```
############################################################
#Measures of dispersion
Sales_Variance <- data.frame(Variance = c(var(VideoGame$na_sales),
var(VideoGame$eu_sales), var(VideoGame$jp_sales), var(VideoGame$other_sales),
var(VideoGame$global_sales)))
row.names(Sales_Variance) <- c("na_sales", "eu_sales", "jp_sales", "other_sales",
"global_sales")
Sales_Variance

Sales_std <- data.frame(std = c(sqrt(var(VideoGame$na_sales)),
sqrt(var(VideoGame$eu_sales)), sqrt(var(VideoGame$jp_sales)),
sqrt(var(VideoGame$other_sales)), sqrt(var(VideoGame$global_sales))))
row.names(Sales_std) <- c("na_sales", "eu_sales", "jp_sales", "other_sales", "global_sales")
Sales_std4

Sales_Dispersion <- data.frame(DM = Sales_Means$Mean, Variance = Sales_Variance$Variance,
std = Sales_std$std)
row.names(Sales_Dispersion) <- c("na_sales", "eu_sales", "jp_sales", "other_sales",
"global_sales")
Sales_Dispersion
```

Modeling Techniques

Linear Models



Linear Models Observations

In the linear model we took sales data ranging from 1980 to 2016 to try and find correlations or relationships between markets. Linear models are used with regression data in an attempt to find dependencies and to predict future sales. The first visualization compares years and global sales over that time. The graph has no slope thus there is no relationship between years and global sales. We can also rule out relationships between Europe and Japan, Japan and Other markets and North America and Japan. The slopes between Japan, Europe and Other markets almost horizontal which means that there is no relationship between the different markets. Each market is acting independently of one another and their tastes in video games are distinct and different.

As we examine Global Sales and North America Sales the slope indicates that there is a relationship, however Global Sales includes North America Sales. North America Sales are added to global sales thus the direct correlation between the two.

The visualization that appears to be promising is comparing North America Sales with Europe. To make a decision on this relationship we have to look at the summary of the linear model for these two markets.

North American and Europe Linear Model Summary

```
Call:
lm(formula = eu_sales ~ na_sales, data = VideoGame)

Residuals:
     Min       1Q   Median       3Q      Max
-12.2141  -0.0474  -0.0212   0.0060   9.2430

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.021188   0.002676   7.918 2.56e-15 ***
na_sales    0.476158   0.003099 153.652  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3253 on 16321 degrees of freedom
Multiple R-squared:  0.5913,    Adjusted R-squared:  0.5912
F-statistic: 2.361e+04 on 1 and 16321 DF,  p-value: < 2.2e-16
```

Examining the residuals, it is difficult to say if there is a standard distribution. There is a cluster between 1Q, Median and 3Q and the Min and Max appear to be outliers or at the polar end of the spectrum. It could be bell shaped. Pr(>|t|) of na_sales have an asterisks and it's not below .05. R-Squared is at .5913 and it's difficult to evaluate its significance. Do we want it to be higher or is .6 significant enough for us to say that there is a relationship between North America sales and Europe Sales? Overall, I think that the evidence does lean more towards there being a relationship between the North American and European markets. We can conclude that both markets will react similarly in purchasing specific video games. We cannot predict what type of video games both markets will purchase by looking at sales figures.

FREQUENCY CHARTS AND TABLES

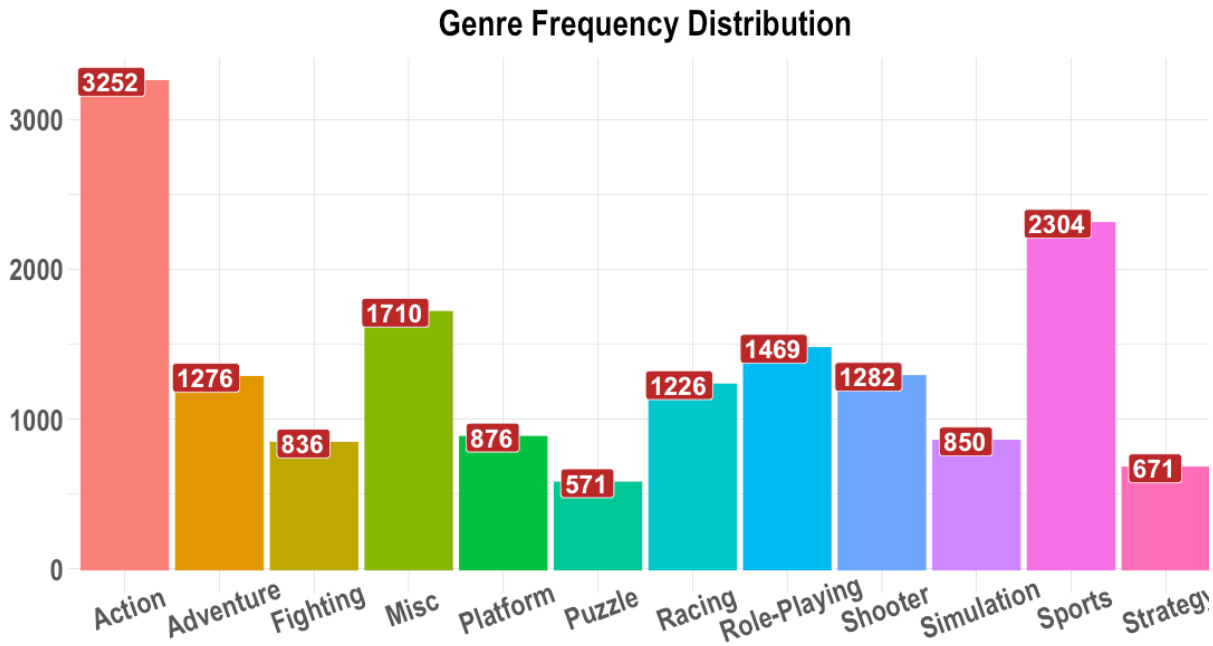
Top 15 frequency Years

	frequency	percent
2009	1431	8.76677081
2008	1428	8.74839184
2010	1259	7.71304295
2007	1202	7.36384243
2011	1139	6.97788397
2006	1008	6.17533542
2005	941	5.76487165
2002	829	5.07872327
2003	775	4.74790173
2004	763	4.67438584
2012	657	4.02499541
2015	614	3.76156344
2014	582	3.56552104
2013	546	3.34497335
2001	482	2.95288856

Frequency of Genre

	frequency	percent
Action	3252	19.922808
Sports	2304	14.115052
Misc	1710	10.476015
Role-Playing	1469	8.999571
Shooter	1282	7.853948
Adventure	1276	7.817190
Racing	1226	7.510874
Platform	876	5.366661
Simulation	850	5.207376
Fighting	836	5.121608
Strategy	671	4.110764
Puzzle	571	3.498131

Genre Frequency Distribution Chart



FREQUENCY OBSERVATIONS

Frequency and Percent were calculated using the following formulas.

