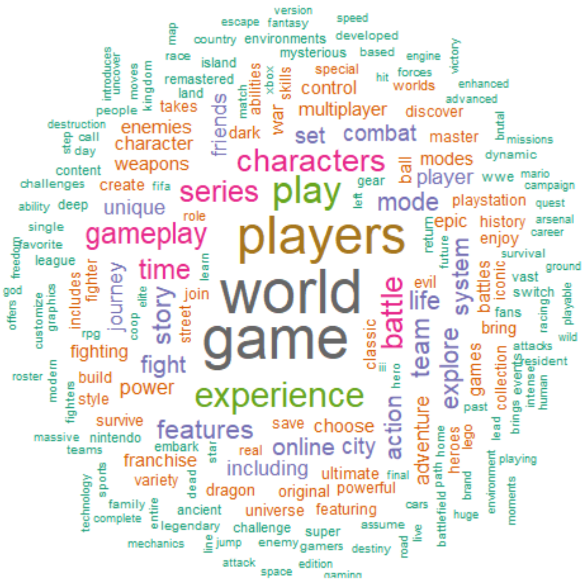
VIDEO GAME SALES DATA ANALYSIS



10/26/2019

Exploring the Multi-Billion Dollar Industry

Aaron Carmichael – A20003997

Emily Jacobs – A20005192

Arka De – A20231272

Logan Davis – A20107383

Video Game Sales Data Analysis

EXPLORING THE MULTI-BILLION DOLLAR INDUSTRY

CONTENTS

TABLES

FIGURES

I. EXECUTIVE SUMMARY

In recent decades, video game popularity has grown exponentially, earning them tremendous influence in our popular culture. However, what attributes do these popular video games have that causes them to resonate so strongly with society? Our goal for this project is to collect and analyze data to find out what makes a video game successful. To quantify success, we will be looking at sales. The research, planning, and development of a videogame can take years and are accompanied by high costs. As a result, it is in the best interest of the companies that develop these games to have successful products. In order to obtain success, companies need to have a solid understanding of the current trends in the industry. Our hope with this project is to establish those trends that create a successful video game for the current generation of consoles. This will be done through the analysis of two perspectives of data; the first data set looks at sales data itself, while the second looks at the attributes that would be important to the consumer, such as genre, content rating, and ranking, among other attributes.

I. STATEMENT OF SCOPE

Our purpose for this project is to collect and analyze data to find out what attributes of a video game correlate with higher sales. This will be done through the analysis of data relating aspects of sales figures, including units sold, as well as consumer information about the game. The data will focus on the current generation of consoles, being the PlayStation 4, the Xbox One, and the Nintendo Switch, and will focus on games sold between approximately 2013 to 2018. Our data will consist of the 30 top selling video games per week of the current generation dating back to their release.

A. Objectives

- Examine the common genres, developers, publishers, release dates, and content ratings that relate to the number of units sold
- Analyze the fluctuations of ranking with the amount of time since the release date according to the number of units sold.
- Gather a sample of approximately 6000 observations of the top 30 selling video games for the current generation of consoles to generalize to modern, current generation video games

B. Target Variables

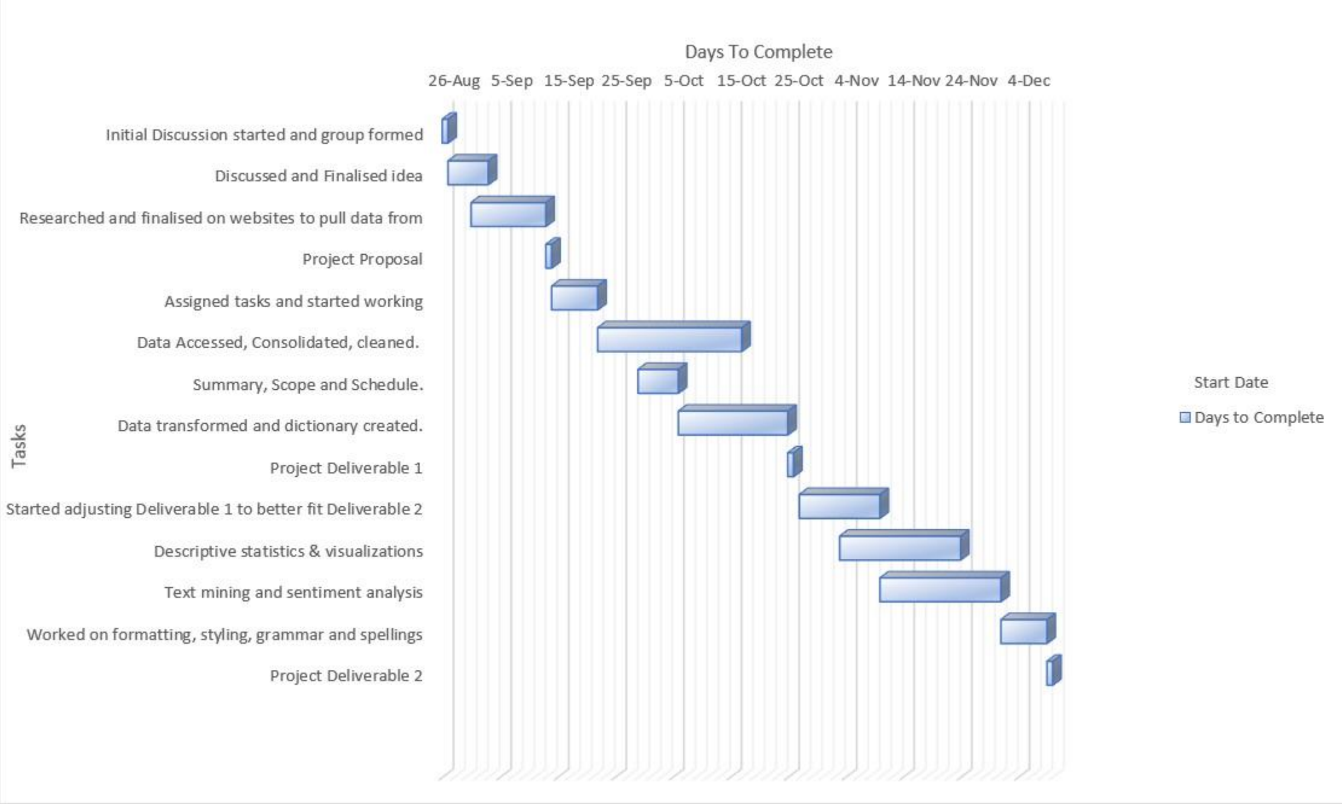
Our target variable is the total units sold per video game. Our objective is to establish a relationship between sales totals and the following:

- Consumer information such as the genre, developer, and key words within an aggregated summary.
- Sales information such as the number of weeks it was a top selling video game, as well as the ranking.

II. PROJECT SCHEDULE

The project is projected and on track to be completed by the conclusion of the semester. Each week, typically on Sunday afternoons, we meet as a team to discuss progress and future steps. The specific times of these meetings are flexible in order to accommodate for the ever-changing schedules of all members. At these meetings, we establish the responsibilities held by each individual member. Aaron oversaw programming by revising all contributions to the code, Arka handled scheduling and delegating, Emily oversaw documentation, and Logan created the data dictionary and kept track of data being used. Each member contributed to the coding of the web crawlers. For the second portion of the project Arka and Logan worked on the statistical analysis and refining the documentation, Aaron worked on the text mining and sentiment analysis and Emily worked on the visualizations. In between meeting times, we communicate several times a week through WhatsApp and have been completing documentation through both a shared Google Drive folder and shared Google Documents. This has allowed us to maintain communication with each other, as well as given each team member the ability to see changes in real time and have access to the most up to date versions of documentation or programs, including data files and scripts. In regards to holidays, we have been maintaining our typical weekly meeting, occurring when we return or completing the meeting through phone calls or online communication accommodating for travel.

Figure 1: GANTT Chart for Project Schedule



III. DATA PREPARATION

A. Data Access

We are using two sources for our analysis, VGChartz and Metacritic. VGChartz is a website that releases weekly sales charts for video games dating back to 2004 that display the top seventy-five selling games, their genre, their publisher, their sales in units for the week, and their total sales in units. This data has contributed to our project by not only providing the sales aspect of our research, but by providing a quantitative measure of success for a video game. This information was scraped through a web crawler that pulled each individual chart and combined them all into one single data frame. Our primary obstacle that we encountered was in regards to VGChartz blocking some data behind a paywall. VGChartz provides access to all data through an expensive subscription, but the top 30 selling games were still fully visible on every chart. To account for this within our web crawler, the program drops all hidden data (the values showed up as "Pro") after obtaining all of the charts. The source code for this crawler can be found in the Appendix. Metacritic has provided the project with data that a potential customer would want to know before purchasing the game, and it was also scraped with a crawler. This crawler obtains variables such as the number of players, content rating, genre, developer, the release date, rating score, the Metacritic score, and a summary. It begins as an extension of the VGChartz data, as it first imports the data exported by the VGChartz web crawler. Within the data frame that houses the VGChartz data is an extra column that uses the title taken from VGChartz to create a URL compatible with Metacritic. This is then used to pass through the Metacritic web crawler to scrape the data. Due to constraints of time and resources in scraping this data, it was divided into five indexes to enable parallel processing in different Rselenium environments.

B. Data Consolidation

In terms of consolidating the data, much of this is handled through the web crawlers themselves. The VGChartz crawler, as it gathers the individual charts, assembles them into one final data frame. The Metacritic crawler takes the final data frame from the VGChartz crawler and, using the url_title column, adds columns on to this data frame, one row at a time. At various points of the script, they directly write the data into a CSV file, and all of these files are kept for backup purposes. After it finishes, the data frame exported by the Metacritic crawler is ready for use.

C. Data Cleaning

Much of our cleaning revolved around the creation of the URL with the title created in the VGChartz web crawler to be used in the Metacritic web crawler. Not all of the created URLs were the correct one to use for Metacritic. Additionally, not every game found on VGChartz can be found on Metacritic. This created some missing values within the data set. The primary means of addressing this issue involved analyzing which faulty titles resulted in the most missing values. Those titles and their respective variables were hard-coded in after the initial scraping, and eliminated most of the missing values. Additionally, the genre and developer pulled from Metacritic contained the words “Genre(s):” and “Developer:” at the beginning, which have been addressed through the use of substrings. Finally, by default, most variables were character strings, so they have been converted to their respective data types. This will help us compare the different values in a more efficient manner. For example, genre is converted to a categorical variable. This way, the genre can be compared with other games in the same genre or against other genres to see which has more sales. Our final data set does contain missing values. However, the ratio of missing values is low, and those observations still have data that can be used, so they were kept. In order to better process the visualizations and other analysis we used an SQL query to narrow the create a new data frame that only listed the video game title once with the highest units sold. The reasoning behind this is a video game could have been present on the chart for multiple weeks. This allowed us to ensure that video games were not being double counted in our analysis.

D. Data Transformation

A few columns were added based on the columns that were scraped. For example, the URL within the VGChartz data was used to create the URL to be used for the Metacritic crawler. Additionally, in order to make the two variables comparable, two columns have been added to transform the user score, which is formatted as a decimal on a scale of 1 to 10 (e.g. 8.4) and MetaScore, which is formatted as a whole number on a scale of 1 to 100 (e.g. 84). One column multiplies the UserScore by 10, and another column measures the difference between the two, providing additional insight to the relationship between sales and rating. Additionally, variables such as genre and publisher were made into categorical variables. This way, sales figures can be easily compared between these categories.

E. Data Reduction

After the initial acquisition of our data, which included sales figures dating back to 2004, we decided to narrow the scope of our data to only the current generation of consoles. This way, the data would focus on more recent and relevant technology as well as sales figures. The video game industry has changed drastically since 2004, and by limiting our data to the current generation of consoles, it can better reflect current trends. By representing more current trends, insights found in the data will be more relevant to our business case, and by extension, will be more useful to a company in the industry. Our goal within this project is to identify trends within the video game industry to establish what attributes make a video game successful in terms of sales, and insights derived from our data can provide information on these trends. This research would allow a company to have a better strategic advantage in their business strategy. The reason that this would be beneficial within this industry is that it would allow them to make educated decisions on the attributes that are the focal point of the game, while also being able to analyze what the gaps in the market are. Another benefit that came from this shift of scope, is it allowed the amount of time spent on the data acquisition portion of the project to decrease fairly significantly. In narrowing the scope, it has allowed the quality of the data to increase as more time was allocated to cleaning and transforming the data.

F. Data Dictionary

Table 1: Data Dictionary

Attribute Name	Description	Data Type	Source	Example
week_pos	The week that the chart was posted	Date	http://www.Vgchartz.com/weekly/	2018-12-29
position	The rank of the video game of that week. 1-30	int	http://www.Vgchartz.com/weekly/	1
title	The full title of the game.	chr	http://www.Vgchartz.com/weekly/	Super Smash Bros. Ultimate
URL-title	The title within the URL connected to the Metacritic website.	chr	http://www.Vgchartz.com/weekly/	super-smash-bros-ultimate
console	The game console the game was created for	factor	http://www.Vgchartz.com/weekly/	Switch
publisher	The creator and distributor of the game.	factor	http://www.Vgchartz.com/weekly/	Nintendo
genre	The genre of the video game.	factor	http://www.Vgchartz.com/weekly/	Fighting
unit_week	The number of units sold during the specific week.	numeric	http://www.Vgchartz.com/weekly/	1103894
unit_total	The total number of units sold.	numeric	http://www.Vgchartz.com/weekly/	8951009

chart_week	The number of weeks the game has been on the charts.	int	http://www.Vgchartz.com/weekly/	4
numplayers	The maximum number of players per game.	factor	https://www.Metacritic.com/game	Up to 8
esrb	The content rating of the game posted by the Entertainment Software Rating Board.	factor	https://www.Metacritic.com/game	E10+
metaGenre	The genre of the game from the Metacritic website.	chr	https://www.Metacritic.com/game	Action, Fighting, 2D
developer	The developer(s) listed on Metacritic.	factor	https://www.Metacritic.com/game	Nintendo, HAL Labs, Bandai Namco Games, Sora Ltd.
release_date	The video games original release date.	Date	https://www.Metacritic.com/game	7-Dec-18
metascore	The score of the game rated by video game critics.	numeric	https://www.Metacritic.com/game	93
user_score	The score of the game rated by users	numeric	https://www.Metacritic.com/game	8.7
user_score_x10	User_score multiplied by 10	numeric	https://www.Metacritic.com/game	87
score_diff	The difference between metascore and user_score_x10	numeric	https://www.Metacritic.com/game	6
summary	The description of the video game.	chr	https://www.Metacritic.com/game	Inklings from the Splatoon series, as well as returning Smash characters....

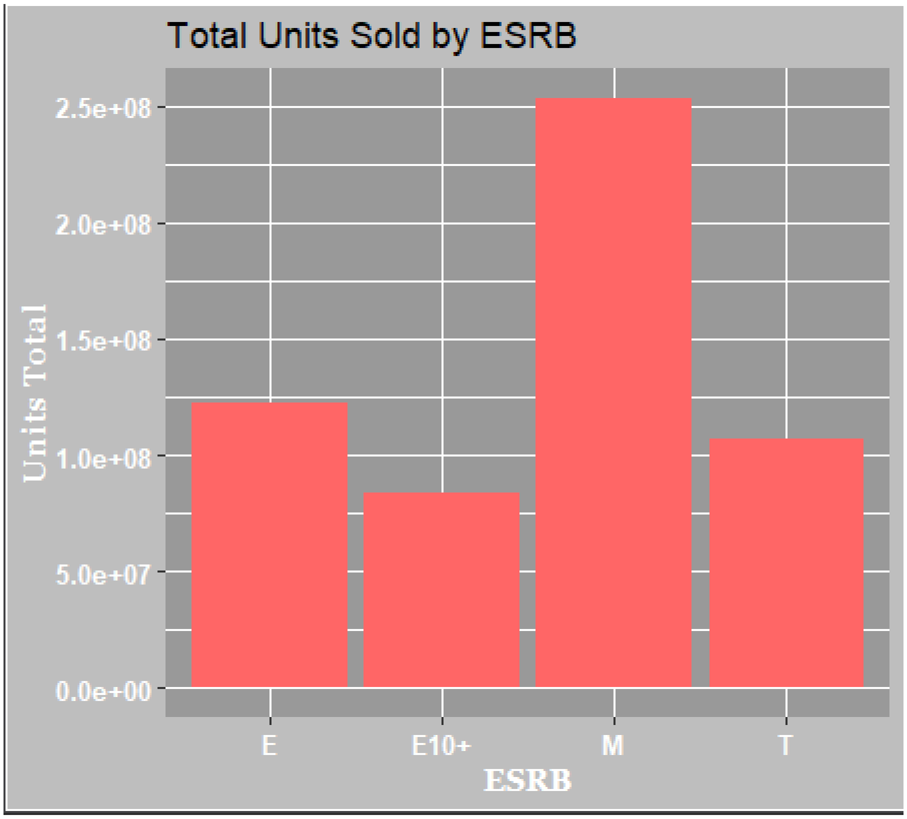
IV. VISUALIZATION

The following visualizations analysis the variables within the dataset. Given the nature of our dataset, there is the potential that a unique video game could be on a chart for multiple weeks. To ensure that the video games were not double counted for this analysis a separate data frame was created using an SQL query (see appendix) that narrowed down the data to only show unique titles with the highest grossing units sold. This has allowed us to verify that our analysis is not screwed by a popular game that was on the charts for a number of weeks. This data frame was used for the visualizations unless otherwise noted.

A. Target Variables with Predicting Variables

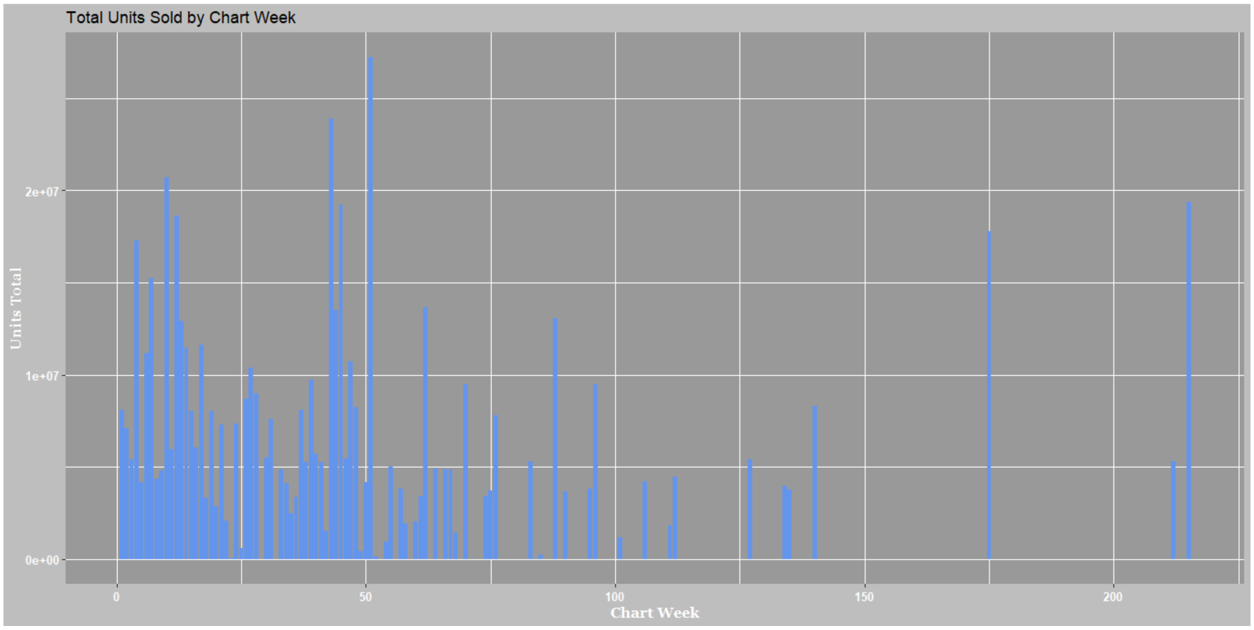
The target variable for this analysis is the number of units sold (total units sold) This is put into comparison with our predictor variables, ESRB, Chart Week, Console, Genre, Score Diff, User Score, MetaScore, Developer and Publisher.

Figure 2: Units Sold by ESRB Rating



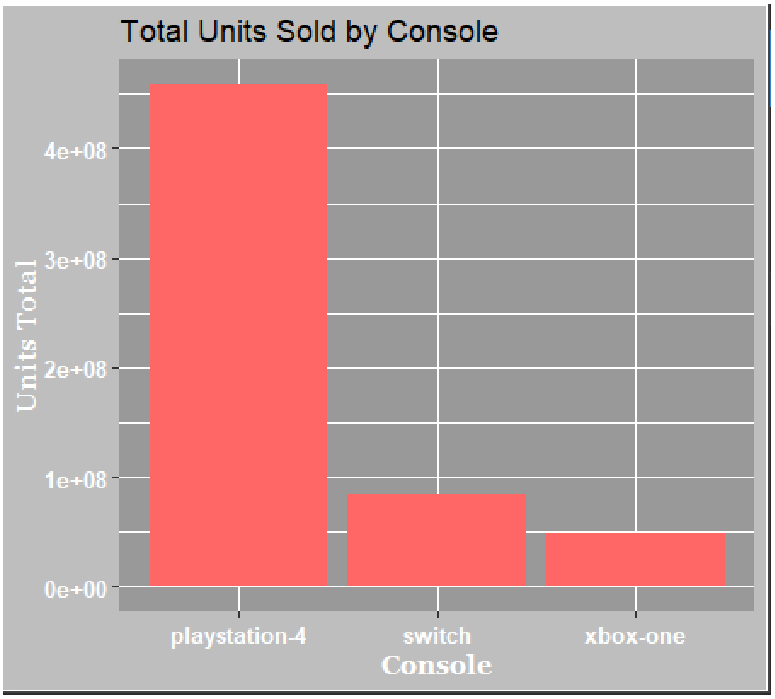
This chart is looking at the units sold per ESRB rating. An ESRB rating is a scale used to establish what audience is appropriate for the game. The ESRB rating that has the highest total units sold is the mature rating.

Figure 3: Total Units Sold by Chart Week



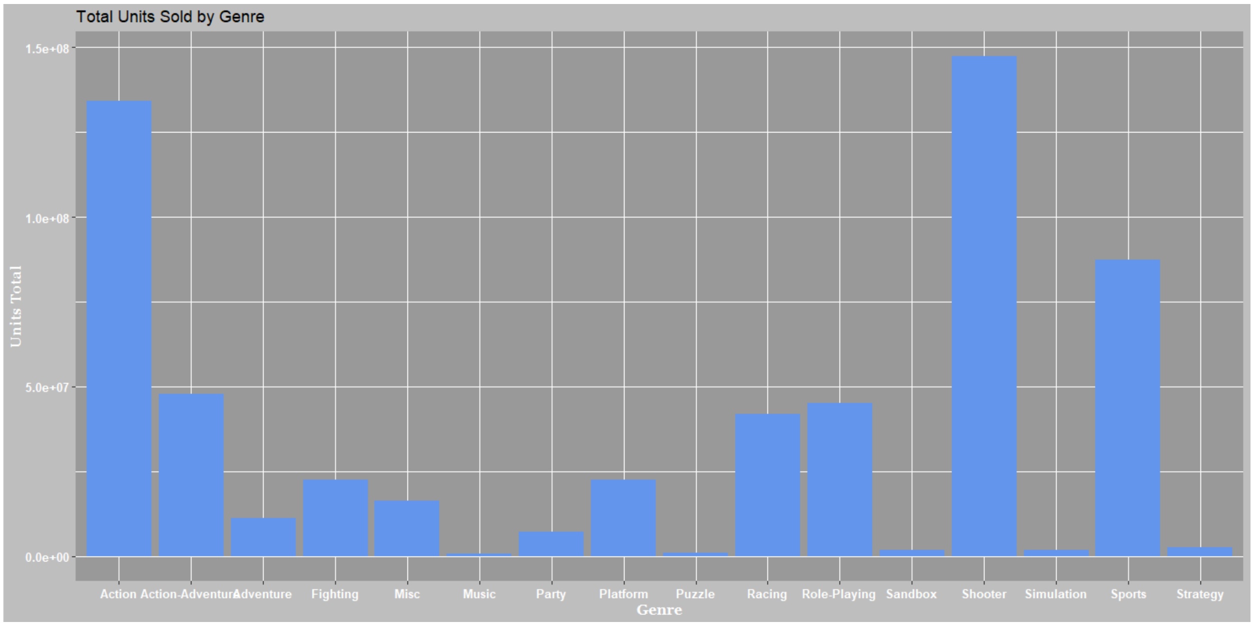
This figure is looking at the total units sold by chart week, the chart week is the number of weeks that the video game has been on the chart. Given that this graph was created with the highest cumulative units sold, this represents the week that this happened. The spread is anywhere from 1-200s range, with the most units sold being found in the 50s.

