THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

Technical Report

The Turtle Games Case Study as an example to demonstrate the analytics capabilities by using the Jupyter Notebook with Python Programming Language and RStudio with RScript

Vadims Suharevs

London. December 23, 2022

Contents

Background

The client (Turtle Games) have instructed the team that there is a requirement to conduct an analysis into its client base including the loyalty points accumulation, gaming patterns and sales by region. The exploratory analysis should give more information into the consumer behaviour trends and buying patterns for a potential marketing campaigns to target the right audience.

Areas of the analysis would consider spending, loyalty points accumulation, natural language processing and sales patterns.

The body of this technical document does not answer the business questions (*please refer to the enclosed PDF version of the PowerPoint Presentation or go to the **Appendix 1** of this report for business solutions and recommendations*).

Methodology

As per the academic requirement the report is a technical step by step document explaining the approach towards implementing the analysis. All the actual insights would be referenced on the relevant appendix.

Please note: The workflow of this report would be closely following the structure of the Jupiter Notebook and RStudio Terminal.

Prework

The team had been provided with the datasets to explore the problem and make recommendations.

For the purposes of this analytics the project the team of analysts would be using Python and RScript environment via the Jupiter Notebook and RStudio with various libraries to expand and support the analysis. The workbook would be saved on the GitHub repository (please follow the link: https://github.com/CTPATEG/Suharevs_Vadims_DA301_Assignment).

1 Datasets and Data Cleaning

1.1 Datasets, Jupyter Notebook & RStudio

The Jupiter notebook analytics flow is structured in the way that all the activities can be run step by step from start to finish without having the need to go back to the prior sections. At the beginning of each section all the necessary libraries would be imported into the Jupiter Notebook and RStudio so that if any extra analysis is required within a specific section – the library is already there and ready for use ***(Figure 1, Figure 2)***.

```python
# Imports
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import seaborn as sns
import statsmodels.api as sm
import statsmodels.stats.api as sms
import sklearn
from statsmodels.formula.api import ols
from statsmodels.stats.outliers_influence import variance_inflation_factor

from sklearn import datasets
from sklearn import metrics
from sklearn import linear_model
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

# Ignore warnings:
import warnings
warnings.filterwarnings('ignore')
```

Figure 1. Importing Libraries into Jupiter Notebook

```r
# 0. Working Directory

# 0.1 Determine the working directory.
getwd()

# 0.2 Select the current directory ('#' for the purposes of the submission).
# setwd(dir = '/Users/VS/Desktop/LSE/3 Course/Assignment 3/Data')

# 0.3 Double-check the working directory.
getwd()

# 1. Load and explore the data

# Install and import Tidyverse.
install.packages('tidyverse')
install.packages('tidyr')

# Import package.
suppressWarnings(library(tidyverse))
# library(tidyverse)
```

Figure 2. Importing libraries into RStudio

Then the analysis took to the exploration of the datasets given to the project team using both Python and RStudio (***Table 1, Table 2***).

Table 1. Data import using Python via Jupiter Notebook

```
# Load the CSV file(s) as reviews.
reviews = pd.read_csv('turtle_reviews.csv')

# View the DataFrame.
reviews.head()
```

	gender	age	remuneration (k£)	spending_score (1-100)	loyalty_points	education	language	platform	product	review	summary
0	Male	18	12.30	39	210	graduate	EN	Web	453	When it comes to a DM's screen, the space on t...	The fact that 50% of this space is wasted on a...
1	Male	23	12.30	81	524	graduate	EN	Web	466	An Open Letter to GaleForce9*:\n\nYour unpaint...	Another worthless Dungeon Master's screen from...
2	Female	22	13.12	6	40	graduate	EN	Web	254	Nice art, nice printing. Why two panels are f...	pretty, but also pretty useless
3	Female	25	13.12	77	562	graduate	EN	Web	263	Amazing buy! Bought it as a gift for our new d...	Five Stars
4	Female	33	13.94	40	366	graduate	EN	Web	291	As my review of GF9's previous screens these w...	Money trap

Table 2. Data Import using RScript via RStudio

```
> head(turtle_sales)
  Ranking Product Platform Year         Genre Publisher NA_Sales EU_Sales Global_Sales
1       1     107      Wii 2006        Sports  Nintendo    34.02    23.80        67.85
2       2     123      NES 1985      Platform  Nintendo    23.85     2.94        33.00
3       3     195      Wii 2008        Racing  Nintendo    13.00    10.56        29.37
4       4     231      Wii 2009        Sports  Nintendo    12.92     9.03        27.06
5       5     249       GB 1996 Role-Playing  Nintendo     9.24     7.29        25.72
6       6     254       GB 1989        Puzzle  Nintendo    19.02     1.85        24.81
```

In order to then explore the datasets various commands have been used on Jupiter Notebook including *".info()"*, *".value_counts()"*, *".describe()"* to understand the data types, the overall composition of the datasets and generate descriptive statistics (***Table 3***). For the similar activities on RStudio the *"str()"*, *"glimpse()"*, *"typeof()"*, *"class()"*, *"dim()"*, *"as_tibble()"* and *"summary()"*commands have been used (please refere to the "RScript.R" file attached to the submission (***Table 4***).

Table 3. Descriptive statistics for reviews dataset (Python)

```
# Descriptive statistics.
reviews.describe()
```

	age	remuneration (k£)	spending_score (1-100)	loyalty_points	product
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	39.495000	48.079060	50.000000	1578.032000	4320.521500
std	13.573212	23.123984	26.094702	1283.239705	3148.938839
min	17.000000	12.300000	1.000000	25.000000	107.000000
25%	29.000000	30.340000	32.000000	772.000000	1589.250000
50%	38.000000	47.150000	50.000000	1276.000000	3624.000000
75%	49.000000	63.960000	73.000000	1751.250000	6654.000000
max	72.000000	112.340000	99.000000	6847.000000	11086.000000

Table 4. Descriptive statistics for turtle sales dataset (RStudio)

```
> # View a summary (descriptive statistics) of the data frame.
> summary(turtle_sales)
    Ranking          Product       Platform             Year         Genre             Publisher
 Min.   :    1.00   Min.   : 107   Length:352         Min.   :1982   Length:352         Length:352
 1st Qu.:   88.75   1st Qu.:1945   Class :character   1st Qu.:2003   Class :character   Class :character
 Median :  176.50   Median :3340   Mode  :character   Median :2009   Mode  :character   Mode  :character
 Mean   : 1428.02   Mean   :3607                      Mean   :2007
 3rd Qu.: 1439.75   3rd Qu.:5436                      3rd Qu.:2012
 Max.   :16096.00   Max.   :9080                      Max.   :2016
                                                      NA's   :2
    NA_Sales          EU_Sales       Global_Sales
 Min.   : 0.0000   Min.   : 0.000   Min.   : 0.010
 1st Qu.: 0.4775   1st Qu.: 0.390   1st Qu.: 1.115
 Median : 1.8200   Median : 1.170   Median : 4.320
 Mean   : 2.5160   Mean   : 1.644   Mean   : 5.335
 3rd Qu.: 3.1250   3rd Qu.: 2.160   3rd Qu.: 6.435
 Max.   :34.0200   Max.   :23.800   Max.   :67.850
```

1.2 Cleaning Datasets (Jupyter Notebook and RStudio)

After having explored the datasets, the next step is to clean the data and get the datasets ready for the initial insights and views. This includes and is not limited to removing the redundant columns "*language*" and "*platform*" for the review dataset using "*.drop()*" function via Python (***Table 5***) as well as "*select()*" function on RStudio with the "-" sign to remove the columns (***Table 6***).

Table 5. Drop columns, create new data frame (Python)

```
# Drop unnecessary columns.
reviews = reviews.drop(['language', 'platform'], axis=1)

# View column names.
reviews.head()
```

	gender	age	remuneration (k£)	spending_score (1-100)	loyalty_points	education	product	review	summary
0	Male	18	12.30	39	210	graduate	453	When it comes to a DM's screen, the space on t...	The fact that 50% of this space is wasted on a...
1	Male	23	12.30	81	524	graduate	466	An Open Letter to GaleForce9*:\n\nYour unpaint...	Another worthless Dungeon Master's screen from...
2	Female	22	13.12	6	40	graduate	254	Nice art, nice printing. Why two panels are f...	pretty, but also pretty useless
3	Female	25	13.12	77	562	graduate	263	Amazing buy! Bought it as a gift for our new d...	Five Stars
4	Female	33	13.94	40	366	graduate	291	As my review of GF9's previous screens these w...	Money trap

Table 6. Drop columns, create new data frame (RStudio)

```
> # Create a new data frame from a subset of the sales data frame.
> # Remove unnecessary columns (Ranking, Year, Genre, Publisher).
> turtle_sales_new <- select(turtle_sales,
+                            -Ranking, -Year, -Genre, -Publisher)
> # View the new data frame.
> head(turtle_sales_new)
  Product Platform NA_Sales EU_Sales Global_Sales
1     107      Wii    34.02    23.80        67.85
2     123      NES    23.85     2.94        33.00
3     195      Wii    13.00    10.56        29.37
4     231      Wii    12.92     9.03        27.06
5     249       GB     9.24     7.29        25.72
6     254       GB    19.02     1.85        24.81
```

After having removed the unnecessary columns – the newly created subsets of data have been saved for future use using "*.to_csv()*" command via Python and "*write_csv()*" on RStudio after having changed the Product variable from numeric to character using the "mutate()" function and validating that there are no missing values.

Source file: "turtle_reviews.csv" ➔

Output files: "reviews.csv", "df3_summary.csv", "df3_review.csv"

Source file: "turtle_sales.csv" ➔

Output files: "turtle_sales_new.csv" ➔ "turtle_sales_group.csv" ➔ turtle_sales_filtered.csv"

2 Data Exploration (Reviews) - Python

2.1 Data Exploration (Linear Regression)

The next step is to slice up the "***reviews.csv***" dataset to get it ready for running the linear regressions and multiple linear regression (***Table 7***).

Table 7. Multiple linear regression ready dataset (Python)

```
# Split out the DataFrame - Loyalty-SPending-Age-Remuneration
reviews_mlr = reviews[['loyalty_points', 'spending_score', 'age', 'remuneration']]

# View DataFrame
reviews_mlr.head()
```

	loyalty_points	spending_score	age	remuneration
0	210	39	18	12.30
1	524	81	23	12.30
2	40	6	22	13.12
3	562	77	25	13.12
4	366	40	33	13.94

The subsets of data are ready for running the simple linear regressions and multiple linear regressions. The insights into the ***Spending Score vs Loyalty***, ***Remuneration vs Loyalty***, ***Age vs Loyalty*** and ***Multiple Linear Regression*** would be communicated on the presentation and referenced on the **Appendix 1**. Example of the code and the visualisation can be evidenced on the (***Figure 3***).

```
# Extract the estimated parameters.
print("Parameters: ", test.params)

# Extract the standard errors.
print("Standard errors: ", test.bse)

# Extract the predicted values.
print("Predicted values: ", test.predict())
```

```
Parameters:  Intercept   -75.052663
x            33.061693
dtype: float64
Standard errors:  Intercept    45.930554
x             0.814419
dtype: float64
Predicted values:  [1214.35337415 2602.94449102  123.31749662 ... 2933.56142361  453.93442921
  189.44088314]
```

```
# Set the X coefficient and the constant to generate the regression table.
y_pred = (-75.052663) + 33.061693 * reviews_spend['spending_score']

# View the output.
y_pred
```

Figure 3. Linear regression testing parameters for Spending Score vs Loyalty.

```
# Plot the graph with a regression line.
# Plot the data points with a scatterplot.
plt.scatter(x, y)

# Plot the regression line (in black).
plt.plot(x, y_pred, color='red')

# Set the x and y limits on the axes.
plt.xlim(0)
plt.ylim(0)

# View the plot.
plt.show()
```

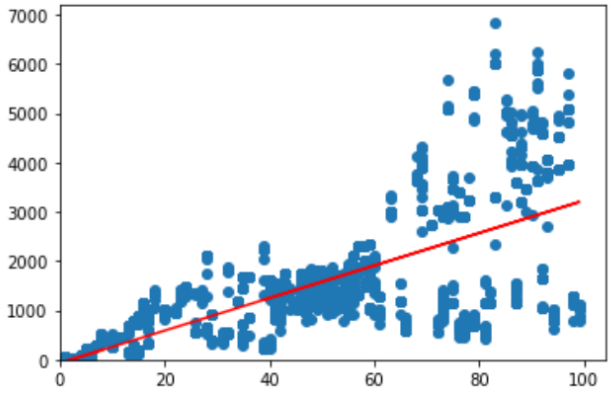


Figure 4. Fitted Scatter plot and Regression line for Spending Score vs Loyalty.

```python
# Training the model using the 'statsmodel' OLS library.
# Fit the model with the added constant.
model = sm.OLS(y_train, sm.add_constant(x_train)).fit()

# Set the predicted response vector.
Y_pred = model.predict(sm.add_constant(x_test))

# Call a summary of the model.
print_model = model.summary()

# Print the summary.
print(print_model)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:         loyalty_points   R-squared:                       0.844
Model:                            OLS   Adj. R-squared:                  0.843
Method:                 Least Squares   F-statistic:                     2513.
Date:                Thu, 22 Dec 2022   Prob (F-statistic):               0.00
Time:                        11:59:32   Log-Likelihood:                -10711.
No. Observations:                1400   AIC:                         2.143e+04
Df Residuals:                    1396   BIC:                         2.145e+04
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          -2185.5298     61.684    -35.431      0.000   -2306.534   -2064.526
spending_score    33.7217      0.538     62.629      0.000      32.666      34.778
remuneration      34.4787      0.591     58.304      0.000      33.319      35.639
age               10.7664      1.027     10.485      0.000       8.752      12.781
==============================================================================
Omnibus:                       10.911   Durbin-Watson:                   2.040
Prob(Omnibus):                  0.004   Jarque-Bera (JB):               11.554
Skew:                           0.172   Prob(JB):                      0.00310
Kurtosis:                       3.283   Cond. No.                         375.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

```python
print(mlr.score(x_train, y_train)*100)
```

```
84.36143267390747
```

Figure 5. Multiple Linear Regression and Code associated with it (R-Squared 84% Train, 83% Test)

The Multiple Linear regression model proved to be reliable as the VIF factors for all the variables are around 1 (***Table 8***).

Table 8. VIF Factor for the investigated variables

VIF Factor	features
20.5	const
1.1	spending_score
1.0	remuneration
1.1	age

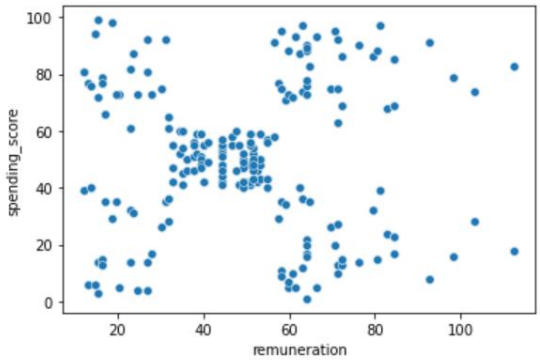
2.2 Data Exploration (K-Means Clustering)

Turtle Games marketing department is eager to understand better the relationship between **Remuneration** and **Spending Score**. Currently Spending Score vs Loyalty Points is out of scope, even though potentially it could yield further targeting and segmentation insights.

Even though the initial plotting exercise does give a form of what to look out for, it would make sense to further evaluate the data (***Figure 6***). In order to deep dive further the K-Means Clustering approach would be used.