P -> Percent

Freq_x -> Frequency of an element x

$\sum_{ni=0}(Freq_i)$∑i=0n(Freqi) -> Sum of all frequencies

$$p=100*Freq_x\sum_{ni=0}(Freq_i)$$p=100∗Freqx∑i=0n(Freqi)

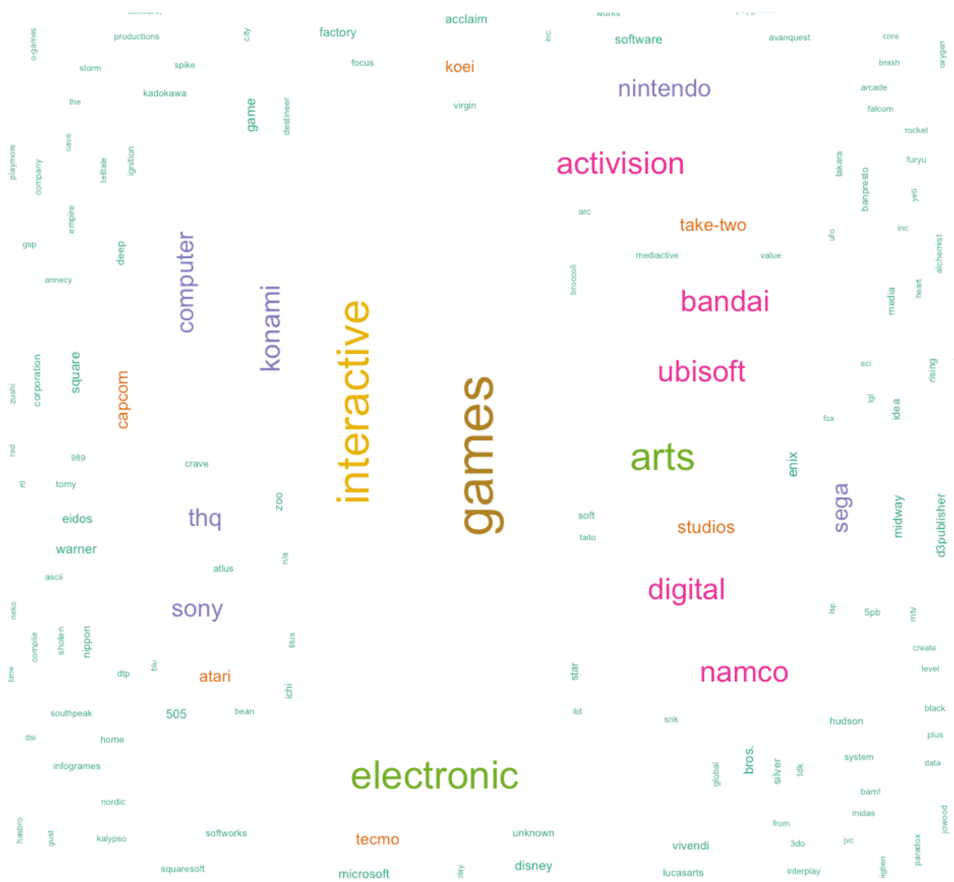
The frequency of video games being made and sold seems to be on a decline. The highest frequency years were 2006 to 2011. A great recession took place in the United States in 2008 but 2008, 2009, 2010 and 2011 are some of the years where video games were published and sold. It could be that people used video games as a coping mechanism to difficult economic times. People might have decided to stay home and enjoy video games as entertainment instead of going out.

The genre frequency distribution chart is informative because it lets us know what is happening in the video game market. It gives us an idea of peoples tastes and what competing video game companies are doing. The most frequent genres are Action, sports, misc, role playing, and shooter genres.

Genre Word Cloud

Video Game Console Word Cloud

PUBLISHER WORD CLOUD



WORDCLOUD OBSERVATIONS

The most common words in the genre category are Action, Sports, Misc and roleplaying. The word cloud matches the genre frequency distribution chart. The most popular console up to 2016 was the PS2. *Sanskrit* has a lot of competition but there are certain video game publishing companies that have stood the test of time: interactive games, electronic arts, activision, Nintendo and Konami are noticeable competitors.

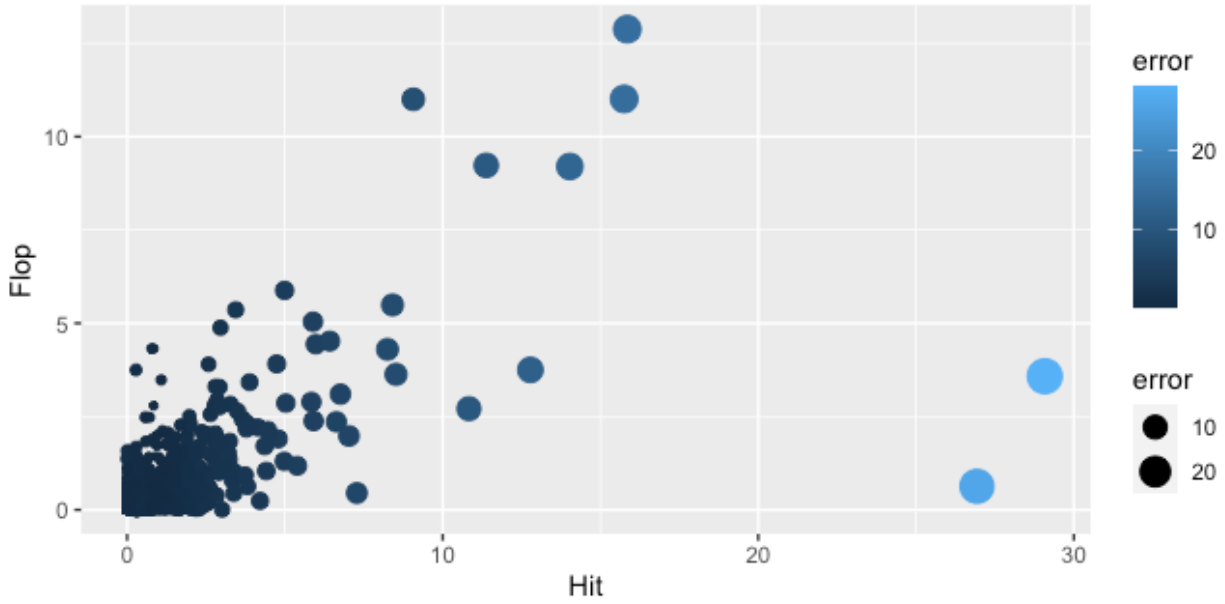
Support Vector Machine Statistics

```
Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.1  cost C = 10

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  162.960037398446

Number of Support Vectors : 1701

Objective Function Value : -1608.759
Training error : 0.016998
Cross validation error : 0.004823
Laplace distr. width : 0.243858
```