Figure 4: Total Units Sold by Console



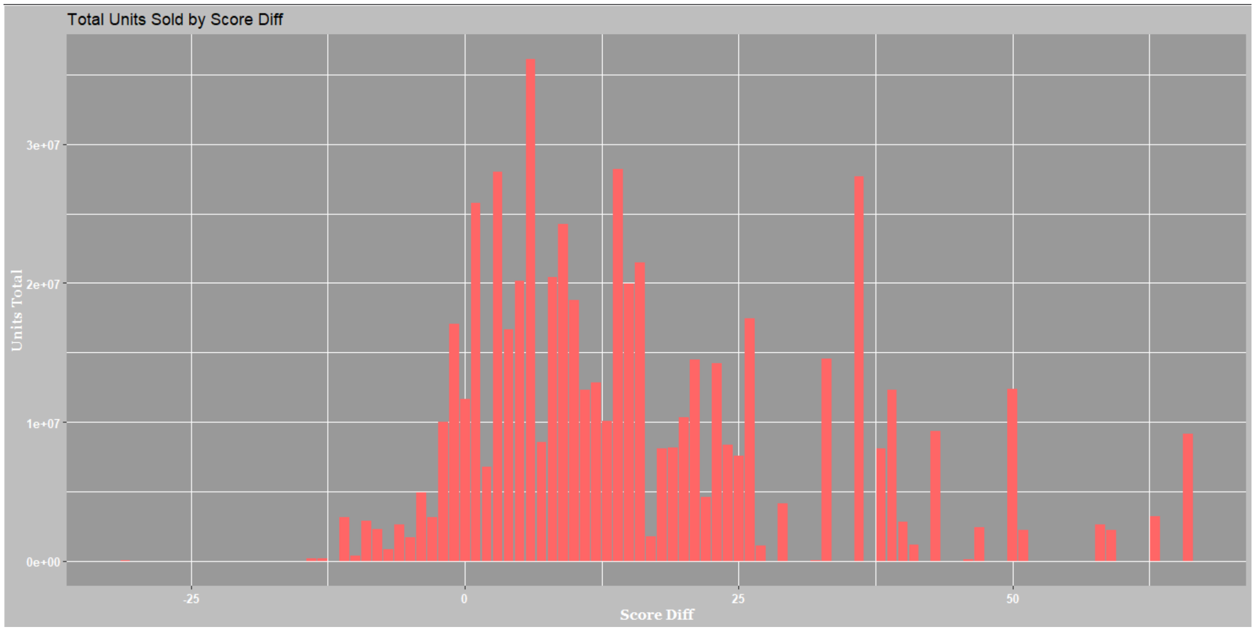
This graph is looking at the total units sold for each console. A disclaimer about this particular graph, is there is a possibility that a video game could be available on more than one console, and VGChartz only has it listed as one. Given the listing of the video games on VGChartz, video games for the PlayStation 4 have the highest total unit sales.

Figure 5: Total Units Sold by Genre



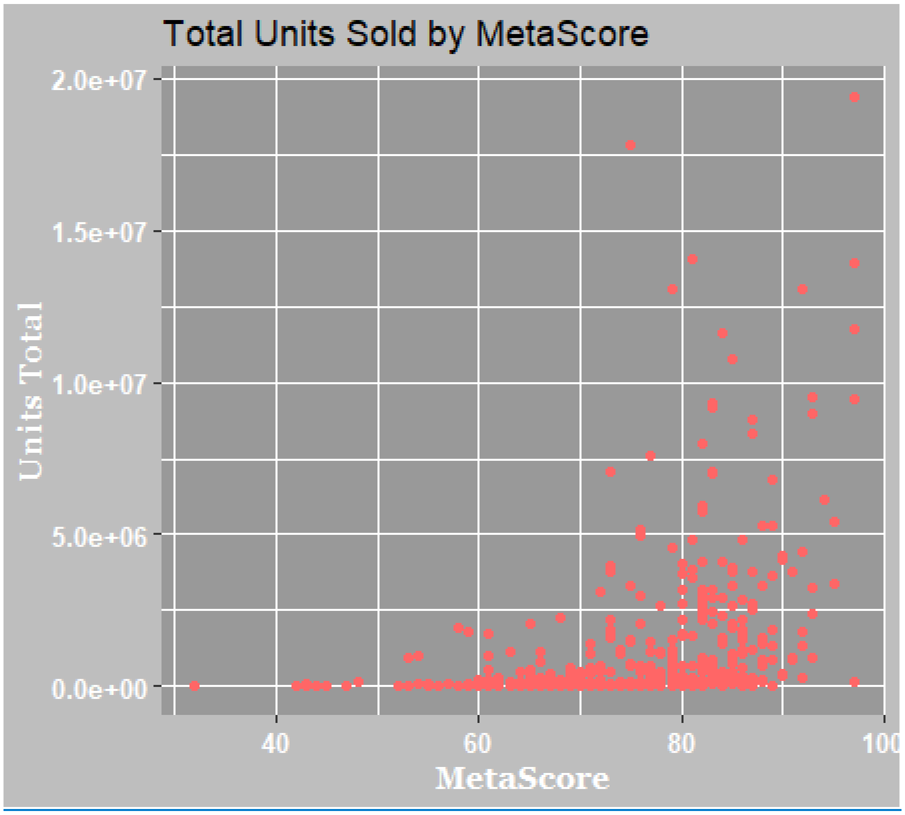
This chart is looking at the total units sold per genre, the highest is shooter games, followed by action and sports, the lowest was music and puzzles.

Figure 6: Total Units Sold by Score Diff



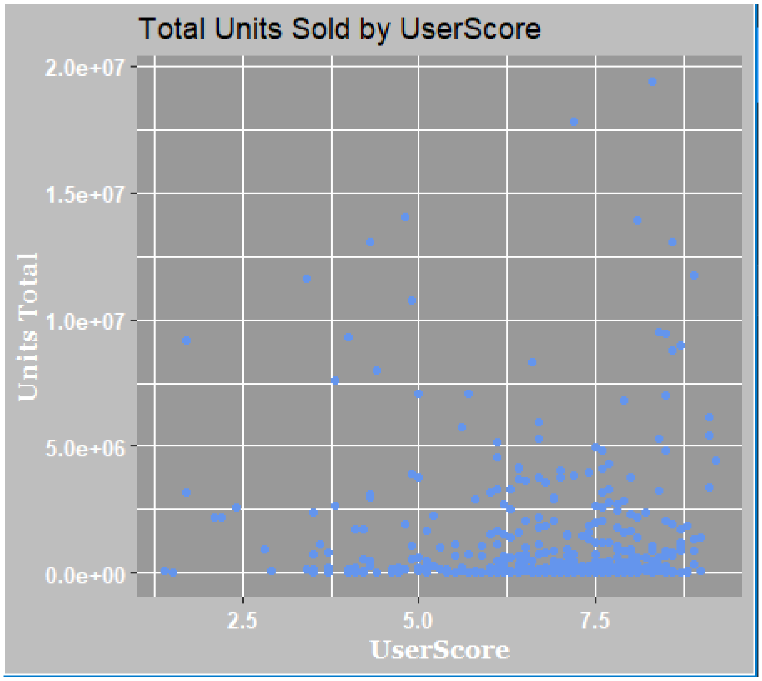
This chart is looking at the score difference and units total. The Score_Diff is the difference between the MetaScore and the userscoreX10. The majority of games have a slight difference in scores, but there are some outliers with abnormally high differences that have still attained commercial success.

Figure 7: Total Units Sold by Metascore



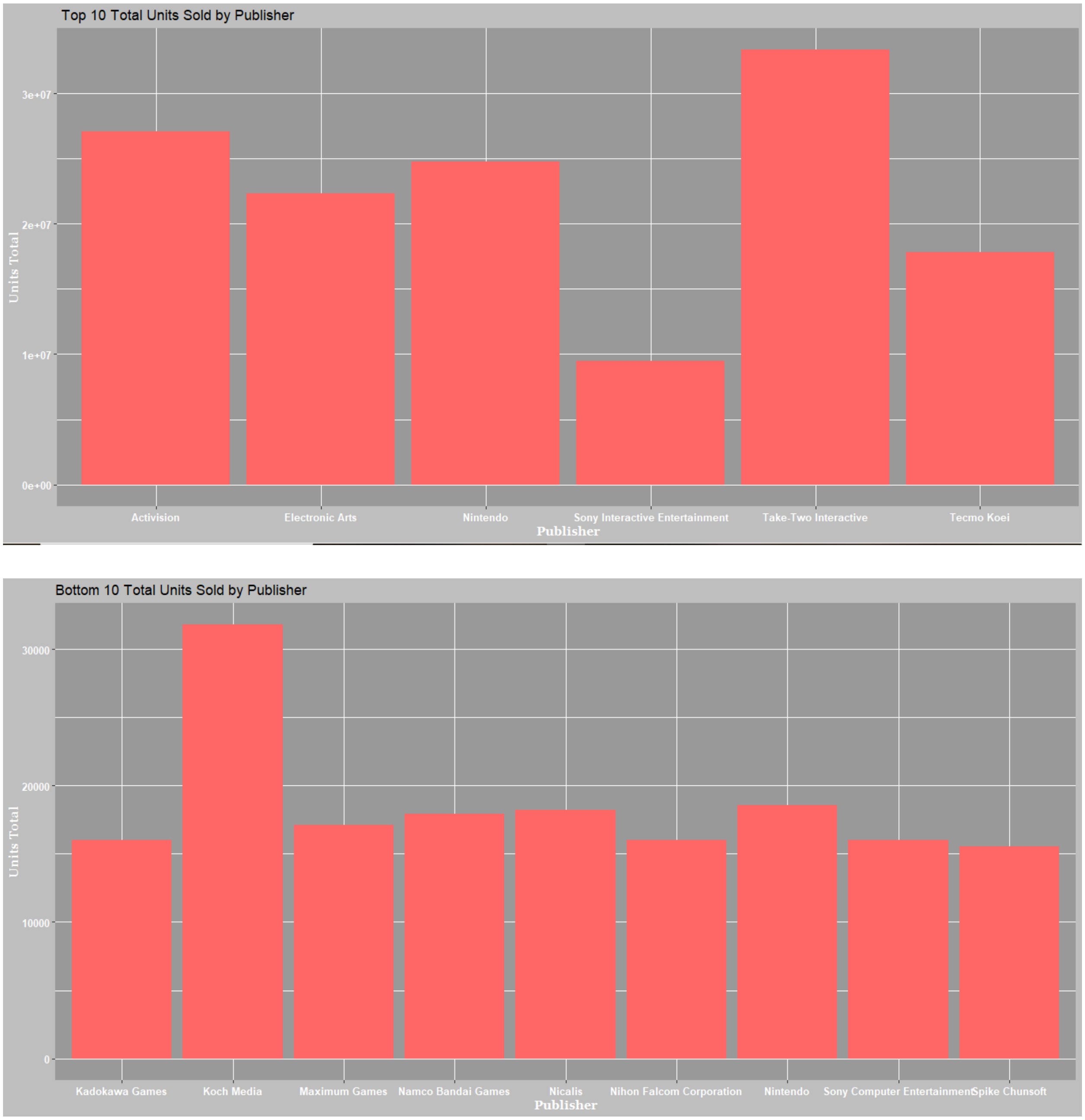
The MetaScore is a rating given to the videogame by a critic. This graph shows an upward trend between the sales and meta score. Meaning there is a relationship between the higher MetaCritic score, resulting in higher sales.

Figure 8: Total Units Sold by User Score



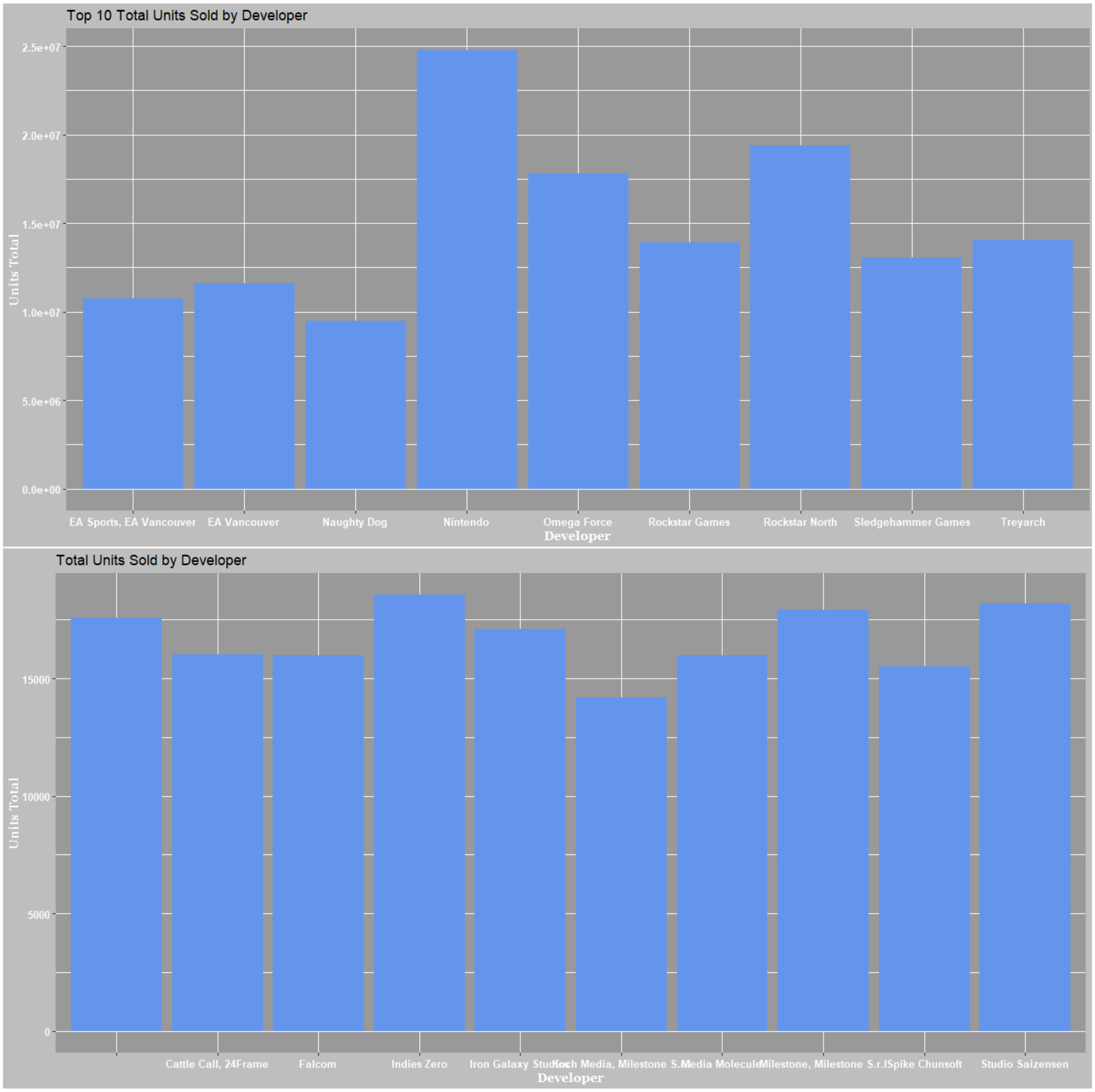
The user score is a rating provided by the users. This chart shows little relationship between the two variables. There is a large clustering toward the lower sales range. There is a slight upward trend. There are several outliers from this trend.

Figure 9: Total Units Sold by Publisher



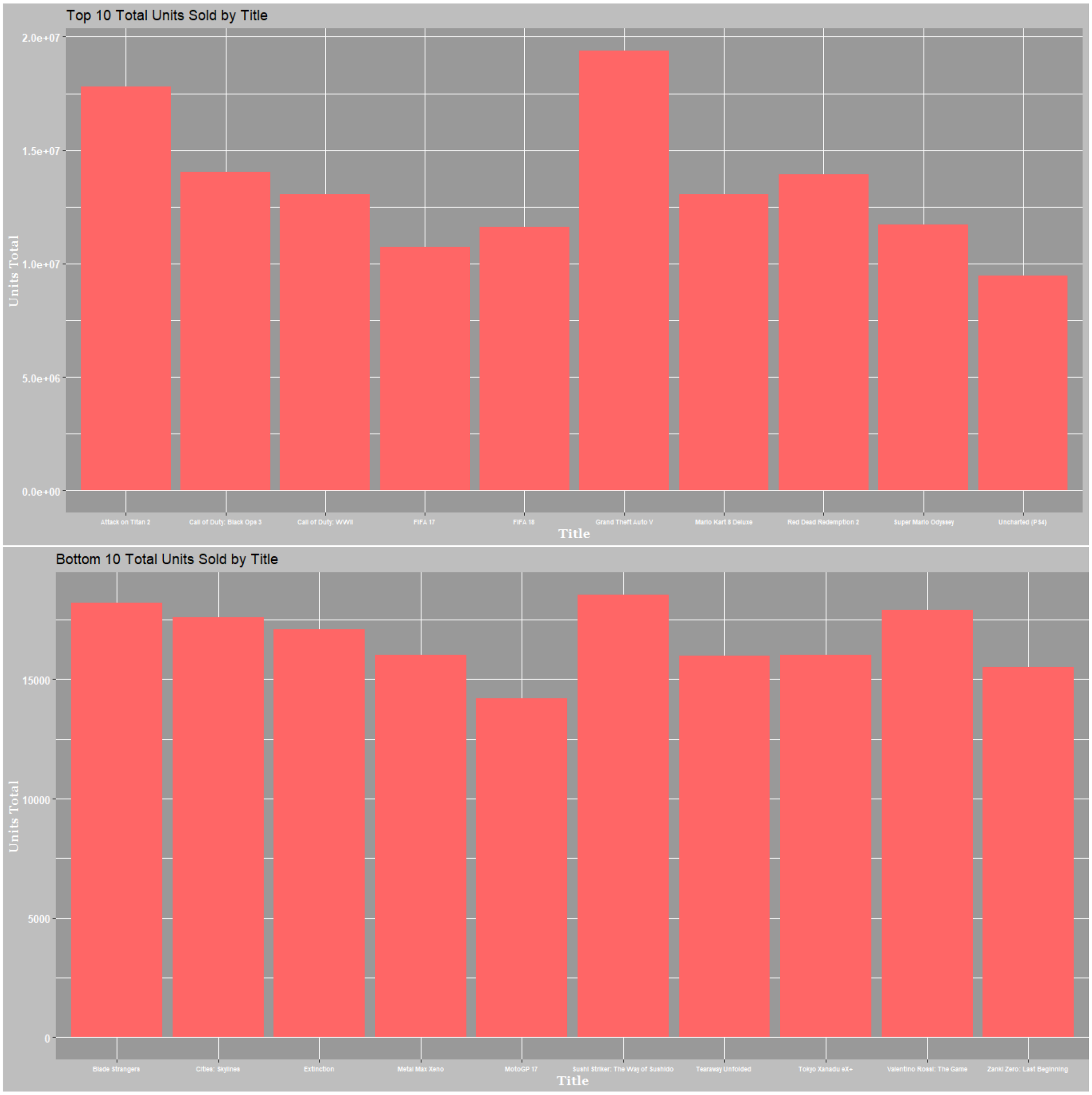
The publisher is the company that published the videogame. Due to the number of publishers in the dataset, two additional graphs were created to better express the publishers with the highest and lowest sales. The highest selling publisher is Take Two Interactive, while the lowest is Chunsoft.

Figure 10: Total Units Sold by Developer



The developer is the company that created the game. Due to the number of developers in the data set, two additional graphs were created to express the highest and lowest developer sales. The highest selling developer was Nintendo and the lowest was Koch Media, Milestone.

Figure 11: Total Units Sold by Title

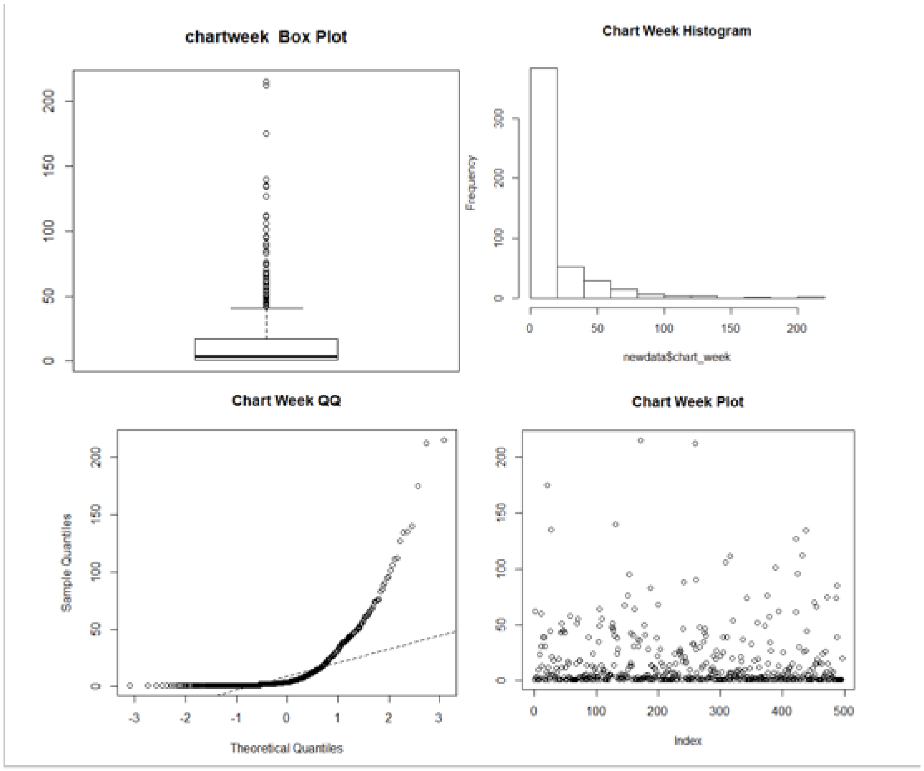


Each game has a uniquely identifying title. The dataset had a large enough number of titles that the dataset was split to express the highest and lowest selling titles. The highest was Grand Theft Auto V and the lowest was MotoGP 17.

B. Numeric Variable Analysis

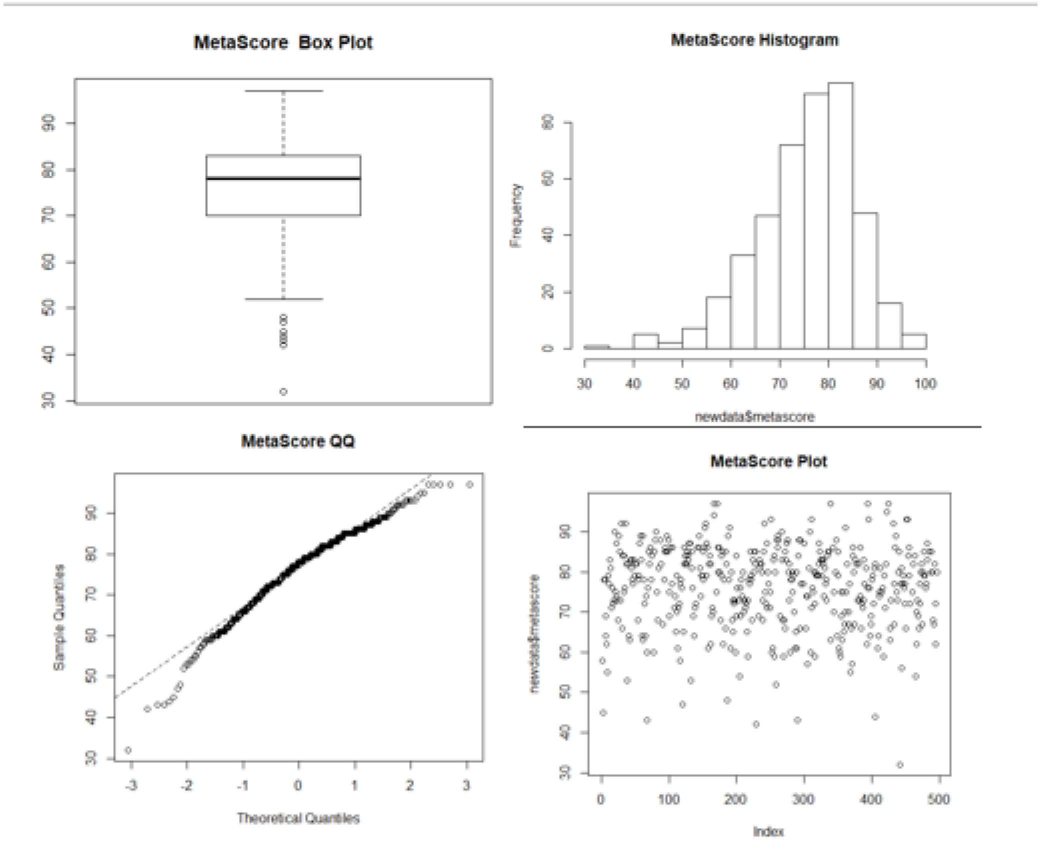
The following charts are looking at the numeric variables and some of their statistical attributes.

Figure 12: Chart Week



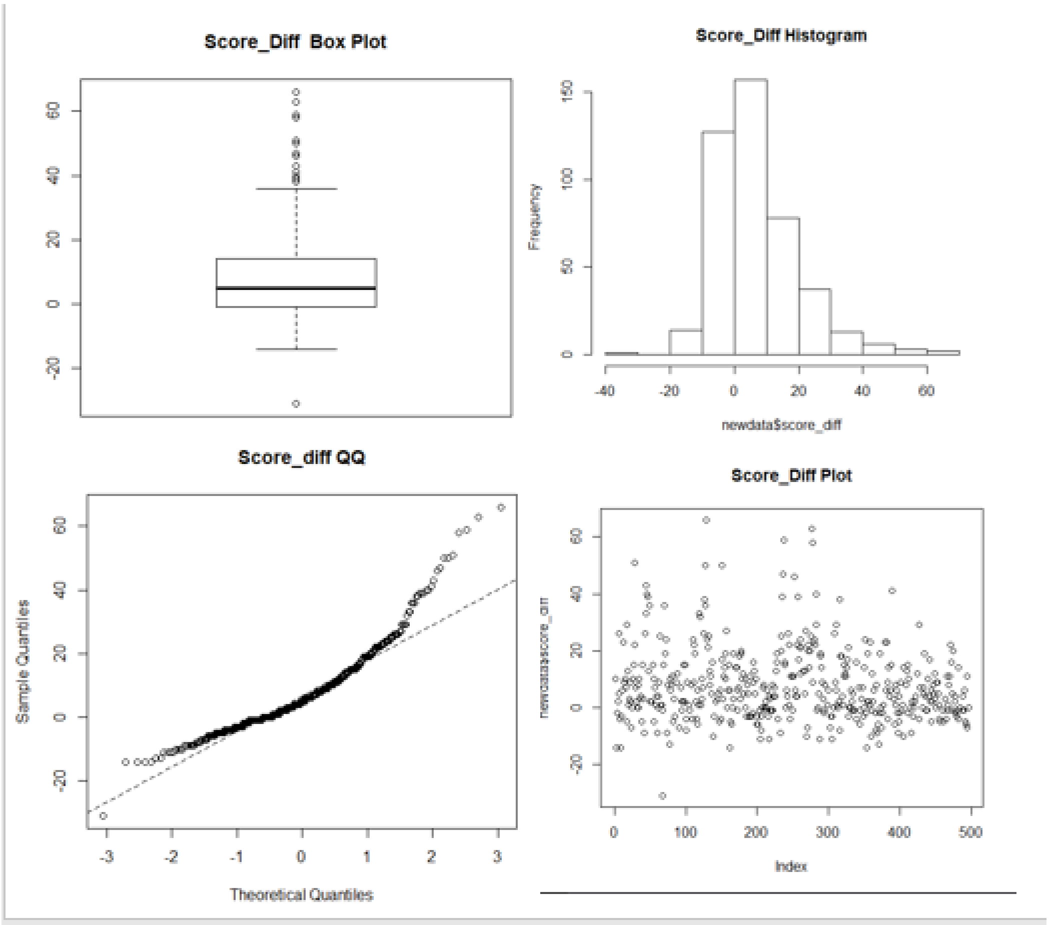
The chart week variable is looking at the number of weeks the video game has been on the charts. When looking at the box plot it is apparent that the average of the data is in the lower portion of the box. This is accompanied by a number of larger outliers. This is mirrored in the histogram, which shows the data is predominantly a positive or right skewness. The QQ Plot shows a slight deviation from the normal line in the beginning, followed by a dip in the middle and a sharp increase at the end.