```
# Create a scatterplot with Seaborn.
sns.scatterplot(x='remuneration',
                y='spending_score',
                data=df2)
```

```
<AxesSubplot:xlabel='remuneration', ylabel='spending_score'>
```




```
# Create a pairplot with Seaborn.
x = df2[['remuneration', 'spending_score']]

sns.pairplot(df2, vars=x, diag_kind='kde')
```

```
<seaborn.axisgrid.PairGrid at 0x217a1446610>
```

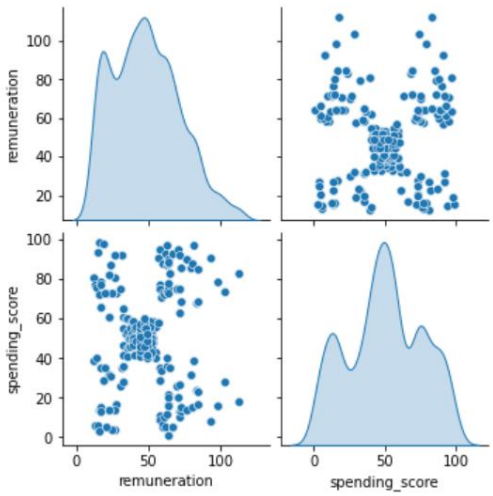


Figure 6. Spending Score vs Remuneration initial evaluation

To better understand the clustering of the dataset it is advisable to use the “Elbow and Silhouette” methods to distinguish the number of clusters and separate them out for further targeting. The method shows 5 as an objective number of clusters (Figure 7).

```python
# Determine the number of clusters: Elbow method.
ss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i,
                    init='k-means++',
                    max_iter=300,
                    n_init=10,
                    random_state=0)
    kmeans.fit(x)
    ss.append(kmeans.inertia_)

# Plot the elbow method.
plt.plot(range(1, 11),
         ss,
         marker='o')

# Insert labels and title.
plt.title("The Elbow Method")
plt.xlabel("Number of clusters")
plt.ylabel("SS distance")

plt.show()
```

```python
# Determine the number of clusters: Silhouette method.
sil = []
kmax = 10

for k in range(2, kmax+1):
    kmeans_s = KMeans(n_clusters=k).fit(x)
    labels = kmeans_s.labels_
    sil.append(silhouette_score(x,
                                labels,
                                metric='euclidean'))

# Plot the silhouette method.
plt.plot(range(2, kmax+1),
         sil,
         marker='o')

# Insert labels and title.
plt.title("The Silhouette Method")
plt.xlabel("Number of clusters")
plt.ylabel("Sil")

plt.show()
```

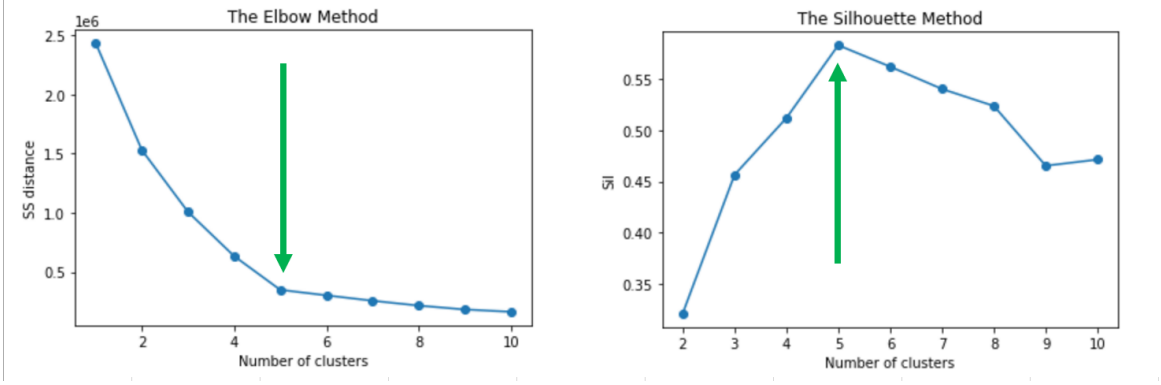


Figure 7. Elbow and Silhouette method in practice

After having run several iterations of clustering between 3 and 6 clusters it had been determined that 5 is the objective number and the final Clustering Model had been plotted (***Figure 8***).

```
# Use five clusters.
kmeans = KMeans(n_clusters = 5,
                max_iter = 15000,
                init='k-means++',
                random_state=0).fit(x)

clusters = kmeans.labels_

x['K-Means Predicted'] = clusters

# Plot the predicted.
sns.pairplot(x,
             hue='K-Means Predicted',
             diag_kind= 'kde')
```

```
<seaborn.axisgrid.PairGrid at 0x1ee72226460>
```

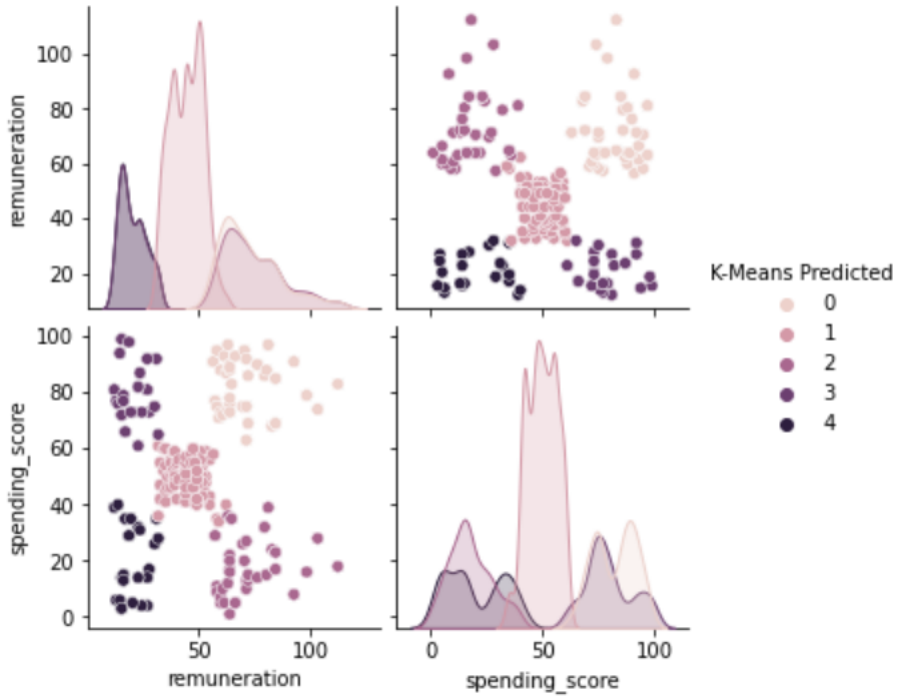


Figure 8. K-Means Clustering - final model validation using 5 clusters

To enhance the visuals we can clearly see 5 clusters separate from one another (***Figure 9***).

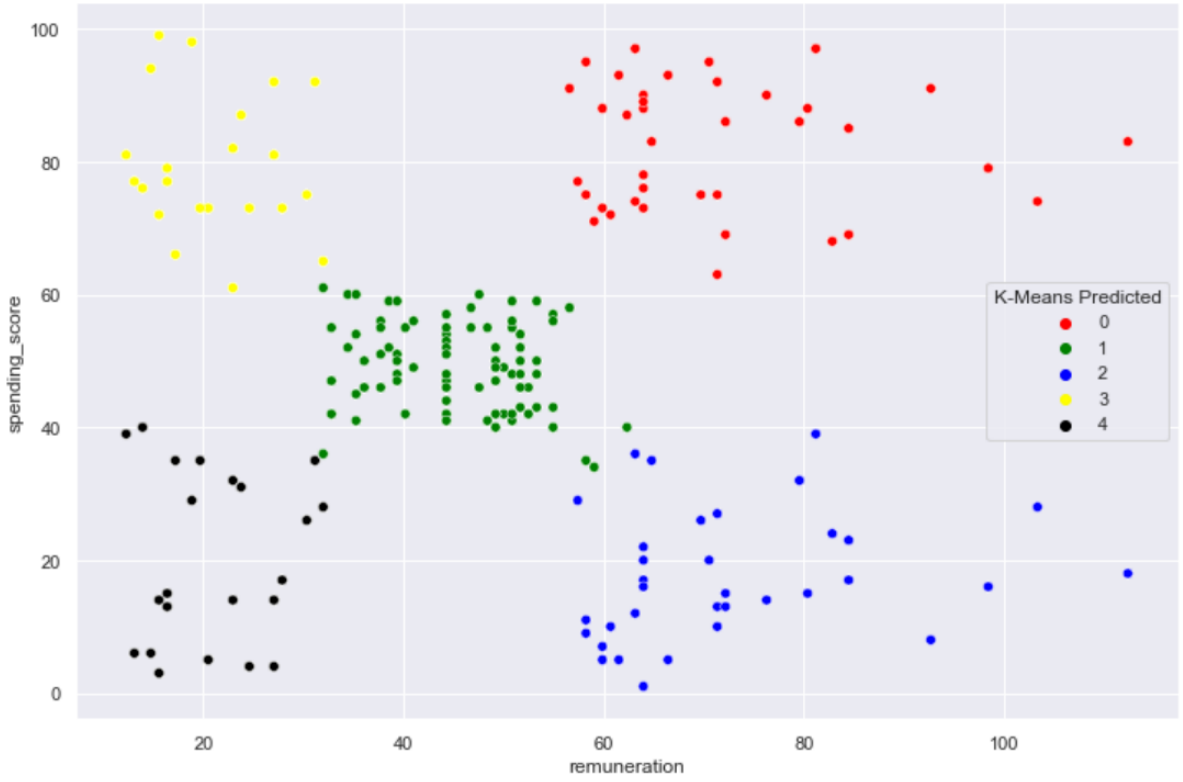


Figure 9. Five clusters clearly visible

Insights and observations related to the findings above would be discussed during the presentation and within the **Appendix 1**.

2.3 Data Exploration (Natural Language Processing)

In order to assess the voice of the customer closer it had been decided to use the Natural Language Processing capabilities of Python which would require additional Libraries to be introduced (***Figure 10***).

```
# Import all the necessary packages.
import pandas as pd
import numpy as np
import nltk
import os
import matplotlib.pyplot as plt
import seaborn as sns

# nltk.download ('punkt').
# nltk.download ('stopwords').

from wordcloud import WordCloud
from nltk.tokenize import word_tokenize
from nltk.tokenize import RegexpTokenizer
from nltk.probability import FreqDist
from nltk.corpus import stopwords
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from textblob import TextBlob
from scipy.stats import norm
from sklearn.feature_extraction.text import CountVectorizer

# Import Counter.
from collections import Counter

import warnings
warnings.filterwarnings('ignore')
```

Figure 10. Importing Python libraries to aid with Natural Language Processing

To follow along with the natural language processing analysis only the essential columns would be left on the dataset ("*review*" and "*summary*"), all the other columns from **turtle_reviews.csv** have been removed. Each of the columns ("*review*" and "*summary*") would be assessed separately, therefore fer the experiment purposes the data frames for each would be separated, treated in parallel and duplicate values would also be treated separately (50 duplicate rows would be dropped on ***df3_review***, and 649 duplicates dropped on ***df3_summary***) (***Figure 11***).

```
# Check the number of duplicate values in the review column.
df3.review.duplicated().sum()
```

```
50
```

```
# Check the number of duplicate values in the summary column.
df3.summary.duplicated().sum()
```

```
649
```

Figure 11. Duplicates on review column and summary column

The "*review*" column analysis would be used as an example for the purposes of this analytics report, the insights for both columns would be included in the final presentation and the **Appendix 1 and** (***Figure 12***).

Change all to Lower Case

```
# Review: Change all to lower case and join with a space.
# Transform data to lowercase.
df3['review'] = df3['review'].apply(lambda x: " ".join(x.lower() for x in x.split()))

# Preview the result.
df3['review'].head()
```

```
0    when it comes to a dm's screen, the space on t...
1    an open letter to galeforce9*: your unpainted ...
2    nice art, nice printing. why two panels are fi...
3    amazing buy! bought it as a gift for our new d...
4    as my review of gf9's previous screens these w...
Name: review, dtype: object
```

Replace all punctuation

```
# Replace all the punctuations in review column.
df3['review'] = df3['review'].str.replace('[^\w\s]','')

# View output.
df3['review'].head()
```

```
0    when it comes to a dms screen the space on the...
1    an open letter to galeforce9 your unpainted mi...
2    nice art nice printing why two panels are fill...
3    amazing buy bought it as a gift for our new dm...
4    as my review of gf9s previous screens these we...
Name: review, dtype: object
```

Drop duplicates in case this is a genuine error

```
# Check the number of duplicate values in the review column.
df3.review.duplicated().sum()
```

```
50
```

Figure 12. Change the words to lower case, remove punctuation and duplicates

Then the data is tokenised and a word cloud is created where it can be seen that the stop words are still there (***Figure 13***).

```
# Review: Plot the WordCloud image.
# Plot the WordCloud image.
plt.figure(figsize = (16, 9), facecolor = None)
plt.imshow(word_cloud_reviews)
plt.axis('off')
plt.tight_layout(pad = 0)
plt.show()

# Note that your word cloud might differ slightly from the one provided.
```

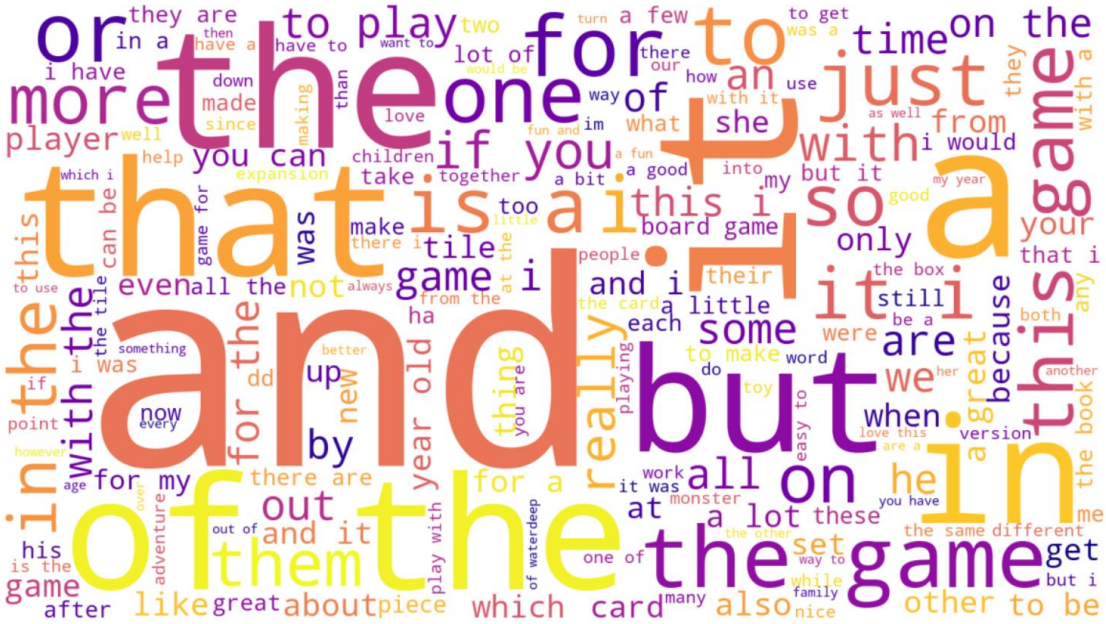


Figure 13. Word cloud with stop words.

In order assess the magnitude of the impact from the stop words the next step is to analyse the frequency distribution and polarity of words within the "*review*" column and then remove stop words and alphanumeric characters that are not adding value to text to generate a clean word cloud (***Figure 14***).

Figure 14. Clean word cloud after removing stop words and alphanumeric characters.

The next step is to visualise the most frequently used words (***Figure 15***).

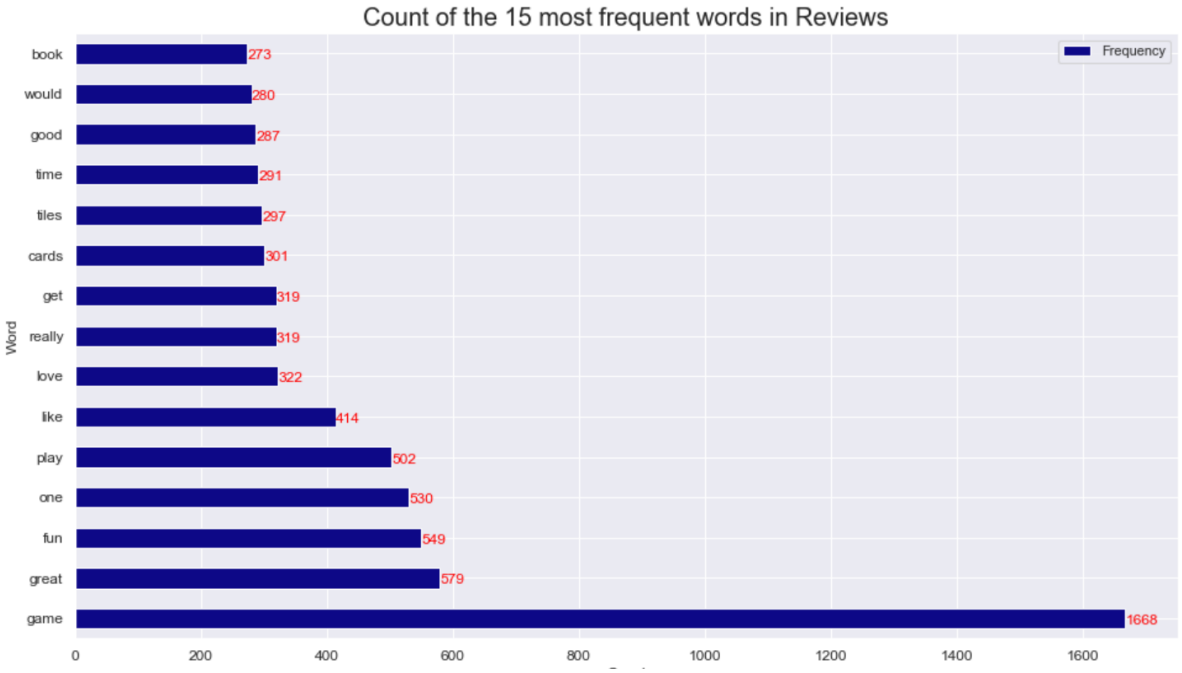


Figure 15. Most frequently used words on Reviews

Then the researcher would need to understand whether the words being used on the reviews are positive or negative and where on the scale the overall polarity towards the product is placed. This is done by creating a histogram for polarity with the score of -1/+1 and visualising the polarity (***Figure 16***).