INTERPRETATION

The training error is 1.6% and the cross-validation error is lower than that. There are 1701 support vectors which is a large number. I believe this sets up a pretty good predictive model.

SVM Visualization to predict if Video Game will be a hit or a flop in North America



INTERPRETATION

The visualizations places Hit video games on the x-axis and video games that Flop on the y-axis. The visualization uses the relationship between North America sales and Europe Sales. To establish if a game is a hit or a flop we used the mean of North America Sales. If the sales are above the mean sales of the North American market than we classify the game as a hit. If the sales are below the mean than we classify the game as a hot. We computed the Root Mean Squared Error (RSME) and used the RSME to color and determine the size of the plots. We see a high density of plots and the error is low. The mean of North American Sales is .26 or 26 million. Our model can predict with good certainty if a video game will be a hit or a flop in North America.

OVERALL INTERPREATIONS AND ACTIONABLE INSIGHTS

Answers to Business Questions

Which video games sold the most per unit and what genre where they in?

The top 5 selling video games were Wii Sports/Sports, Super Mario Bros/Platform, Mario Kart Wii/Racing, Wii Sports Resort/Sports and Pokemon Red/Pokemon Blue/Role-Playing. This determination was made based on global sales. This information was visible in the dataset because it ranked video games based on global sales.

Which video games sold the least per unit and what genre where they in?

This question cannot be answered because if a video game sold less than 100,00 units it was classified in the data base as a zero. There are a whole bunch of games with a zero which makes it difficult to say which games were the least selling games per unit and what genre they were in.

What do consumers in North America, Japan, and Europe prefer?

The sales by genre graphs show a correlation in taste between North America and Europe. There is a difference in sales, the sale numbers are higher in North America than in Europe and both markets have a higher mean in sales when compared to Japan. The sales by genre graph matches the frequency by genre visualization. The frequency by genre visualization is a visualization of all global sales and it appears that the North American and European market impact global sales. Consumers in North America and Europe prefer **Action, Sports and Shooter games.** Other genre's that are high in frequency and in sales are platform games and Misc.

Japan has a different taste than North American and Europe. **Role playing games** are the highest selling games in Japan followed by **Action, Sports and Fighting.** Overall as a market Japan does not sell as many video games as North America and Europe(according to this database).

Should we focus on a specific genre in a specific market?

There is plenty of evidence that points to *Sanskrit* focusing on North America and the European Markets. Video Games are highest in North America followed by Europe. This conclusion is supported by the **mean of sales and by the sales in market by genre.** There is also a correlation/relationship between the North American and European markets. We discussed this correlation in the previous question, but it is also supported by the **linear model.** North America and Europe are the only markets that have a relationship according to linear model analysis. We can also do a SVM prediction on weather a video game will be a hit, or a flop based on the relationship between North America and Europe. To establish if a video game will be a hit or a flop, we used the north American sales mean as the line of demarcation. Games that sale above the mean are considered a hit and games that sell below the mean are a flop. The SVM model was also built on Root Mean Squared Error and the dots are colored based on the error. **The recommendation is that *Sanskrit* sell in North America and Europe if it can.**

If *Sanskrit* wants to sell in both North America and Europe there is evidence that it should focus on Action, sports and shooter games. These three genres are being made in higher frequency in these two markets and are selling the most. We have discussed the evidence above and there are similar tastes. The owners of *Sanskrit* were divided on whether they should focus on Sports or Drama themed games. We will classify role playing games as games that fall into Drama and **if these are the two options the recommendation is to focus on sports games.** Sports sell high in both North America and Europe. *Sanskrit* should follow the model of EA sports and play sports games that focus on different sports.

