Predicting Future Outcomes

A Report for Turtle Games

Background

An initial set of questions has been provided by Turtle Games, as part of a wider project to improve overall sales performance. The responses to those questions follow:

How customers accumulate loyalty points

Using Python, Pearson's Correlation Coefficient showed a correlation of 0.67231 between spending score and loyalty points which can be considered a strong relationship (Table 1). A scatterplot was produced which supports this (Figure 1). A simple linear regression model (NumPy polyval) was applied to the data to see if loyalty points could be predicted based on spending score (Figure 2).

The Ordinary Least Squares method was also applied (Table 2 & Table 3) and a regression line was applied in Figure 3.

These methods were also applied to remuneration (Tables 4, 5 & 6 and Figure 4, 5 & 6) and age (Tables 7, 8 & 9 and Figures 7, 8 & 9) in relation to loyalty points.

In summary, remuneration also had a strong correlation with loyalty points (0.616065) although not as strong as spending score. Age had a low degree correlation with loyalty points (-0.042445) which is a very small correlation.

How groups within the customer base can be used to target specific market segments

Clustering was undertaken with k-means using Python. This focussed on remuneration and spending score. Initially spending score and remuneration were plotted against each other (Figure 10) and a Seaborn pairplot was created (Figure 11). The Elbow and Silhouette methods were applied (Figure 12 & 13 respectively) which both indicated that 5 was the optimal number of clusters (the elbow or knee of Figure 12 and the highest point of Figure 13).

This was tested using clusters of 3, 4 and 5 (as shown in figures 14, 15 & 16 respectively). It was clear that the model using 5 as the value of k-means showed clearly distinct clusters and that was used for the final model. As the marketing department could now target five specific market segments, the number of observations per class was identified in order to assist with prioritisation (Table 10).

The clusters were then plotted for interpretation (Table 11 & Figure 17).

How social data can be used to inform marketing campaigns

The social data available was customer reviews, and summary data of the reviews, so natural language processing was employed. The data was prepared for NLP using Python by changing to lowercase, joining, removing punctuation, dropping duplicates and applying tokenisation. Wordclouds were created with the output for the customer reviews (Figure 18) and summary (Figure 19).

Frequency distribution was then established and alphanumeric characters and stopwords were removed. Wordclouds were created from the resulting data for customer reviews (Figure 20) and summary (Figure 21).

The 15 most commons words in each column were identified (Tables 12 & 13).

Sentiment intensity was then analysed for the respective columns.

For *review* a boxplot was created which showed generally positive sentiment (Figure 22), the polarity of views (Figure 23) and a histogram with 15 bins (for ease of interpretation – Figure 24). The same process was applied for *summary* (Figure 25-27).

The top 20 negative and top 20 positive records were identified (Tables 14 & 15 and Tables 16 and 17).

The review data and sentiment analysis is clearly of greater benefit to the marketing department. The summary data may have been produced by the customer service team in an attempt to address/resolve customer issues and as such most sentiment has been removed. There is a more neutral sentiment score on the summary data.

Positive sentiment can better inform marketing exercises as they will focus on the positive attributes of the products, in comparison to customer service persons who will focus on resolving issues (negative) for existing customers.

The impact that each product has on sales

The data was imported in RStudio and a new dataframe created for initial exploration. A scatterplot (Figure 28), histogram (Figure 29) and boxplot (Figure 30) were created to provide an initial insight into the sales figures.

While the initial scatterplot and histogram indicated that there was value in clustering the data using R at a later stage, the boxplot demonstrated that the bulk of products contributed less than £10m pounds in revenue.

Descriptive statistics were identified (Table 18).

A number of products are sold on multiple platforms (querying unique values) and the data was aggregated to group by product and the sum of revenue contribution.

A scatterplot (Figure 31) and boxplot (Figure 33) showed broadly similar results to the output of the earlier exploration. However, the histogram (Figure 32) was created using bins to crudely group the data into four categories, examining showing potential for clustering. At this exploratory stage, further dataframes were creating from subsetting the data based on those products which contributed more than £20m (Table 19), between £10m and £20m, and less than £10m to revenue.