Figure 13: Metascore



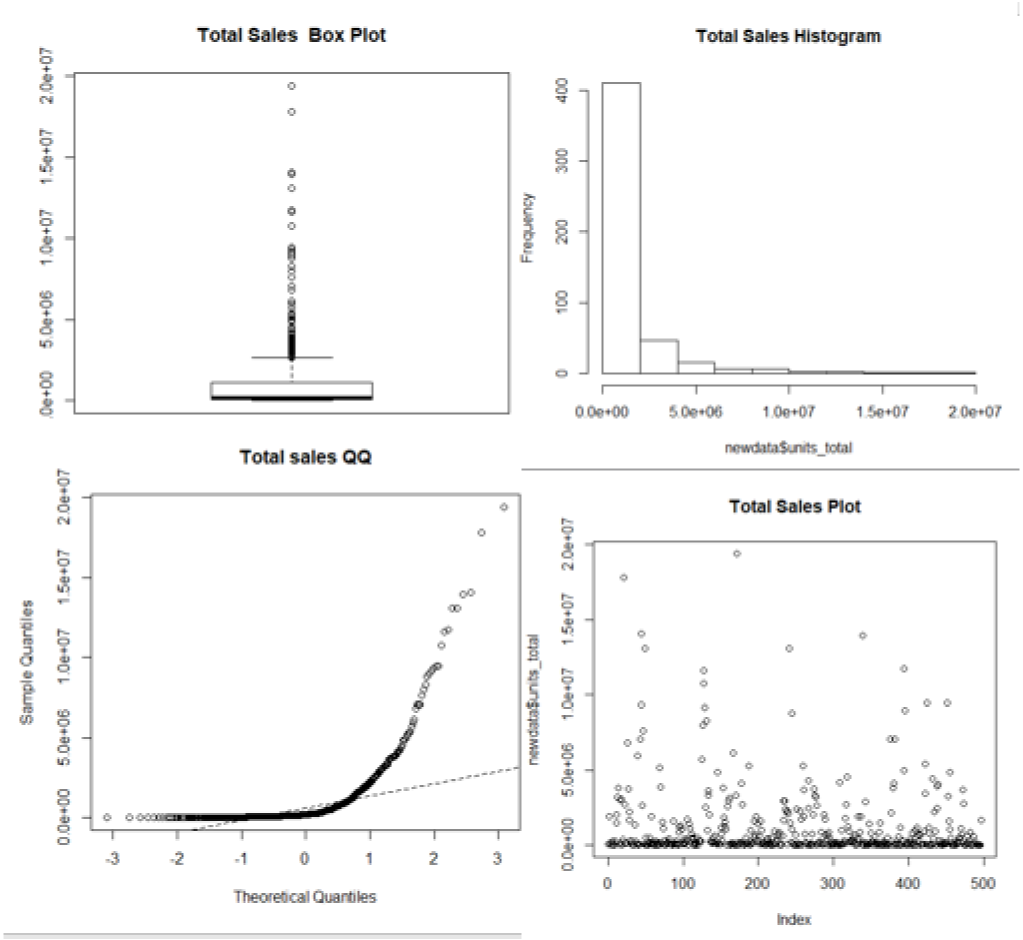
The MetaScore is the average of the ratings assigned to a game by critics. The box plot shows that the mean of the data is slightly higher than center on the box, and there are several outliers on the lower end of the data. The histogram shows a negative or left skew that is near normal, this is also shown in the QQplot, while the data follows the normal closely it does deviate at several points.

Figure 14: Score Diff



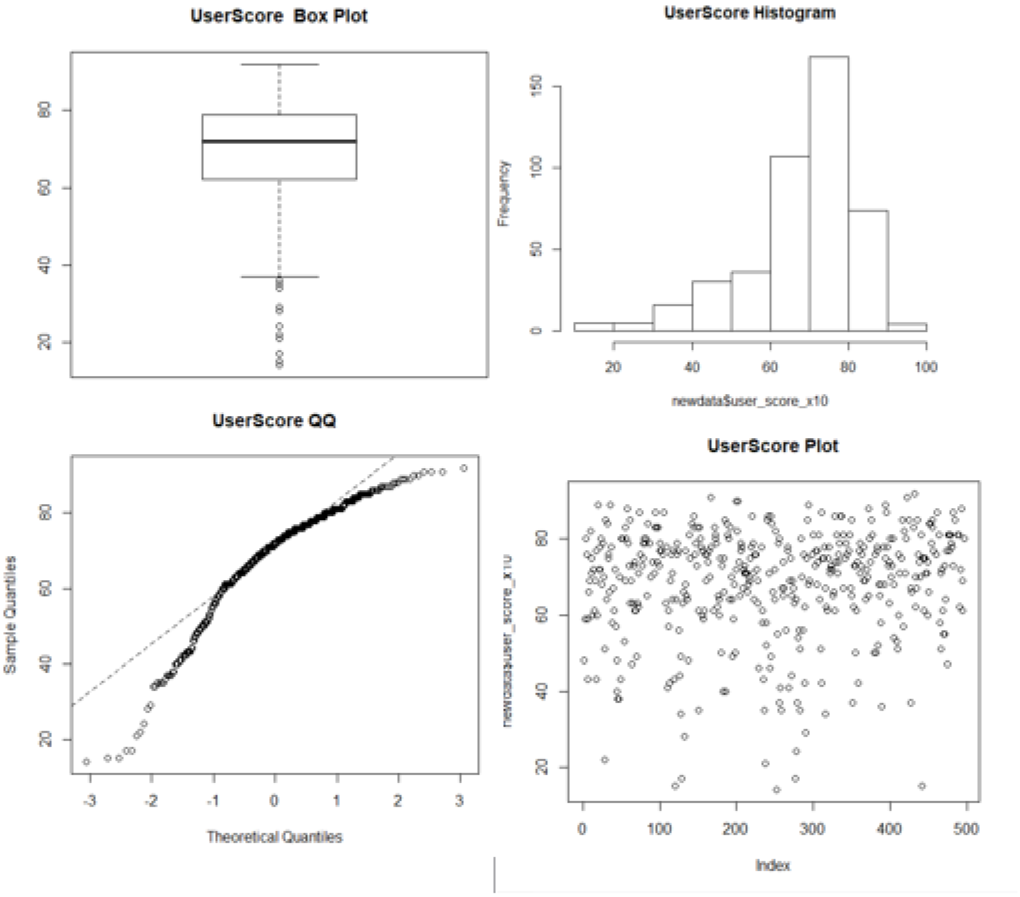
The score difference is looking at the difference between the MetaScore or critic score and the user score. As the boxplot expresses the mean is relatively center in the box and the data does have some higher outliers. The data has a relatively positive or right skew. The QQ plot shows a deviation from the normal line at the upper and lower regions of the data.

Figure 15: Total Units



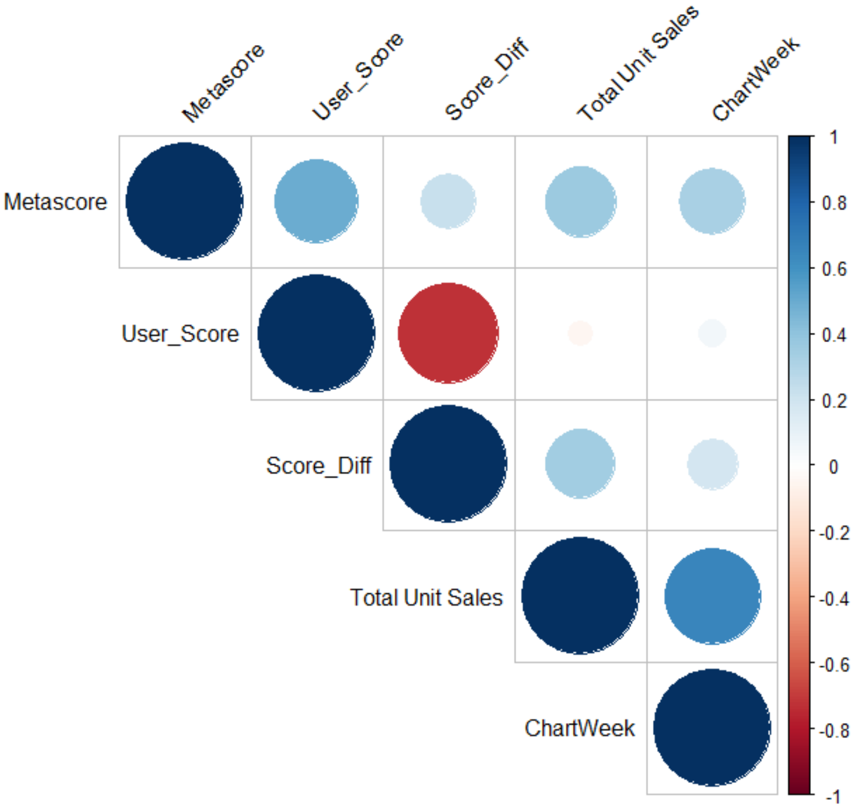
Total sales is looking at the total number of units sold per video game. The boxplot shows that the data has a fairly low mean with a large number of outliers. This is also represented in the histogram as it has a positive or right skew in the data. The QQ plot also shows deviation from the normal line, specifically on the upper end of the data, with a dip in the middle.

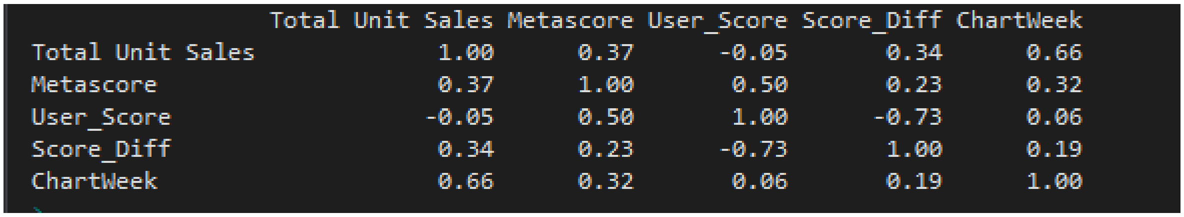
Figure 16: User Score



The user score is the rating provided by the users of the videogame on MetaCritic. The box plot shows that the mean is in the upper middle portion of the box. The data has several outliers on the lower end of the data. The histogram shows a negative or left skew. The QQ Plot shows deviation from the normal line at both the upper and lower range.

Figure 17: Correlation Plot





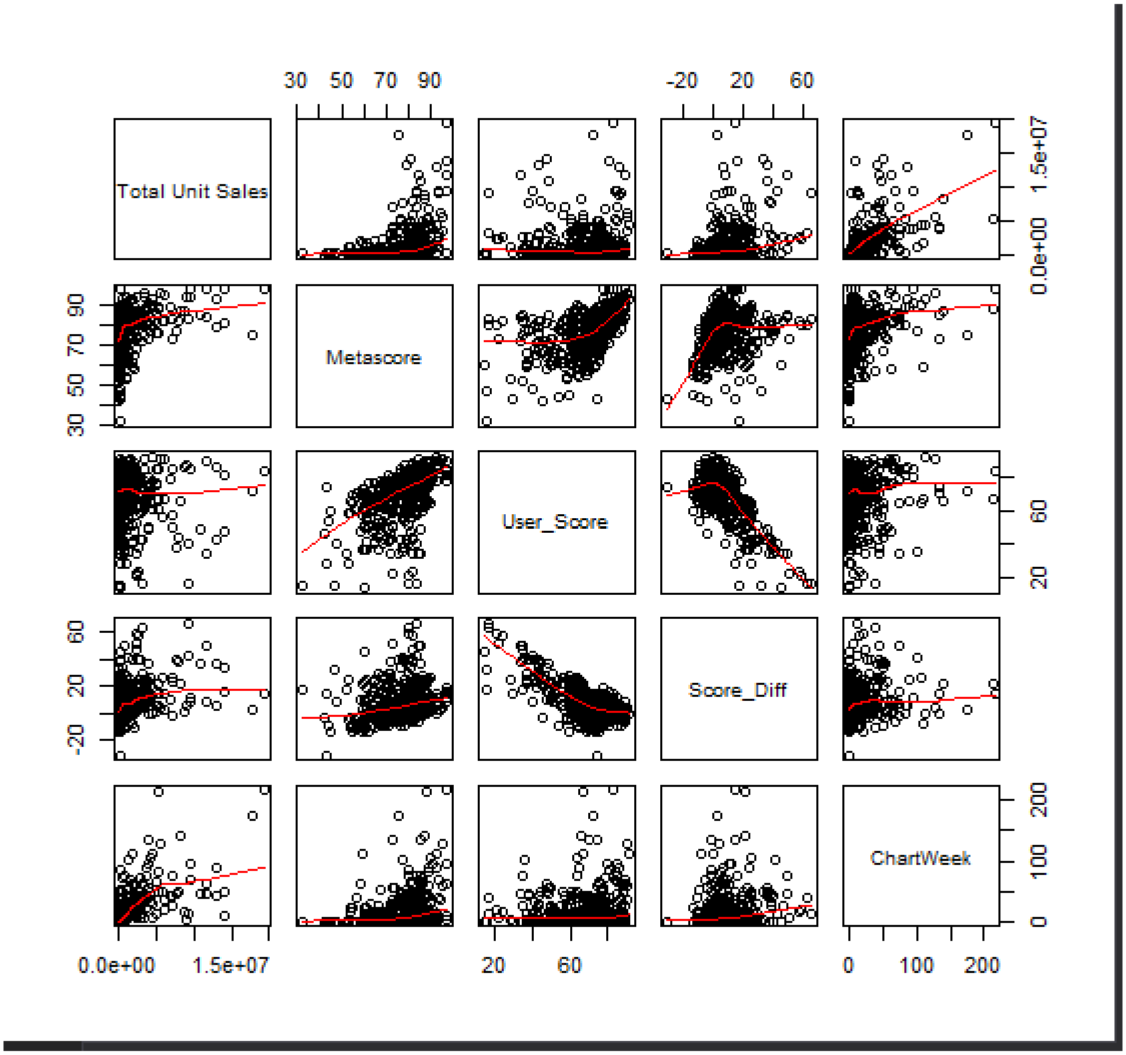
```
                 Total Unit Sales Metascore User_Score Score_Diff ChartWeek
Total Unit Sales             1.00      0.37      -0.05       0.34      0.66
Metascore                    0.37      1.00       0.50       0.23      0.32
User_Score                  -0.05      0.50       1.00      -0.73      0.06
Score_Diff                   0.34      0.23      -0.73       1.00      0.19
ChartWeek                    0.66      0.32       0.06       0.19      1.00
```

This correlation matrix and plot expresses the strength of the correlation between the numeric variables. This ultimately shows which of the variables are most influenced by the others. This expresses that the variables with the most correlation are Score_Diff and user score. This makes sense given that the user score is a component to the calculation of the Score_Diff. A theory behind why the user score has a higher correlation is through the way that the two numbers were subtracted from each other. The calculation for Score_Diff was:

Score_Diff = MetaScore - UserScore

This equation explains the strong negative correlation between Score_Diff and UserScore, and a larger UserScore leads to a smaller Score_Diff. Additionally, the critics' score is often lower than the MetaScore. Another strong correlation is between the ChartWeek and the total sales. This makes sense as the higher the chart week number is the longer the game has been on the chart means that the higher number of weeks, the sales would theoretically be higher.

Figure 18: Comparison Plot

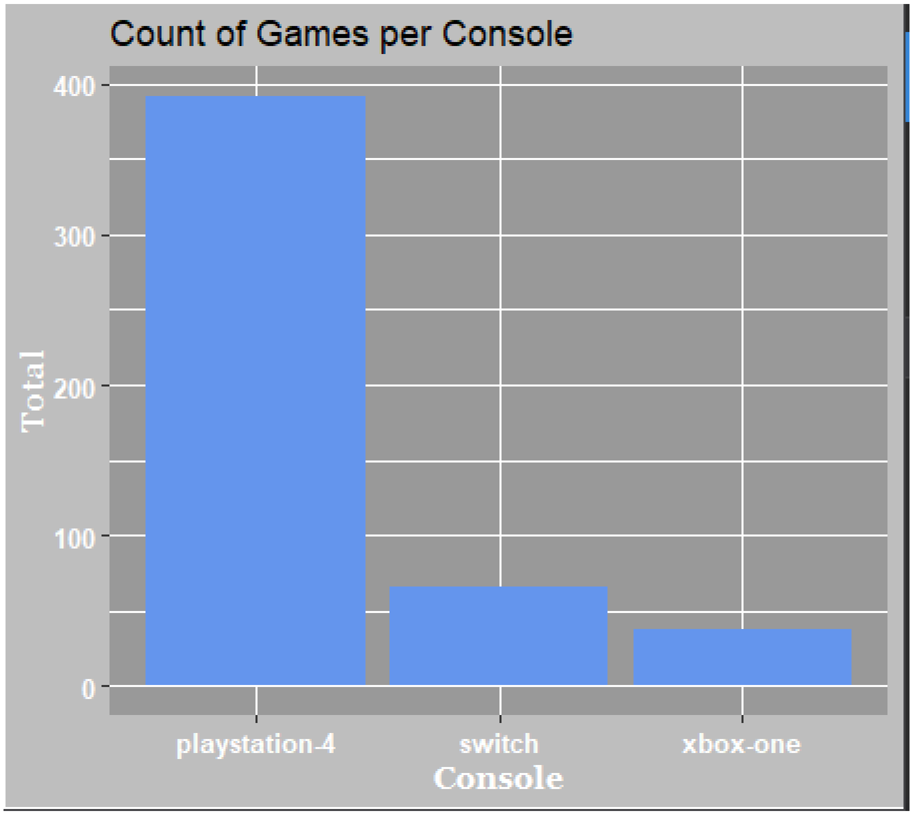


This graph shows the relationship between each variable. Some of the findings from these relationships is that there is a positive relationship between the MetaScore and the user score, chart week and total sale. There is a negative relationship between the user score and the score diff.

C. Variable Comparison

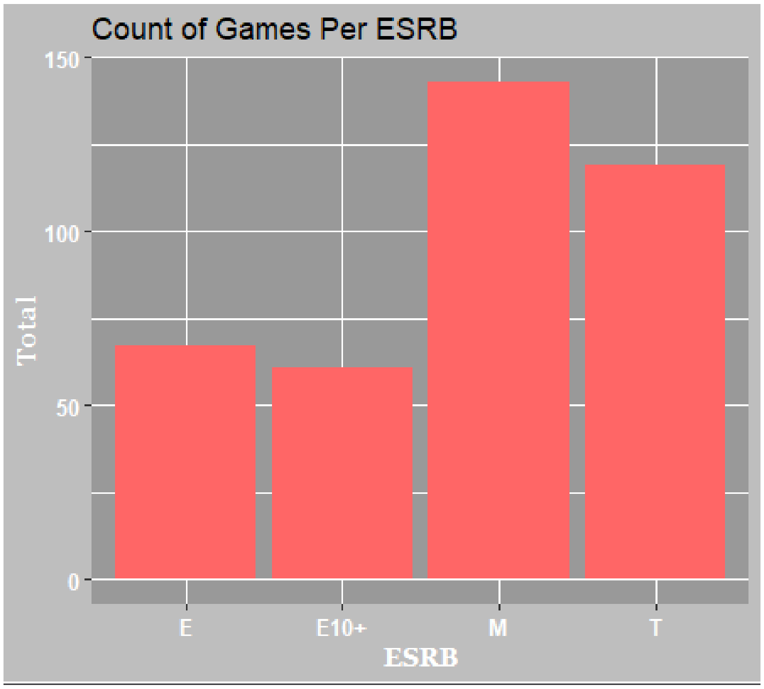
The following charts provide additional overview and comparison of that variables within the dataset.

Figure 19: Count of Games Per Console



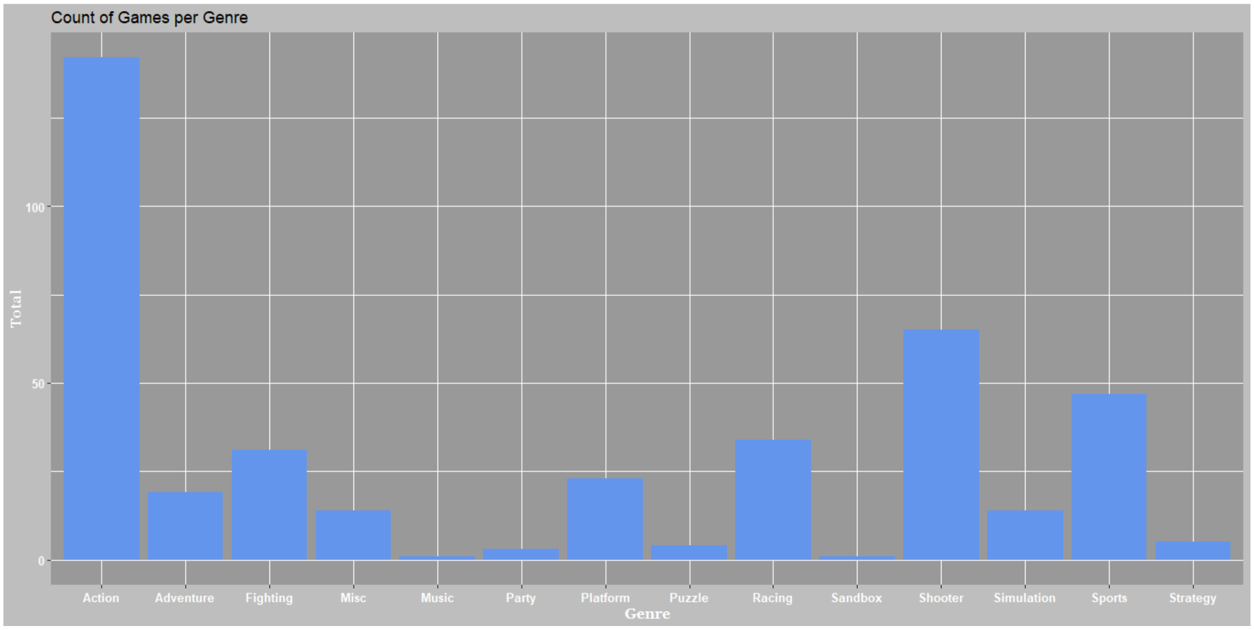
This chart is looking at the counts of games per console. This charts conclusion is very similar to Figure 3A. Expressing that the PlayStation 4 has the highest number of games on the list. This chart does once again have the disclaimer of a video game could possibly be offered on more than one console; this chart only expresses that which VGChartz has it labeled as.

Figure 20: Count of Games Per ESRB



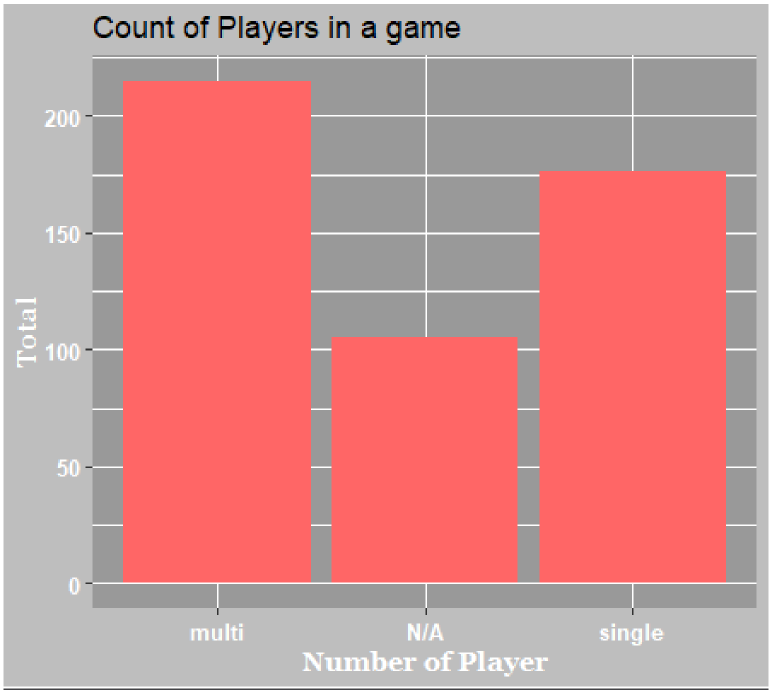
This chart is looking at the total number of games per each ESRB. This shows that the greatest number of videogames have a mature rating. This is mirrored in Figure 1A, where the highest units total sold is also those with a mature rating. Unlike Figure 1A, the second highest total units sold is E and here is the third in terms of count of games.

Figure 21: Count of Games Per Genre



This chart is looking at the count of games per genre. This is contrasted with Figure 4A, as the highest number of video games fall under the genre action and the second highest is shooter, whereas when compared to the units sold, this is switched.

Figure 22: Count of Players in a game



This chart is looking at whether video games are single or multiplayer. To do this, we took the Number of Players variable and condensed it into a two-level factor to determine whether a game is single player or multiplayer. This shows that there is a larger number of multiplayer games than there is single player.