```python
# Review: Create a histogram plot with bins = 15.
# Histogram of polarity
# Set the number of bins.
num_bins = 15

# Set the plot area.
plt.figure(figsize=(16,9))

# Define the bars.
n, bins, patches = plt.hist(tokens_rev['polarity'], num_bins, facecolor='red', alpha=0.6)

# Set the labels.
plt.xlabel('Polarity', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.title('Histogram of sentiment score polarity for Reviews', fontsize=20)

plt.show()
```

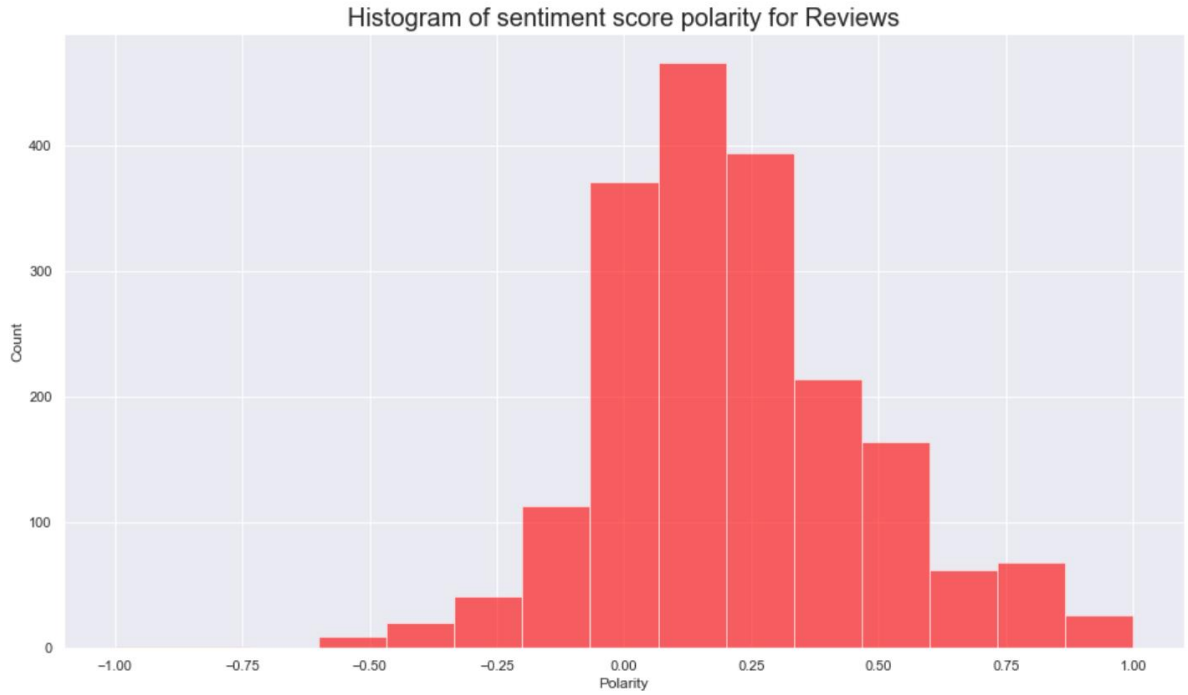


Figure 16. Creation of polarity histogram for Reviews

Then the most positive reviews can be investigated further, as well as the **vector analysis** (***Figure 17***) and **document term matrix** (***Figure 18***) can be used to further enhance the analysis.

```python
from spacy import displacy

# Iterate through a selection of comments.
for i in range(750,1750):
    # Check whether the corresponding comment has a named entity.
    if tokens_rev['named_entities'][i]:
        # Highlight the entity in the comment.
        displacy.render(nlp(tokens_rev['review'][i]), style='ent', jupyter=True)
```

this is not an actual judy clock which should have visible functioning gears maintain correct hourhand and minutehand relationships this clock has a single brad for the hour TIME and minutehands so they move independently of each other its fine as a simple nongeared clock its just not a judy clock

love this i am able to work one CARDINAL on one with a child that is struggling with mastering how to tell time

its small enough to fit in my three year DATE olds purse so she can take anywhere great product she did leave the marker open and it dried out but she had other markers that size so no biggie weve had this product for over four months DATE and its held up very very well its difficult to bend wont tear unless you use megaforce a wipe will clean off the dry erase perfectly

laminated and sturdy with clock hands that work well the marker is in great condition check the size it is small enough to fit in one hand i do not recommend this for teaching a group of children as the product is too small it works well for one CARDINAL on one CARDINAL

Figure 17. Vector word analysis

```
# Create a DataFrame.
dt = pd.DataFrame(cvs.todense()).iloc[:15]

# Name the columns.
dt.columns = cv.get_feature_names()

# Transpose columns and headings.
document_term_matrix = dt.T

# Update the column names.
document_term_matrix.columns = ['Doc '+str(i) for i in range(1, 16)]

# Get the totals.
document_term_matrix['total_count'] = document_term_matrix.sum(axis=1)

# Identify the top 10 words
document_term_matrix = document_term_matrix.sort_values(by ='total_count',
                                                        ascending=False)[:10]

# Display the results.
print(document_term_matrix.drop(columns=['total_count']).head(10))
```

```
            Doc 1  Doc 2  Doc 3  Doc 4  Doc 5  Doc 6  Doc 7  Doc 8  Doc 9  \
screen          2      2      2      0      0      0      1      0      1
adventure       0      2      4      0      0      0      0      0      0
map             0      1      3      0      0      0      0      0      1
useful          0      2      5      0      0      0      0      0      0
use             0      1      4      0      0      0      0      0      0
screens         0      2      2      0      1      0      1      0      0
dm              0      2      1      1      0      0      0      0      0
space           2      0      1      0      0      0      0      0      0
completely      2      0      0      0      1      0      0      0      0
dms             1      0      1      0      0      0      0      0      1
```

Figure 18. Document Term Matrix

The insights with regard to the natural language processing would be communicated within the presentation, as we as available on the **Appendix 1**.

3 Data Exploration (Sales) – RScript

3.1 Data Exploration (Visualisations using “qplot()”)

The datasets used with the RScript file had been assessed and cleaned as per the methodology within the **Section 1 Datasets and Data Cleaning**. The “turtle_sales.csv” file had been treated and saved as the “turtle_sales_new.csv”.

Initial data exploration initiatives warrant checking out various combinations of variables on different plot types to understand the data as per examples below (***Figure 19-22***). The initial insights are available on the **Appendix 1** and within the supporting presentation.

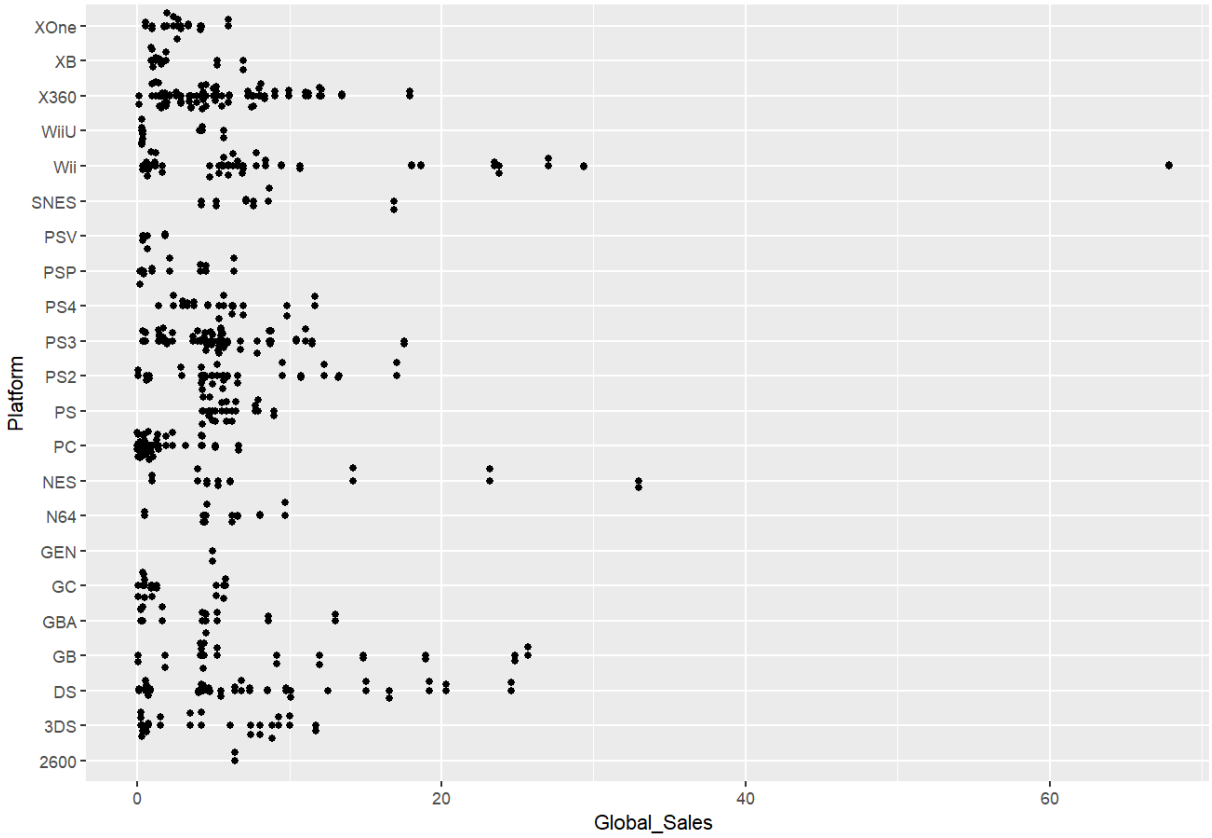


Figure 19. Platform by Global Sales Scatter visualisation

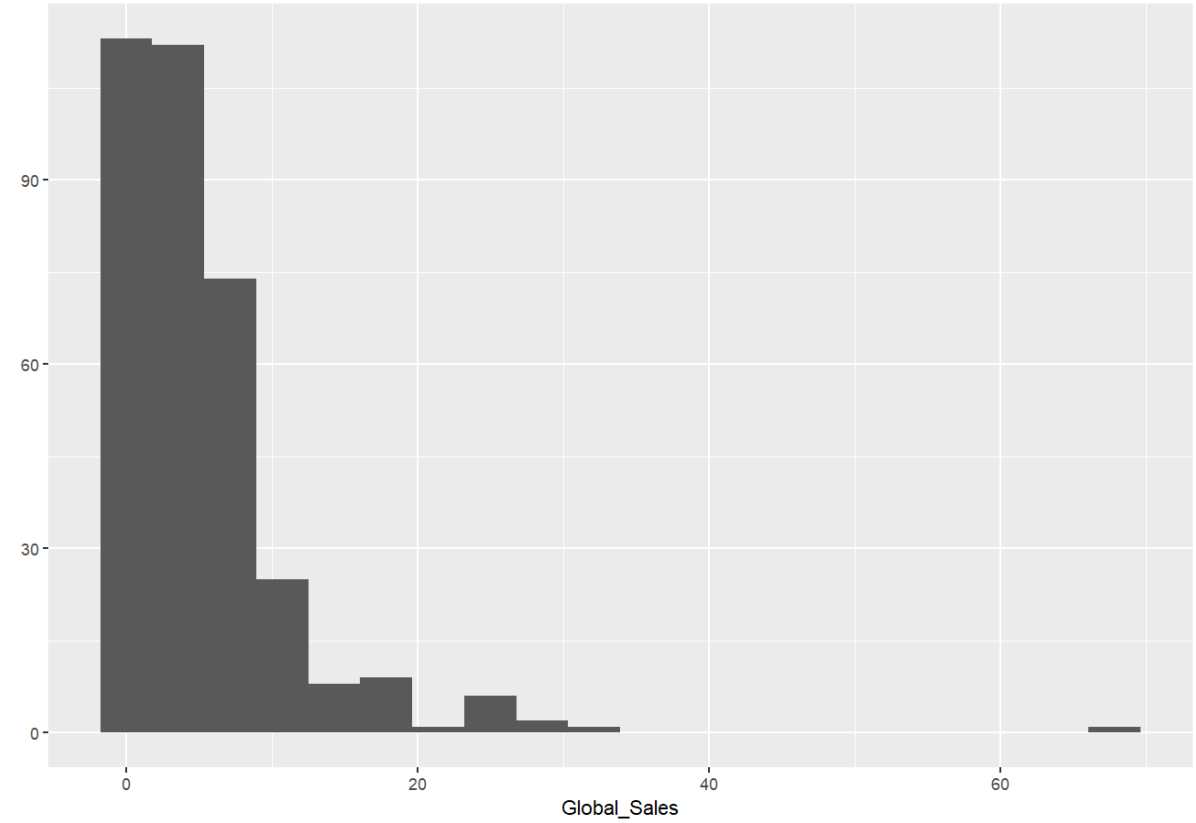


Figure 20. Global Sales Histogram to understand the shape of data

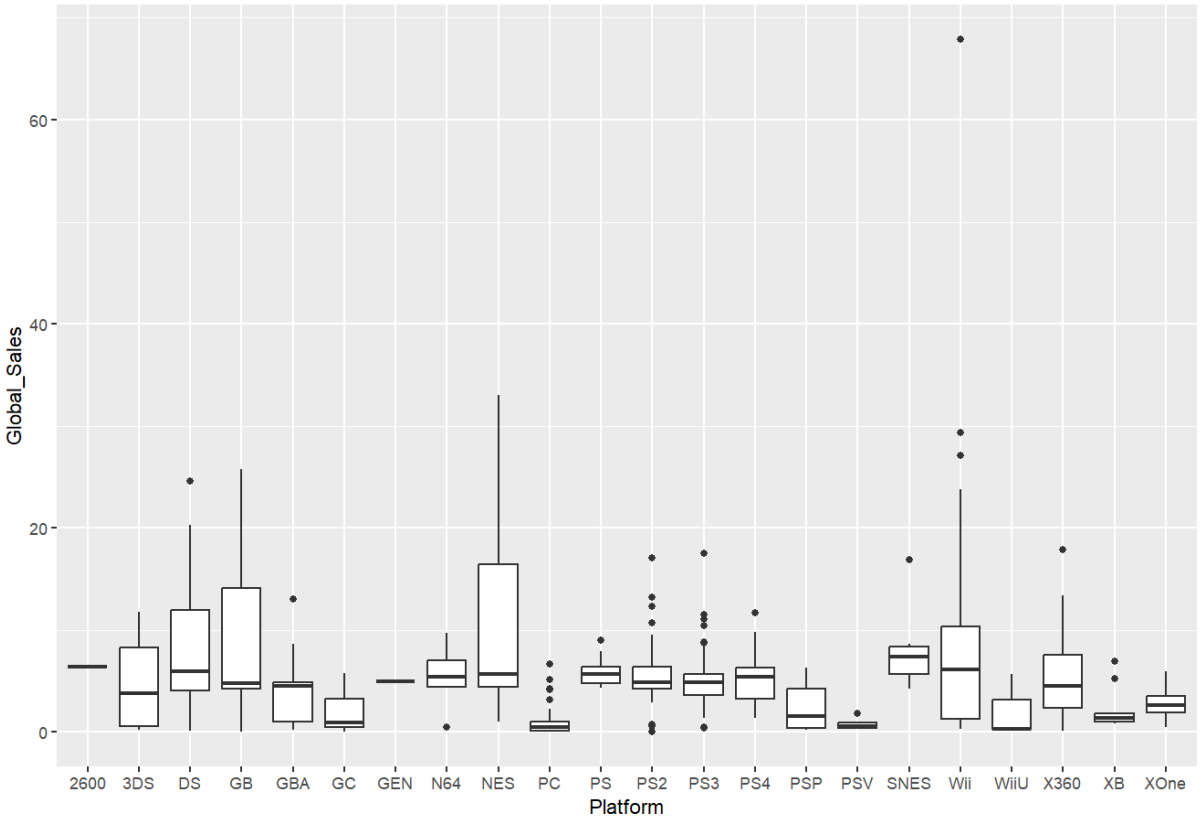


Figure 21. Boxplot Global sales by Platform to understand the sales by Platform

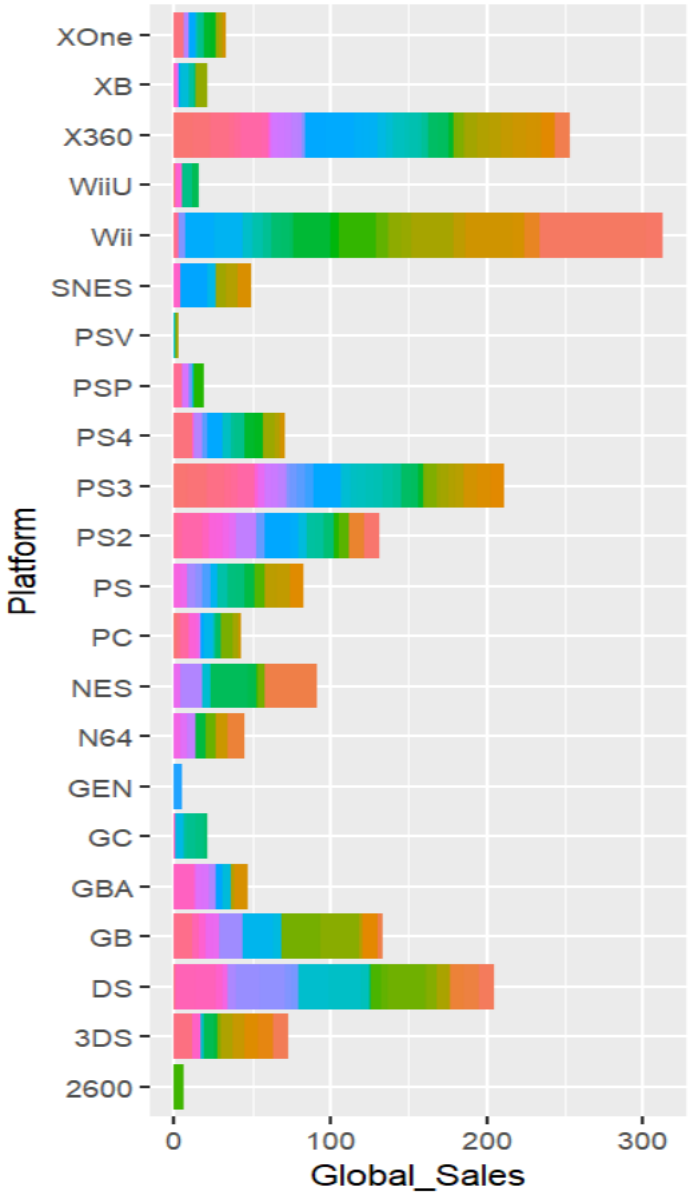


Figure 22. Bar plot for Global sales by Platform

3.2 Data Exploration (Enhanced Visuals using "ggplot()")

After seeing the initial insights and descriptive statistics (***Table 9***) the client had decided to remove the "*Console*" variable so an aggregation for the dataset was required.

Table 9. Descriptive statistics for the Sales dataset