Modelling Techniques and Visualization Appendix

```
#Create new data table for frequency and percent.
Freq_year <- data.frame(cbind(Frequency = table(VideoGame$year), Percent =
prop.table(table(VideoGame$year)) * 100))
freq_year<- freq_year[order(freq_year$Frequency, decreasing = TRUE), ]
freq_year


freq_name <- data.frame(cbind(Frequency = table(VideoGame$name), Percent =
prop.table(table(VideoGame$name))* 100))
freq_name <- head(freq_name[order(freq_name$Frequency, decreasing = T), ], 5)
freq_name


freq_genre <- data.frame(cbind(Frequency = table(VideoGame$genre), Percent =
prop.table(table(VideoGame$genre)) * 100))
freq_genre <-freq_genre[order(freq_genre$Frequency, decreasing = T), ]
freq_genre

setnames(freq_year, old = c("Frequency", "Percent"), new = c("frequency", "percent"))
freq_year

setnames(freq_genre, old = c("Frequency", "Percent"), new = c("frequency", "percent"))
freq_genre

setnames(freq_name, old = c("Frequency", "Percent"), new = c("frequency", "percent"))
freq_name

#Frequency Distribution of Genre Graph.  Code provided by Murilao on Kaggle.com

options(repr.plot.width = 14, repr.plot.height = 6)
ggplot(data = freq_genre, mapping = aes(x = frequency, y = row.names(freq_genre))) +
  geom_bar(stat = "identity", mapping = aes(fill = row.names(freq_genre), color =
row.names(freq_genre)), 27ngli = .7,
         size = 1.1) +
        geom_label(mapping = aes(label = frequency), fill = "#B22222", size = 6, color = "white",
fontface = "bold", hjust = .7)+
        ggtitle("Genre Frequency Distribution") + xlab(" ") +
ylab("")+theme_ipsum()+coord_flip()+
        theme(plot.title = element_text(size = 24, hjust = .5, face = "bold"), axis.title =
element_text(size = 24, hjust = .5, face = "italic"),
              axis.title.x = element_text(size = 24, hjust = .5, face = "italic"),
```

```
    axis.title.y = element_text(size = 24, hjust = .5, face = “italic”),
    axis.text.x = element_text(size = 20, face = “bold”, angle = 20),
    axis.text.y = element_text(size = 20, face = “bold”),
    legend.position = “none”)
```

```
#Building and plotting linear models
#Plotting to find dependencies
plot(VideoGame$global_sales, VideoGame$na_sales)
plot(VideoGame$global_sales, VideoGame$eu_sales)
plot(VideoGame$global_sales, VideoGame$jp_sales)
plot(VideoGame$global_sales, VideoGame$other_sales)



plot(VideoGame$year, VideoGame$na_sales)
plot(VideoGame$year, VideoGame$eu_sales)
plot(VideoGame$year, VideoGame$jp_sales)
plot(VideoGame$year, VideoGame$other_sales)

#building linear models
model1<- lm(formula=na_sales~global_sales, data=VideoGame)
summary(model1)
plot(VideoGame$global_sales, VideoGame$na_sales)
abline(model1)
NorthAmerica_World<-ggplot(VideoGame, aes(x=global_sales,
y=na_sales))+geom_point()+stat_smooth(method=”lm”, col = “red”)
NorthAmerica_World

#adding year to the variable.  There appears to be no correlation between years and video
game sales.  There is no slope.  There does appear to be a correlation between global sales and
sales in indidividual markets.  How about sales between different markets?
Model2<-lm(formula=global_sales~year, data=VideoGame)
model2
plot(VideoGame$year, VideoGame$global_sales)
abline(model2)
Year_Global<-ggplot(VideoGame, aes(x=year,
y=global_sales))+geom_point()+stat_smooth(method=”lm”, col = “red”)
Year_Global


#comparing markets
model3<-lm(formula=eu_sales~na_sales, data=VideoGame)
summary(model3)
```

```
plot(VideoGame$na_sales, VideoGame$eu_sales)
abline(model3)
NorthAmerica_Europe<-ggplot(VideoGame, aes(x=na_sales,
y=eu_sales))+geom_point()+stat_smooth(method="lm", col = "red")
NorthAmerica_Europe

model4<-lm(formula=eu_sales~jp_sales, data=VideoGame)
summary(model4)
plot(VideoGame$eu_sales, VideoGame$jp_sales)
abline(model4)
Europe_Japan<-ggplot(VideoGame, aes(x=jp_sales,
y=eu_sales))+geom_point()+stat_smooth(method="lm", col = "red")
Europe_Japan


model5<-lm(formula=other_sales~jp_sales, data=VideoGame)
summary(model5)
plot(VideoGame$jp_sales, VideoGame$other_sales)
abline(model5)
Japan_Other<-ggplot(VideoGame, aes(x=jp_sales,
y=other_sales))+geom_point()+stat_smooth(method="lm", col = "red")
Japan_Other

model6<-lm(formula=jp_sales~na_sales, data=VideoGame)
summary(model6)
plot(VideoGame$na_sales, VideoGame$jp_sales)
abline(model6)
NorthAmerica_Japan<-ggplot(VideoGame, aes(x=na_sales,
y=jp_sales))+geom_point()+stat_smooth(method="lm", col = "red")
NorthAmerica_Japan


grid.arrange(Year_Global,
NorthAmerica_World,NorthAmerica_Europe,Europe_Japan,Japan_Other,NorthAmerica_Japan,
ncol=3,nrow=3)
```

```
#Visualization Word clouds

text<-VideoGame$genre
docs<-Corpus(VectorSource(text))