V. DESCRIPTIVE STATISTICS AND ANALYSIS

A. Numeric Descriptive Statistics

Table 2: Numeric Descriptive Statistics

Variable	Number	Min	Max	Median	Mean	Std Dev	Skewness	Kurtosis
Position	496	1	30.00	23.00	20.98	7.519773	-0.693144	2.473856
Units Week	496	11732	1103894	26821	51191	80787.34	7.0209	73.52716
Units Total	496	14190	19392497	185366	1191430	2446959	3.72421	19.8803
Chart Week	496	1	215	4.0	15.67	27.2091	3.458406	18.98777
Metascore	496	32	97.00	78	76.11	10.23848	-0.766328	3.953898
User Score	496	1.4	9.2	7.2	6.854	1.458112	-1.24862	4.631629
User Score x10	496	14	92	72	68.54	14.58112	-1.24862	4.631629
Score Difference	496	-31	66.000	5.00	7.626	12.96471	1.326964	5.855385

B. Categorical Descriptive Statistics

Table 3: Categorical Descriptive Statistics

Variable	N	Missing	Number of Levels	Mode	Mode %	Mode Frequency
Console	5427	0	3	PlayStation 4	61.1%	3317
Genre	5427	0	16	Action	25.9%	1404
ESRB	5427	0	6	M	43.9%	2190
Number of Players	5427	0	20	No Online Multiplayer	26.9%	1339

VI. TEXT MINING AND SENTIMENT ANALYSIS

A. Overview

Our target for text mining and sentiment analysis were the summaries for all games. Whereas it is not the only text-based variable, it is the variable that contains the most text and the widest variety of text. By analyzing these summaries, we can also explore any trends within the summaries. What words are common among these summaries? Are they more positive or negative?

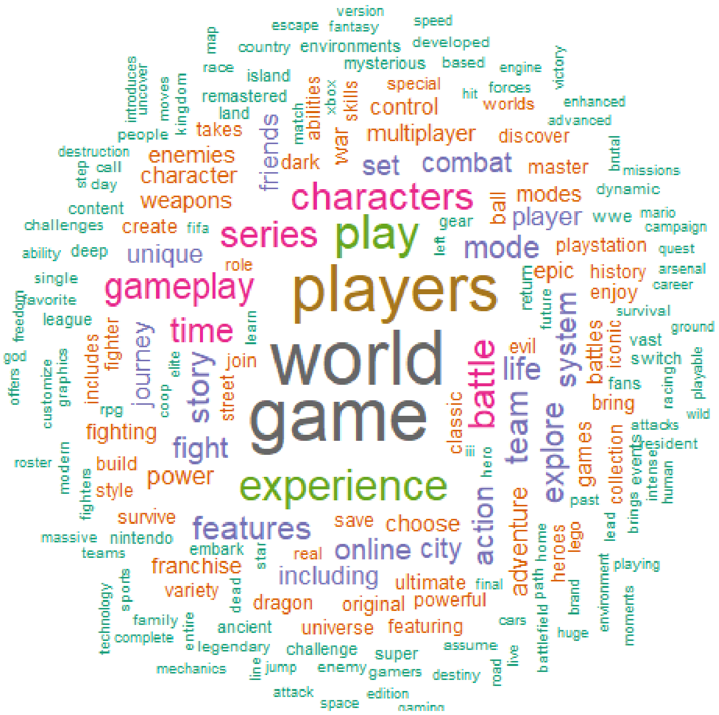
In order to do this, we started by combining all summaries into one unit of text and tokenizing it. After tokenizing and cleaning, we analyzed the top words (outside of stop words) that appeared in the summaries.

```
tidy_summaries <- tidy_summaries %>%
   count(word) %>%
   arrange(desc(n)) %>%
   mutate(proportion = (n / sum(n) * 100))

> tidy_summaries
# A tibble: 4,865 x 3
  word        n proportion
  <chr>   <int>      <dbl>
 1 game     311       1.23
 2 player   304       1.20
 3 world    289       1.14
 4 play     187      0.738
 5 charact  164      0.647
 6 battl    160      0.631
 7 experi    142      0.560
 8 fight     135      0.532
 9 featur    132      0.521
10 mode      126      0.497
```

Any missing letters are a result of the stemming process that was performed during the cleaning. We can see here that the most commonly appearing words can definitely be attributed to video games, so there are no unexpected appearances. Two words, being “battle” and “fight” support the observation that action-oriented games are generally selling better since those words appear more often. Below is a word cloud constructed from the unique summaries:

```
tdm <- TermDocumentMatrix(corp)
m <- as.matrix(tdm)
v <- sort(rowSums(m), decreasing = TRUE)
d <- data.frame(word = names(v), freq = v)

d <- d %>%
  anti_join(stop_words)

set.seed(1234)
wordcloud(words = d$word, freq = d$freq, min.freq = 1,
          max.words = 200, random.order = FALSE, rot.per = 0.35,
          colors = brewer.pal(8, "Dark2"))
```

Figure 23: Word Cloud - General



B. Sentiment Analysis

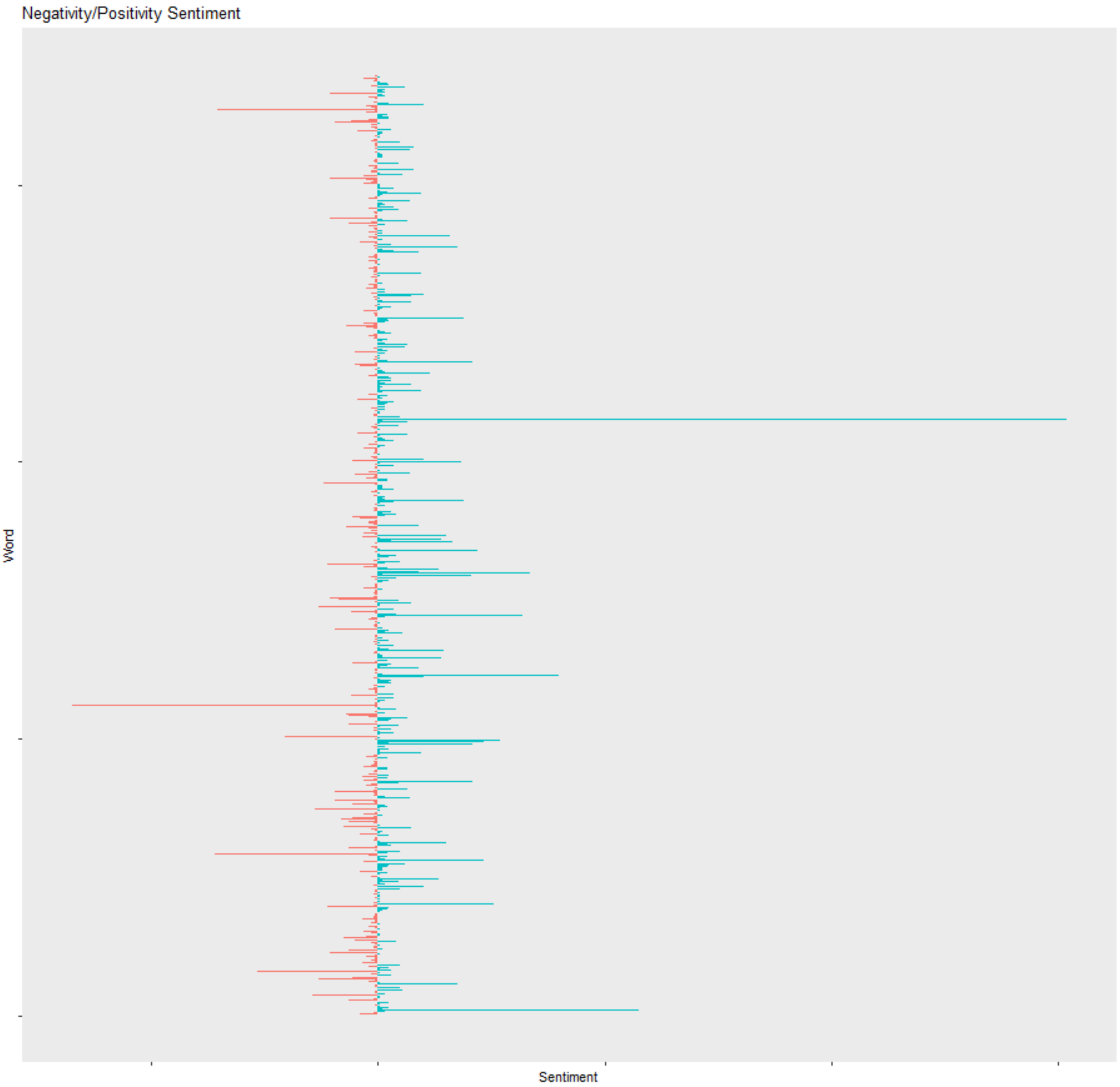
Next, we analyzed the emotional content of these summaries. We decided to compare the positivity and negativity, as well as the joy and sadness via the NRC lexicon to accomplish this. We chose these emotions as we feel that they were the most prevalent sentiment pairs that apply to video game summaries. Whereas positive and negative obviously do not mean good or bad in this context, they can be interpreted to be along the lines of lighthearted and heavy. Joy and sadness can be interpreted in a similar manner.

Positivity/Negativity

Below is an examination of the positivity and negativity contained in the unique game summaries:

```
  > head(summary_posneg)
# A tibble: 6 x 7
  word    proportion negative positive contentment linenumber sentiment
  <chr>        <dbl>    <dbl>    <dbl>       <dbl>      <int> <lgl>
1 action       0.454        0      115         115          4 TRUE
2 friend       0.316        0       80          80        246 TRUE
3 journey      0.264        0       67          67        320 TRUE
4 hero         0.252        0       64          64        289 TRUE
5 epic         0.213        0       54          54        199 TRUE
6 build        0.201        0       51          51         81 TRUE
> tail(summary_posneg)
# A tibble: 6 x 7
  word   proportion negative positive contentment linenumber sentiment
  <chr>       <dbl>    <dbl>    <dbl>       <dbl>      <int> <lgl>
1 evil        0.162       41        0         -41        202 FALSE
2 attack      0.209       53        0         -53         32 FALSE
3 war         0.280       71        0         -71        654 FALSE
4 combat      0.284       72        0         -72        117 FALSE
5 fight       0.532      135        0        -135        224 FALSE
6 player      1.20         0      304         304        431 TRUE
> sum(summary_posneg$contentment)
[1] 1184
> summary(summary_posneg$contentment)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
-135.000   -2.000    0.000    1.744    3.000  304.000
```

Figure 24: Negativity/Positivity Sentiment



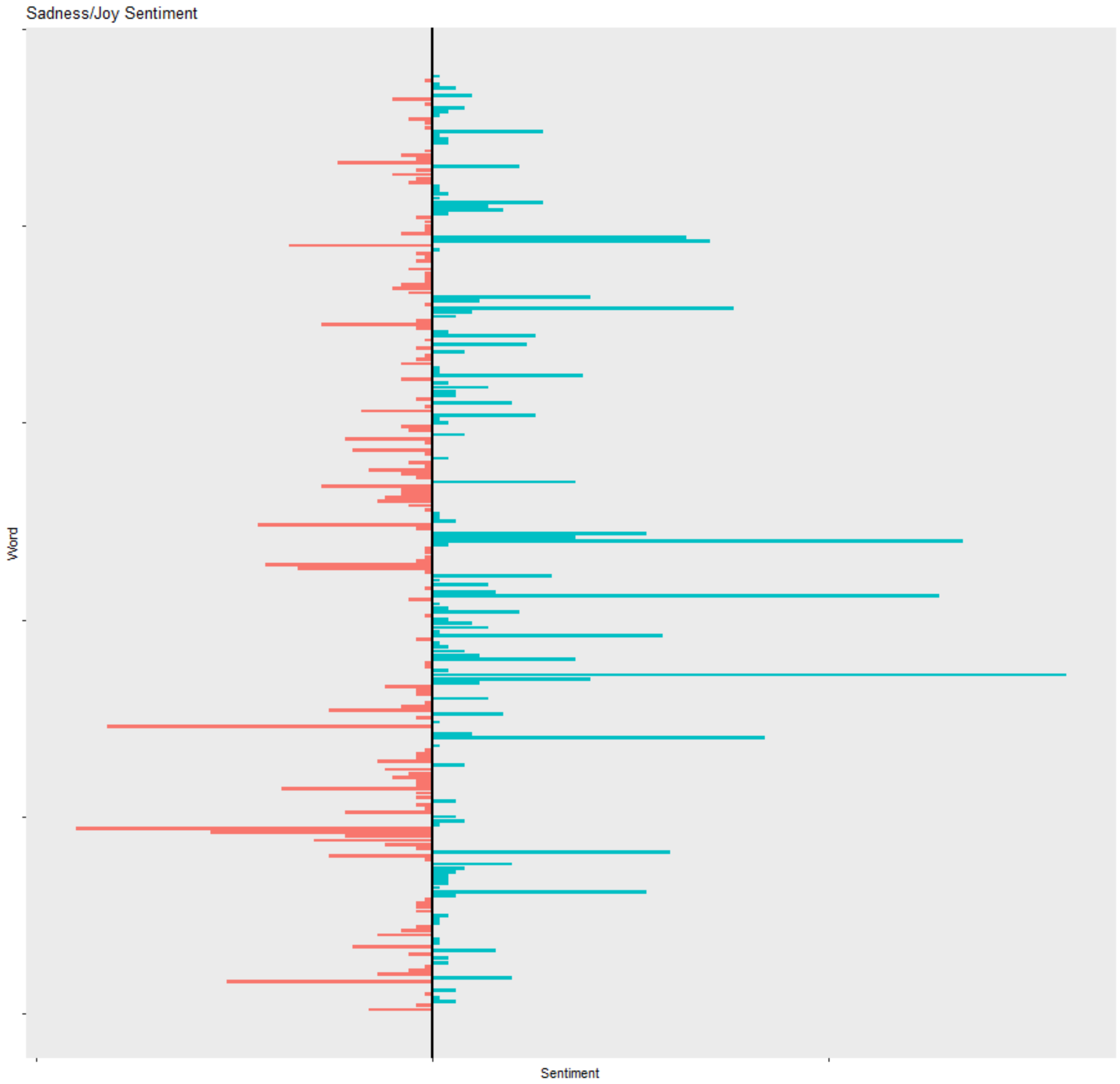
From this we can see that generally, the terms used in the summaries skew positive. However, there are a few outliers. Initially, the word "player" was being counted in a negative context (hence its appearance at the end of the dataset) but it was corrected to display correctly on the plot. The other negative outliers are correctly placed and seem to correlate with action-oriented games.

Joy/Sadness

Below is an examination of the joy and sadness contained within the unique game summaries:

```
> head(summary_joysad)
# A tibble: 6 x 7
  word    proportion   joy sadness contentment linenumber sentiment
  <chr>        <dbl> <dbl>   <dbl>       <dbl>      <int> <lgl>
1 friend       0.316    80       0          80         86 TRUE
2 journey      0.264    67       0          67        120 TRUE
3 hero         0.252    64       0          64        106 TRUE
4 enjoy        0.166    42       0          42         70 TRUE
5 save         0.150    38       0          38        179 TRUE
6 special      0.138    35       0          35        196 TRUE
> tail(summary_joysad)
# A tibble: 6 x 7
  word     proportion   joy sadness contentment linenumber sentiment
  <chr>         <dbl> <dbl>   <dbl>       <dbl>      <int> <lgl>
1 hunter       0.0828     0      21         -21        114 FALSE
2 kill         0.0868     0      22         -22        124 FALSE
3 assassin     0.103      0      26         -26          8 FALSE
4 danger       0.110      0      28         -28         46 FALSE
5 evil         0.162      0      41         -41         73 FALSE
6 dark         0.177      0      45         -45         47 FALSE
> sum(summary_joysad$contentment)
[1] 309
> summary(summary_joysad$contentment)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-45.000  -2.000  -1.000   1.298   3.000  80.000
```

Figure 25: Sadness/Joy Sentiment



Observing the joy and sadness of the summaries yields the same conclusions as the positivity and negativity. The main difference is that based on the descriptive statistics, there is a less dramatic difference between the two, as the sum of the contentment value (difference between the two sentiment values) is less. However, this could also be caused by different words being used by the NRC lexicon.

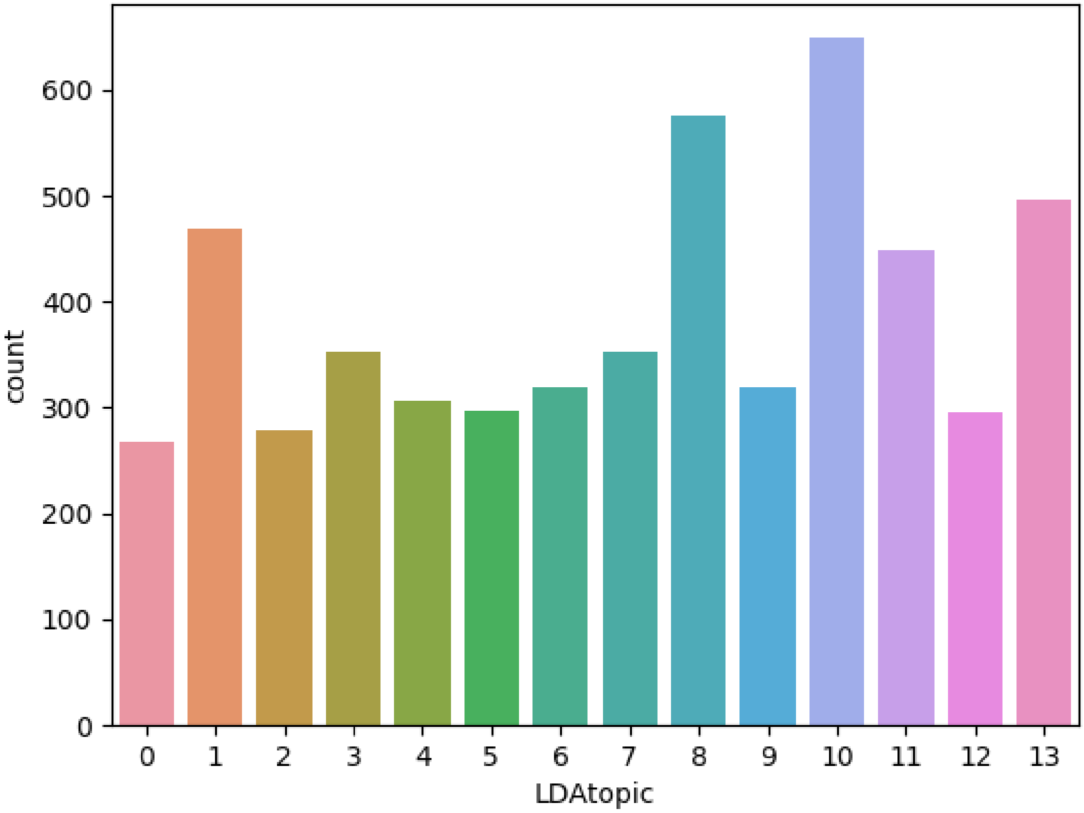
C. Topic Classification

Using Python, we wanted to be able to group games together by the content of their summaries. These topics can work in a similar manner as a genre, but the genre is determined solely on the content of the summary. The number of different topics was determined via trial and error, as the goal was to prevent as much overlap between categories. We settled on 14 categories, as this provided the least overlap while still providing some specificity. These are the results:

```
Top 10 words for topic #0:
['ups', 'nan', 'mode', 'gameplay', 'madden', 'world', 'modes', 'game', 'new', 'play']


Top 10 words for topic #1:
['gameplay', 'players', 'embark', 'experience', 'action', 'ancient', 'explore', 'new', 'game', 'world']


Top 10 words for topic #2:
['mario', 'track', 'returning', 'friends', 'race', 'version', 'play', 'mode', 'new', 'battle']


Top 10 words for topic #3:
['warfare', 'vast', 'choose', 'unique', 'pokémon', 'rainbow', 'new', 'world', 'game', 'players']


Top 10 words for topic #4:
['wild', 'wolf', 'adventure', 'new', 'need', 'll', 'link', 'world', 'way', 'enemies']


Top 10 words for topic #5:
['uncharted', 'combat', 'experience', 'franchise', 'world', 'military', 'drake', 'advanced', 'future', 'nba']


Top 10 words for topic #6:
['feel', 'destiny', 'combat', 'dynamic', 'player', 'team', 'play', 'game', 'players', 'new']


Top 10 words for topic #7:
['success', 'life', 'high', 'way', 'world', 'set', 'series', 'crew', 'street', 'time']


Top 10 words for topic #8:
['play', 'action', 'war', 'players', 'return', 'battles', 'multiplayer', 'turf', 'game', 'new']
```

```
Top 10 words for topic #9:
['control', 'players', 'battle', 'war', 'weather', 'movement', 'world', 'play', 'new', 'battlefield']


Top 10 words for topic #10:
['playing', 'build', 'unique', 'wars', 'story', 'new', 'star', 'player', 'game', 'players']


Top 10 words for topic #11:
['fifa', '19', 'duty', 'uefa', 'champions', 'league', 'experience', 'war', 'new', 'world']


Top 10 words for topic #12:
['player', 'adventure', 'music', 'events', 'horizon', 'cars', 'new', 'world', 'friends', 'drake']


Top 10 words for topic #13:
['real', 'players', 'world', 'player', 'bandicoot', 'gameplay', 'mario', 'crash', 'game', 'new']

 LDAtopic
0       267
1       469
2       278
3       352
4       307
5       298
6       319
7       353
8       575
9       320
10       649
11       448
12       295
13       497
```

Figure 26: Topic Classification: Topic Count



Categories 1, 8, 10, 11, and 13 have a spike in the number of games in those categories, implying either more games in that category or more overlap in that category. Here is our analysis on these specific categories.

Category 1: The words “action”, “embark”, and “ancient” point towards RPG’s in an ancient setting. These games may be more oriented toward cinematic, story rich experiences.

Category 8: The key word in this category is “multiplayer.” Paired with “battles” and “war,” this category likely contains multiplayer shooters, which are among the top selling games.

Category 10: Similar to category 8, this category likely contains shooters. However, these shooters probably contain single-player campaigns due to the inclusion of “story.”

Category 11: Most of category 11 more than likely contains the FIFA games, given the inclusion of “fifa”, “uefa”, “champions”, and “league.”

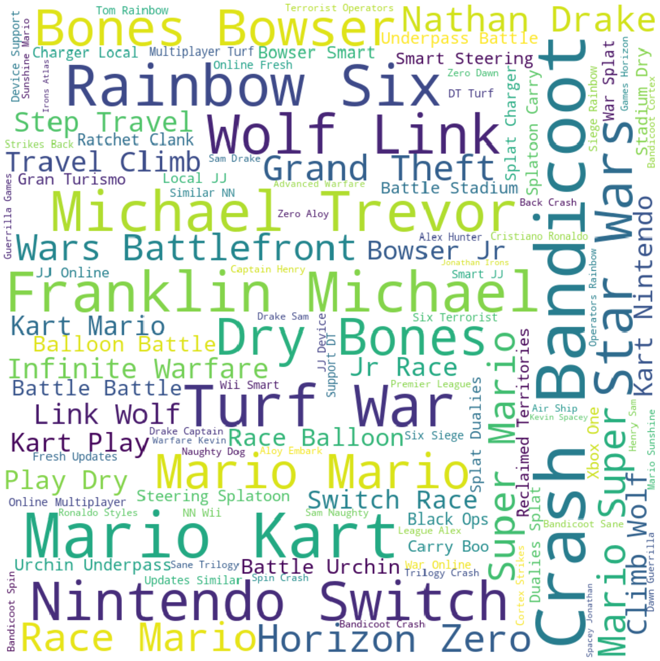
Category 13: “Mario”, “crash”, and “bandicoot” point towards single-player platforming games that are part of these classic series’.

D. Named Entity Recognition

Utilizing Python, we explored the named entities for people, organizations, and locations. With it, we were able to construct word clouds illustrating the entities that make up these categories. The first three word clouds were constructed with all game summaries.

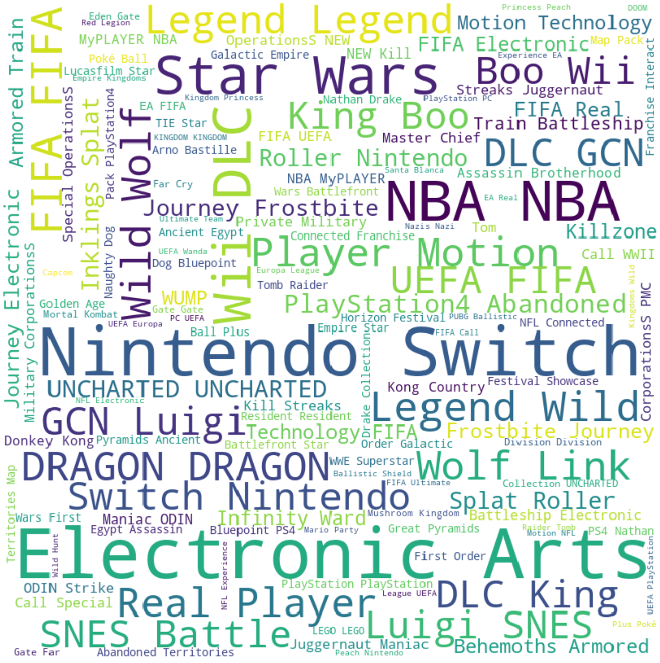
People:

Figure 27: Entities: People - Full



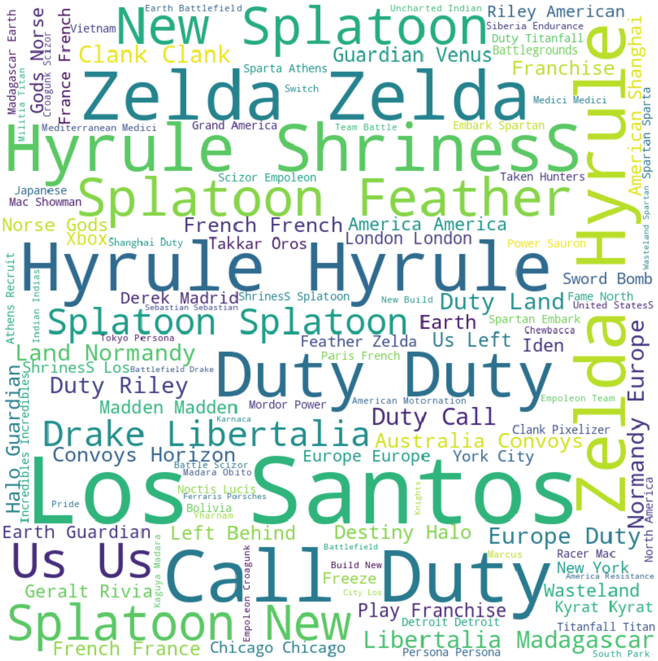
Organizations

Figure 28: Entities: Organizations - Full



Locations

Figure 29: Entities: Locations - Full

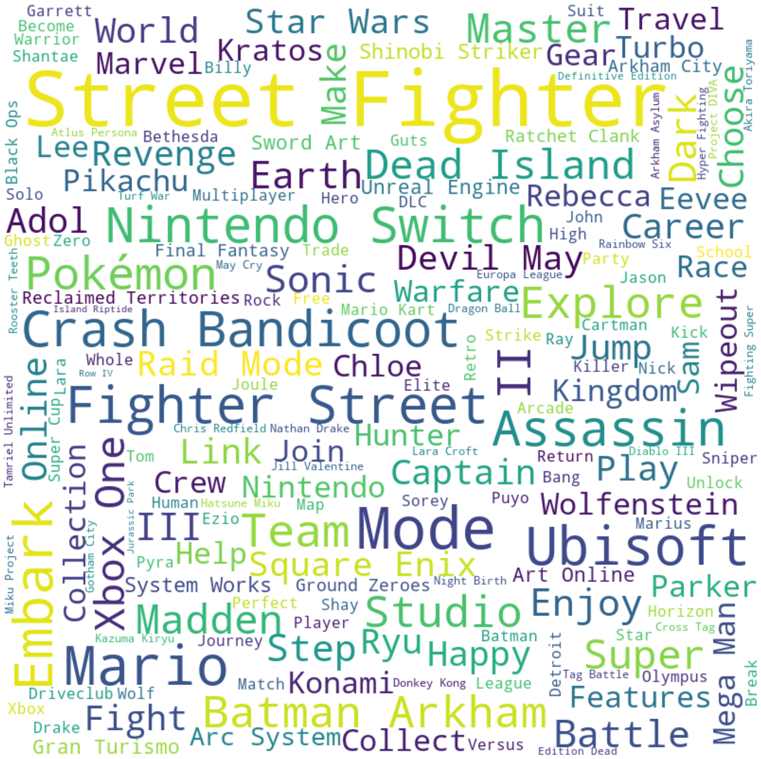


The next three word clouds are the same as the previous three, but they were constructed from only the unique summaries. Additionally, the top appearing entities have been calculated using OpenNLP in R. We were not able to do this analysis on the full list of summaries due to a lack of computing power.