```
> summary(turtle_sales_new)
   Product            Platform           NA_Sales          EU_Sales       Global_Sales
 Length:352         Length:352         Min.   : 0.0000   Min.   : 0.000   Min.   : 0.010
 Class :character   Class :character   1st Qu.: 0.4775   1st Qu.: 0.390   1st Qu.: 1.115
 Mode  :character   Mode  :character   Median : 1.8200   Median : 1.170   Median : 4.320
                                       Mean   : 2.5160   Mean   : 1.644   Mean   : 5.335
                                       3rd Qu.: 3.1250   3rd Qu.: 2.160   3rd Qu.: 6.435
                                       Max.   :34.0200   Max.   :23.800   Max.   :67.850
```

Several options have been explored and it had been decided to use "*group_by()*" function (***Figure 23***) and then save the progress as a separate dataset "turtle_sales_group.csv".

```
# Group_By Totals
group_by(turtle_sales_new) %>% summarise(NA_Total=sum(NA_Sales),
                                         EU_Total=sum(EU_Sales),
                                         Global_Total=sum(Global_Sales))

# Group_By Detail by Product for Global Sales, NA and EU
turtle_sales_group <- turtle_sales_new %>%  group_by(Product) %>%
  summarise(NA_Total=sum(NA_Sales),
            EU_Total=sum(EU_Sales),
            Global_Total=sum(Global_Sales),
            .groups='drop')
```

Figure 23. "group_by()" function in action

The next step is to discover the new dataset further. Examples of the advanced graphs used for the analysis can be evidenced below (***Figure 24-27***).

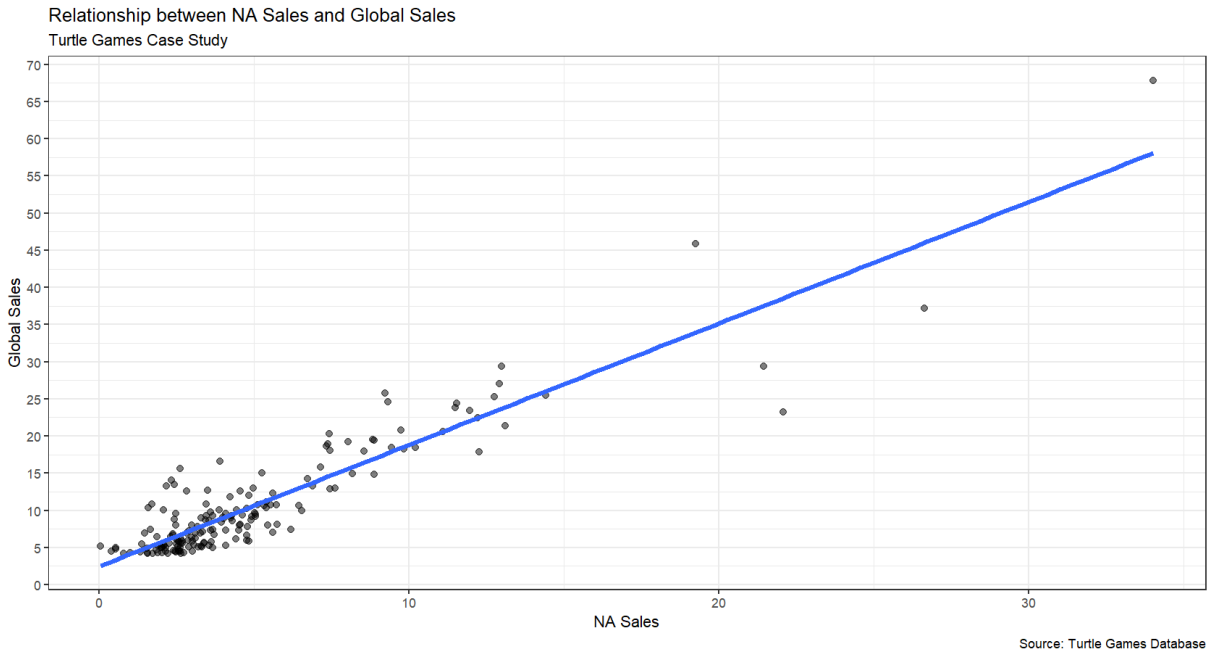


Figure 24. Scatter plot with a trend line for aggregated NA Sales vs Global Sales

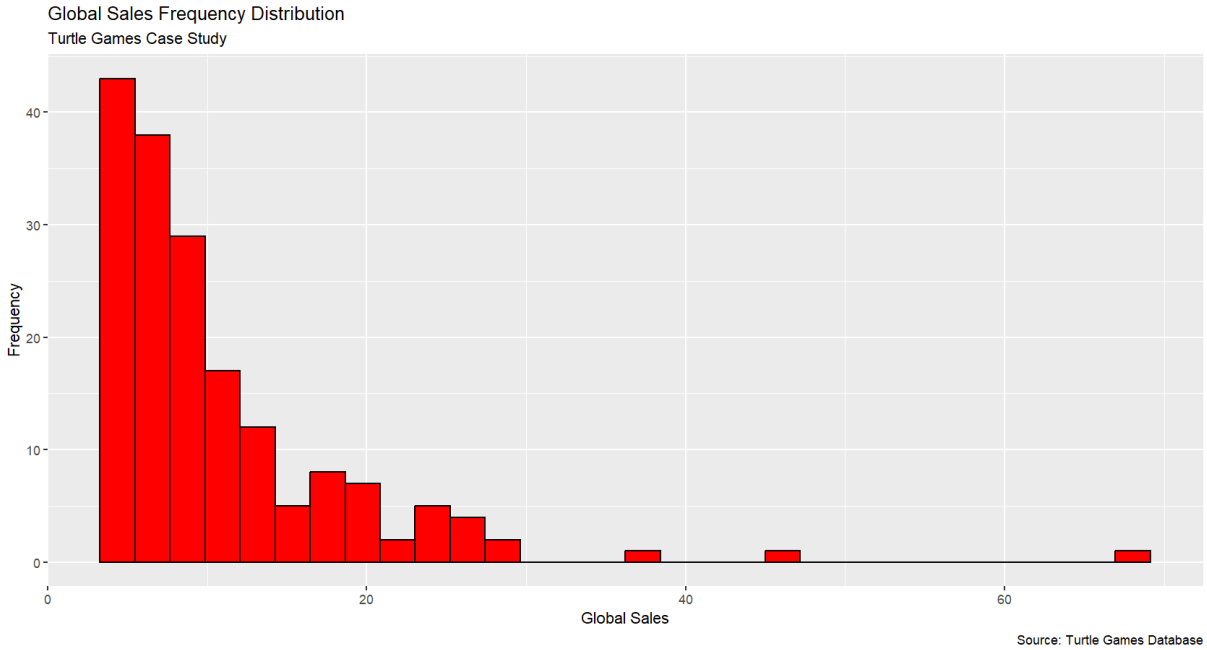


Figure 25. Histogram for aggregated Global Sales

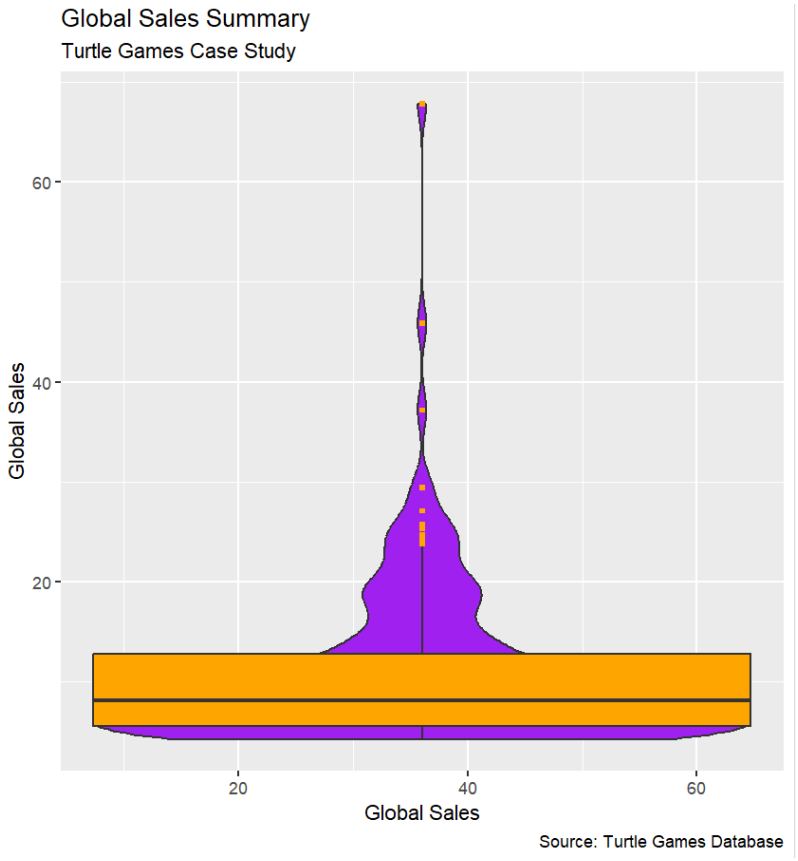


Figure 26. Boxplot with Violin plot for Global Sales

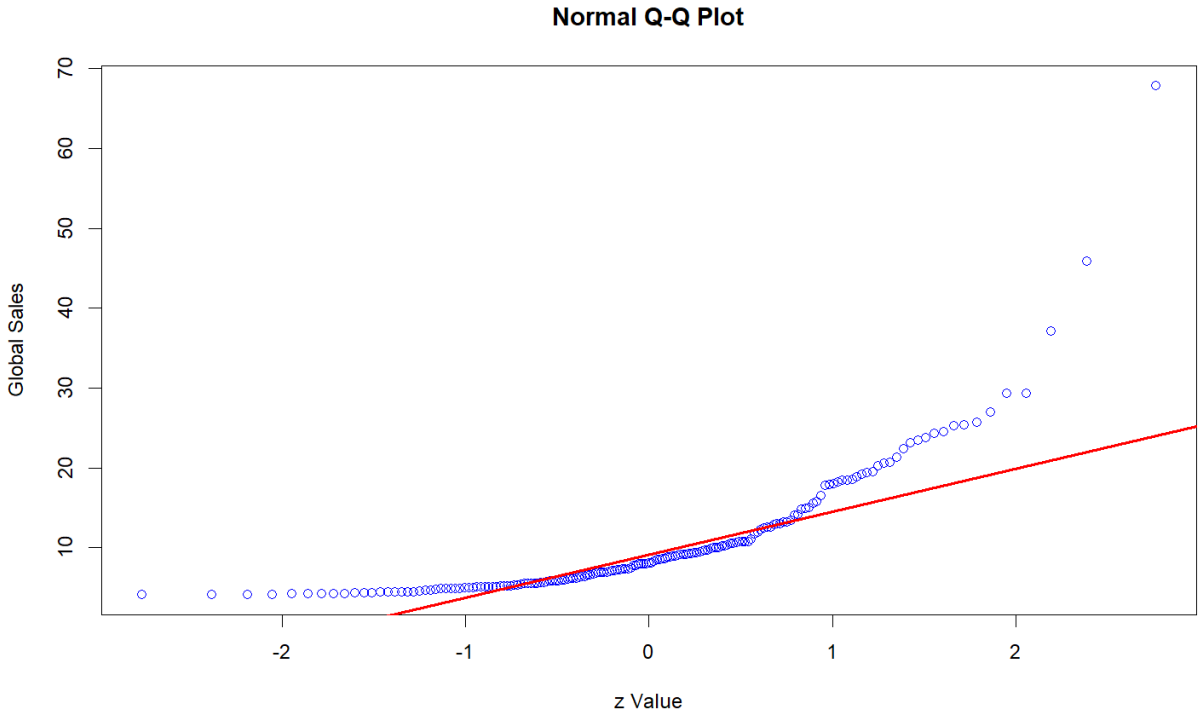


Figure 27. Q-Q Plot for Global Sales

Based on the already gathered information - the next step is to run a Shapiro-Wilk test and check for Skewness and Kurtosis on all the variables (***Figure 28***).