summary(docs)
docs<-docs%>%
```

```
  tm_map(removeNumbers) %>%
  tm_map(removePunctuation) %>%
  tm_map(stripWhitespace)
docs<-tm_map(docs, content_transformer(tolower))
docs<-tm_map(docs, removeWords, stopwords("30nglish"))
View(docs)
summary(docs)


#Step 3 Create a document-term matrix
dtm<- TermDocumentMatrix(docs)
matrix <-as.matrix(dtm)
words<-sort(rowSums(matrix), decreasing = TRUE)
wordsdf<-data.frame(word = names(words), freq = words)

set.seed(1234)
wordcloud(words = wordsdf$word, freq = wordsdf$freq, min.freq = 1, max.words = 1500,
random.order = FALSE, rot.per=.35, colors = brewer.pal(8, "Dark2"))



#Platform Word Cloud
text<-VideoGame$genre
docs<-Corpus(VectorSource(text))

summary(docs)
docs<-docs%>%
  tm_map(removeNumbers) %>%
  tm_map(removePunctuation) %>%
  tm_map(stripWhitespace)
docs<-tm_map(docs, content_transformer(tolower))
docs<-tm_map(docs, removeWords, stopwords("30nglish"))
View(docs)
summary(docs)


#Step 3 Create a document-term matrix
text1<-VideoGame$platform
docs1<-Corpus(VectorSource(text1))

summary(docs1)

dtm2<- TermDocumentMatrix(docs1)
matrix2 <-as.matrix(dtm2)
```

```
words2<-sort(rowSums(matrix2), decreasing = TRUE)
wordsdf2<-data.frame(word = names(words2), freq = words2)

set.seed(1234)
wordcloud(words = wordsdf2$word, freq = wordsdf2$freq, min.freq = 1, max.words = 300,
random.order = FALSE, rot.per=.35, colors = brewer.pal(8, “Dark2”))


#Publisher Word Cloud
text<-VideoGame$genre
docs<-Corpus(VectorSource(text))

summary(docs)
docs<-docs%>%
  tm_map(removeNumbers) %>%
  tm_map(removePunctuation) %>%
  tm_map(stripWhitespace)
docs<-tm_map(docs, content_transformer(tolower))
docs<-tm_map(docs, removeWords, stopwords(“31nglish”))
View(docs)
summary(docs)


#Step 3 Create a document-term matrix
text2<-VideoGame$publisher
docs2<-Corpus(VectorSource(text2))

summary(docs2)

dtm3<- TermDocumentMatrix(docs2)
matrix3 <-as.matrix(dtm3)
words3<-sort(rowSums(matrix3), decreasing = TRUE)
wordsdf3<-data.frame(word = names(words3), freq = words3)

set.seed(1234)
wordcloud(words = wordsdf3$word, freq = wordsdf3$freq, min.freq = 1, max.words = 300,
random.order = FALSE, rot.per=.35, colors = brewer.pal(8, “Dark2”))
```

```
#Using support vector Machines to predict if a video game will be a hit or a flop.
#Creating testing database and focusing on na_sales and eu_sales.  From our lm there is a
possibility for a correlation between the two markets.
#Doing predictive analysis on both of these markets.
```

```
Northerndf<-data.frame(VideoGame$na_sales, VideoGame$eu_sales)
#Creating cut off point
randIndex1<-sample(1:dim(Northerndf)[1])
summary(randIndex1)
head(randIndex1)
#creating a 2/3 cut point
cutPoint2_3<-floor(2*dim(Northerndf)[1]/3)
cutPoint2_3
#10882which is about 2/3 of 153
#Creating train data
trainData1<-Northerndf[randIndex1[1:cutPoint2_3],]

summary(trainData1)
#Creating test data
testData1<-Northerndf[randIndex1[(cutPoint2_3+1):dim(Northerndf)[1]],]

summary(testData1)

#Creating hit variable
hit_na_videogame <-c()

for (I in 1:nrow(trainData1)) {
  if (trainData1$VideoGame.na_sales[i] < mean(trainData1$VideoGame.na_sales)){
    #cat(I, “hitVideoGame”, “\n”)
    trainData1$hit_na_videogame[i] <- 0
  }
  else {
    trainData1$hit_na_videogame[i] <- 1
    #cat(I, “flopVideoGame”, “\n”)
  }
}

#creating hit variable for Europe

hit_eu_videogame <-c()

for (I in 1:nrow(trainData1)) {
  if (trainData1$VideoGame.eu_sales[i] < mean(trainData1$VideoGame.eu_sales)){
    #cat(I, “hitEUVideoGame”, “\n”)
    trainData1$hit_eu_videogame[i] <- 0
  }
  else {
```

```
   trainData1$hit_eu_videogame[i] <- 1
   #cat(I, “flopEUVideoGame”, “\n”)
  }
}


#Support Vector Machine Output

trainData1<-trainData1[,-4]
trainData1
PredictHitVideoGame <- ksvm(hit_na_videogame~., data = trainData1, kernel = “rbfdot”, kpar =
“automatic”,C=10, cross = 10, prob.model = TRUE)

PredictHitVideoGame
summary(PredictHitVideoGame)


NAHitPredictor <- predict(PredictHitVideoGame, testData1, type = “votes”)
str(NAHitPredictor)
cTable2 <- data.frame(testData1[,1], NAHitPredictor[,1])
colnames(cTable2) <- c(“hit”,”flop”)
cTable2
sqrt(mean((cTable2$test-cTable2$Pred)^2))
cTable2$error <- abs(cTable2$hit – cTable2$flop)
HitsvmPlot <- data.frame(cTable2$error, testData1$VideoGame.na_sales,
testData1$VideoGame.eu_sales)
colnames(HitsvmPlot) <- c(“error”,”Hit”,”Flop”)
HitsvmPlot
HitVGM<-ggplot(HitsvmPlot, aes(x=Hit,y=Flop)) + geom_point(aes(size=error, color=error))
HitVGM

fig8 <- plot_ly(data = HitsvmPlot, x=HitsvmPlot$Hit, y=HitsvmPlot$Flop, color =
HitsvmPlot$error)
fig8

fig8 <- plot_ly(data = HitsvmPlot, x= ~Hit, y= ~Flop, color = ~error)
fig8
```