People

```
entities_person %>%
+     count(entity, sort = TRUE)
# A tibble: 221 x 2
   entity            n
   <fct>          <int>
 1 Mario             6
 2 Harry Potter      5
 3 Sam               4
 4 Chris Redfield    3
 5 Diablo III        3
 6 Dragon Ball       3
 7 Jill Valentine    3
 8 Mario Kart        3
 9 Saints Row IV     3
10 Wolf              3
# ... with 211 more rows
```

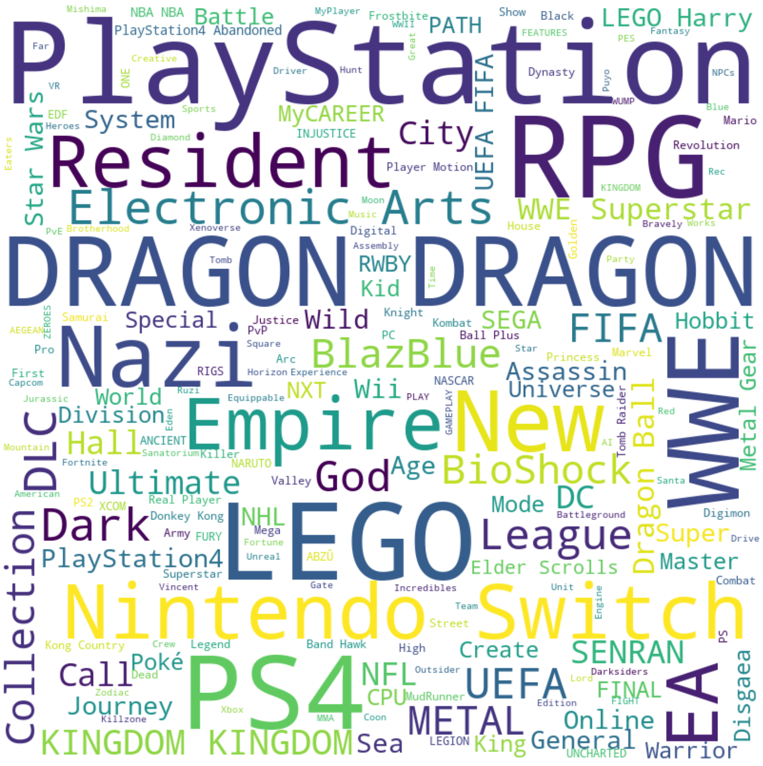
Figure 30: Entities: People - Unique



Organizations

```
entities_org %>%
+     count(entity, sort = TRUE)
# A tibble: 202 x 2
   entity                    n
   <fct>                 <int>
 1 NBA                      14
 2 Electronic Arts          13
 3 UEFA Champions League    10
 4 Arc System Works          6
 5 NFL                       5
 6 RPG                       5
 7 Division                  4
 8 French Revolution         4
 9 NHL                       4
10 DLC                       3
# ... with 192 more rows
```

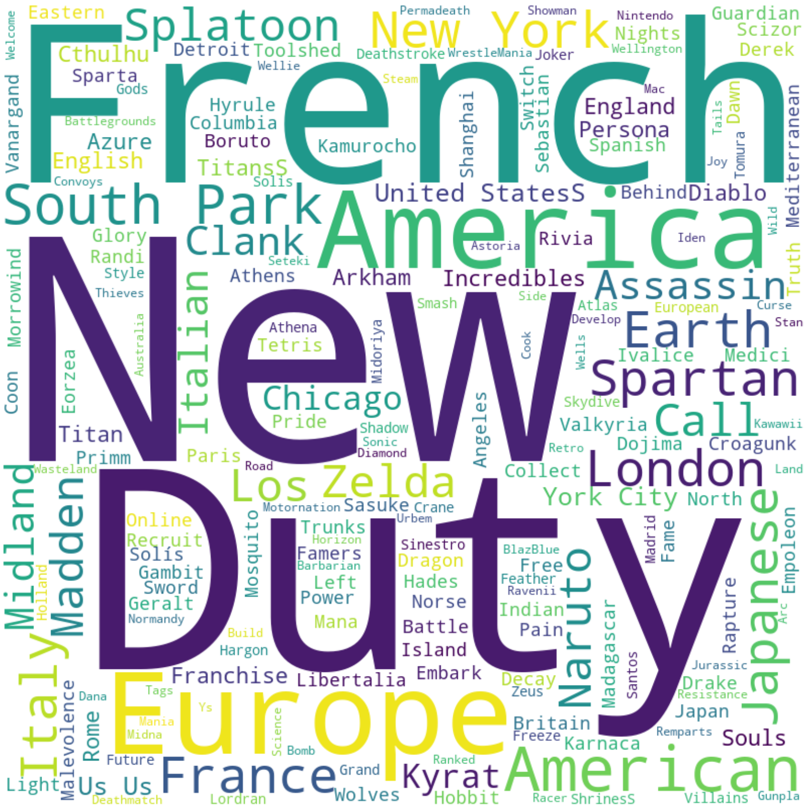
Figure 31: Entities: Organizations - Unique



Locations

```
entities_loc %>%
+     count(entity, sort = TRUE)
# A tibble: 134 x 2
   entity            n
   <fct>          <int>
 1 Earth            11
 2 Europe            8
 3 London            8
 4 South Park        8
 5 France            5
 6 Chicago           4
 7 Detroit           4
 8 Devil May         4
 9 Italy             4
10 New York City     4
# ... with 124 more rows
```

Figure 32: Entities: Locations - Unique



The differentiation between the total summaries and unique summaries is to examine the differences between the words that appear often due to the game remaining on the charts and the words that appear often due to showing up in different games' summaries. For the most part, the words appear in the cloud where they should. Of course, "New" and "Duty" probably should not appear under locations. The appearance of "Michael," "Trevor," and "Franklin" emphasize the success of Grand Theft Auto V, and the common appearances of "Mario" and its variations, as well as "Crash Bandicoot" illustrates how popular those series' were. The unique people word cloud does not have any one entity that overshadows the rest, which implies that there are no game characters that appear far more often than any others across different games. Both organization clouds show that PlayStation, Nintendo, and Electronic Arts appear far more often than other organizations, which shows the popularity of all of those organizations. An interesting appearance is "Nazi" on the unique organizations word cloud, which could presumably refer to the popularity of WWII-era games, such as *Wolfenstein*. The locations cloud made from all summaries shows the popularity of Grand Theft Auto V by including "Los Santos," and it shows the popularity of the Legend of Zelda series, as it includes many variations of "Hyrule." Lastly, any differences between the R output and the word clouds can be attributed to the different packages used to calculate the entities.

VII. APPENDIX

1. Data Sources

Vgchartz: http://www.vgchartz.com/weekly/43464/Global/
Metacritic: https://www.metacritic.com/game

2. Vgchartzscrape_final.R

```
library(RSelenium)
library(stringr)
library(dplyr)

#Set up Vgchartz scraping
vgURL <- "http://www.Vgchartz.com/weekly/43464/Global/"

rD <- rsDriver(verbose = FALSE, browser = 'firefox', port = 4568L)
remDr <- rD$client
remDr$navigate(vgURL)

#Help prevent timeout errors
remDr$setTimeout(type = "page load", milliseconds = 100000)

#Get all values for the chart URLs
weekElements <- remDr$findElements(using = "tag", "option")
weekVals <- c()
for (i in c(1:length(weekElements))) {
  weekVals <- c(weekVals, weekElements[[i]]$getElementAttribute("value"))
}
```

```
#Chart is missing on this page
weekVals[[68]] <- NULL

global_chart_df <- data.frame(matrix(ncol = 10, nrow = 0))
colnames(global_chart_df) <- c("chart_date", "position", "title", "Url_title",
"console", "developer", "genre", "units_week", "units_total", "chart_week")


for (val in weekVals) {
  URL <- paste("https://Vgchartz.com/weekly/", val, "/Global", sep = "")
  remDr$navigate(URL)
  remDr$setTimeout(type = "page load", milliseconds = 100000)

  #Get chart and save all text results
  chart <- remDr$findElement(using = "class", "chart")
  results <- str_split(chart$getElementText(), "\n")

  #Parse out text results into the different variables
  pos_list <- c()
  titleConsole_list <- c()
  genreDev_list <- c()
  unitsWeek_list <- c()

  #The first result contains the column headers
  for (i in c(2:length(results[[1]]))) {
    if (str_detect(results[[1]][i], "^[0-9]{1,2}$")) {
      pos_list <- c(pos_list, results[[1]][i])
    }
    else if (str_detect(results[[1]][i], "\\)$")) {
      titleConsole_list <- c(titleConsole_list, results[[1]][i])
    }
    else if (str_detect(results[[1]][i], "[a-z]$")) {
      genreDev_list <- c(genreDev_list, results[[1]][i])
    }
    else if (str_detect(results[[1]][i], "\\s[0-9]*$")) {
      unitsWeek_list <- c(unitsWeek_list, results[[1]][i])
    }
  }

  #Split titleConsole into title and console, genreDev into genre and developer, and
unitsWeek into units weekly, units total, and week
```

```
title_list <- c()
Url_title_list <- c()
console_list <- c()
dev_list <- c()
genre_list <- c()
unitsWeekly_list <- c()
unitsTotal_list <- c()
week_list <- c()

for (i in c(1:length(pos_list))) {
   title_split <- str_split(titleConsole_list[i], " ")
   len <- length(title_split[[1]]) - 1
   title <- paste(title_split[[1]][1:len], collapse = " ")
   console <- title_split[[1]][len + 1]

   #Certain titles are different between Vgchartz and Metacritic
   if (title == "Super Smash Bros. (2018)") {
      title <- "Super Smash Bros. Ultimate"
   }
   else if (str_detect(title, "IIII$")) {
      title <- "Call of Duty: Black Ops 4"
   }
   else if (title == "Spider-Man (PS4)") {
      title <- "Marvel's Spider-Man"
   }
   else if (title == "Crash Bandicoot N.Sane Trilogy") {
      title <- "Crash Bandicoot N Sane Trilogy"
   }
   else if (title == "Minecraft") {
      if (console == "(NS)") {
         title = "Minecraft: Switch Edition"
      }
      else if (console == "(PS4)") {
         title = "Minecraft: PlayStation 4 Edition"
      }
      else if (console == "(XOne)") {
         title = "Minecraft: Xbox One Edition"
      }
      else if (console == "(X360)") {
         title = "Minecraft: Xbox 360 Edition"
      }
```

```
    else if (console == "(PSV)") {
        title = "Minecraft: PlayStation Vita Edition"
    }
    else if (console == "(WiiU)") {
        title = "Minecraft: Wii U Edition"
    }
}

title_list <- c(title_list, title)

#Alter the title to use in a Metacritic URL
Url_title <- tolower(title)
Url_title <- trimws(sub("\\(ps4\\)", "", Url_title))
Url_title <- gsub(" ", "-", Url_title)
Url_title <- gsub("[\\(|\\)]", "", Url_title)
Url_title <- gsub("[\\.|\'|\\,|\\:]", "", Url_title)
Url_title_list <- c(Url_title_list, Url_title)

#Change console to how it appears in a Metacritic URL
if (console == "(NS)") {
    console = "switch"
}
else if (console == "(PS4)") {
    console = "playstation-4"
}
else if (console == "(XOne)") {
    console = "xbox-one"
}
else if (console == "(3DS)") {
    console = "3ds"
}
else if (console == "(PC)") {
    console = "pc"
}
else if (console == "(Wii)") {
    console = "wii"
}
else if (console == "(WiiU)") {
    console = "wii-u"
}
else if (console == "(X360)") {
```

```
            console = "xbox-360"
        }
        else if (console == "(PSV)") {
            console = "playstation-vita"
        }
        else if (console == "(PS3)") {
            console = "playstation-3"
        }
        else if (console == "(PSP)") {
            console = "psp"
        }
        else if (console == "(DS)") {
            console = "ds"
        }
        else if (console == NA) {
            console
        }

        console_list <- c(console_list, console)

        #Assemble other lists
        dev_list <- c(dev_list, str_split(genreDev_list[i], ", ")[[1]][1])
        genre_list <- c(genre_list, str_split(genreDev_list[i], ", ")[[1]][2])
        unitsWeekly_list <- c(unitsWeekly_list, str_split(unitsWeek_list[i], " ")[[1]][1])
        unitsTotal_list <- c(unitsTotal_list, str_split(unitsWeek_list[i], " ")[[1]][2])
        week_list <- c(week_list, str_split(unitsWeek_list[i], " ")[[1]][3])
    }

    #Get text for the week of the chart
    weekSelector = paste("option[value='", val,"']", sep = "")
    chartWeek <- remDr$findElement(using = "css", weekSelector)$getElementText()

    #Assemble individual rows based on the previous lists and add them to the global
data frame
    for (i in c(1:75)) {
        newrow <- data.frame("week_posted" = chartWeek[[1]], "position" = pos_list[i],
"title" = title_list[i], "Url_title" = Url_title_list[i], "console" = console_list[i],
"developer" = dev_list[i],
                "genre" = genre_list[i], "units_week" = unitsWeekly_list[i], "units_total" =
unitsTotal_list[i],
                "chart_week" = week_list[i], stringsAsFactors = FALSE)
        global_chart_df <- rbind(global_chart_df, newrow)
    }

    #Print the week of the chart for reference in the console
    print(paste("Added ", chartWeek, " chart."))
}

#Narrow the full data frame to the top 30 of each chart to keep only rows with all data
global_top30_df <- global_chart_df[!(global_chart_df$units_week == "Pro"),]

#The script defaulted to 75 rows per chart, even though some of the later charts don't
have 75 rows. This deletes the resulting NA rows
global_top30_clean_df <- na.omit(global_top30_df)

#Export data
setwd("C:\\Users\\Aaron\\Google Drive\\School Stuff\\Fall 2019\\Data Science
Programming I\\DSP1 Semester Project\\Data")
write.csv(global_chart_df, "global_chart_full.csv")
write.csv(global_top30_df, "global_top30.csv")
write.csv(global_top30_clean_df, "top30_noNA.csv")

#Close the session
remDr$close()
rD$server$stop()
```