```
        Shapiro-Wilk normality test

data:  turtle_sales_group$Global_Total
W = 0.70955, p-value < 2.2e-16

> shapiro.test(turtle_sales_group$NA_Total)

        Shapiro-Wilk normality test

data:  turtle_sales_group$NA_Total
W = 0.69813, p-value < 2.2e-16

> shapiro.test(turtle_sales_group$EU_Total)

        Shapiro-Wilk normality test

data:  turtle_sales_group$EU_Total
W = 0.74058, p-value = 2.987e-16

> ## 3c) Determine Skewness and Kurtosis
> # Skewness and Kurtosis Global.
> skewness(turtle_sales_group$Global_Total)
[1] 3.066769
> kurtosis(turtle_sales_group$Global_Total)
[1] 17.79072
> # Skewness and Kurtosis NA.
> skewness(turtle_sales_group$NA_Total)
[1] 3.048198
> kurtosis(turtle_sales_group$NA_Total)
[1] 15.6026
> # Skewness and Kurtosis EU.
> skewness(turtle_sales_group$EU_Total)
[1] 2.886029
> kurtosis(turtle_sales_group$EU_Total)
[1] 16.22554
> ## 3d) Determine correlation
> # Determine correlation Global vs NA, Global vs EU, NA vs EU.
> cor(turtle_sales_group$Global_Total, turtle_sales_group$NA_Total)
[1] 0.9162292
> cor(turtle_sales_group$Global_Total, turtle_sales_group$EU_Total)
[1] 0.8486148
> cor(turtle_sales_group$NA_Total, turtle_sales_group$EU_Total)
[1] 0.6209317
```

Figure 28. Initial Shapiro-Wilk and Skewness, Kurtosis tests

After seeing the results for Shapiro-Wilk and Skewness and Kurtosis tests, it became apparent that the outliers have a significant impact on the dataset which resulted in applying various iterations of filtering to better understand the impact (***Figure 29***). Please refer to the **Appendix 1** for the detail with regard to the insights and consult with the PDF version of the presentation.

Data	
turtle_sales	352 obs. of 9 variables
turtle_sales_agg	175 obs. of 2 variables
turtle_sales_filtered	172 obs. of 4 variables
turtle_sales_filtered_2	156 obs. of 4 variables
turtle_sales_group	175 obs. of 4 variables
turtle_sales_new	352 obs. of 5 variables

Figure 29. Various Data sets saved as supporting csv files

3.3 Data Exploration (Regression Modelling)

The final step for the current analytics project is to implement the regression analysis and then run predictions based on the values given by the Turtle Games to fit the multiple linear regression model. For the purposes of this analysis the filtered dataset would be used which excludes the 3 blockbuster level products.

Residuals have got no clearly defined pattern which makes the model well fitted as the errors are independent and normally distributed (***Figure 30***). Then it is safe to go on and plot the Linear Regression model with the Regression line (***Figure 31***). The next step is to test out the Logarithmic version of the Linear Regression model (***Figure 32***).

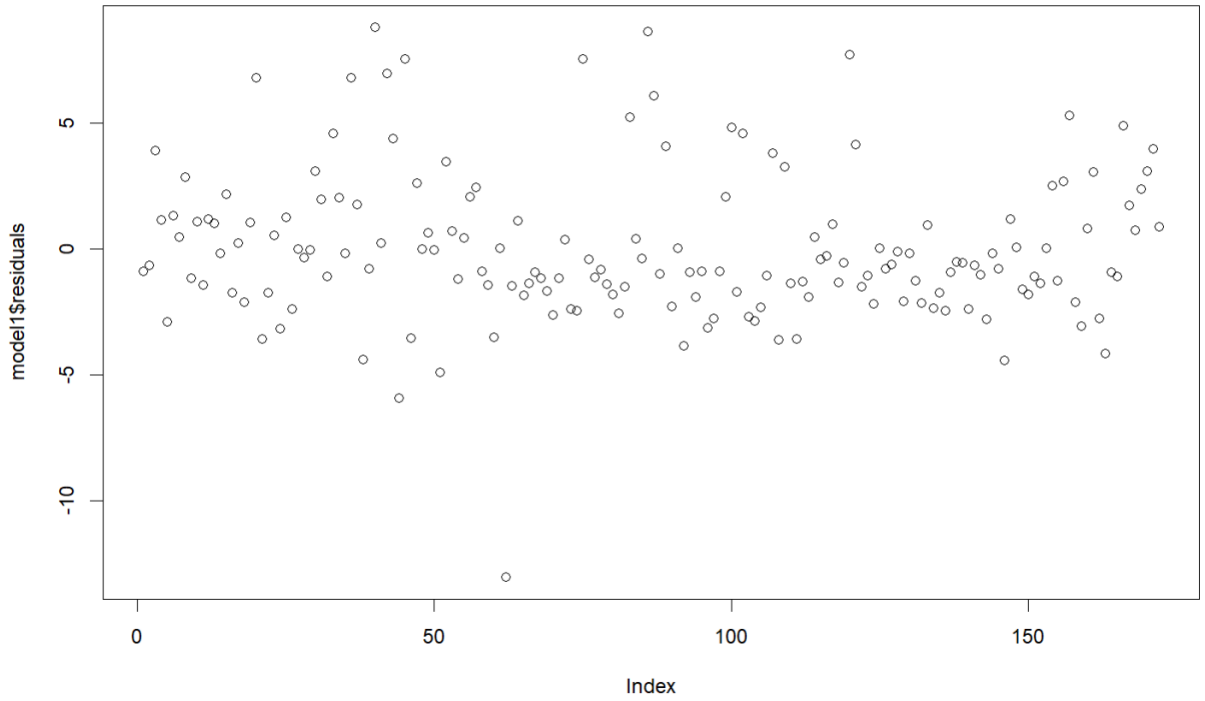


Figure 30. Residuals check for Linear Regression

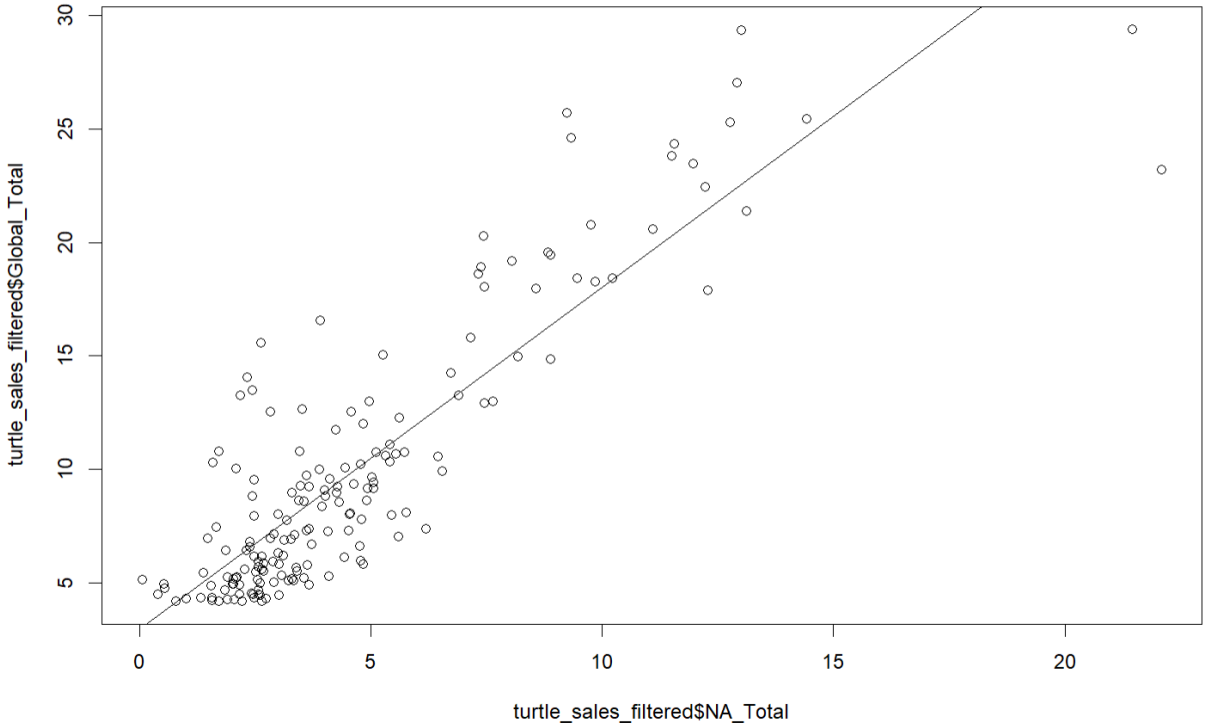


Figure 31. Linear Regression with the Regression line

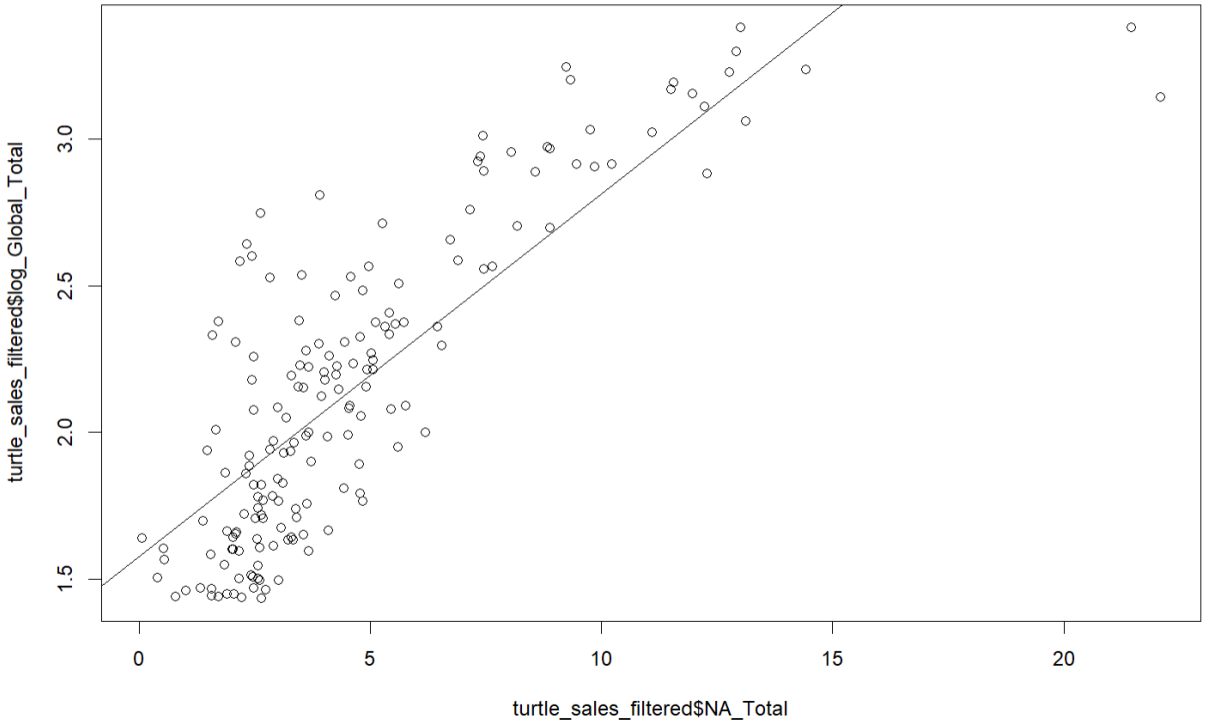


Figure 32. Log model for Linear Regression with the Regression Line

In order to calculate the adjusted R-Squared for the multiple linear regression the model should be fitted with the right variables where Global_Total would be used as a "Y" variable, and NA and EU as independent variables (***Figure 33***).

```
# 3.2 Create a multiple linear regression model
# Select only numeric columns from the treated data frame.

cor(turtle_sales_filtered)
corPlot(turtle_sales_filtered, cex=2)

# Create a new object and specify the lm function and the variables.
model_mlr_1 = lm(Global_Total ~ NA_Total + EU_Total, data=turtle_sales_filtered)

# Multiple linear regression model.
summary(model_mlr_1)
```

Figure 33. Setting up the MLR

The correlation plot shows the relationships between the variables, whether positive, negative or neutral (***Figure 34***).

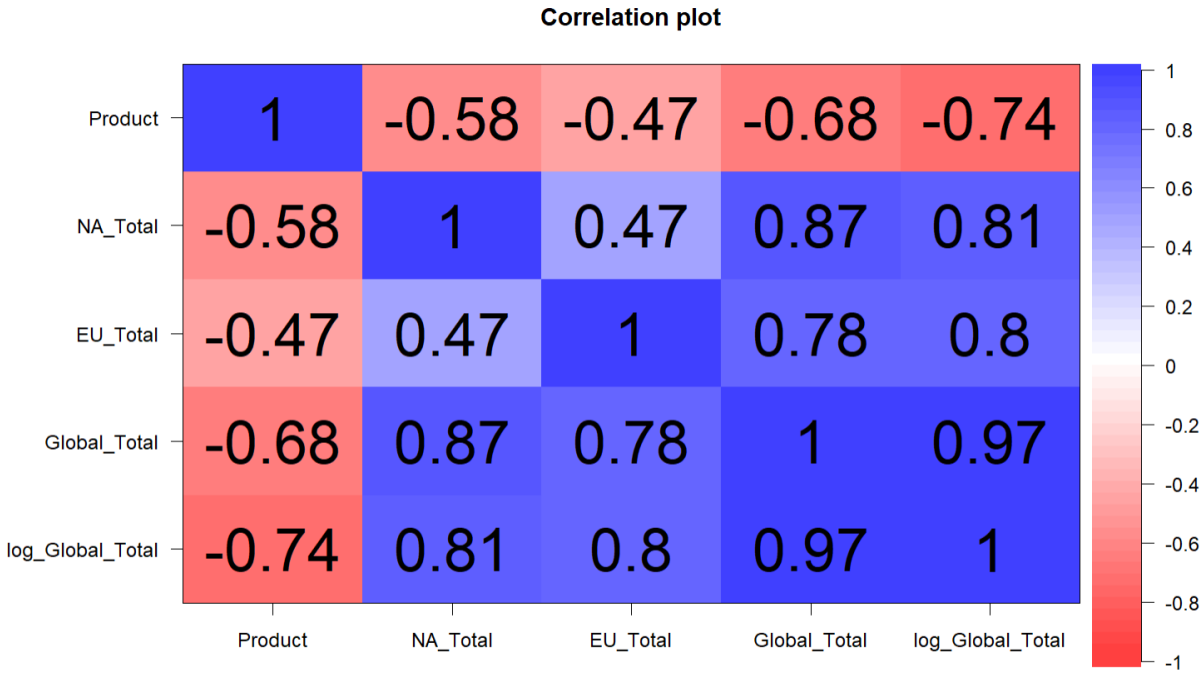


Figure 34. Correlation Plot for MLR

The adjusted R-Squared is at 93.8% which shows it is a very high quality model confirming that both NA and EU sales can explain the Global sales with the 93.8% accuracy and are significant variables (***Figure 35***).