ENTIRE PROJECT SOURCE CODE

```
########################################
#IST 687
```

```
#Final Project
#Rafael Hernandez
#3/12/21
#######################################
#Libraries: ggplot2, ggmap, data.table, cowplot, maps, mapproj, lubridate, plotly
#Libraries cntd: hrbrthemes, gganimate, gapminder, babynames, ggthemes, gridExtra,
#Continued: tidyr, plyr, tm, SnowballC, wordcloud, RColorBrewer, kernlab.
install.packages("arules")
library(arules)
library(ggplot2)
library(ggmap)
install.packages("data.table")
library(data.table)
install.packages("cowplot")
library(cowplot)
install.packages("maps")
library(maps)
install.packages("mapproj")
library(mapproj)
install.packages("hrbrthemes")
library(hrbrthemes)
install.packages("gganimate")
library(gganimate)
install.packages("gapminder")
library(gapminder)
install.packages("babynames")
library(babynames)
install.packages("ggthemes")
library(ggthemes)
install.packages("e1071")
library(e1071)
install.packages("gridExtra")
library(gridExtra)
library(tidyr)
library(plyr)
install.packages("tm")
install.packages("SnowballC")
install.packages("wordcloud")
install.packages("RColorBrewer")
library(tm)
library(SnowballC)
library(wordcloud)
library(RColorBrewer)
install.packages("kernlab")
```

```
library(kernlab)
library(plotly)

VideoGame<-(vgsales)
str(VideoGame)

VideoGame$Year<-as.Date(VideoGame$Year)

install.packages("lubridate")
library(lubridate)

VideoGame$Year<-as.Date(as.character(VideoGame$Year), format = "%Y")
VideoGame$Year<-year(VideoGame$Year)

#Removing the Rank Column
VideoGame$Rank <- NULL
summary(VideoGame)

#Filtering only the records of interest for this study, removing the records wth Year=nan and
records with the eyar above 2016

VideoGame<- VideoGame[VideoGame$Year != "N/A" & VideoGame$Year != "2017" &
VideoGame$Year != "2020", ]

VideoGame$Year <- factor(VideoGame$Year)
head(VideoGame, 6)

#Renaming columns and lower casing headings.
setnames(VideoGame, old = c("Name", "Platform", "Year", "Genre", "Publisher", "NA_Sales",
"EU_Sales", "JP_Sales", "Other_Sales", "Global_Sales"), new = c("name", "platform", "year",
"genre", "publisher", "na_sales", "eu_sales", "jp_sales", "other_sales", "global_sales"))
summary(VideoGame)

#Create new data table for frequency and percent.
freq_year <- data.frame(cbind(Frequency = table(VideoGame$year), Percent =
prop.table(table(VideoGame$year)) * 100))
freq_year<- freq_year[order(freq_year$Frequency, decreasing = TRUE), ]
freq_year


freq_name <- data.frame(cbind(Frequency = table(VideoGame$name), Percent =
prop.table(table(VideoGame$name))* 100))
freq_name <- head(freq_name[order(freq_name$Frequency, decreasing = T), ], 5)
freq_name
```

```
freq_genre <- data.frame(cbind(Frequency = table(VideoGame$genre), Percent =
prop.table(table(VideoGame$genre)) * 100))
freq_genre <-freq_genre[order(freq_genre$Frequency, decreasing = T), ]
freq_genre

setnames(freq_year, old = c("Frequency", "Percent"), new = c("frequency", "percent"))
freq_year

setnames(freq_genre, old = c("Frequency", "Percent"), new = c("frequency", "percent"))
freq_genre

setnames(freq_name, old = c("Frequency", "Percent"), new = c("frequency", "percent"))
freq_name

#Frequency Distribution of Genre Graph.  Code provided by Murilao on Kaggle.com

options(repr.plot.width = 14, repr.plot.height = 6)
ggplot(data = freq_genre, mapping = aes(x = frequency, y = row.names(freq_genre))) +
  geom_bar(stat = "identity", mapping = aes(fill = row.names(freq_genre), color =
row.names(freq_genre)), aplha = .7,
         size = 1.1) +
        geom_label(mapping = aes(label = frequency), fill = "#B22222", size = 6, color = "white",
fontface = "bold", hjust = .7)+
        ggtitle("Genre Frequency Distribution") + xlab(" ") +
ylab("")+theme_ipsum()+coord_flip()+
        theme(plot.title = element_text(size = 24, hjust = .5, face = "bold"), axis.title =
element_text(size = 24, hjust = .5, face = "italic"),
              axis.title.x = element_text(size = 24, hjust = .5, face = "italic"),
              axis.title.y = element_text(size = 24, hjust = .5, face = "italic"),
              axis.text.x = element_text(size = 20, face = "bold", angle = 20),
              axis.text.y = element_text(size = 20, face = "bold"),
              legend.position = "none")

summary(VideoGame)

install.packages("plotly")
library(plotly)

fig<-plot_ly(
  x=c(VideoGame$genre),
  y=c(VideoGame$na_sales),
  name = "Genre Sales",
```

```
  type = "bar"
)
fig

#Code provided by plotly.com
fig1<-plot_ly(VideoGame, x =~genre, y=~na_sales, type = 'bar', name = 'North America Sales')
fig1 <-fig1 %>% add_trace(y = ~eu_sales, name = 'Europe Sales')
fig1 <-fig1 %>% add_trace(y = ~jp_sales, name = 'Japan Sales')
fig1 <-fig1 %>% add_trace(y = ~other_sales, name = 'Other Markets')
fig1 <-fig1 %>% add_trace(y = ~global_sales, name = 'Global Sales')
fig1 <- fig1 %>% layout(yaxis =list(title = 'Sales'), barmode = 'group')
fig1

fig2<-plot_ly(VideoGame, x =~genre, y=~na_sales, type = 'bar', name = 'North America Sales')
fig2 <-fig2 %>% add_trace(y = ~eu_sales, name = 'Europe Sales')
fig2 <- fig2 %>% layout(yaxis =list(title = 'Sales'), barmode = 'group')
fig2

fig3<-plot_ly(VideoGame, x =~genre, y=~na_sales, type = 'bar', name = 'North America Sales')
fig3 <-fig3 %>% add_trace(y = ~jp_sales, name = 'Japan Sales')
fig3 <- fig3 %>% layout(yaxis =list(title = 'Sales'), barmode = 'group')
fig3


fig4<-plot_ly(VideoGame, x =~genre, y=~na_sales, type = 'bar', name = 'North America Sales')
fig4 <-fig4 %>% add_trace(y = ~other_sales, name = 'Other Markets')
fig4 <- fig4 %>% layout(yaxis =list(title = 'Sales'), barmode = 'group')
fig4

#################################################################
#Mean Median and Mode
#Median code provided by Murilao via Kaggle.com
median_Df<-data.frame(Median = c(median(VideoGame$na_sales),
median(VideoGame$eu_sales),
                      median(VideoGame$jp_sales), median(VideoGame$other_sales),
                      median(VideoGame$global_sales)))

row.names(median_Df)<- c("na_sales", "eu_sales", "jp_sales", "other_sales", "global_sales")
median_Df

#Mode code provided by Mrilao via Kaggle.com
mode_df<-data.frame(Mode = c(mode(VideoGame$na_sales), mode(VideoGame$eu_sales),
                     mode(VideoGame$jp_sales), mode(VideoGame$other_sales),
                     mode(VideoGame$global_sales)))
```

```
row.names(mode_df)<- c("na_sales", "eu_sales", "jp_sales", "other_sales", "global_sales")
mode_df

#Central Tendencies