3. Metacritic_final.R

```
library(RSelenium)
library(stringr)
library(plyr)
library(dplyr)
library(data.table)

setwd("C:\\Users\\Aaron\\Google Drive\\School Stuff\\Fall 2019\\Data Science Programming I\\DSP1
Semester Project\\Data")

#Import Vgchartz data
vgData <- read.csv("top30_noNA.csv")

#21963 rows

current_gen <- c('switch', 'playstation-4', 'xbox-one')
currrent_gen_data <- vgData[(vgData$console %in% current_gen),]
```

```
write.csv(currrent_gen_data, "top30_currentGen.csv")

#5476 rows

#Split between multiple R environments to speed up the process.
#In 5 groups of 1095

index1 <- c(1:1095)
index2 <- c(1096:2190)
index3 <- c(2191:3285)
index4 <- c(3286:4380)
index5 <- c(4381:5476)

rD <- rsDriver(verbose = FALSE, browser = 'firefox', port = 4568L)
remDr <- rD$client

#Prevent timeout errors
remDr$setTimeout(type = "page load", milliseconds = 100000)

meta_df <- data.frame(matrix(ncol = 18, nrow = 0))
colnames(meta_df) <- c("week_posted", "position", "title", "Url_title",
        "console", "developer", "genre", "units_week", "units_total", "chart_week",
        "numPlayers", "esrb", "metaGenre", "metaDevs", "release_date", "metascore",
        "user_score", "summary")

#Change index depending on environment and run simultaneously
for (i in index3) {
   metaURL <- paste("https://www.Metacritic.com/game/", as.character(currrent_gen_data[i,]$console),
"/", currrent_gen_data[i,]$Url_title, sep = "")

   tryCatch({
      remDr$navigate(metaURL)
   }, warning = function(w) {
      print(paste("Warning: ", w))
   }, error = function(e) {
      print(paste("Error: ", e))
   }, finally = {
      print(paste("Started scan for ", currrent_gen_data[i,]$title, " on Metacritic."))
   })

   metascore <- tryCatch({
      remDr$findElement(using = "css", "span[itemprop='ratingValue']")$getElementText()[[1]]
   }, warning = function(w) {
      print(paste("Warning: ", w))
      metascore <- 999
   }, error = function(e) {
      print(paste("Error: ", e))
```

```
        metascore <- 999
    }, finally = {
        print(paste("Scanned MetaScore for ", currrent_gen_data[i,]$title, " on Metacritic."))
    })

    userscore <- tryCatch({
        remDr$findElement(using = "css", "div[class='metascore_w user large game positive'],
div[class='metascore_w user large game negative'], div[class='metascore_w user large game
mixed']")$getElementText()[[1]]
    }, warning = function(w) {
        print(paste("Warning: ", w))
        userscore <- 999
    }, error = function(e) {
        print(paste("Error: ", e))
        userscore <- 999
    }, finally = {
        print(paste("Scanned user score for ", currrent_gen_data[i,]$title, " on Metacritic."))
    })

    numPlayers <- tryCatch({
        remDr$findElement(using = "css", "li.summary_detail.product_players > span:nth-
child(2)")$getElementText()[[1]]
    }, warning = function(w) {
        print(paste("Warning: ", w))
        numPlayers <- "WARNING"
    }, error = function(e) {
        print(paste("Error: ", e))
        numPlayers <- "ERROR"
    }, finally = {
        print(paste("Scanned # of players for ", currrent_gen_data[i,]$title, " on Metacritic."))
    })

    releaseDate <- tryCatch({
        remDr$findElement(using = "css", ".release_data > span:nth-child(2)")$getElementText()[[1]]
    }, warning = function(w) {
        print(paste("Warning: ", w))
        releaseDate <- "WARNING"
    }, error = function(e) {
        print(paste("Error: ", e))
        releaseDate <- "ERROR"
    }, finally = {
        print(paste("Scanned release date for ", currrent_gen_data[i,]$title, " on Metacritic."))
    })

    esrb_rating <- tryCatch({
        remDr$findElement(using = "css", "li.summary_detail:nth-child(5) > span:nth-
child(2)")$getElementText()[[1]]
```

```
  }, warning = function(w) {
    print(paste("Warning: ", w))
    esrb_rating <- "WARNING"
  }, error = function(e) {
    print(paste("Error: ", e))
    esrb_rating <- "ERROR"
  }, finally = {
    print(paste("Scanned esrb rating for ", currrent_gen_data[i,]$title, " on Metacritic."))
  })

  #Check for existance of expanding button. If it exists, click and get the summary. If not, get the summary
  #Embedded try/catch should execute if the expand button is not present, throwing a NoSuchElement Exception
  if (length(remDr$findElements(using = "css", ".product_summary > span:nth-child(2) > span:nth-child(1) > span:nth-child(4)")) > 0) {
    remDr$findElement(using = "css", ".product_summary > span:nth-child(2) > span:nth-child(1) > span:nth-child(4)")$clickElement()
    tryCatch({
      summary <- remDr$findElement(using = "css", ".inline_expanded > span:nth-child(2)")$getElementText()[[1]]
    }, warning = function(w) {
      print(paste("Warning: ", w))
      summary <- "Warning"
    }, error = function(e) {
      print(paste("Error: ", e))
      summary <- "ERROR"
    })

    print(paste("Scanned summary for ", currrent_gen_data[i,]$title, " on Metacritic."))
  }

  else if (length(remDr$findElements(using = "css", ".summary_detail.product_summary > span:nth-child(2)")) > 0) {
    tryCatch({
      summary <- remDr$findElement(using = "css", ".summary_detail.product_summary > span:nth-child(2)")$getElementText()[[1]]
    }, warning = function(w) {
      print(paste("Warning: ", w))
      summary <- "Warning"
    }, error = function(e) {
      print(paste("Error: ", e))
      summary <- "ERROR"
    }, finally = {
      print(paste("Scanned summary for ", currrent_gen_data[i,]$title, " on Metacritic."))
    })
  }
```

```
else {
  summary <- "No Summary Found"
  print("No Summary Found")
}

metagenre <- tryCatch({
  remDr$findElement(using = "css", ".product_genre")$getElementText()[[1]]
}, warning = function(w) {
  print(paste("Warning: ", w))
  metagenre <- "WARNING"
}, error = function(e) {
  print(paste("Error: ", e))
  metagenre <- "ERROR"
}, finally = {
  print(paste("Scanned metagenre for ", currrent_gen_data[i,]$title, " on Metacritic."))
})

metadev <- tryCatch({
  remDr$findElement(using = "css", ".developer")$getElementText()[[1]]
}, warning = function(w) {
  print(paste("Warning: ", w))
  metadev <- "WARNING"
}, error = function(e) {
  print(paste("Error: ", e))
  metadev <- "ERROR"
}, finally = {
  print(paste("Scanned metadev for ", currrent_gen_data[i,]$title, " on Metacritic."))
})

newrow <- data.frame("week_posted" = currrent_gen_data[i,]$week_posted,
             "position" = currrent_gen_data[i,]$position,
             "title" = currrent_gen_data[i,]$title,
             "Url_title" = currrent_gen_data[i,]$Url_title,
             "console" = currrent_gen_data[i,]$console,
             "developer" = currrent_gen_data[i,]$developer,
             "genre" = currrent_gen_data[i,]$genre,
             "units_week" = currrent_gen_data[i,]$units_week,
             "units_total" = currrent_gen_data[i,]$units_total,
             "chart_week" = currrent_gen_data[i,]$chart_week,
             "numPlayers" = numPlayers,
             "esrb" = esrb_rating,
             "metaGenre" = metagenre,
             "metaDevs" = metadev,
             "release_date" = releaseDate,
             "metascore" = metascore,
             "user_score" = userscore,
                    "summary" = summary)

    meta_df <- rbind(meta_df, newrow)
}


#Change name depending on index, will end with 5 different csv's
write.csv(meta_df, "vgMetaFullI3.csv")


#Combine the five individual indices
I1_df <- read.csv("vgMetaFullI1.csv")
I2_df <- read.csv("vgMetaFullI2.csv")
I3_df <- read.csv("vgMetaFullI3.csv")
I4_df <- read.csv("vgMetaFullI4.csv")
I5_df <- read.csv("vgMetaFullI5.csv")

vgMetaFull_df <- rbind(I1_df, I2_df, I3_df, I4_df, I5_df)
write.csv(vgMetaFull_df, "vgMetaFull.csv")

#These games are not on Metacritic
gamesNotFound = c("Super Robot Wars X", "Super Robot Wars V", "Super Robot Wars OG: The Moon
Dwellers", "Hokuto ga Gotoku",
                "Girls und Panzer: Dream Tank Match", "Gintama Ranbu", "City Shrouded in Shadow",
                "Fortune Street: Dragon Quest & Final Fantasy 30th Anniversary", "Monster Hunter Double
Cross",
                "Fate/Extella Link", "Fate/Extella: The Umbral Star", "Seiken Densetsu Collection",
"Deformers",
                "Musou Stars", "Blue Reflection: Maboroshi ni Mau - Shoujo no Ken",
                "Dragon Quest Heroes I & II for Nintendo Switch", "Yonmegami Online: Cyber Dimension
Neptune",
                "Yakuza Zero: The Place of Oath", "New Danganronpa V3: Minna no Koroshiai Shin
Gakki",
                "Utawarerumono: Futari no Hakuoro", "Utawarerumono: itsuwari no kamen",
"Utawarerumono: chiriyukusha e no komoriuta",
                "The Idolmaster: Platinum Stars", "Jikkyou Powerful Pro Baseball 2016", "Jikkyou Powerful
Pro Baseball 2018",
                "Gundam Breaker 3", "Kamen Rider: Battride War Genesis", "Dragon's Dogma Online",
"Sengoku Basara 4: Sumeragi",
                "Hyperdimension Neptunia victory II", "Yakuza: Ishin", "The Legend of Heroes: Trails of
Cold Steel IV",
                "Little Dragon Cafe", "Pro Yakyuu Famista Evolution", "Persona Dancing: All-Star Triple
Pack",
                "The Snack World: Trejarers Gold", "Attack on Titan (KOEI)")

#Make a copy of the main data frame as a backup
vgMetaFull_df_clean <- copy(vgMetaFull_df)
```

```
#Eliminate games not on Metacritic
vgMetaFull_df_clean <- vgMetaFull_df_clean[!(vgMetaFull_df_clean$title %in% gamesNotFound),]

#Fix popular games with faulty URLs
for (i in c(1:nrow(vgMetaFull_df_clean))) {
  if (vgMetaFull_df_clean[i, ]$title == "Call of Duty: Black Ops 3" && vgMetaFull_df_clean[i,
]$summary == "No Summary Found") {
      vgMetaFull_df_clean[i,]$Url_title <- "call-of-duty-black-ops-iii"
      vgMetaFull_df_clean[i,]$numPlayers <- "Up to 18"
      vgMetaFull_df_clean[i,]$esrb <- "M"
      vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Action, Shooter, First-Person, Acrcade"
      vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Treyarch"
      vgMetaFull_df_clean[i,]$release_date <- "Nov 6, 2015"
      vgMetaFull_df_clean[i,]$metascore <- "81"
      vgMetaFull_df_clean[i,]$user_score <- "4.8"
      vgMetaFull_df_clean[i,]$summary <- "In Black Ops 3, Treyarch introduces a new momentum-
based chained-movement system which enables players to fluidly move through the environment with
finesse, using controlled thrust jumps, slides, wall runs and mantling abilities in a myriad of
combinations, all while continuously keeping full control over their weapon. Maps are designed from
the ground-up for the new movement system, enabling players to be successful with traditional
movement, as well as with advanced tactics and maneuvers."
  }
  else if (vgMetaFull_df_clean[i, ]$title == "Destiny 2" && vgMetaFull_df_clean[i, ]$summary == "No
Summary Found") {
      vgMetaFull_df_clean[i,]$Url_title <- "destiny-2"
      vgMetaFull_df_clean[i,]$numPlayers <- "Up to 12"
      vgMetaFull_df_clean[i,]$esrb <- "T"
      vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Action, Shooter, First-Person, Acrcade"
      vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Bungie"
      vgMetaFull_df_clean[i,]$release_date <- "Sep 6, 2017"
      vgMetaFull_df_clean[i,]$metascore <- "85"
      vgMetaFull_df_clean[i,]$user_score <- "4.9"
      vgMetaFull_df_clean[i,]$summary <- "In Destiny 2, the last safe city on Earth has fallen and lays
in ruins, occupied by a powerful new enemy and his elite army, the Red Legion. Every player creates his
own character called a “Guardian,” humanity’s chosen protectors. As a Guardian in Destiny 2, players
must master new abilities and weapons to reunite the city’s forces, stand together and fight back to
reclaim their home. In Destiny 2 players embark on a fresh story filled with new destinations around our
solar system to explore, and an expansive amount of activities to discover. There is something for almost
every type of gamer in Destiny 2, including gameplay for solo, cooperative and competitive players set
within a vast, evolving and exciting universe. [Bungie]"
  }
  else if (vgMetaFull_df_clean[i, ]$title == "Doom (2016)" && vgMetaFull_df_clean[i, ]$summary ==
"No Summary Found") {
      vgMetaFull_df_clean[i,]$Url_title <- "doom"
      vgMetaFull_df_clean[i,]$numPlayers <- "Up to 12"
      vgMetaFull_df_clean[i,]$esrb <- "M"
```

```
        vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Sci-Fi, Third-Person, Action, Shooter, First-
Person, Acrcade"
        vgMetaFull_df_clean[i,]$metaDevs <- "Developer: id Software"
        vgMetaFull_df_clean[i,]$release_date <- "May 13, 2016"
        vgMetaFull_df_clean[i,]$metascore <- "85"
        vgMetaFull_df_clean[i,]$user_score <- "8.3"
        vgMetaFull_df_clean[i,]$summary <- "There is no taking cover or stopping to regenerate health in
campaign mode as you beat back Hell’s raging demon hordes. Combine your arsenal of futuristic and
iconic guns, upgrades, movement and an advanced melee system to knock-down, slash, stomp, crush,
and blow apart demons in creative and violent ways. In multiplayer, dominate your opponents in
DOOM’s signature, fast-paced arena-style combat. In both classic and all-new game modes, annihilate
your enemies utilizing your personal blend of skill, powerful weapons, vertical movement, and unique
power-ups that allow you to play as a demon. DOOM SnapMap is an easy-to-use game and level editor
that allows for limitless gameplay experiences on every platform. Anyone can snap together and
customize maps, add pre-defined or custom gameplay, and edit game logic to create new modes.
Instantly play your creation or make it available to players around the world. [ID/Bethesda]"
    }
    else if (vgMetaFull_df_clean[i, ]$title == "Dragon Ball Fighter Z" && vgMetaFull_df_clean[i,
]$summary == "No Summary Found") {
        vgMetaFull_df_clean[i,]$Url_title <- "dragon-ball-fighterz"
        vgMetaFull_df_clean[i,]$numPlayers <- "2"
        vgMetaFull_df_clean[i,]$esrb <- "T"
        vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Action, Fighting, 2D"
        vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Arc System Works"
        vgMetaFull_df_clean[i,]$release_date <- "May 13, 2016"
        vgMetaFull_df_clean[i,]$metascore <- "85"
        vgMetaFull_df_clean[i,]$user_score <- "8.3"
        vgMetaFull_df_clean[i,]$summary <- "After the success of the Xenoverse series, its time to
introduce a new classic 2D DRAGON BALL fighting game for this generations consoles. DRAGON
BALL FighterZ is born from what makes the DRAGON BALL series so loved and famous: endless
spectacular fights with its allpowerful fighters. Partnering with Arc System Works, DRAGON BALL
FighterZ maximizes high end Anime graphics and brings easy to learn but difficult to master fighting
gameplay to audiences worldwide. Key Features: * 3vs3 TAG/SUPPORT Allows players to train and
master more than one fighter/style which brings deeper gameplay * HIGH-END ANIME GRAPHICS
Using the power of the Unreal engine and the talented team at Arc System Works, DRAGON BALL
FighterZ is a visual tour-de-force * SPECTACULAR FIGHTS Experience aerial combos, destructible
stages, famous scenes from the DRAGON BALL anime reproduced in 60FPS and 1080p resolution
(Higher resolution will be supported on PS4 Pro and Xbox Scorpio)"

    }
    else if (vgMetaFull_df_clean[i, ]$title == "Dragon Quest XI" && vgMetaFull_df_clean[i, ]$summary
== "No Summary Found") {
        vgMetaFull_df_clean[i,]$Url_title <- "dragon-quest-xi-echoes-of-an-elusive-age"
        vgMetaFull_df_clean[i,]$numPlayers <- "No Online Multiplayer"
        vgMetaFull_df_clean[i,]$esrb <- "T"
        vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Role-Playing, Japanese-Style"
        vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Square Enix"
        vgMetaFull_df_clean[i,]$release_date <- "Sep 4, 2018"
        vgMetaFull_df_clean[i,]$metascore <- "86"
        vgMetaFull_df_clean[i,]$user_score <- "8.7"
        vgMetaFull_df_clean[i,]$summary <- "DRAGON QUEST XI: Echoes of an Elusive Age tells a captivating tale of a hunted hero and is the long-awaited role-playing game from series creator Yuji Horii, character designer Akira Toriyama and composer Koichi Sugiyama. While it is the eleventh mainline entry in the critically acclaimed series, DRAGON QUEST XI is a completely standalone experience that features entirely new characters, a beautifully detailed world, finely tuned turn-based combat, and an immersive story that promises to appeal to longtime fans and franchise newcomers alike."
    }
    else if (vgMetaFull_df_clean[i, ]$title == "F1 2016 (Codemasters)" && vgMetaFull_df_clean[i, ]$summary == "No Summary Found") {
        vgMetaFull_df_clean[i,]$Url_title <- "f1-2016"
        vgMetaFull_df_clean[i,]$numPlayers <- "Up to 22"
        vgMetaFull_df_clean[i,]$esrb <- "E"
        vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Racing, Simulation, Automobile"
        vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Codemasters"
        vgMetaFull_df_clean[i,]$release_date <- "Aug 19, 2016"
        vgMetaFull_df_clean[i,]$metascore <- "82"
        vgMetaFull_df_clean[i,]$user_score <- "7.9"
        vgMetaFull_df_clean[i,]$summary <- "Includes the full 2016 season calendar of 21 tracks, including the brand new Baku circuit in Azerbaijan, and the full roster of 22 drivers and 11 teams, including the new Haas F1 Team. F1 2016 immerses you in not only the on-track excitement of a FORMULA ONE career, including the inclusion of the iconic Safety Car and also the Virtual Safety Car for the first time, but also uniquely offers the drama and vehicle development that goes on behind the scenes. Work with your agent, engineer and team to develop your car in the deepest ever career experience, spanning up to ten seasons. Forge your own path to glory, and rise to be the champion."
    }
    else if (vgMetaFull_df_clean[i, ]$title == "FIFA Soccer 14" && vgMetaFull_df_clean[i, ]$summary == "No Summary Found") {
        vgMetaFull_df_clean[i,]$Url_title <- "fifa-14"
        vgMetaFull_df_clean[i,]$numPlayers <- "Up to 22"
        vgMetaFull_df_clean[i,]$esrb <- "E"
        vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Sports, Traditional, Team, Soccer, Sim, Sim"
        vgMetaFull_df_clean[i,]$metaDevs <- "Developer: EA Canada"
        vgMetaFull_df_clean[i,]$release_date <- "Nov 12, 2013"
        vgMetaFull_df_clean[i,]$metascore <- "87"
        vgMetaFull_df_clean[i,]$user_score <- "6.3"
        vgMetaFull_df_clean[i,]$summary <- "Fuelled by EA SPORTS IGNITE, FIFA 14 feels alive with players who think, move, and behave like world-class footballers, and dynamic stadiums that come to life. Players have four times the decision making ability and feel alive with human-like reactions, anticipation, and instincts. With 10 times more animation depth and detail than previous consoles, FIFA 14 delivers the dynamic movements and biomechanics of the worlds best footballers. Players are agile and athletic as they plant, pivot and cut, and explode out of each step. The increased fidelity has a game-changing effect on gameplay. Plus, the artistry and athleticism of footballers will come to life with the addition of hundreds of new types of skills and hundreds of new behaviors. And for the first time ever,
fans will feel the electricity of a living stadium as the emotions of new 3D crowds rise and fall around
the stories on the pitch."
    }
    else if (vgMetaFull_df_clean[i, ]$title == "Gears of War" && vgMetaFull_df_clean[i, ]$summary ==
"No Summary Found") {
        vgMetaFull_df_clean[i,]$Url_title <- "gears-of-war-ultimate-edition"
        vgMetaFull_df_clean[i,]$numPlayers <- "Up to 8"
        vgMetaFull_df_clean[i,]$esrb <- "M"
        vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Action, Shooter, Third-Person, Arcade"
        vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Splash Damage, The Coalition"
        vgMetaFull_df_clean[i,]$release_date <- "Aug 25, 2015"
        vgMetaFull_df_clean[i,]$metascore <- "82"
        vgMetaFull_df_clean[i,]$user_score <- "7.6"
        vgMetaFull_df_clean[i,]$summary <- "The shooter has been remastered in 1080P and modernized
for Xbox One, and is augmented with new content including five campaign chapters never released on
Xbox."
    }
    else if (vgMetaFull_df_clean[i, ]$title == "Hitman (2016)" && vgMetaFull_df_clean[i, ]$summary
== "No Summary Found") {
        vgMetaFull_df_clean[i,]$Url_title <- "hitman"
        vgMetaFull_df_clean[i,]$numPlayers <- NA
        vgMetaFull_df_clean[i,]$esrb <- "M"
        vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Modern, Action Adventure, General"
        vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Io Interactive"
        vgMetaFull_df_clean[i,]$release_date <- "Oct 31, 2016"
        vgMetaFull_df_clean[i,]$metascore <- "85"
        vgMetaFull_df_clean[i,]$user_score <- "7.5"
        vgMetaFull_df_clean[i,]$summary <- "HITMAN empowers players to perform contract hits on
powerful, high-profile targets in exotic locations around the globe, from Paris and sunny coast of Italy to
the dust and hustle of the markets in Marrakesh in this stealth action title. Gameplay centers on taking
out targets in vast sandbox levels with complete freedom of approach. Where to go, when to attack and
who to kill is at your discretion."
    }
    else if (vgMetaFull_df_clean[i, ]$title == "Hyrule Warriors" && vgMetaFull_df_clean[i, ]$summary
== "No Summary Found") {
        vgMetaFull_df_clean[i,]$Url_title <- "hyrule-warriors-definitive-edition"
        vgMetaFull_df_clean[i,]$numPlayers <- "No Online Multiplayer"
        vgMetaFull_df_clean[i,]$esrb <- "T"
        vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Action, Beat-'Em-Up, 3D"
        vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Koei Tecmo Games"
        vgMetaFull_df_clean[i,]$release_date <- "May 18, 2018"
        vgMetaFull_df_clean[i,]$metascore <- "78"
        vgMetaFull_df_clean[i,]$user_score <- "8.1"
        vgMetaFull_df_clean[i,]$summary <- "A new, ultimate version of the exhilarating action game set
in the Zelda universe will include every map and mission, plus all 29 playable characters from both the
Wii U and Nintendo 3DS versions of the game, along with all of the previous paid downloadable
content. Play as Link, Zelda, Midna, Skull Kid and dozens more in action-packed battles at home or on
```

```
the go. Additionally, the game includes new outfits for Link and Zelda based on the Legend of Zelda:
Breath of the Wild game."
    }
    else if (vgMetaFull_df_clean[i, ]$title == "Kingdom Hearts 1.5 + 2.5 Remix" &&
vgMetaFull_df_clean[i, ]$summary == "No Summary Found") {
        vgMetaFull_df_clean[i,]$Url_title <- "kingdom-hearts-hd-i5-+-ii5-remix"
        vgMetaFull_df_clean[i,]$numPlayers <- "No Online Multiplayer"
        vgMetaFull_df_clean[i,]$esrb <- "E10+"
        vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Miscellaneous, Compilation"
        vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Square Enix"
        vgMetaFull_df_clean[i,]$release_date <- "Mar 28, 2017"
        vgMetaFull_df_clean[i,]$metascore <- "84"
        vgMetaFull_df_clean[i,]$user_score <- "9.0"
        vgMetaFull_df_clean[i,]$summary <- "Kingdom Hearts HD 1.5 + 2.5 Remix is an HD remastered
collection of 6 Kingdom Hearts experiences available for the first time on PS4."
    }
    else if ((vgMetaFull_df_clean[i, ]$title == "MineCraft" && vgMetaFull_df_clean[i, ]$summary ==
"No Summary Found") && (vgMetaFull_df_clean[i, ]$console == "xbox-one" &&
vgMetaFull_df_clean[i, ]$summary == "No Summary Found")) {
        vgMetaFull_df_clean[i,]$Url_title <- "miencraft-xbox-one-edition"
        vgMetaFull_df_clean[i,]$numPlayers <- "Up to 8"
        vgMetaFull_df_clean[i,]$esrb <- "E10+"
        vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Action Adventure, Adventure, 3D, Sandbox,
First-Person, Fantasy, Fantasy"
        vgMetaFull_df_clean[i,]$metaDevs <- "Developer: 4J Studios, Mojang AB"
        vgMetaFull_df_clean[i,]$release_date <- "Nov 18, 2014"
        vgMetaFull_df_clean[i,]$metascore <- "88"
        vgMetaFull_df_clean[i,]$user_score <- "6.7"
        vgMetaFull_df_clean[i,]$summary <- "Minecraft: Xbox One Edition is a game about placing
blocks to build anything you can imagine and was revealed at E3 that it will be coming to the Xbox
One."
    }
    else if (vgMetaFull_df_clean[i, ]$title == "Need for Speed (2015)" && vgMetaFull_df_clean[i,
]$summary == "No Summary Found") {
        vgMetaFull_df_clean[i,]$Url_title <- "need-for-speed"
        vgMetaFull_df_clean[i,]$numPlayers <- NA
        vgMetaFull_df_clean[i,]$esrb <- "T"
        vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Racing, Simulation, Automobile"
        vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Ghost Games"
        vgMetaFull_df_clean[i,]$release_date <- "Nov 3, 2015"
        vgMetaFull_df_clean[i,]$metascore <- "65"
        vgMetaFull_df_clean[i,]$user_score <- "7.6"
        vgMetaFull_df_clean[i,]$summary <- "Jump behind the wheel of some iconic cars and put the
pedal to the metal through Ventura Bay, a sprawling urban playground. Explore overlapping stories as
you establish your reputation – and your dream car – and become the top racing icon. Play repeatedly
this time, you have five distinct ways to win."
    }
```

```
  else if (vgMetaFull_df_clean[i, ]$title == "Nintendo Labo: Toy-Con 01 Variety Kit" &&
vgMetaFull_df_clean[i, ]$summary == "No Summary Found") {
      vgMetaFull_df_clean[i,]$Url_title <- "nintendo-labo-toycon-01-variety-kit"
      vgMetaFull_df_clean[i,]$numPlayers <- "No Online Multiplayer"
      vgMetaFull_df_clean[i,]$esrb <- "E"
      vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Miscellaneous, Party / Minigame"
      vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Nintendo"
      vgMetaFull_df_clean[i,]$release_date <- "Apr 20, 2018"
      vgMetaFull_df_clean[i,]$metascore <- "77"
      vgMetaFull_df_clean[i,]$user_score <- "6.9"
      vgMetaFull_df_clean[i,]$summary <- "The Variety Kit includes five different projects to Make,
Play, and Discover: two Toy-Con RC Cars, a Toy-Con Fishing Rod, a Toy-Con House, a Toy-Con
Motorbike, and a Toy-Con Piano. Bring each Toy-Con creation to life with the power of the Nintendo
Switch console and Joy-Con controllers."
  }
  else if (vgMetaFull_df_clean[i, ]$title == "Ni-Oh" && vgMetaFull_df_clean[i, ]$summary == "No
Summary Found") {
      vgMetaFull_df_clean[i,]$Url_title <- "nioh"
      vgMetaFull_df_clean[i,]$numPlayers <- "2"
      vgMetaFull_df_clean[i,]$esrb <- "M"
      vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S):  Role-Playing, Action RPG"
      vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Team Ninja"
      vgMetaFull_df_clean[i,]$release_date <- "Feb 7, 2017"
      vgMetaFull_df_clean[i,]$metascore <- "88"
      vgMetaFull_df_clean[i,]$user_score <- "8.5"
      vgMetaFull_df_clean[i,]$summary <- "Take up your sword and travel to Japans blood-bathed
Sengoku period an era ravaged by warring states and dark, malevolent forces and cut a violent path
through the land as the masterless samurai, William. Cross blades in brutal hand-to-hand combat,
wielding swords, axes, spears and even war hammers against foes both human and demon. Endure the
vicious encounters and learn from your mistakes: each death will bring you resurrection and each
resurrection a greater resolve to overcome your foes."
  }
  else if (vgMetaFull_df_clean[i, ]$title == "Overwatch" && vgMetaFull_df_clean[i, ]$summary ==
"No Summary Found") {
      vgMetaFull_df_clean[i,]$Url_title <- "overwatch"
      vgMetaFull_df_clean[i,]$numPlayers <- "Up to 12"
      vgMetaFull_df_clean[i,]$esrb <- "T"
      vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Action, Shooter, First-Person, Tactical"
      vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Blizzard Entertainment"
      vgMetaFull_df_clean[i,]$release_date <- "May 23, 2016"
      vgMetaFull_df_clean[i,]$metascore <- "90"
      vgMetaFull_df_clean[i,]$user_score <- "6.4"
      vgMetaFull_df_clean[i,]$summary <- "Overwatch is a highly stylized team-based shooter set on
earth in the near future. Every match is an intense multiplayer showdown pitting a diverse cast of
soldiers, mercenaries, scientists, adventurers, and oddities against each other in an epic, globe-spanning
conflict."
  }
  else if (vgMetaFull_df_clean[i, ]$title == "Pokken Tournament" && vgMetaFull_df_clean[i, ]$summary == "No Summary Found") {
      vgMetaFull_df_clean[i,]$Url_title <- "pokken-tournament-dx"
      vgMetaFull_df_clean[i,]$numPlayers <- "2"
      vgMetaFull_df_clean[i,]$esrb <- "E10+"
      vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Action, Fighting, 3D"
      vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Bandai Namco Games"
      vgMetaFull_df_clean[i,]$release_date <- "Sep 22, 2017"
      vgMetaFull_df_clean[i,]$metascore <- "79"
      vgMetaFull_df_clean[i,]$user_score <- "7.5"
      vgMetaFull_df_clean[i,]$summary <- "The popular Pokemon fighting game comes to Nintendo Switch with added Pokemon fighters and new ways to battle other players. Take direct control of one of over 20 prized Pokemon fighters to defeat other Pokemon in action-packed arena fights. With Nintendo Switch, you can battle at home or on the go to become the Ferrum League champion. Master the new fighting styles of Croagunk, Scizor, Empoleon, Darkrai, and the newly added Decidueye. New modes give you the ability to challenge your friends in intense competition. Duke it out with your favorite Pokemon Fighters in Team Battle. Then record and upload your favorite fights with the new replay feature. Take the battle on the road with the portable Nintendo Switch system or find competition online in the new Group Match Mode. Do you have what it takes to claim the title of Pokken Tournament DX champion? There's only one way to find out, and that's by stepping into the battle arena. * Fight as Decidueye, Darkrai , Scizor, Empoleon, Croagunk, and more * Stunning Pokemon battles come to life like never before * Execute powerful Pokemon moves with simple button combinations * Enjoy an intuitive fighting system for all skill levels"
  }
  else if (vgMetaFull_df_clean[i, ]$title == "Prey (2017)" && vgMetaFull_df_clean[i, ]$summary == "No Summary Found") {
      vgMetaFull_df_clean[i,]$Url_title <- "prey"
      vgMetaFull_df_clean[i,]$numPlayers <- "No Online Multiplayer"
      vgMetaFull_df_clean[i,]$esrb <- "M"
      vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): General, Action, Shooter, First-Person, Arcade"
      vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Arkane Studios"
      vgMetaFull_df_clean[i,]$release_date <- "May 5, 2017"
      vgMetaFull_df_clean[i,]$metascore <- "79"
      vgMetaFull_df_clean[i,]$user_score <- "7.8"
      vgMetaFull_df_clean[i,]$summary <- "You awaken aboard Talos I, a space station orbiting the moon in the year 2032. You are the key subject in an experiment meant to change humanity forever – but things have gone terribly wrong. The space station has been overrun by hostile aliens and you are now being hunted. As you delve into the dark secrets of Talos I and your own past, you have to survive using the tools found on the station, your wits, weapons, and mind-bending abilities. [Bethesda Softworks]"
  }
  else if (vgMetaFull_df_clean[i, ]$title == "Rainbow Six: Siege" && vgMetaFull_df_clean[i, ]$summary == "No Summary Found") {
      vgMetaFull_df_clean[i,]$Url_title <- "tom-clancys-rainbow-six-siege"
      vgMetaFull_df_clean[i,]$numPlayers <- "Up to 10"
      vgMetaFull_df_clean[i,]$esrb <- "M"
```

```
        vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Modern, Action, Shooter, First-Person, Tactical"
        vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Ubisoft Montreal"
        vgMetaFull_df_clean[i,]$release_date <- "Dec 1, 2015"
        vgMetaFull_df_clean[i,]$metascore <- "73"
        vgMetaFull_df_clean[i,]$user_score <- "7.4"
        vgMetaFull_df_clean[i,]$summary <- "Tom Clancy’s Rainbow Six Siege is inspired by real world counter-terrorist organizations, and inserts players in the middle of lethal close-quarters engagements. For the first time in a Tom Clancy’s Rainbow Six game, players can choose from a variety of unique Counter Terrorist Operators and engage in tangible sieges, a new style of assault in which enemies have the means to transform their environments into modern strongholds while Rainbow Six teams lead the assault to breach the enemy’s position."
    }
    else if ((vgMetaFull_df_clean[i, ]$title == "Ratchet & Clank" && vgMetaFull_df_clean[i, ]$summary == "No Summary Found") || (vgMetaFull_df_clean[i, ]$title == "Ratchet & Clank (2016)" && vgMetaFull_df_clean[i, ]$summary == "No Summary Found")) {
        vgMetaFull_df_clean[i,]$Url_title <- "ratchet-clank"
        vgMetaFull_df_clean[i,]$numPlayers <- "No Online Multiplayer"
        vgMetaFull_df_clean[i,]$esrb <- "E10+"
        vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S):  Action, Platformer, 3D"
        vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Insomniac Games"
        vgMetaFull_df_clean[i,]$release_date <- "Apr 12, 2016"
        vgMetaFull_df_clean[i,]$metascore <- "85"
        vgMetaFull_df_clean[i,]$user_score <- "8.6"
        vgMetaFull_df_clean[i,]$summary <- "Play the game, based on the movie, based on the game. Ratchet & Clank (PS4) is a new game based on elements from the original Ratchet & Clank (PS2). Developed alongside the major motion CG-animated picture coming to theatres in 2016, Ratchet & Clank (PS4) marks the PlayStation 4 debut of PlayStation's greatest heroes. Join Ratchet, Clank, Captain Qwark and new friends as they battle to save the Solana Galaxy from the evil Chairman Drek. With an hour of new cinematics (including footage from the film), Ratchet & Clank (PS4) takes a deeper look at the characters' origin stories and modernizes the original gameplay. Explore the galaxy in a game that features new planets, new and updated gameplay segments, all-new bosses, all-new Clank gameplay, all-new flight sequences, and much more --- with completely new visuals built to demonstrate the power of the PS4. Battle your enemies with an out-of-this-world arsenal, including new weapons and fan-favorite tools of destruction from the Ratchet & Clank Future series. From the brand new Pixelizer (which transforms enemies into explosive 8-bit pixels) to franchise favorites like the dance-party-inducing Groovitron and wise-cracking robotic bodyguard Mr. Zurkon, Ratchet & Clank's arsenal has never been better. Help Ratchet and Clank save the galaxy for the first time, again, exclusively on PlayStation 4."
    }
    else if (vgMetaFull_df_clean[i, ]$title == "Resident Evil VII: Biohazard" && vgMetaFull_df_clean[i, ]$summary == "No Summary Found") {
        vgMetaFull_df_clean[i,]$Url_title <- "resident-evil-7-biohazard"
        vgMetaFull_df_clean[i,]$numPlayers <- "No Online Multiplayer"
        vgMetaFull_df_clean[i,]$esrb <- "M"
        vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Action Adventure, Survival"
        vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Capcom"
        vgMetaFull_df_clean[i,]$release_date <- "Aug 25, 2015"
        vgMetaFull_df_clean[i,]$metascore <- "86"
        vgMetaFull_df_clean[i,]$user_score <- "7.9"
        vgMetaFull_df_clean[i,]$summary <- "While Resident Evil 7 draws from the series’ roots of atmospheric survival horror, it also delivers a new level of terror. In the Resident Evil games of yesteryear, players braced for fear in the first-person via the creepy door-opening scenes, and Resident Evil 7 ramps up that tension with an immersive first-person view and a photorealistic graphical style. Capcom is able to achieve a higher degree of visual fidelity thanks to the new proprietary in-house RE Engine that includes VR oriented tools. With the RE Engine plus industry leading audio and visual technologies, you experience every abhorrent detail up close and personal in Resident Evil 7. Playing the game in the PlayStation VR Mode escalates the unsettling feeling of presence to a level that horror fans have never experienced. The full gameplay experience is available in the included PlayStation VR Mode from beginning to end. No Resident Evil you’ve ever survived could prepare you for this. [Capcom]"
    }
    else if (vgMetaFull_df_clean[i, ]$title == "Rise of the Tomb Raider" && vgMetaFull_df_clean[i, ]$summary == "No Summary Found") {
        vgMetaFull_df_clean[i,]$Url_title <- "rise-of-the-tomb-raider-20-year-celebration"
        vgMetaFull_df_clean[i,]$numPlayers <- "2"
        vgMetaFull_df_clean[i,]$esrb <- "M"
        vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Action Adventure, Linear"
        vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Crystal Dynamics, Nixxes Software"
        vgMetaFull_df_clean[i,]$release_date <- "OCt 11, 2016"
        vgMetaFull_df_clean[i,]$metascore <- "88"
        vgMetaFull_df_clean[i,]$user_score <- "8.1"
        vgMetaFull_df_clean[i,]$summary <- "Lara Croft embarks on her first Tomb Raiding expedition to the most treacherous and remote regions of Siberia. Rise of the Tomb Raider: 20 Year Celebration also includes the new “Blood Ties” single player content, VR support, new Co-op play for Endurance mode, and more."
    }
    else if (vgMetaFull_df_clean[i, ]$title == "Star Wars Battlefront (2015)" && vgMetaFull_df_clean[i, ]$summary == "No Summary Found") {
        vgMetaFull_df_clean[i,]$Url_title <- "star-wars-battlefront"
        vgMetaFull_df_clean[i,]$numPlayers <- "Up to 40"
        vgMetaFull_df_clean[i,]$esrb <- "T"
        vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): General, General, Action, Shooter, First-Person, Tactical"
        vgMetaFull_df_clean[i,]$metaDevs <- "Developer: EA DICE"
        vgMetaFull_df_clean[i,]$release_date <- "Nov 17, 2015"
        vgMetaFull_df_clean[i,]$metascore <- "73"
        vgMetaFull_df_clean[i,]$user_score <- "5.0"
        vgMetaFull_df_clean[i,]$summary <- "Star Wars Battlefront lets players live out a broad array of heroic moments and intense battle fantasies of their own – firing blasters, riding speeder bikes and snow speeders, commanding AT-ATs and piloting TIE fighters and the Millennium Falcon. These battles will take place on the iconic planets of the Star Wars universe, including Endor, Hoth, Tatooine and the previously unexplored planet, Sullust. Gamers can play as the memorable characters from the original trilogy such as Darth Vader and Boba Fett. Star Wars Battlefront features a wide range of modes tailored
```

```
for different types of battles, from larger 40-person competitive multiplayer to crafted missions that are
played solo, with a partner via split-screen offline or co-operatively online. [EA/Lucasfilm]"
   }
  else if (vgMetaFull_df_clean[i, ]$title == "Star Wars Battlefront II (2017)" &&
vgMetaFull_df_clean[i, ]$summary == "No Summary Found") {
      vgMetaFull_df_clean[i,]$Url_title <- "star-wars-battlefront-ii"
      vgMetaFull_df_clean[i,]$numPlayers <- "Up to 40"
      vgMetaFull_df_clean[i,]$esrb <- "T"
      vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Action, Shooter, First-Person, Tactical"
      vgMetaFull_df_clean[i,]$metaDevs <- "Developer: EA DICE"
      vgMetaFull_df_clean[i,]$release_date <- "Nov 14, 2017"
      vgMetaFull_df_clean[i,]$metascore <- "82"
      vgMetaFull_df_clean[i,]$user_score <- "7.6"
      vgMetaFull_df_clean[i,]$summary <- " Star Wars Battlefront II allows players to experience the
untold story of an Imperial elite special forces soldier in an all-new single player campaign. In epic
multiplayer battles, players pilot a First Order TIE fighter through intense dogfights in space, and play as
ground troopers or iconic heroes and villains, such as Yoda and Darth Maul, across all three Star Wars
eras. Featuring new characters created by Motive in close partnership with the story group at Lucasfilm,
the Star Wars Battlefront II single player campaign delivers a new story to the Star Wars universe.
Taking place in the 30 years between the destruction of Death Star II and through the rise of the First
Order, the single player story introduces Iden Versio, the leader of Inferno Squad, an elite special forces
unit of the Galactic Empire, who are equally lethal on the ground and in space. In addition to Iden,
players also encounter and play as other iconic Star Wars heroes and villains like Luke Skywalker and
Kylo Ren during the campaign."
   }
  else if (vgMetaFull_df_clean[i, ]$title == "Star Wars: Battlefront (2015)" && vgMetaFull_df_clean[i,
]$summary == "No Summary Found") {
      vgMetaFull_df_clean[i,]$Url_title <- "star-wars-battlefront"
      vgMetaFull_df_clean[i,]$numPlayers <- "Up to 40"
      vgMetaFull_df_clean[i,]$esrb <- "T"
      vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): General, General, Action, Shooter, First-Person,
Tactical"
      vgMetaFull_df_clean[i,]$metaDevs <- "Developer: EA DICE"
      vgMetaFull_df_clean[i,]$release_date <- "Nov 17, 2015"
      vgMetaFull_df_clean[i,]$metascore <- "73"
      vgMetaFull_df_clean[i,]$user_score <- "5.0"
      vgMetaFull_df_clean[i,]$summary <- "Star Wars Battlefront lets players live out a broad array of
heroic moments and intense battle fantasies of their own – firing blasters, riding speeder bikes and snow
speeders, commanding AT-ATs and piloting TIE fighters and the Millennium Falcon. These battles will
take place on the iconic planets of the Star Wars universe, including Endor, Hoth, Tatooine and the
previously unexplored planet, Sullust. Gamers can play as the memorable characters from the original
trilogy such as Darth Vader and Boba Fett. Star Wars Battlefront features a wide range of modes tailored
for different types of battles, from larger 40-person competitive multiplayer to crafted missions that are
played solo, with a partner via split-screen offline or co-operatively online. [EA/Lucasfilm]"
   }
  else if (vgMetaFull_df_clean[i, ]$title == "The Elder Scrolls V: Skyrim" && vgMetaFull_df_clean[i,
]$summary == "No Summary Found") {
      vgMetaFull_df_clean[i,]$Url_title <- "the-elder-scrolls-v-skyrim-special-edition"
      vgMetaFull_df_clean[i,]$numPlayers <- "No Online Multiplayer"
      vgMetaFull_df_clean[i,]$esrb <- "M"
      vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Role-Playing, Western-Style"
      vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Bethesda Game Studios"
      vgMetaFull_df_clean[i,]$release_date <- "Oct 28, 2016"
      vgMetaFull_df_clean[i,]$metascore <- "81"
      vgMetaFull_df_clean[i,]$user_score <- "7.1"
      vgMetaFull_df_clean[i,]$summary <- "Skyrim Special Edition brings the epic fantasy to life in detail. The Special Edition includes the game and add-ons with all-new features like remastered art and effects, volumetric god rays, dynamic depth of field, screen-space reflections, and more."
  }
  else if (vgMetaFull_df_clean[i, ]$title == "The Evil Within II" && vgMetaFull_df_clean[i, ]$summary == "No Summary Found") {
      vgMetaFull_df_clean[i,]$Url_title <- "the-evil-within-2"
      vgMetaFull_df_clean[i,]$numPlayers <- "No Online Multiplayer"
      vgMetaFull_df_clean[i,]$esrb <- "M"
      vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Action Adventure, Survival"
      vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Tango Gameworks"
      vgMetaFull_df_clean[i,]$release_date <- "Oct 13, 2017"
      vgMetaFull_df_clean[i,]$metascore <- "76"
      vgMetaFull_df_clean[i,]$user_score <- "8.5"
      vgMetaFull_df_clean[i,]$summary <- "You are Detective Sebastian Castellanos and at your lowest point. But when given a chance to save your daughter, you must enter a world filled with nightmares and discover the dark origins of a once-idyllic town to bring her back. Horrifying threats emerge from every corner as the world twists and warps around you. Will you face adversity head on with weapons and traps, or sneak through the shadows to survive?"
  }
  else if (vgMetaFull_df_clean[i, ]$title == "The Last of Us" && vgMetaFull_df_clean[i, ]$summary == "No Summary Found") {
      vgMetaFull_df_clean[i,]$Url_title <- "the-last-of-us-remastered"
      vgMetaFull_df_clean[i,]$numPlayers <- "Up to 8"
      vgMetaFull_df_clean[i,]$esrb <- "M"
      vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): Action Adventure, General, Modern"
      vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Naughty Dog"
      vgMetaFull_df_clean[i,]$release_date <- "Jul 29, 2014"
      vgMetaFull_df_clean[i,]$metascore <- "95"
      vgMetaFull_df_clean[i,]$user_score <- "9.1"
      vgMetaFull_df_clean[i,]$summary <- "The Last of Us has been rebuilt for the PlayStation4 system. Now featuring full 1080p, higher resolution character models, improved shadows and lighting, in addition to several other gameplay improvements. 20 years after a pandemic has radically changed known civilization, infected humans run wild and survivors are killing each other for food, weapons; whatever they can get their hands on. Joel, a violent survivor, is hired to smuggle a 14 year-old girl, Ellie, out of an oppressive military quarantine zone, but what starts as a small job soon transforms into a brutal journey across the U.S. The Last of Us Remastered includes the Abandoned Territories Map Pack, Reclaimed Territories Map Pack, and the critically acclaimed The Last of Us: Left Behind Single Player campaign that combines themes of survival, loyalty, and love with tense, survival-action gameplay.
```

```
Remastered Features: -Explore a brutal post-pandemic world, fully realized with the power of
PlayStation4 system -Includes additional game content: over $30 in value -Delve into Ellies past in Left
Behind, the single-player prequel chapter -Eight new multiplayer maps in the Abandoned and Reclaimed
Territories packs -In-game cinematic commentary from the cast and creative director"
   }
   else if (vgMetaFull_df_clean[i, ]$title == "The Order 1866" && vgMetaFull_df_clean[i, ]$summary
== "No Summary Found") {
      vgMetaFull_df_clean[i,]$Url_title <- "the-order-1866"
      vgMetaFull_df_clean[i,]$numPlayers <- "No Online Multiplayer"
      vgMetaFull_df_clean[i,]$esrb <- "M"
      vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): General, General, Action, Action Adventure,
Shooter, Linear, Third-Person, Sci-Fi, Arcade"
      vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Ready at Dawn, SCE Santa Monica"
      vgMetaFull_df_clean[i,]$release_date <- "Feb 20, 2015"
      vgMetaFull_df_clean[i,]$metascore <- "63"
      vgMetaFull_df_clean[i,]$user_score <- "6.7"
      vgMetaFull_df_clean[i,]$summary <- "The Order: 1886 introduces us to a unique vision of
Victorian-Era London in which Man uses advanced technology to battle a powerful and ancient enemy.
As Galahad, a member of an elite order of Knights, join a centuries-old war against a powerful threat
that will determine the course of history in this third-person action-adventure shooter."
   }
   else if (vgMetaFull_df_clean[i, ]$title == "Uncharted (PS4)" && vgMetaFull_df_clean[i, ]$summary
== "No Summary Found") {
      vgMetaFull_df_clean[i,]$Url_title <- "uncharted-4-a-thiefs-end"
      vgMetaFull_df_clean[i,]$numPlayers <- "Up to 10"
      vgMetaFull_df_clean[i,]$esrb <- "T"
      vgMetaFull_df_clean[i,]$metaGenre <- "Genre(S): General, Modern, Action Adventure, Linear"
      vgMetaFull_df_clean[i,]$metaDevs <- "Developer: Naughty Dog"
      vgMetaFull_df_clean[i,]$release_date <- "May 10, 2016"
      vgMetaFull_df_clean[i,]$metascore <- "93"
      vgMetaFull_df_clean[i,]$user_score <- "8.4"
      vgMetaFull_df_clean[i,]$summary <- " Set 3 years after the events of Uncharted 3, Nathan Drake
has apparently left the world of fortune hunting behind. However, it doesn’t take long for adventure to
come calling when Drake’s brother, Sam, re-emerges asking for his help to save his own life and
offering an adventure Drake cannot resist. On the hunt for Captain Henry Avery’s long-lost treasure,
Sam and Drake embark on a journey to find Libertalia, the pirate utopia deep in the forests of
Madagascar. Uncharted 4: A Thief’s End takes players around the globe, through jungle isles, urban
cities and snow-capped peaks on the search for Avery’s fortune. [Naughty Dog]"
   }
}

#Vgchartz provided the publisher rather than developer
colnames(vgMetaFull_df_clean)[which(names(vgMetaFull_df_clean) == "developer")] <- "publisher"
colnames(vgMetaFull_df_clean)[which(names(vgMetaFull_df_clean) == "metaDevs")] <- "developer"

#Set data types
vgMetaFull_df_clean %>% mutate_if(is.factor, as.character) -> vgMetaFull_df_clean
```

```
vgMetaFull_df_clean$week_posted <- sapply(vgMetaFull_df_clean$week_posted, gsub, pattern = "th",
replacement = "")
vgMetaFull_df_clean$week_posted <- sapply(vgMetaFull_df_clean$week_posted, gsub, pattern = "nd",
replacement = "")
vgMetaFull_df_clean$week_posted <- sapply(vgMetaFull_df_clean$week_posted, gsub, pattern = "st",
replacement = "")
vgMetaFull_df_clean$week_posted <- sapply(vgMetaFull_df_clean$week_posted, gsub, pattern = "rd",
replacement = "")
vgMetaFull_df_clean$week_posted <- as.Date(vgMetaFull_df_clean$week_posted, "%d %b %Y")
vgMetaFull_df_clean$console <- as.factor(vgMetaFull_df_clean$console)
vgMetaFull_df_clean$publisher <- as.factor(vgMetaFull_df_clean$publisher)
vgMetaFull_df_clean$genre <- as.factor(vgMetaFull_df_clean$genre)
vgMetaFull_df_clean$units_total <- as.numeric(gsub(",", "", vgMetaFull_df_clean$units_total))
vgMetaFull_df_clean$units_week <- as.numeric(gsub(",", "", vgMetaFull_df_clean$units_week))
vgMetaFull_df_clean$numPlayers <- as.factor(vgMetaFull_df_clean$numPlayers)
vgMetaFull_df_clean$esrb <- as.factor(vgMetaFull_df_clean$esrb)
vgMetaFull_df_clean$metaGenre <- as.factor(vgMetaFull_df_clean$metaGenre)
vgMetaFull_df_clean$developer <- as.factor(vgMetaFull_df_clean$developer)
vgMetaFull_df_clean$release_date <- sapply(vgMetaFull_df_clean$release_date, gsub, pattern = ",",
replacement = "")
vgMetaFull_df_clean$release_date <- as.Date(vgMetaFull_df_clean$release_date, "%b %d %Y")
vgMetaFull_df_clean$metascore <- as.numeric(vgMetaFull_df_clean$metascore)
vgMetaFull_df_clean$user_score <- as.numeric(vgMetaFull_df_clean$user_score)


#Add two columns. One for the user score multiplied by 10, and another for the difference between the
two columns
vgMetaFull_df_clean$user_score_x10 <- vgMetaFull_df_clean$user_score * 10
vgMetaFull_df_clean$score_diff <- vgMetaFull_df_clean$metascore -
vgMetaFull_df_clean$user_score_x10

#Convert errors to NAs for functionality
revalue(vgMetaFull_df_clean$esrb, c("ERROR" = NA)) -> vgMetaFull_df_clean$esrb
revalue(vgMetaFull_df_clean$numPlayers, c("ERROR" = NA)) -> vgMetaFull_df_clean$numPlayers

for (i in c(1:nrow(vgMetaFull_df_clean))) {
   if (vgMetaFull_df_clean[i, ]$metaGenre == "") {
      vgMetaFull_df_clean[i,]$numPlayers = NA
   }
   if (vgMetaFull_df_clean[i, ]$developer == "") {
      vgMetaFull_df_clean[i,]$metaDevs = NA
   }
   if (vgMetaFull_df_clean[i, ]$summary == "No Summary Found") {
      vgMetaFull_df_clean[i,]$summary = NA
   }
}
```

```
#Clean metaGenre; remove "Genre(s): "
vgMetaFull_df_clean$metaGenre <- sapply(vgMetaFull_df_clean$metaGenre, substr, start = 11, stop =
1000000L)

#Clean metaDev; remove "Developer: "
vgMetaFull_df_clean$developer <- sapply(vgMetaFull_df_clean$developer, substr, start = 12, stop =
1000000L)

#Reorder columns
vgMetaFull_df_clean <- vgMetaFull_df_clean %>% select(2:18, 20, 21, 19)

write.csv(vgMetaFull_df_clean, "vgMetaFullCleanFinal.csv")

remDr$close()

rD$server$stop()
```