```
Residuals:
    Min      1Q  Median      3Q     Max
-3.1342 -1.0194 -0.3870  0.6781  6.6955

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.08202    0.20983   5.157 6.97e-07 ***
NA_Total     1.11570    0.03725  29.953  < 2e-16 ***
EU_Total     1.20676    0.05419  22.271  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.499 on 169 degrees of freedom
Multiple R-squared:  0.939,     Adjusted R-squared:  0.9382
F-statistic:  1300 on 2 and 169 DF,  p-value: < 2.2e-16
```

Figure 35. Coefficients Analysis

The final step to fulfil the requirements of the Turtle Games client is to fit the created MLR model with the provided values.

```
> # a. NA_Sales_sum of 34.02 and EU_Sales_sum of 23.80.
> Global_Total_Forecast <- data.frame(NA_Total=c(34.02), EU_Total=c(23.80))
> predict(model_mlr_1, newdata = Global_Total_Forecast,
+         interval='confidence')
       fit      lwr      upr
1 67.7591 65.49936 70.01883
> # b. NA_Sales_sum of 3.93 and EU_Sales_sum of 1.56.
> Global_Total_Forecast <- data.frame(NA_Total=c(3.93), EU_Total=c(1.56))
> predict(model_mlr_1, newdata = Global_Total_Forecast,
+         interval='confidence')
       fit      lwr      upr
1 7.349275 7.080392 7.618158
> # c. NA_Sales_sum of 2.73 and EU_Sales_sum of 0.65.
> Global_Total_Forecast <- data.frame(NA_Total=c(2.73), EU_Total=c(0.65))
> predict(model_mlr_1, newdata = Global_Total_Forecast,
+         interval='confidence')
       fit      lwr      upr
1 4.912281 4.589107 5.235455
> # d. NA_Sales_sum of 2.26 and EU_Sales_sum of 0.97.
> Global_Total_Forecast <- data.frame(NA_Total=c(2.26), EU_Total=c(0.97))
> predict(model_mlr_1, newdata = Global_Total_Forecast,
+         interval='confidence')
       fit      lwr      upr
1 4.774066 4.464075 5.084056
> # e. NA_Sales_sum of 22.08 and EU_Sales_sum of 0.52.
> Global_Total_Forecast <- data.frame(NA_Total=c(22.08), EU_Total=c(0.52))
> predict(model_mlr_1, newdata = Global_Total_Forecast,
+         interval='confidence')
       fit     lwr      upr
1 26.34421 24.8969 27.79152
```

Figure 36. Fit the MLR Model with the Turtle Games Requirements

For further insights please refer to the **Appendix 1** or the presentation in PDF format.

4 Answering the Turtle Games Questions

Please refer to the enclosed PDF version of the PowerPoint Presentation or go to the **Appendix 1** of this report.

Please also refer to the Jupyter Notebook and RScript files enclosed and run them sequentially step by step as a walkthrough complimenting this report.

Technical Analysis Conclusions

All the required and advanced libraries on Jupyter Notebook (***sklearn, nltk, os, textblob, scipy***) and on RStudio (***tidyverse, tidyr, dplyr, moments, psych***) as well as supporting functions have been used to aid with the analysis.

New Jupyter Notebook commands used extensively for linear regression analysis, also sm.OLS().fit() and KMeans().

Data exploration functions on RStudio such as str(), glimpse(), typeof(), head(), class(), dim(), summary() have been used to sense check the data loaded correctly.

Data wrangling functions including aggregate(), group_by(), sum() have been used to treat the data and get it ready for further analysis.

Statistical analysis aided with the functions such as shapiro.test(), skewness(), kurtosis(), cor(), corPlot(), lm(), abline(), predict().

Formatting via mutate() implemented to change the format.

Data visualisation techniques using qplot() and ggplot() used extensively to create histograms, barplots, scatterplots, boxplots to aid with the analysis.

Formatting for the visualisations used on ggplot() to set the background, as well as format axis and apply more readable legend and naming conventions.

Several subsets have been saved as csv files for future use for quick access to avoid importing the base files and running the analysis from the beginning, instead focussing on the specific section.

The detailed outcomes of the Turtle Games Reviews and Sales analysis would be available within the **Appendix 1** and followed up on within the PPT presentation.

Appendix 1 (Insights)

Section 2.1

- After having completed linear regressions between Spending Score vs Loyalty, Remuneration vs Loyalty, Age vs Loyalty and Multiple Linear Regression it can be concluded that:
- There is a positive correlation between Remuneration and Loyalty Points accumulation, which could mean higher salary leads to more disposable income and this could drive extra spending.
- The more customers spend, the higher the loyalty points they accumulate, however some of the high spenders do not accumulate sufficient loyalty points therefore the loyalty programme should be revisited.
- Majority of the consumers are located within the 40-60 spending score region with loyalty points in some cases lower that for those spending less.
- Target consumers with remuneration between £60k and £100k with low loyalty points – opportunity to grow sales.
- However, there seemingly is no real correlation between Age and Loyalty, even though more of those aged between 30 and 40 are clustered together with substantially higher numbers of Loyalty Points than any of the other age groups.
- Multiple linear regression model R-Squared 84% Train, 83% Test datasets shows high accuracy, as well as the VIF factor hovering around 1 which shows the reliability of the constructed model.

Section 2.2

- Elbow and silhouette methods have been used to determine that the best number of unique clusters is 5.
- To validate this then several K-Means clustering simulations had been run ranging from 3 to 6 which yielded a clear confirmation that 5 is the best option.
- Further to that each of the clusters would need to be investigated further. The dataset subsets for each of the clusters would need to be investigated and targeted separately based on the demographics, buying patterns and other cluster specific attributes. Further information from the Turtle Games collected about its customers would be required to assess each of the clusters deeper.

Section 2.3

- After reviewing the clean version of the word cloud and also consulting the most frequently used words. Most common ones are Game, Great, Fun, Love, Like on both the Review and Summary columns at the top of the list which at a glance shows positivity towards to gaming products on offer.
- To validate the above success the polarity of the most frequently words used needs to be assessed and it can clearly be evidenced that in most part the reviews are either positive or positively neutral on both the review and summary columns. The majority of the distribution for review column is between 0 and 0.5 which is somewhat positive, ranging all the way to 1.

- The summary column is more fragmented ranging from predominantly neutral to chunks of words standing at 0.25, 0.5 and 0.75, which also signals for the positive sentiment.
- There are only a tiny fraction of negative reviews and summary words compared to the overall distribution of the token words.
- After further consulting the document term matrix it is quite difficult to gauge the sentiment and the words resemble the gaming slang more than anything else, so would not be particularly useful for the given analysis.
- However, to deep dive further - the most positive and the most negative reviews need to be thoroughly reviewed and analysed to keep doing well the things that are going well for the business and understand and improve the areas where the offering is not delivering what the audience expects.
- Vector words could also be taken into account for further analysis in conjunction with the most negative reviews.

Section 3.1

- The initial insights show that there are clear outliers present and there are quite a few of them.
- There is a wide spread of sales across different platforms
- The data is grouped together for kurtosis and histogram shows skew on the lower side of sales for Global sales
- Additional insights could be gained from the variables that were requested by the Turtle Sales team to be removed, i.e. Genre or Publisher

Section 3.2

- Filter out the three blockbuster games which skew the data and heavily hinder normality and should be separated out of the general cohort as the blockbuster games can be deemed to be in the league of their own and should be compared to the similar best sellers. 3 Blockbusters removed to create a new dataframe.
- There are quite a lot of the data points at total level above the 1+ z value, as well as below -1 z value - removing all these datapoints would invalidate the whole analysis as too many observations would be removed.
- Several titles above the QQ line on the positive side fall within the blockbuster territory, and skew the entire dataset.
- For a more objective approach towards the analysis it would make sense to slice the data further and discuss the grouping of the different titles with the client (Turtle Games).
- An example would be isolating the Top 10 to Top 20 Titles and running a separate analysis for the top performers, as well as running separate analysis for the average performers and also check out the low performers.
- Even though, seemingly, there are substantially more outliers on the positive side than the 3 Blockbusters – they still form a portion of the global sales. Tested the different filtering between 20 and 30 for global sales based on the global sales boxplot outliers visualisation, but decided to leave them within the dataset.
- Depending on the title and the marketing campaign, or the distribution after launch of the product, it would make sense to not only look at the global picture, but rather target different

regions separately and run the analysis separately, which would help working with the outliers specific to a given region.

- An example, Product 254 – NA Sales 19.03, vs EU Sales of 1.85. If the sales were somewhat proportionate, the analysis would have been objective, otherwise, due to demographics, local norms or regulations, preferences, marketing etc the results could be vastly different depending on the region and hence could skew the data one way or the other.

Section 3.3

- The residuals are random and let us plot the linear regression without issue.
- Based on the plotted values the NA Sales (0.87) as well as the EU Sales (0.78) have got a strong positive correlation with the Global Sales. It makes sense as these two variables comprise the total sales figure. There is a moderate correlation between EU and NA sales (0.47) so one might not necessarily be affected by the other.
- Simple linear regression for NA Total vs Global Total generates the R-Squared of 0.76 whilst the logarithmic model R-Squared actually makes the model worse with R-Squared at 0.66. For the EU it barely changes (0.61 ➔ 0.63) vs Global and NA vs EU (0.22 ➔ 0.18).
- The adjusted R-Squared for MLR is at 93.8% which shows it being a high quality model confirming that both NA and EU sales can explain the Global sales with the 93.8% accuracy and are significant variables.

Fitted Results as requested:

NA_Sales_sum of 34.02 and EU_Sales_sum of 23.80

fit	lwr	upr
67.7591	65.49936	70.01883

NA_Sales_sum of 3.93 and EU_Sales_sum of 1.56

fit	lwr	upr
7.349275	7.080392	7.618158

NA_Sales_sum of 2.73 and EU_Sales_sum of 0.65

fit	lwr	upr
4.912281	4.589107	5.235455

NA_Sales_sum of 2.26 and EU_Sales_sum of 0.97

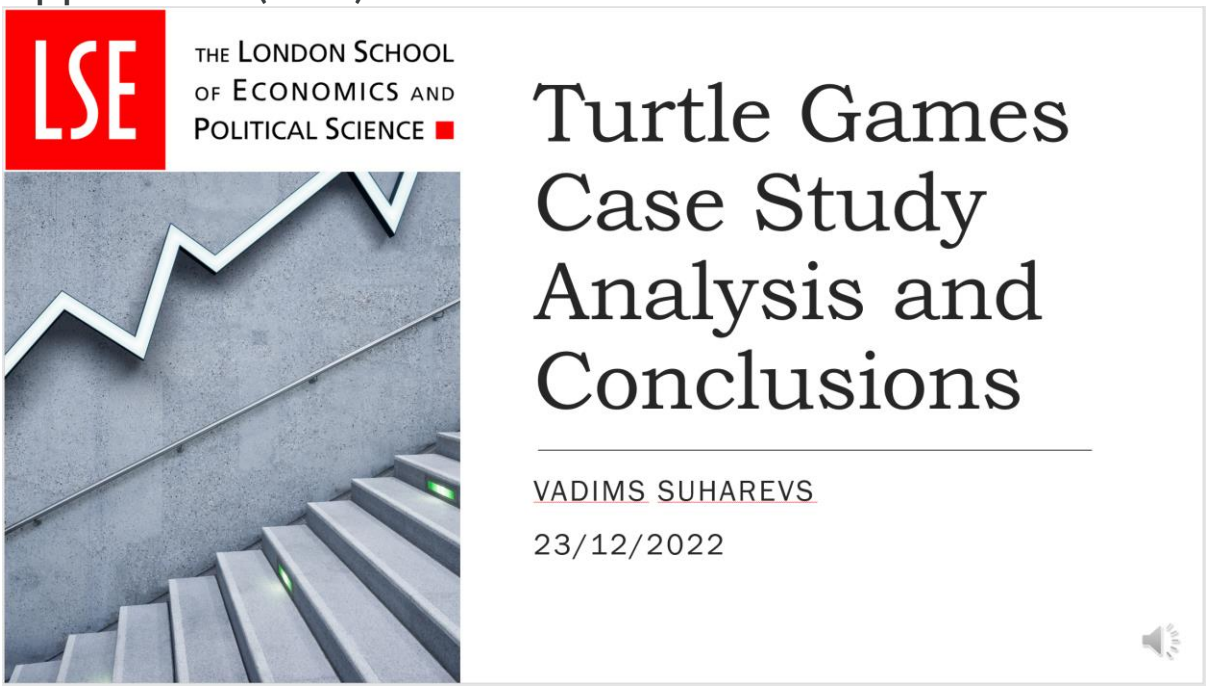
fit	lwr	upr
4.774066	4.464075	5.084056

NA_Sales_sum of 22.08 and EU_Sales_sum of 0.52

fit	lwr	upr
26.34421	24.8969	27.79152

All the requested fitted lines fall within the Upper and Lower limits which means the model is stable.

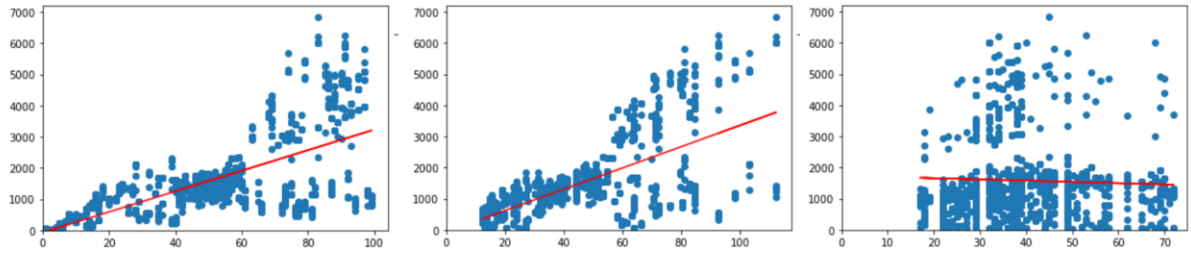
Appendix 2 (PPT)



Loyalty Points Analysis (reviews)

Remuneration
= £ thousand

Correlation and regression for Loyalty Points and Spending Score, Remuneration, Age



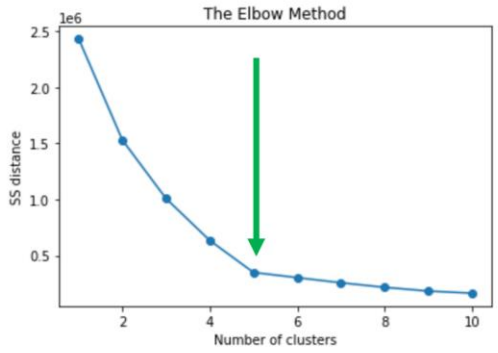
Spending Score | Remuneration | Age

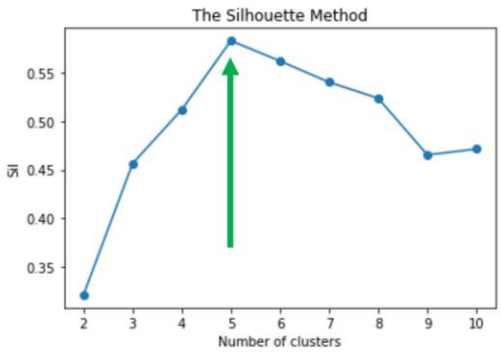
Multiple Linear Regression Results → R-Squared Train: 84% Test: 83% →

VIF Factor	features
20.5	const
1.1	spending_score
1.0	remuneration
1.1	age

Prework for Segmentation Targeting (reviews)

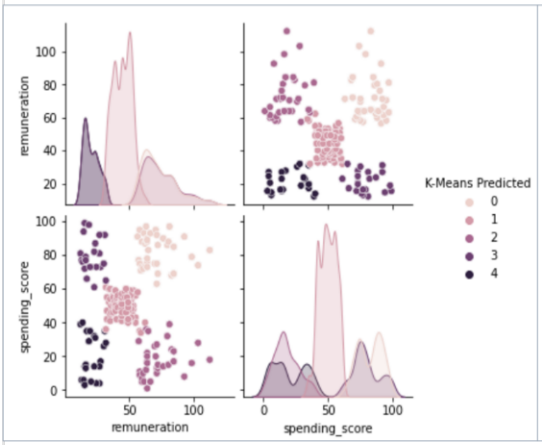
Elbow and Silhouette Method to determine Clusters

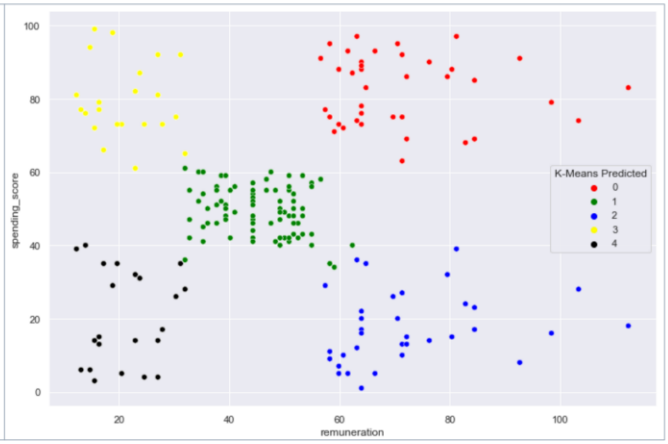




Customer Profiling (reviews)

Clustering in Practice using 5 determined Clusters





Natural Language Processing (word cloud)

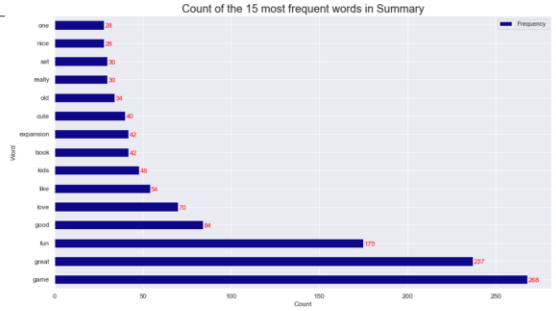
Review Column

Summary Column

Natural Language Processing (word freq)

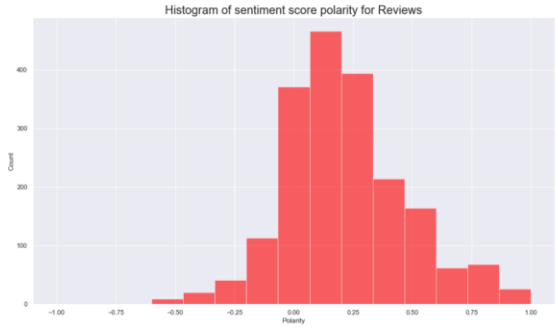
Review Column

Summary Column

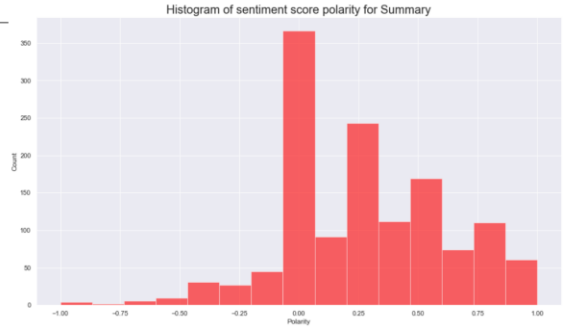


Natural Language Processing (polarity)

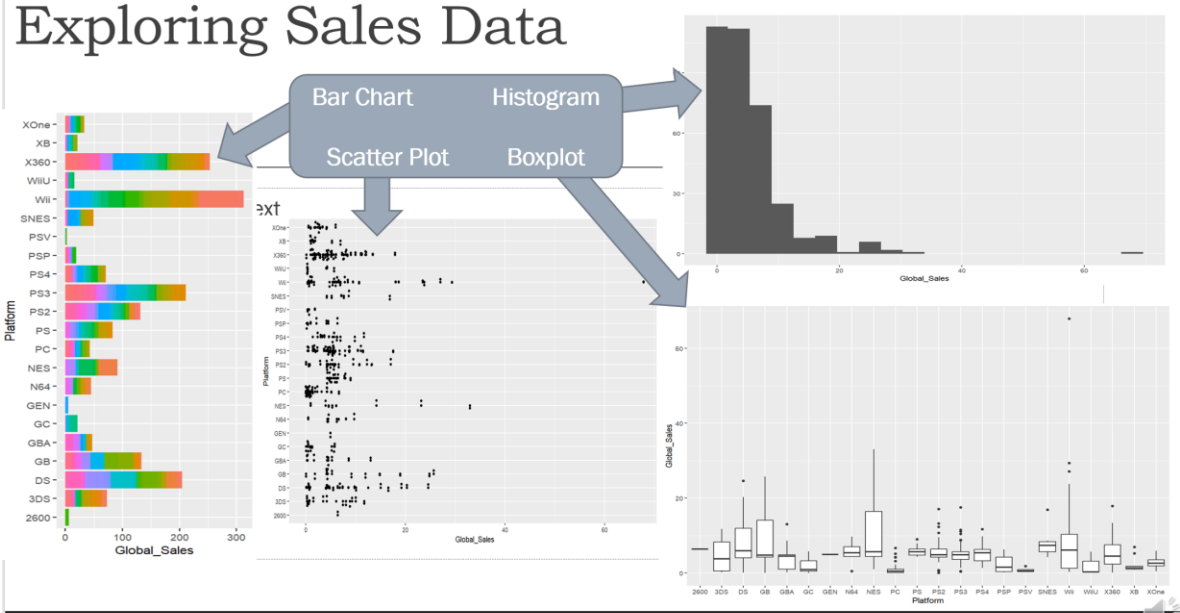
Review Column



Summary Column



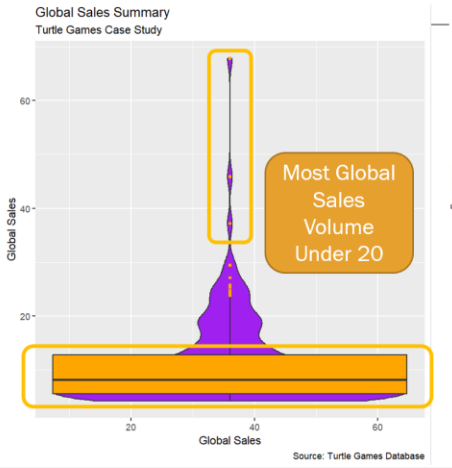
Exploring Sales Data



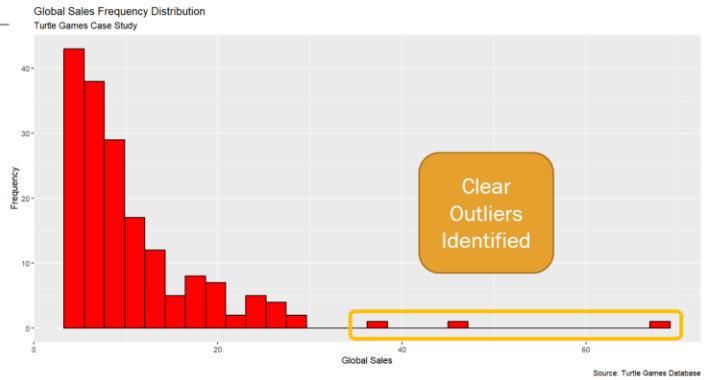
Exploring Sales Data (cont.)

Sales Unit of Measure = £ million

Data Summary



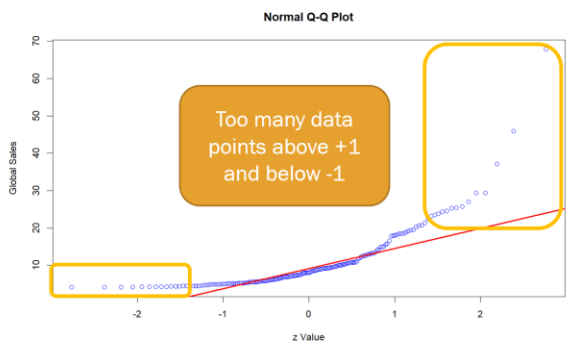
Global Sales data distribution



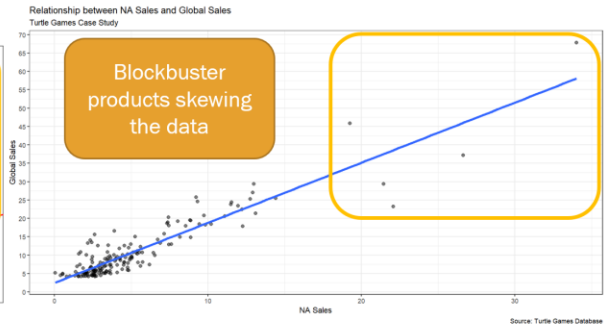
Exploring Sales Data (cont.)

Sales Unit of Measure = £ million

Q-Q Plot Check for Normality



Global Sales vs NA Scatter



Exploring Sales Data (cont.)

No Treatment

(175 Observations)