central_tend<-data.frame(means_df, median_Df, mode_df)
central_tend
row.names(central_tend)<-c("Mean", "Median", "Mode")

central_tend<-central_tend[,-3]
central_tend

options(repr.plot.width = 14, repr.plot.height = 6)
a <- ggplot(data = means_df, mapping = aes(x = Mean, y = row.names(means_df))) +
 geom_line(group = 1, size = 1.2, linetype = "dashed", color = "blue") +
 geom_point(size = 5, shape = 21, stroke = 1.5, mapping = aes(fill = row.names(means_df))) +
 theme_minimal() +
 ylab("") +
 theme(plot.title = element_text(size = 24, hjust = .5, face = "bold"),
      axis.title.x = element_text(size = 24, hjust = .5, face = "italic"),
      axis.title.y = element_text(size = 24, hjust = .5, face = "italic"),
      axis.text.x = element_text(size = 18, face = "bold"),
      axis.text.y = element_text(size = 18, face = "bold"),
      legend.position = "none")

b <- ggplot(data = means_df, mapping = aes(x = Mean, y = row.names(means_df))) +
 geom_line(group = 1, size = 1.2, linetype = "dashed", color = "blue") +
 geom_point(size = 5, stroke = 1.5, shape = 21, mapping = aes(fill = row.names(means_df))) +
 theme_minimal() +
 ylab("") +
 xlab("") +
 theme(plot.title = element_text(size = 24, hjust = .5, face = "bold"),
      axis.title.x = element_text(size = 24, hjust = .5, face = "italic"),
      axis.title.y = element_text(size = 24, hjust = .5, face = "italic"),
      axis.text.x = element_text(size = 18, face = "bold"),
      axis.text.y = element_text(size = 18, face = "bold"),
      legend.position = "bottom",
      legend.title = element_text(color = "white"),
      legend.text = element_text(size = 12, face = "bold"))

plot_grid(a, b + coord_polar(), nrow = 1, ncol = 2)
```

```
#Code provided by plotly
Sales_Means<- data.frame("Categorie"=rownames(means_df), means_df)
data<-Sales_Means[c('Categorie', 'Mean')]
fig5 <- plot_ly(data, labels = ~Categorie, values = ~Mean, type = 'pie')
fig5 <- fig5%>% layout(title = 'Video Game Sales by Market Means',
                       xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
                       yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))

fig5


# NA_Sales #code provided by murilao via Kaggle.com
t_v_name_NA <- aggregate(list(na_sales = VideoGame$na_sales), list(Name =
VideoGame$name), sum)
t_v_name_NA <- t_v_name_NA[order(t_v_name_NA$na_sales, decreasing = T), ]
t_v_name_NA

t_v_name_EU<- aggregate(list(eu_sales= VideoGame$eu_sales), list(Name =
VideoGame$name), sum)
t_v_name_EU<-t_v_name_EU[order(t_v_name_EU$eu_sales, decreasing = T),]
t_v_name_EU


#Code provided by plotly
fig6<- plot_ly(data= head(t_v_name_NA, 10), x=~Name, y =~na_sales, type = 'bar', name =
'North America Top')
fig6 <- fig6 %>% layout(yaxis =list(title = 'North America Sales in Millions'))
fig6

fig9<- plot_ly(data=head(t_v_name_EU, 10), x=~Name, y =~eu_sales, type = 'bar', name =
'Europe Top')
fig9<- fig9 %>% layout(yaxis = list(title = 'Europe Sales in Millions'))
fig9

# Global_Sales code provided by murilao via kaggle.com
t_v_name_Global <- aggregate(list(global_sales = VideoGame$global_sales), list(Name =
VideoGame$name), sum)
t_v_name_Global <- t_v_name_Global[order(t_v_name_Global$global_sales, decreasing = T), ]
t_v_name_Global

fig7<- plot_ly(data= head(t_v_name_Global, 10), x=~Name, y =~global_sales, type = 'bar', name
= 'Global Top 10 Sales')
```

```
fig7 <- fig7 %>% layout(yaxis =list(title = 'Global Sales in Millions'))
fig7



#################################################################
#Measures of dispersion
Sales_Variance <- data.frame(Variance = c(var(VideoGame$na_sales),
var(VideoGame$eu_sales), var(VideoGame$jp_sales), var(VideoGame$other_sales),
var(VideoGame$global_sales)))
row.names(Sales_Variance) <- c("na_sales", "eu_sales", "jp_sales", "other_sales",
"global_sales")
Sales_Variance

Sales_std <- data.frame(std = c(sqrt(var(VideoGame$na_sales)),
sqrt(var(VideoGame$eu_sales)), sqrt(var(VideoGame$jp_sales)),
sqrt(var(VideoGame$other_sales)), sqrt(var(VideoGame$global_sales))))
row.names(Sales_std) <- c("na_sales", "eu_sales", "jp_sales", "other_sales", "global_sales")
Sales_std4

Sales_Dispersion <- data.frame(DM = Sales_Means$Mean, Variance = Sales_Variance$Variance,
std = Sales_std$std)
row.names(Sales_Dispersion) <- c("na_sales", "eu_sales", "jp_sales", "other_sales",
"global_sales")
Sales_Dispersion



#####################################################################
#Building and plotting linear models
#Plotting to find dependencies
plot(VideoGame$global_sales, VideoGame$na_sales)
plot(VideoGame$global_sales, VideoGame$eu_sales)
plot(VideoGame$global_sales, VideoGame$jp_sales)
plot(VideoGame$global_sales, VideoGame$other_sales)



plot(VideoGame$year, VideoGame$na_sales)
plot(VideoGame$year, VideoGame$eu_sales)
plot(VideoGame$year, VideoGame$jp_sales)
plot(VideoGame$year, VideoGame$other_sales)

#building linear models
```

```
model1<- lm(formula=na_sales~global_sales, data=VideoGame)
summary(model1)
plot(VideoGame$global_sales, VideoGame$na_sales)
abline(model1)
NorthAmerica_World<-ggplot(VideoGame, aes(x=global_sales,
y=na_sales))+geom_point()+stat_smooth(method="lm", col = "red")
NorthAmerica_World

#adding year to the variable.  There appears to be no correlation between years and video
game sales.  There is no slope.  There does appear to be a correlation between global sales and
sales in indidividual markets.  How about sales between different markets?
model2<-lm(formula=global_sales~year, data=VideoGame)
model2
plot(VideoGame$year, VideoGame$global_sales)
abline(model2)
Year_Global<-ggplot(VideoGame, aes(x=year,
y=global_sales))+geom_point()+stat_smooth(method="lm", col = "red")
Year_Global


#comparing markets
model3<-lm(formula=eu_sales~na_sales, data=VideoGame)
summary(model3)
plot(VideoGame$na_sales, VideoGame$eu_sales)
abline(model3)
NorthAmerica_Europe<-ggplot(VideoGame, aes(x=na_sales,
y=eu_sales))+geom_point()+stat_smooth(method="lm", col = "red")
NorthAmerica_Europe

model4<-lm(formula=eu_sales~jp_sales, data=VideoGame)
summary(model4)
plot(VideoGame$eu_sales, VideoGame$jp_sales)
abline(model4)
Europe_Japan<-ggplot(VideoGame, aes(x=jp_sales,
y=eu_sales))+geom_point()+stat_smooth(method="lm", col = "red")
Europe_Japan


model5<-lm(formula=other_sales~jp_sales, data=VideoGame)
summary(model5)
plot(VideoGame$jp_sales, VideoGame$other_sales)
abline(model5)
Japan_Other<-ggplot(VideoGame, aes(x=jp_sales,
y=other_sales))+geom_point()+stat_smooth(method="lm", col = "red")
```