4. Visualization.R

```
library(RSelenium)
library(stringr)
library(forcats)
library(ggplot2)
library(zoo)
library(extrafont)
library(grid)
library(dplyr)
library(dbplyr)
library(sqldf)
library(corrplot)
#Font
font_import()
loadfonts(device = "win")

#############################################GETTING DATA READY FOR
PLOTS################################

#working directory
setwd("C:\\Users\\emcat\\Desktop\\Data Science Programming")
#Read the tables
data <- read.csv("vgMetaFullCleanFinal.csv ", header = TRUE)
#SQL to find the highest sales value for the each distint name
#Testing number of distinct titles
distincttitle=sqldf("SELECT Distinct title
FROM data ;")
unitstotal= sqldf("SELECT Max(units_total), title
FROM data ;")
#testing sql query
test = sqldf("SELECT units_total,title
FROM data
Group By title;")
#Unique genres to find the number of genre for later counts
genretest = sqldf("SELECT genre
FROM data
Group By Genre;")
#Unique publishers to find the number of publishers for later counts
publishertest = sqldf("SELECT publisher
FROM data
Group By publisher;
")
#Unique developers to find the number of developers in later couunts
developertest = sqldf("SELECT developer
FROM data
Group By developer;
")
#sql query that narrows dataset to the top count sold per game
newdata = sqldf("SELECT *
FROM data
Group By title;")
#New dataframe of only the top count sold per game
write.csv(newdata, 'test.csv')
#numeric values- dataframe of only the numeric values
numeric = data.frame(newdata$units_total, newdata$metascore, newdata$user_score_x10,
newdata$score_diff, newdata$chart_week)
numeric
#Changing names of the columns of the dataframe
colnames(numeric) = c("Total Unit Sales", "Metascore", "User_Score", "Score_Diff", "ChartWeek")
colnames(numeric)
#Creates count of number of games per esrb score
esrb_game <- newdata %>%
  count(esrb, sort = TRUE) %>%
  #filters out NA's and otherwise useless values
  filter(esrb %in% c('M', 'T', 'E', 'E10+'))
#Creates filtered ESRB sales
esrb_game_sales <- newdata %>%
  select(esrb, units_total) %>%
  group_by(esrb) %>%
  filter(esrb %in% c('M', 'T', 'E', 'E10+')) %>%
  summarise(Units_Sold = sum(units_total))
#Multiplayer sales DOES NOT WORK
multiplayer_game_sales <- newdata %>%
    select(multi_count, units_total) %>%
    group_by(multi_count) %>%
    summarise(Units_Sold = sum(units_total))
#Creates count of number of games per console
Console_game <- newdata %>%
    count(console, sort = TRUE) %>%
    #filters out NA's and otherwise useless values
    filter(console %in% c('switch', 'playstation-4', 'xbox-one'))
#Top ten title
top10_title <- newdata %>%
    select(title, units_total) %>%
    arrange(desc(units_total)) %>%
    head(n = 10)
#botton 10 title
bottom10_title <- newdata %>%
    select(title, units_total) %>%
    arrange(desc(units_total)) %>%
    tail(n = 10)
#Top 10 developer
top10_developer <- newdata %>%
    select(developer, units_total) %>%
    arrange(desc(units_total)) %>%
    head(n = 10)
#Bottom 10 developer
bottom10_developer <- newdata %>%
    select(developer, units_total) %>%
    arrange(desc(units_total)) %>%
    tail(n = 10)

#Top 10 publisher
top10_publisher <- newdata %>%
    select(publisher, units_total) %>%
    arrange(desc(units_total)) %>%
    head(n = 10)
#Bottom 10 publisher
bottom10_publisher <- newdata %>%
    select(publisher, units_total) %>%
    arrange(desc(units_total)) %>%
    tail(n = 10)
#Creates count of number of genres
Genre_game <- newdata %>%
    count(genre, sort = TRUE) %>%
    #filters out NA's and otherwise useless values
    filter(genre %in% c('Action','Action - Adventure','Adventure','Fighting','Misc',
'Music',
 'Party',
 'Platform',
 'Puzzle',
 'Racing',
 'Role - Playing',
 'Sandbox',
 'Shooter',
'Simulation',
'Sports',
 'Strategy'))
#Creates count of number of publishers
Publishier_game <- newdata %>%
    count(publisher, sort = TRUE) %>%
    #filters out NA's and otherwise useless values
    filter(publisher %in% c(
 '505 Games'
, 'Activision'
, 'Aqua Plus'
, 'Arc System Works'
, 'Atlus'
, 'Bandai Namco Entertainment'
, 'Bethesda Softworks'
, 'Big Ben Interactive'
, 'CCP'
, 'Capcom'
, 'Capcom Entertainment'
, 'City Interactive'
, 'Codemasters'
, 'Crytek'
, 'D3Publisher'
, 'Deep Silver'
, 'Disney Interactive Studios'
, 'Dusenberry Martin Racing'
, 'Electronic Arts'
, 'Focus Home Interactive'
, 'From Software'
, 'Frontier Developments'
, 'FuRyu Corporation'
, 'Gearbox Software'
, 'Grey Box'
, 'Gun Media'
, 'Harmonix Music Systems'
, 'Hello Games'
, 'Idea Factory'
, 'Insomniac Games'
, 'Kadokawa Games'
, 'Kalypso Media'
, 'Koch Media'
, 'Konami Digital Entertainment'
, 'Marvelous'
, 'Maximum Games'
, 'Merge Games'
, 'Microsoft Game Studios'
, 'Microsoft Studios'
, 'Milestone'
, 'Milestone S.r.l.'
, 'Mojang'
, 'Namco Bandai Games'
, 'Natsume'
, 'Nicalis'
, 'Nighthawk Interactive'
, 'Nihon Falcom Corporation'
, 'Nintendo'
, 'Nippon Ichi Software'
, 'PQube'
, 'Rebellion Developments'
, 'Sega'
, 'Slightly Mad Studios'
, 'Sold Out'
, 'Sony Computer Entertainment'
, 'Sony Computer Entertainment America'
, 'Sony Computer Entertainment Europe'
, 'Sony Interactive Entertainment'
, 'Spike Chunsoft'
, 'Square Enix'
, 'Studio Wildcard'
, 'THQ Nordic'
, 'Take - Two Interactive'
, 'Tecmo Koei'
, 'Telltale Games'
, 'Ubisoft'
, 'Unknown'
, 'Warner Bros. Interactive Entertainment'
, 'Xseed Games'))

#Creates Count of games per developer
developer_game <- newdata %>%
   count(developer, sort = TRUE) %>%
   filter(!developer %in% c(""))



#add a boolean column for singleplayer/multiplayer
single <- c("2", "No Online Multiplayer")
newdata$multiplayer <- ""
for (i in c(1:nrow(newdata))) {
  if (newdata[i, ]$numPlayers %in% single) {
    newdata[i,]$multiplayer = "single"
  } else if (is.na(newdata[i, ]$numPlayers)) {
    newdata[i,]$multiplayer = "N/A"
  } else {
    newdata[i,]$multiplayer = "multi"
  }
}

newdata$multiplayer <- as.factor(newdata$multiplayer)

multi_count <- newdata %>%
  count(multiplayer, sort = TRUE)
newdata
#Omitting nulls for regression
ompleterecords <- na.omit(numeric)

##############################################################################################
###########

###################GRAPHS BY SALES(TARGET VARIABLES) BY PREDICTOR
VARIABLES########################

#Sales with metascore
salesmeetascore <- ggplot(data = newdata, aes(x = newdata$metascore, y = newdata$units_total)) +
  geom_point(color="#FF6666")

salesmeetascore + ggtitle("Total Units Sold by MetaScore") +
  xlab("MetaScore") + ylab("Units Total") +
  theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))

#Sales with user score
salesmeetascore <- ggplot(data = newdata, aes(x = newdata$user_score, y = newdata$units_total)) +
  geom_point(color = "cornflowerblue")

salesmeetascore + ggtitle("Total Units Sold by UserScore") +
  xlab("UserScore") + ylab("Units Total") +
    theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))

#total units sold with esrb
salesesrb <- ggplot(data = esrb_game_sales, aes(x = esrb, y = Units_Sold)) +
    geom_bar(stat = "identity", fill = "#FF6666")

salesesrb + ggtitle("Total Units Sold by ESRB") +
    xlab("ESRB") + ylab("Units Total") +
    theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))


#total units sold with title
salestitle <- ggplot(data = newdata, aes(x = newdata$title, y = newdata$units_total)) +
    geom_bar(stat = "identity", fill = "#FF6666")
salestitle + ggtitle("Total Units Sold by Title") +
    xlab("Title") + ylab("Units Total") +
    theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 6, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))
```

```
# top 10 total units sold with title
top10salestitle <- ggplot(data = top10_title, aes(x = title, y = units_total)) +
    geom_bar(stat = "identity", fill = "#FF6666")
top10salestitle + ggtitle("Top 10 Total Units Sold by Title") +
    xlab("Title") + ylab("Units Total") +
    theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 6, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))

#Bottom total units sold with title
bottomsalestitle <- ggplot(data = bottom10_title, aes(x = title, y = units_total)) +
    geom_bar(stat = "identity", fill = "#FF6666")
bottomsalestitle + ggtitle("Bottom 10 Total Units Sold by Title") +
    xlab("Title") + ylab("Units Total") +
    theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 6, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))

#Units total with console
salesconsole <- ggplot(data = newdata, aes(x = newdata$console, y = newdata$units_total)) +
    geom_bar(stat = "identity", fill = "#FF6666")
salesconsole + ggtitle("Total Units Sold by Console") +
    xlab("Console") + ylab("Units Total") +
    theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))

#sales publisher
salespublisher <- ggplot(data = newdata, aes(x = newdata$publisher, y = newdata$units_total)) +
    geom_bar(stat = "identity", fill = "#FF6666")
salespublisher + ggtitle("Total Units Sold by Publisher") +
    xlab("Publisher") + ylab("Units Total") +
    theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))

#sales publisher top 10
salespublishertopten <- ggplot(data = top10_publisher, aes(x = publisher, y = units_total)) +
    geom_bar(stat = "identity", fill = "#FF6666")
salespublishertopten + ggtitle(" Top 10 Total Units Sold by Publisher") +
    xlab("Publisher") + ylab("Units Total") +
    theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))

#sales publisher bottom 10
salespublishertopten <- ggplot(data = bottom10_publisher, aes(x = publisher, y = units_total)) +
    geom_bar(stat = "identity", fill = "#FF6666")
salespublishertopten + ggtitle("Bottom 10 Total Units Sold by Publisher") +
    xlab("Publisher") + ylab("Units Total") +
    theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))



#total sales with developer
salesdeveloper <- ggplot(data = newdata, aes(x = newdata$developer, y = newdata$units_total)) +
   geom_bar(stat = "identity", fill = "cornflowerblue")
salesdeveloper + ggtitle("Total Units Sold by Developer") +
   xlab("Developer") + ylab("Units Total") +
   theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))


# Top 10 total sales with developer
salesdeveloper <- ggplot(data = top10_developer, aes(x = developer, y = units_total)) +
   geom_bar(stat = "identity", fill = "cornflowerblue")
salesdeveloper + ggtitle("Top 10 Total Units Sold by Developer") +
   xlab("Developer") + ylab("Units Total") +
   theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))

#Bottom 10 total sales with developer
```

```
salesdeveloper <- ggplot(data = bottom10_developer, aes(x = developer, y = units_total)) +
    geom_bar(stat = "identity", fill = "cornflowerblue")
salesdeveloper + ggtitle("Total Units Sold by Developer") +
    xlab("Developer") + ylab("Units Total") +
    theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))


#Total Sales by Genre
salesgenre <- ggplot(data = newdata, aes(x = newdata$genre, y = newdata$units_total)) +
    geom_bar(stat = "identity", fill = "cornflowerblue")
salesgenre + ggtitle("Total Units Sold by Genre") +
    xlab("Genre") + ylab("Units Total") +
    theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))


#Sales by Chart week
saleschartweek <- ggplot(data = newdata, aes(x = newdata$chart_week, y = newdata$units_total)) +
    geom_bar(stat = "identity", fill = "cornflowerblue")
saleschartweek + ggtitle("Total Units Sold by Chart Week") +
    xlab("Chart Week") + ylab("Units Total") +
    theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))

#Sales by score_diff
salesscorediff <- ggplot(data = newdata, aes(x = newdata$score_diff, y = newdata$units_total)) +
   geom_bar(stat = "identity", fill = "#FF6666")
salesscorediff + ggtitle("Total Units Sold by Score Diff") +
   xlab("Score Diff") + ylab("Units Total") +
   theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))



salesmultiplayer <- ggplot(data = multi_count, aes(x = multiplayer, y = newdata$units_total)) +
   geom_bar(stat = "identity", fill = "#FF6666")
salesmultiplayer + ggtitle("Total Units Sold by Title") +
   xlab("Title") + ylab("Units Total") +
   theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 6, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))


####################################################################################
########
```

```
###############################STATISITCS
PlOTS#############################################################
#Scatterplots
#Units total plot
plot(newdata$units_total, main = "Total Sales Plot")
#metascore plot
plot(newdata$metascore, main = "MetaScore Plot")
#user score plot
plot(newdata$user_score_x10, main = "UserScore Plot")
#Score_diff plot
plot(newdata$score_diff, main = "Score_Diff Plot")
#Chart week plot
plot(newdata$chart_week, main = "Chart Week Plot")
##########################Boxplots
#boxplot All
boxplot(numeric, main = "Box Plot All")
#Metascore boxplot
boxplot(newdata$metascore, main = "MetaScore  Box Plot")
#units total sold boxplot
boxplot(newdata$units_total, main = "Total Sales  Box Plot")
#user score boxplot
boxplot(newdata$user_score_x10, main = "UserScore  Box Plot")
#score diff boxplot
boxplot(newdata$score_diff, main = "Score_Diff  Box Plot")
#Chart week boxplots
boxplot(newdata$chart_week, main = "chartweek  Box Plot")
#####Scatterplot pairs###################################
pairs(numeric, panel = panel.smooth)
###############################################QQPlots
#Units total
qqnorm(newdata$units_total, main = "Total sales QQ")
qqline(newdata$units_total, lty = 2)
shapiro.test(newdata$units_total)
#Metascore
qqnorm(newdata$metascore, main = "MetaScore QQ")
qqline(newdata$metascore, lty = 2)
shapiro.test(newdata$metascore)
#Userscore
qqnorm(newdata$user_score_x10, main = "UserScore QQ")
qqline(newdata$user_score_x10, lty = 2)
shapiro.test(newdata$user_score_x10)
#Score_diff
qqnorm(newdata$score_diff, main="Score_diff QQ")
qqline(newdata$score_diff, lty = 2)
shapiro.test(newdata$score_diff)
#Chart week
```

```
qqnorm(newdata$chart_week, main = "Chart Week QQ")
qqline(newdata$chart_week, lty = 2)
shapiro.test(newdata$chart_week)
######################################Histograms
hist(newdata$units_total, main = "Total Sales Histogram")
hist(newdata$metascore, main = "MetaScore Histogram")
hist(newdata$user_score_x10, main = "UserScore Histogram")
hist(newdata$score_diff, main = "Score_Diff Histogram")
hist(newdata$chart_week, main = "Chart Week Histogram")
##############################################################################################
#########

##############################################ADDITIONAL
PLOTS############################################
#Count of each ESRB
esrb_game_plot <- ggplot(esrb_game, aes(x = esrb, y = n)) +
  geom_bar(stat = "identity", fill = "#FF6666")
esrb_game_plot + ggtitle("Count of Games Per ESRB") +
  xlab("ESRB") + ylab("Total") +
  theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))
#Count of each console
Console_game_plot <- ggplot(Console_game, aes(x = console, y = n)) +
  geom_bar(stat = "identity", fill = "cornflowerblue")
Console_game_plot + ggtitle("Count of Games per Console") +
  xlab("Console") + ylab("Total") +
  theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))
#Count of each genre
```

```
Genre_game_plot <- ggplot(Genre_game, aes(x = genre, y = n)) +
  geom_bar(stat = "identity", fill = "cornflowerblue")
Genre_game_plot + ggtitle("Count of Games per Genre") +
  xlab("Genre") + ylab("Total") +
  theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),

#panel color
panel.background = element_rect(fill = "gray60"))

#Count of each publisher
publisher_game_plot <- ggplot(Publishier_game, aes(x = publisher, y = n)) +
  geom_bar(stat = "identity", fill = "#FF6666")
publisher_game_plot + ggtitle("Count of Games per Publisher") +
  xlab("Publisher") + ylab("Total") +
  theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 6, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))
#Count of each developer
developer_game_plot <- ggplot(developer_game, aes(x = developer, y = n)) +
  geom_bar(stat = "identity", fill = "cornsilk")
developer_game_plot + ggtitle("Count of Games per Developer") +
  xlab("Developer") + ylab("Units Total") +
  theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))
#correlation matrix/correlation plot
res <- cor(ompleterecords)
round(res, 2)
corrplot(res, type = "upper", order = "hclust",
        tl.col = "black", tl.srt = 45)
#Multiplayer Count
Multiplayer_game_plot <- ggplot(multi_count, aes(x = multiplayer, y = n)) +
   geom_bar(stat = "identity", fill = "#FF6666")
Multiplayer_game_plot + ggtitle("Count of Players in a game") +
   xlab("Number of Player") + ylab("Total") +
   theme(
#changes colors and fonts and size of axis labels
axis.title.x = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
axis.title.y = element_text(color = "white", size = 12, face = "bold", family = "Georgia"),
#changes colors and fonts and size of axis
axis.text.x = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
axis.text.y = element_text(color = "white", size = 10, face = "bold", family = "Arial"),
#background color
plot.background = element_rect(fill = "gray"),
#panel color
panel.background = element_rect(fill = "gray60"))
```