```
        Shapiro-Wilk normality test

data:  turtle_sales_group$Global_Total
W = 0.70955, p-value < 2.2e-16

> shapiro.test(turtle_sales_group$NA_Total)

        Shapiro-Wilk normality test

data:  turtle_sales_group$NA_Total
W = 0.69813, p-value < 2.2e-16

> shapiro.test(turtle_sales_group$EU_Total)

        Shapiro-Wilk normality test

data:  turtle_sales_group$EU_Total
W = 0.74058, p-value = 2.987e-16

> ## 3c) Determine Skewness and Kurtosis
> # Skewness and Kurtosis Global.
> skewness(turtle_sales_group$Global_Total)
[1] 3.066769
> kurtosis(turtle_sales_group$Global_Total)
[1] 17.79072
> # Skewness and Kurtosis NA.
> skewness(turtle_sales_group$NA_Total)
[1] 3.048198
> kurtosis(turtle_sales_group$NA_Total)
[1] 15.6026
> # Skewness and Kurtosis EU.
> skewness(turtle_sales_group$EU_Total)
[1] 2.886029
> kurtosis(turtle_sales_group$EU_Total)
[1] 16.22554
> ## 3d) Determine correlation
> # Determine correlation Global vs NA, Global vs EU, NA vs EU.
> cor(turtle_sales_group$Global_Total, turtle_sales_group$NA_Total)
[1] 0.9162292
> cor(turtle_sales_group$Global_Total, turtle_sales_group$EU_Total)
[1] 0.8486148
> cor(turtle_sales_group$NA_Total, turtle_sales_group$EU_Total)
[1] 0.6209317
```

3 Blockbusters Removed

(172 Observations)

```
        Shapiro-Wilk normality test

data:  turtle_sales_filtered$Global_Total
W = 0.8316, p-value = 8.212e-13

> shapiro.test(turtle_sales_filtered$NA_Total)

        Shapiro-Wilk normality test

data:  turtle_sales_filtered$NA_Total
W = 0.80028, p-value = 4.618e-14

> shapiro.test(turtle_sales_filtered$EU_Total)

        Shapiro-Wilk normality test

data:  turtle_sales_filtered$EU_Total
W = 0.85912, p-value = 1.42e-11

> # Skewness and Kurtosis Global.
> skewness(turtle_sales_filtered$Global_Total)
[1] 1.352425
> kurtosis(turtle_sales_filtered$Global_Total)
[1] 4.033009
> # Skewness and Kurtosis NA.
> skewness(turtle_sales_filtered$NA_Total)
[1] 2.097169
> kurtosis(turtle_sales_filtered$NA_Total)
[1] 8.877347
> # Skewness and Kurtosis EU.
> skewness(turtle_sales_filtered$EU_Total)
[1] 1.247008
> kurtosis(turtle_sales_filtered$EU_Total)
[1] 3.719443
> cor(turtle_sales_filtered$Global_Total, turtle_sales_filtered$NA_Total)
[1] 0.8716652
> cor(turtle_sales_filtered$Global_Total, turtle_sales_filtered$EU_Total)
[1] 0.7841482
> cor(turtle_sales_filtered$NA_Total, turtle_sales_filtered$EU_Total)
[1] 0.4713353
```

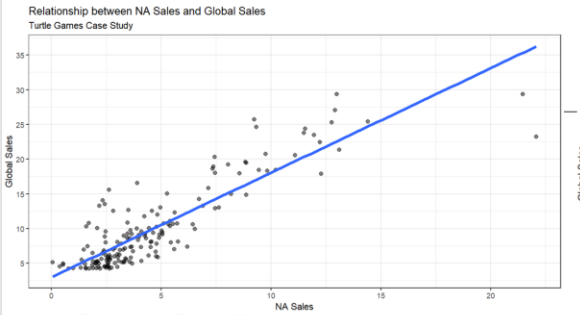
3 Blockbusters Removed

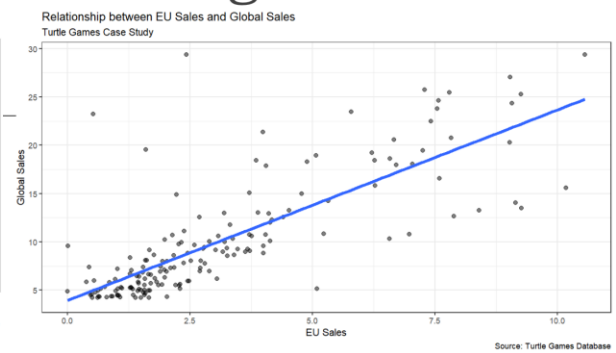
(156 Obersvations)

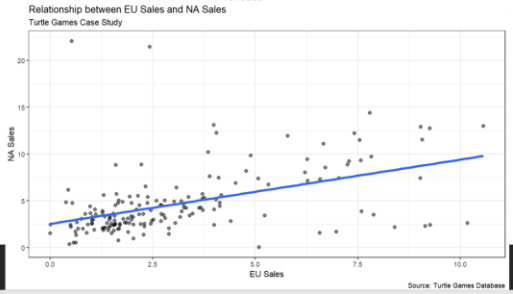
```
        Shapiro-Wilk normality test

data:  turtle_sales_filtered_2$Global_Total
W = 0.87058, p-value = 2.228e-10

> shapiro.test(turtle_sales_filtered_2$NA_Total)

        Shapiro-Wilk normality test

data:  turtle_sales_filtered_2$NA_Total
W = 0.91425, p-value = 5.724e-08

> shapiro.test(turtle_sales_filtered_2$EU_Total)