Japan_Other

```
model6<-lm(formula=jp_sales~na_sales, data=VideoGame)
summary(model6)
plot(VideoGame$na_sales, VideoGame$jp_sales)
abline(model6)
NorthAmerica_Japan<-ggplot(VideoGame, aes(x=na_sales,
y=jp_sales))+geom_point()+stat_smooth(method="lm", col = "red")
NorthAmerica_Japan


grid.arrange(Year_Global,
NorthAmerica_World,NorthAmerica_Europe,Europe_Japan,Japan_Other,NorthAmerica_Japan,
ncol=3,nrow=3)

############################################################
#Visualization Word clouds

text<-VideoGame$genre
docs<-Corpus(VectorSource(text))

summary(docs)
docs<-docs%>%
  tm_map(removeNumbers) %>%
  tm_map(removePunctuation) %>%
  tm_map(stripWhitespace)
docs<-tm_map(docs, content_transformer(tolower))
docs<-tm_map(docs, removeWords, stopwords("english"))
View(docs)
summary(docs)


#Step 3 Create a document-term matrix
dtm<- TermDocumentMatrix(docs)
matrix <-as.matrix(dtm)
words<-sort(rowSums(matrix), decreasing = TRUE)
wordsdf<-data.frame(word = names(words), freq = words)

set.seed(1234)
wordcloud(words = wordsdf$word, freq = wordsdf$freq, min.freq = 1, max.words = 1500,
random.order = FALSE, rot.per=.35, colors = brewer.pal(8, "Dark2"))
```

```
#Platform Word Cloud
text<-VideoGame$genre
docs<-Corpus(VectorSource(text))

summary(docs)
docs<-docs%>%
  tm_map(removeNumbers) %>%
  tm_map(removePunctuation) %>%
  tm_map(stripWhitespace)
docs<-tm_map(docs, content_transformer(tolower))
docs<-tm_map(docs, removeWords, stopwords("english"))
View(docs)
summary(docs)


#Step 3 Create a document-term matrix
text1<-VideoGame$platform
docs1<-Corpus(VectorSource(text1))

summary(docs1)

dtm2<- TermDocumentMatrix(docs1)
matrix2 <-as.matrix(dtm2)
words2<-sort(rowSums(matrix2), decreasing = TRUE)
wordsdf2<-data.frame(word = names(words2), freq = words2)

set.seed(1234)
wordcloud(words = wordsdf2$word, freq = wordsdf2$freq, min.freq = 1, max.words = 300,
random.order = FALSE, rot.per=.35, colors = brewer.pal(8, "Dark2"))


#Publisher Word Cloud
text<-VideoGame$genre
docs<-Corpus(VectorSource(text))

summary(docs)
docs<-docs%>%
  tm_map(removeNumbers) %>%
  tm_map(removePunctuation) %>%
  tm_map(stripWhitespace)
docs<-tm_map(docs, content_transformer(tolower))
docs<-tm_map(docs, removeWords, stopwords("english"))
View(docs)
summary(docs)
```

```
#Step 3 Create a document-term matrix
text2<-VideoGame$publisher
docs2<-Corpus(VectorSource(text2))

summary(docs2)

dtm3<- TermDocumentMatrix(docs2)
matrix3 <-as.matrix(dtm3)
words3<-sort(rowSums(matrix3), decreasing = TRUE)
wordsdf3<-data.frame(word = names(words3), freq = words3)

set.seed(1234)
wordcloud(words = wordsdf3$word, freq = wordsdf3$freq, min.freq = 1, max.words = 300,
random.order = FALSE, rot.per=.35, colors = brewer.pal(8, "Dark2"))


############################################################
#Using support vector Machines to predict if a video game will be a hit or a flop.
#Creating testing database and focusing on na_sales and eu_sales.  From our lm there is a
possibility for a correlation between the two markets.
#Doing predictive analysis on both of these markets.


Northerndf<-data.frame(VideoGame$na_sales, VideoGame$eu_sales)
#Creating cut off point
randIndex1<-sample(1:dim(Northerndf)[1])
summary(randIndex1)
head(randIndex1)
#creating a 2/3 cut point
cutPoint2_3<-floor(2*dim(Northerndf)[1]/3)
cutPoint2_3
#10882which is about 2/3 of 153
#Creating train data
trainData1<-Northerndf[randIndex1[1:cutPoint2_3],]

summary(trainData1)
#Creating test data
testData1<-Northerndf[randIndex1[(cutPoint2_3+1):dim(Northerndf)[1]],]

summary(testData1)

#Creating hit variable
```

```
hit_na_videogame <-c()

for (i in 1:nrow(trainData1)) {
  if (trainData1$VideoGame.na_sales[i] < mean(trainData1$VideoGame.na_sales)){
    #cat(i, "hitVideoGame", "\n")
    trainData1$hit_na_videogame[i] <- 0
  }
  else {
    trainData1$hit_na_videogame[i] <- 1
    #cat(i, "flopVideoGame", "\n")
  }
}

#creating hit variable for Europe

hit_eu_videogame <-c()

for (i in 1:nrow(trainData1)) {
  if (trainData1$VideoGame.eu_sales[i] < mean(trainData1$VideoGame.eu_sales)){
    #cat(i, "hitEUVideoGame", "\n")
    trainData1$hit_eu_videogame[i] <- 0
  }
  else {
    trainData1$hit_eu_videogame[i] <- 1
    #cat(i, "flopEUVideoGame", "\n")
  }
}


#Support Vector Machine Output

trainData1<-trainData1[,-4]
trainData1
PredictHitVideoGame <- ksvm(hit_na_videogame~., data = trainData1, kernel = "rbfdot", kpar =
"automatic",C=10, cross = 10, prob.model = TRUE)

PredictHitVideoGame
summary(PredictHitVideoGame)


NAHitPredictor <- predict(PredictHitVideoGame, testData1, type = "votes")
str(NAHitPredictor)
cTable2 <- data.frame(testData1[,1], NAHitPredictor[,1])
colnames(cTable2) <- c("hit","flop")
```

```
cTable2
sqrt(mean((cTable2$test-cTable2$Pred)^2))
cTable2$error <- abs(cTable2$hit - cTable2$flop)
HitsvmPlot <- data.frame(cTable2$error, testData1$VideoGame.na_sales,
testData1$VideoGame.eu_sales)
colnames(HitsvmPlot) <- c("error","Hit","Flop")
HitsvmPlot
HitVGM<-ggplot(HitsvmPlot, aes(x=Hit,y=Flop)) + geom_point(aes(size=error, color=error))
HitVGM

fig8 <- plot_ly(data = HitsvmPlot, x=HitsvmPlot$Hit, y=HitsvmPlot$Flop, color =
HitsvmPlot$error)
fig8

fig8 <- plot_ly(data = HitsvmPlot, x= ~Hit, y= ~Flop, color = ~error)
fig8
```