5. Descriptive Numeric.R

```
setwd("C:\\Users\\arkad\\OneDrive\\Desktop\\Assignments\\MSIS 5600  DSP\\FINAL
PROJECT")
int_data = read.csv("test.csv")
library("moments")
summary(int_data)
Position <- int_data$position
Unit_week <- int_data$units_week
Unit_total <- int_data$units_total
Chart_week <- int_data$chart_week
Metascore <- int_data$metascore
User_score <- int_data$user_score
User_score_x_10 <- int_data$user_score_x10
Score_diff <- int_data$score_diff
summary(int_data$position)
summary(int_data$units_week)
summary(int_data$units_total)
summary(int_data$chart_week)
summary(int_data$metascore)
summary(int_data$user_score)
summary(int_data$user_score_x10)
summary(int_data$score_diff)


skewness(Position)
skewness(Unit_week)
skewness(Unit_total)
skewness(Chart_week)
skewness(Metascore, na.rm = TRUE)
skewness(User_score, na.rm = TRUE)
skewness(User_score_x_10, na.rm = TRUE)
skewness(Score_diff, na.rm = TRUE)

kurtosis(Position)
kurtosis(Unit_week)
kurtosis(Unit_total)
kurtosis(Chart_week)
kurtosis(Metascore, na.rm = TRUE)
kurtosis(User_score, na.rm = TRUE)
kurtosis(User_score_x_10, na.rm = TRUE)
kurtosis(Score_diff, na.rm = TRUE)

sd(Position)
sd(Unit_week)
sd(Unit_total)
sd(Chart_week)
sd(Metascore, na.rm = TRUE)
sd(User_score, na.rm = TRUE)
sd(User_score_x_10, na.rm = TRUE)
sd(Score_diff, na.rm = TRUE)
```

6. CatDescriptiveStats.R

```
setwd("C:\\Users\\d_log\\OneDrive\\MSIS5600- data")

data_all = read.csv("C:\\Users\\d_log\\OneDrive\\MSIS5600-
data\\vgMetaFullCleanFinal.csv")

head(data_all)

nrow(data_all)
```

```
##mode for console
table(data_all$console)
prop.table(table(data_all$console))
round(100*prop.table(table(data_all$console)),1)

# mode for Genre
table(data_all$genre)
prop.table(table(data_all$genre))
round(100*prop.table(table(data_all$genre)),1)


# mode for ESRB
table(data_all$esrb)
prop.table(table(data_all$esrb))
round(100*prop.table(table(data_all$esrb)),1)

# mode for number of players
table(data_all$numPlayers)
prop.table(table(data_all$numPlayers))
round(100*prop.table(table(data_all$numPlayers)),1)




mode(data_all$console)
class(data_all$week_posted)
class(data_all$position)
class(data_all$title)
class(data_all$url_title)
levels(data_all$position)
```

7. VG_SummClassifer.py

```
8. import pandas as pd
9. pd.set_option('display.max_columns', 10)
10. import numpy as np
11. import matplotlib.pyplot as plt
12. import seaborn as sns
13. import os
14. import nltk
15. import regex
16.
17.
```

```
#nltk.download('punkt')
#nltk.download('averaged_perceptron_tagger')
#nltk.download('maxent_ne_chunker')
#nltk.download('words')


from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import NMF

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

from nltk.stem import PorterStemmer
from nltk import word_tokenize, pos_tag, ne_chunk
from nltk.chunk import conlltags2tree, tree2conlltags

from wordcloud import WordCloud, STOPWORDS

os.chdir(f"C:\\Users\Aaron\Google Drive\School Stuff\Fall 2019\Data Science Programming I\DSP1 Semester Project\Deliverable 1 Materials")

full_data = pd.read_csv("vgMetaFullCleanFinal.csv", encoding = "ISO-8859-1")

full_summaries = full_data['summary']

full_summaries_unique = pd.Series(full_data.summary.unique())


vectorizer = CountVectorizer(max_df = 0.8, min_df=5, stop_words='english')

#classification of all summaries
dtm = vectorizer.fit_transform(full_summaries.values.astype('U'))

LDA = LatentDirichletAllocation(n_components = 14, random_state=35)

LDA.fit(dtm)

for i, topic in enumerate(LDA.components_):
    print(f'Top 10 words for topic #{i}:')
    print([vectorizer.get_feature_names()[i]for i in topic.argsort()[-10:]])
    print('\n')

LDA_values = LDA.transform(dtm)
LDA_values.shape
```

```

full_data['LDAtopic'] = LDA_values.argmax(axis = 1)

values = full_data["LDAtopic"].value_counts().sort_index().to_frame()

#named entity recognition and word clouds, all summaries
textbank_full = ""
for summary in full_summaries:
    textbank_full = str(textbank_full) + " " + str(summary)



stopwords = set(STOPWORDS)
post1 = pos_tag(word_tokenize(textbank_full))
tree1 = ne_chunk(post1)

entityp_full = []
entityo_full = []
entityg_full = []
entitydesc_full = []

for x in str(tree1).split('\n'):
    if 'PERSON' in x:
        entityp_full.append(x)
    elif 'ORGANIZATION' in x:
        entityo_full.append(x)
    elif 'GPE' in x or 'GSP' in x:
        entityg_full.append(x)
    elif '/NN' in x:
        entitydesc_full.append(x)

people_words = ""
for x in entityp_full:
    people_words = people_words + " " + x

people_words = people_words.replace('PERSON', ")
people_words = regex.sub('[\\(|\\)]', ", people_words)
people_words = regex.sub('\\/[A-Z]{2,3}', ", people_words)

org_words = ""
for x in entityo_full:
    org_words = org_words + " " + x

org_words = org_words.replace('ORGANIZATION', ")
org_words = regex.sub('[\\(|\\)]', ", org_words)
org_words = regex.sub('\\/[A-Z]{2,3}', ", org_words)

```

```
loc_words = ""
for x in entityg_full:
    loc_words = loc_words + " " + x

loc_words = loc_words.replace('GPE', '')
loc_words = loc_words.replace('GSP', '')
loc_words = regex.sub('[\\(|\\)]', '', loc_words)
loc_words = regex.sub('\\/[A-Z]{2,3}', '', loc_words)

wordcloud_people = WordCloud(width = 800, height = 800,
                             background_color = 'white',
                             stopwords = stopwords,
                             min_font_size = 10).generate(people_words)

plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud_people)
plt.axis('off')
plt.tight_layout(pad = 0)

plt.show()

wordcloud_orgs = WordCloud(width = 800, height = 800,
                           background_color = 'white',
                           stopwords = STOPWORDS,
                           min_font_size = 10).generate(org_words)

plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud_orgs)
plt.axis('off')
plt.tight_layout(pad = 0)

plt.show()

wordcloud_loc = WordCloud(width = 800, height = 800,
                          background_color = 'white',
                          stopwords = STOPWORDS,
                          min_font_size = 10).generate(loc_words)

plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud_loc)
plt.axis('off')
plt.tight_layout(pad = 0)

plt.show()


#NER, unique summaries
```

```
textbank_unique = ""
for summary in full_summaries_unique:
    textbank_unique = str(textbank_unique) + " " + str(summary)


post2 = pos_tag(word_tokenize(textbank_unique))
tree2 = ne_chunk(post2)

entityp_unique = []
entityo_unique = []
entityg_unique = []
entitydesc_unique = []

for x in str(tree2).split('\n'):
    if 'PERSON' in x:
        entityp_unique.append(x)
    elif 'ORGANIZATION' in x:
        entityo_unique.append(x)
    elif 'GPE' in x or 'GSP' in x:
        entityg_unique.append(x)
    elif '/NN' in x:
        entitydesc_unique.append(x)

people_words_unique = ""
for x in entityp_unique:
    people_words_unique = people_words_unique + " " + x

people_words_unique = people_words_unique.replace('PERSON', ")
people_words_unique = regex.sub('[\\(|\\)]', ", people_words_unique)
people_words_unique = regex.sub('\\/[A-Z]{2,3}', ", people_words_unique)

org_words_unique = ""
for x in entityo_unique:
    org_words_unique = org_words_unique + " " + x

org_words_unique = org_words_unique.replace('ORGANIZATION', ")
org_words_unique = regex.sub('[\\(|\\)]', ", org_words_unique)
org_words_unique = regex.sub('\\/[A-Z]{2,3}', ", org_words_unique)

loc_words_unique = ""
for x in entityg_unique:
    loc_words_unique = loc_words_unique + " " + x

loc_words_unique = loc_words_unique.replace('GPE', ")
loc_words_unique = loc_words_unique.replace('GSP', ")
loc_words_unique = regex.sub('[\\(|\\)]', ", loc_words_unique)
loc_words_unique = regex.sub('\\/[A-Z]{2,3}', ", loc_words_unique)
```

```

wordcloud_people_unique = WordCloud(width = 800, height = 800,
                    background_color = 'white',
                    stopwords = stopwords,
                    min_font_size = 10).generate(people_words_unique)

plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud_people_unique)
plt.axis('off')
plt.tight_layout(pad = 0)

plt.show()

wordcloud_orgs_unique = WordCloud(width = 800, height = 800,
                  background_color = 'white',
                  stopwords = STOPWORDS,
                  min_font_size = 10).generate(org_words_unique)

plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud_orgs_unique)
plt.axis('off')
plt.tight_layout(pad = 0)

plt.show()

wordcloud_loc_unique = WordCloud(width = 800, height = 800,
                  background_color = 'white',
                  stopwords = STOPWORDS,
                  min_font_size = 10).generate(loc_words_unique)

plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud_loc_unique)
plt.axis('off')
plt.tight_layout(pad = 0)

plt.show()
```

8. CatDescriptiveStats.R

```
library(wordcloud)
library(udpipe)
library(lattice)
library(dplyr)
library(tidytext)
library(tm)
library(ggplot2)
library(stringr)
library(tidyverse)


library(SnowballC)
library(hunspell)


#ud_model = udpipe_download_model(language = "english")
#install.packages('textdata')

data("stop_words")

setwd("C:\\Users\\Aaron\\Google Drive\\School Stuff\\Fall 2019\\Data Science
Programming I\\DSP1 Semester Project\\Deliverable 1 Materials")
full_data <- read.csv("vgMetaFullCleanFinal.csv", header = TRUE)

#This script takes the unique summaries found in the data and performs a sentiment
analysis and constructs a word cloud


summaries <- as.data.frame(full_data$summary)
colnames(summaries) <- c("summary")
summaries$summary <- as.character(summaries$summary)
summaries_unique <- distinct(summaries, summary)

#Extract unique summaries into one block of text
summaries_full <- ""
for (i in c(1:length(summaries_unique$summary))) {
  summaries_full <- paste(summaries_full, summaries_unique$summary[i], sep = " ")
}

patterndigits <- '\\b[0-9]+\\b'

#remove stop words
tidy_summaries <- summaries_unique %>%
  unnest_tokens(word, summary) %>%
  anti_join(stop_words)

#remove spaces and digits
tidy_summaries$word <- tidy_summaries$word %>%
  str_replace_all(patterndigits, '') %>%
   tolower() %>%
   str_replace_all('[:space:]', '')

#remove blank words
tidy_summaries <- tidy_summaries %>%
   filter(!(word == ''))

#stem words
tidy_summaries <- tidy_summaries %>%
   mutate_at("word", funs(wordStem((.), language = "en")))

#find most commonly appearing words, while calculating the proportion of the
individual words to the total
tidy_summaries <- tidy_summaries %>%
   count(word) %>%
   arrange(desc(n)) %>%
   mutate(proportion = (n / sum(n) * 100))

plot <- ggplot(tidy_summaries, aes(x = proportion, y = word)) +
   geom_abline(color = "gray40", lty = 2) +
   geom_jitter(alpha = 0.1, size = 2.5, width = 0.3, height = 0.3) +
   geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5) +
   scale_color_gradient(limits = c(0, 0.001), low = "darkslategray4", high = "gray75")
+
   theme(legend.position = "none") +
   labs(y = 'Word', x = 'Proportion')

plot

#positive and negative sentiments from the NRC Lexicon
nrc_posneg <- get_sentiments('nrc') %>%
   filter(sentiment == 'positive' | sentiment == 'negative')

summary_posneg <- tidy_summaries %>%
   inner_join(nrc_posneg) %>%
   spread(sentiment, n, fill = 0) %>%
   mutate(contentment = positive - negative, linenumber = row_number()) %>%
   #sentiment acts as a binary variable, is it more positive or negative, this is
primarily used for the plot
   mutate(sentiment = (positive > 0)) %>%
   arrange(desc(contentment))
```

```
#Player is counted as negative, should be positive in this context
summary_posneg[summary_posneg$word == 'player',]$contentment = 304
summary_posneg[summary_posneg$word == 'player',]$negative = 0
summary_posneg[summary_posneg$word == 'player',]$positive = 304
summary_posneg[summary_posneg$word == 'player',]$sentiment = TRUE

contentplot_posneg <- ggplot(summary_posneg, aes(x = linenumber, y = contentment,
fill = sentiment)) +
  geom_col() +
  scale_x_continuous() +
  coord_flip() +
  labs(x = 'Word', y = 'Sentiment') +
  theme(legend.position = 'none',
        panel.grid = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank()) +
        ggtitle("Negativity/Positivity Sentiment")
        geom_hline(yintercept = 0, color = "black", size = 1)

nrc_joysad <- get_sentiments('nrc') %>%
  filter(sentiment == 'joy' | sentiment == 'sadness')

summary_joysad <- tidy_summaries %>%
  inner_join(nrc_joysad) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(contentment = joy - sadness, linenumber = row_number()) %>%
  mutate(sentiment = (joy > 0)) %>%
  arrange(desc(contentment))

contentplot_joysad <- ggplot(summary_joysad, aes(x = linenumber, y = contentment,
fill = sentiment)) +
  geom_col() +
  scale_x_continuous() +
  coord_flip() +
  labs(x = 'Word', y = 'Sentiment') +
  theme(legend.position = 'none',
        panel.grid = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank()) +
        ggtitle("Sadness/Joy Sentiment") +
        geom_hline(yintercept = 0, color = "black", size = 1)


#wordcloud
corp <- Corpus(VectorSource(summaries_full))
corp <- corp %>%
   tm_map(content_transformer(tolower)) %>%
   tm_map(removeNumbers) %>%
   tm_map(removeWords, stopwords("english")) %>%
   tm_map(removePunctuation) %>%
   tm_map(stripWhitespace)

tdm <- TermDocumentMatrix(corp)
m <- as.matrix(tdm)
v <- sort(rowSums(m), decreasing = TRUE)
d <- data.frame(word = names(v), freq = v)

d <- d %>%
   anti_join(stop_words)

set.seed(1234)
wordcloud(words = d$word, freq = d$freq, min.freq = 1,
        max.words = 200, random.order = FALSE, rot.per = 0.35,
        colors = brewer.pal(8, "Dark2"))
```

9. NerCounts.R

```
library(tidyverse)
library(NLP)
library(openNLP)

full_data <- read.csv("C:\\Users\\Aaron\\Google Drive\\School Stuff\\Fall 2019\\Data
Science Programming I\\DSP1 Semester Project\\Deliverable 1
Materials\\vgMetaFullCleanFinal.csv", TRUE, stringsAsFactors = FALSE)

full_summaries <- full_data$summary


#Performing NER on all summaries results in a vector that is too big for my computers
to handle
full_summaries_unique <- as.String(unique(full_summaries))
```

```
sent_token_annotator = Maxent_Sent_Token_Annotator()
word_token_annotator = Maxent_Word_Token_Annotator()
pos_tag_annotator = Maxent_POS_Tag_Annotator()



entity_annotator_person = Maxent_Entity_Annotator(kind = 'person')
entity_annotator_organization = Maxent_Entity_Annotator(kind = 'organization')
entity_annotator_location = Maxent_Entity_Annotator(kind = 'location')

anno1 = annotate(full_summaries_unique, list(sent_token_annotator,
word_token_annotator))


anno2_person = annotate(full_summaries_unique, entity_annotator_person, anno1)
anno3_person = subset(anno2_person, type == "entity")

entities_person = data.frame(full_summaries_unique[anno3_person])
colnames(entities_person) = c("entity")

anno2_org = annotate(full_summaries_unique, entity_annotator_organization, anno1)
anno3_org = subset(anno2_org, type == "entity")

entities_org = data.frame(full_summaries_unique[anno3_org])
colnames(entities_org) = c("entity")

anno2_loc = annotate(full_summaries_unique, entity_annotator_location, anno1)
anno3_loc = subset(anno2_loc, type == "entity")

entities_loc = data.frame(full_summaries_unique[anno3_loc])
colnames(entities_loc) = c("entity")


entities_person %>%
   count(entity, sort = TRUE)

entities_org %>%
   count(entity, sort = TRUE)

entities_loc %>%
count(entity, sort = TRUE)
```