        Shapiro-Wilk normality test

data:  turtle_sales_filtered_2$EU_Total
W = 0.86102, p-value = 7.815e-11

> # Skewness and Kurtosis Global.
> skewness(turtle_sales_filtered_2$Global_Total)
[1] 1.116184
> kurtosis(turtle_sales_filtered_2$Global_Total)
[1] 3.435866
> # Skewness and Kurtosis NA.
> skewness(turtle_sales_filtered_2$NA_Total)
[1] 1.201097
> kurtosis(turtle_sales_filtered_2$NA_Total)
[1] 4.600045
> # Skewness and Kurtosis EU.
> skewness(turtle_sales_filtered_2$EU_Total)
[1] 1.492756
> kurtosis(turtle_sales_filtered_2$EU_Total)
[1] 5.112208
> cor(turtle_sales_filtered_2$Global_Total, turtle_sales_filtered_2$NA_Total)
[1] 0.784706
> cor(turtle_sales_filtered_2$Global_Total, turtle_sales_filtered_2$EU_Total)
[1] 0.7652949
> cor(turtle_sales_filtered_2$NA_Total, turtle_sales_filtered_2$EU_Total)
[1] 0.3637491
> # Check the dimensions of the data frame
```

Sales Correlations between regions

Sales Unit of Measure = £ million



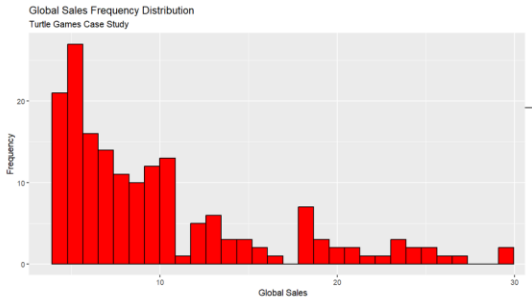


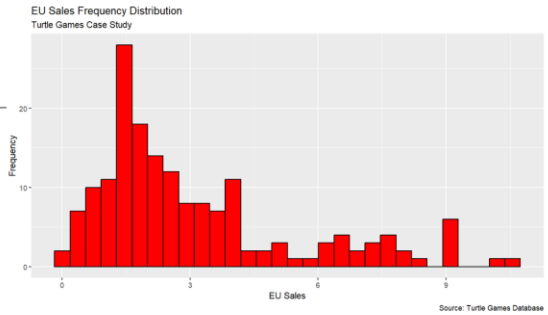


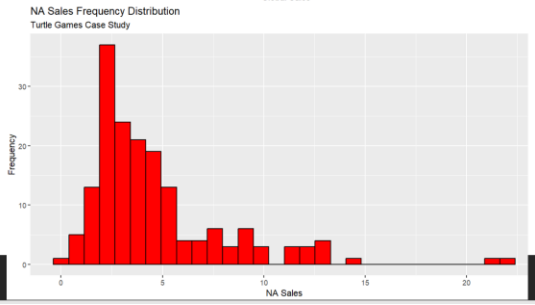
Strong Positive Correlation with Global Across All Regions
NA Sales (0.87)
EU Sales (0.78)
(3 blockbusters excluded)

Sales Frequency Distributions

Sales Unit of Measure = £ million





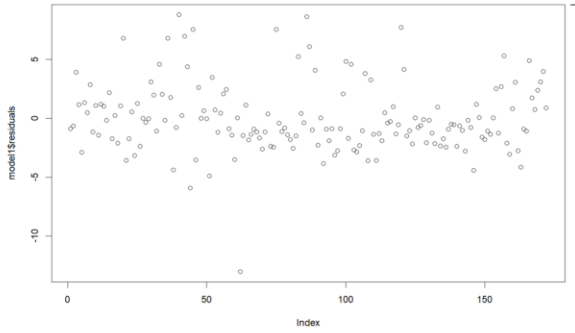


Majority of the Sales Data is located below £20m, some top performers
(3 blockbusters excluded)

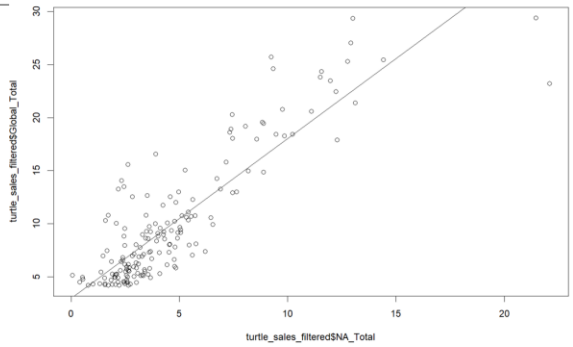
Simple Linear Regression

Sales Unit of Measure = £ million

Residuals Check



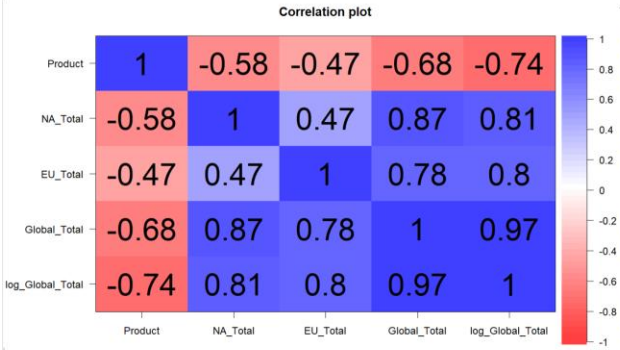
Simple Linear Regression



R-Squared of 0.76
Regression model is highly accurate

Multiple Linear Regression

Correlation Plot



Requested Fit

```
-       fit      lwr      upr
1 67.7591 65.49936 70.01883
> # b. NA_Sales_sum of 3.93 and EU_Sales_sum of 1.56.
> Global_Total_Forecast <- data.frame(NA_Total=c(3.93), EU_Total=c(1.56))
> predict(model_mlr_1, newdata = Global_Total_Forecast,
+         interval='confidence')
       fit      lwr      upr
1 7.349275 7.080392 7.618158
> # c. NA_Sales_sum of 2.73 and EU_Sales_sum of 0.65.
> Global_Total_Forecast <- data.frame(NA_Total=c(2.73), EU_Total=c(0.65))
> predict(model_mlr_1, newdata = Global_Total_Forecast,
+         interval='confidence')
       fit      lwr      upr
1 4.912281 4.589107 5.235455
> # d. NA_Sales_sum of 2.26 and EU_Sales_sum of 0.97.
> Global_Total_Forecast <- data.frame(NA_Total=c(2.26), EU_Total=c(0.97))
> predict(model_mlr_1, newdata = Global_Total_Forecast,
+         interval='confidence')
       fit      lwr      upr
1 4.774066 4.464075 5.084056
> # e. NA_Sales_sum of 22.08 and EU_Sales_sum of 0.52.
> Global_Total_Forecast <- data.frame(NA_Total=c(22.08), EU_Total=c(0.52))
> predict(model_mlr_1, newdata = Global_Total_Forecast,
+         interval='confidence')
       fit      lwr      upr
1 26.34421 24.8969 27.79152
```

Conclusions

Positive correlation between Remuneration and Loyalty Points accumulation, which could mean higher salary leads to more disposable income and this could drive extra spending.

The more customers spend, the higher the loyalty points they accumulate, however some of the high spenders do not accumulate sufficient loyalty points therefore the loyalty programme should be revisited.

Majority of the consumers are located within the 40-60 spending score region with loyalty points in some cases lower that for those spending less.

Five distinct clusters of consumers based on the loyalty points and spending score.

Positive sentiment - Most common words Great, Fun, Love, Like on both the Review and Summary columns at the top of the list

In most part the reviews are either positive or positively neutral, the mostp ositive and most negative comments should be investigated further

Conclusions (cont)

The residuals are random and let us plot the linear regression without issue.

Strong positive correlation for the Global Sales vs other sales figures

More than 70% of the variation can be explained by each variable in a simple linear regression

NA and EU MLR can explain the Global sales with the 93.8% accuracy

Outliers with disproportionately high sales - the blockbuster games, should be compared to the similar best sellers

Quite a lot of the data points at total level above the 1+ z value, as well as below -1 z value - removing all these datapoints would invalidate the whole analysis as too many observations would be removed

Potentially isolate the Top 10 to Top 20 Titles and running a separate analysis for top performers

All the requested fitted lines fall within the Upper and Lower limits which means the MLR is stable.

Recommendations

The current research opens the door for asking further questions and gives an indication where to look next for insights and how to approach various issues stumbled upon during research (i.e. dealing with outliers which are not necessarily outliers, but Blockbuster products)

Collate a more detailed deep dive into the customer segmentation by implementing further analysis of the 5 clusters identified based on the attributes specific to those clusters and tailoring marketing campaigns accordingly.

Target the consumers possessing the spending score between 40 and 60 with marketing campaigns to improve the purchasing behaviour as the vast number of consumers is located within this region.

Target consumers with remuneration between £60k and £100k with low loyalty points – opportunity to grow sales.

Consumers radiate positive sentiment towards the Turtle Games products – this needs to be explored and exploited

The next phase for the Turtle Games would be to provide the project team with clear guidance on whether to focus on specific region and specific product range with the most significant revenue potential

Focus on different regions separately and run the analysis separately, which would help working with the outliers specific to a given region due to demographics, local norms or regulations, preferences, marketing etc

Appendix 3 (Speech)

1

Hello, my name is Vadim and I would like to walk you through the Turtle Games Case study analysis and conclusions.

2

The Turtle Games have instructed the team that there is a requirement to conduct an analysis into the accumulation of loyalty point by their customers, reliability of the data, targeting specific consumer market segments, understand the customers through the analysis of product reviews, regional and global sales to try and improve sales performance.

3

After having completed linear regressions between Spending Score vs Loyalty, Remuneration vs Loyalty, Age vs Loyalty and Multiple Linear Regression it can be concluded that:

There is a positive correlation between Remuneration and Loyalty Points accumulation, which could mean higher salary leads to more disposable income and this could drive extra spending.

The more customers spend, the higher the loyalty points they accumulate, however some of the high spenders do not accumulate sufficient loyalty points therefore the loyalty programme should be revisited.

Majority of the consumers are located within the 40-60 spending score region with loyalty points lower in some cases than for those spending less.

Multiple linear regression model R-Squared 84% Train, 83% Test datasets shows high accuracy, as well as the VIF factor hovering around 1 which shows the reliability of the constructed model.

4

Elbow and silhouette methods have been used to determine that the best number of unique clusters is 5. To validate this then several K-Means clustering simulations had been run ranging from 3 to 6 which yielded a clear confirmation that 5 is the best option.

5

Each of the clusters would need to be investigated further. The dataset subsets for each of the clusters would need to be investigated and targeted separately based on the demographics, buying patterns and other cluster specific attributes. Further information from the Turtle Games collected about its customers would be required to assess each of the clusters deeper.

6

After reviewing the clean version of the word cloud and also consulting the most frequently used words analysis it was determined that most common ones are Game, Great, Fun, Love, Like on both the Review and Summary columns at the top of the list which at a glance shows positivity towards to gaming products on offer.

7

The detail can be seen on this slide.

8

To validate the above success - polarity of the most frequently used words needs to be assessed and it can clearly be evidenced that in most part the reviews are either positive or positively neutral on both the review and summary columns. The majority of the distribution for review column is between 0 and 0.5 which is somewhat positive, ranging all the way to 1.

The summary column is more fragmented ranging from predominantly neutral to chunks of words standing at 0.25, 0.5 and 0.75, which also signals for the positive sentiment. There is only a tiny fraction of negative reviews and summary words compared to the overall distribution of the token words.

To deep dive further - the most positive and the most negative reviews need to be thoroughly reviewed and analysed to keep doing well the things that are going well for the business and understand and improve the areas where the offering is not delivering what the audience expects.

9

The initial insights into sales show that there are clear outliers present and there are quite a few of them. There is a wide spread of sales across different gaming platforms.

The data is grouped together for kurtosis and histogram shows skew to the lower side of sales

Additional insights could be gained from the variables that were requested by the Turtle Games team to be removed, i.e. Genre or Publisher

10

The clear outliers - the three blockbuster games on the right hand side histogram and top of the boxplot skew the data and heavily hinder normality and should be separated out of the general cohort. The blockbuster games can be deemed to be in the league of their own and should be compared to the similar best sellers.

11

After having removed the 3 Blockbusters and creating the new dataframe the next exploration tool is Q-Q plot. There are quite a lot of the data points at total level above the 1+ z value, as well as below -1 z value - removing all these datapoints would invalidate the whole analysis as too many observations would be removed that form substantial value creation to the business. Several titles above the QQ line on the positive side fall within the blockbuster territory, and skew the dataset.

For a more objective approach towards the analysis it would make sense to slice the data further and discuss the grouping of the different titles based on performance and price with the Turtle Games.

12

The different filtering between £20 million and £30 million was tested for global sales based on the boxplot outliers visualisation, but decision was made to leave them within the dataset and keep working with the dataset in the middle as the skewedness and kurtosis do not change much between middle and the right hand side datasets and fall within somewhat manageable level.

13

Based on the plotted values the NA Sales (0.87) as well as the EU Sales (0.78) have got a strong positive correlation with the Global Sales. It makes sense as these two variables comprise the total sales figure.

14

Majority of the data falls between £4m and £15m for the Global Sales with a separate cluster above £20m.

15

The residuals are random and let us plot the linear regression without issue.

Simple linear regression for NA Total vs Global Total generates the R-Squared of 0.76 whilst the logarithmic model R-Squared actually makes the model worse with R-Squared at 0.66. For the EU it barely changes (0.61 ➔ 0.63) vs Global and NA vs EU (0.22 ➔ 0.18).

16

The adjusted R-Squared for Multiple Linear Regression is at 93.8% which shows it being a high quality model confirming that both NA and EU sales can explain the Global sales with the 93.8% accuracy and are significant variables.

All the requested fitted lines fall within the Upper and Lower limits which means the model is stable.

17

Conclusions

Positive correlation between Remuneration and Loyalty Points accumulation could mean higher salary leads to more disposable income and this could drive extra spending.

The more customer spend, the higher the loyalty points they accumulate, however some of the high spenders do not accumulate sufficient loyalty points therefore the loyalty programme should be revisited.

Majority of the consumers are located within the 40-60 spending score region with loyalty points in some cases lower that for those spending less.

We identified five distinct clusters of consumers based on the loyalty points and spending score.

Positive sentiment - Most common words Great, Fun, Love, Like on both the Review and Summary columns at the top of the list

In most part the reviews are either positive or positively neutral

18

The residuals are random and let us plot the linear regression without issue.

Strong positive correlation with the Global Sales

More than 70% of the variation can be explained by each variable in a simple linear regression

NA and EU MLR can explain the Global sales with the 93.8% accuracy

Outliers with disproportionately high sales - the blockbuster games, should be compared to the similar best sellers

Quite a lot of the data points at total level above the 1+ z value, as well as below -1 z value - removing all these datapoints would invalidate the whole analysis as too many observations would be removed

Potentially isolate the Top 10 to Top 20 Titles and running a separate analysis for top performers

All the requested fitted lines fall within the Upper and Lower limits which means the MLR is stable.

19

Recommendations

The current research opens the door for asking further questions and gives an indication where to look next for insights and how to approach various issues stumbled upon during research (i.e. dealing with outliers which are not necessarily outliers, but Blockbuster products)

Collate a more detailed deep dive into the customer segmentation by implementing further analysis of the 5 clusters identified based on the attributes specific to those clusters and tailoring marketing campaigns accordingly.

Target the consumers possessing the spending score between 40 and 60 with marketing campaigns to improve the purchasing behaviour as the vast number of consumers is located within this region.

Target consumers with remuneration between £60k and £100k with low loyalty points – opportunity to grow sales.

Consumers radiate positive sentiment towards the Turtle Games products – this needs to be explored and exploited

The next phase for the Turtle Games would be to provide the project team with clear guidance on whether to focus on specific region and specific product range with the most significant revenue potential

Focus on different regions separately and run the analysis separately, which would help working with the outliers specific to a given region due to demographics, local norms or regulations, preferences, marketing etc

Thank you for your time