It is clear that some products are markedly more/less successful than others and this will facilitate future decision making in relation to both stock and also the focus of marketing campaigns. This was tested (see Figure 34).

How reliable the data is

The data was tested for its reliability using a QQ plot (Figure 35) and the Shapiro-Wilk test (Table 20).

Skewness (Table 21) and Kurtosis (Table 22) were also established.

The QQ test shows that the data is not normally distributed, and that there are significant outliers in particular with the more highly performing products. The Shapiro-Wilk test also supports this as the result is less than 0.05 and so deviates significantly from a normal distribution.

The Skewness test demonstrates that the data is highly skewed. This is typically the case if it is greater than 1 and in this instance it is greater than 3.

Similarly, the Kurtosis test indicates that the data has very heavy tails. A heavy tails result is considered to be one greater than 3 and in this instance it is 17.

Note that these results do not necessarily indicate that the data is unreliable. It may be the case, that due to the nature of the product, that some products do contribute much more significantly to revenue than others. This is supported by the initial analysis and also by the correlation coefficient of -0.6 which indicates a strong relationship between product and revenue (Table 23).

What the relationship(s) is/are (if any) between North American, European, and global sales?

A simple linear regression was undertaken to examine the relationship between Year and Global Sales and the R value of 0.05 demonstrates a low fit (Figure 36).

In contrast, the simple linear regression between North American and global sales provided a very strong fit (R-squared) of 0.83 (Figure 37), and a SLR between European and global sales also demonstrated a strong fit of 0.77 (Figure 38). The fit between North American sales and European sales following SLR showed a medium strength fit of 0.49 (Figure 39). Correlations are summarised in Table 24.

The level of correlation between all numeric variables were established and plotted (Figure 40). This showed two distinct types of data – data relating to the products (Ranking, Product and Year) and data relating to sales (Global sales, North American sales and European sales).

The is a high correlation between North American sales, European sales and global sales indicating a very strong relationship. There is a low correlation between the Product, Ranking, Year and Global sales.

Multiple linear regression was carried out on the sales data, producing an adjusted r-squared of 0.96, higher than the North American sales or European sales analysed separately.

A second multiple linear regression was carried out on Global Sales, Year, Ranking and Product – Adjusted R-squared 0.36 – more significant than year (0.05) or ranking (0.15) or product (0.19) alone.

The sales data was used to develop predictions as to future Global Sales for a listed set of products (Table 25).

Wordcount: 1,180

Appendix 1 – Technical process

Using Python

The relevant packages were imported and the csv file loaded using **pd.read.csv**.

The data was checked for missing values, **reviews_na.shape** and explored, **reviews.info()**

The descriptive statistics were queried, **reviews.describe()**

Unnecessary columns were dropped, e.g. **reviews.drop('language', inplace=True, axis=1)**

Columns were renamed, **reviews.rename(columns={'remuneration (k£)': 'remuneration', 'spending_score (1-100)': 'spending_score'}, inplace=True)**

The dataframe was exported as a csv using **.to_csv()** and then imported as a csv **pd.read_csv()**

A new dataframe was created to examine spending score and loyalty points.

The Pandas correlation method was applied, **.corr()**

The maximum values for both independent and dependent variables was established using **.max()**

As seaborn scatterplot was created before running linear regression using **sns.scatterplot()** and the axis values were set using **plt.ylim()** and **plt.xlim()**

Polyfit was used for simple linear regression to predict the loyalty points based on spending score using **np.polyfit()**

A trendline was added using **np.polyval()** and plotted using **plt.plot()**

The Ordinarily Least Squares method was also applied by creating a formula **f = 'y ~ x'** and then using **ols().fit()**

The estimated parameters, standard error and predicted values were extracted using **test.params**, **test.bse** and **test.predict()** respectively.

The X coefficient and the constant were set to generate the regression table. **y_pred = (-75.0526) + 33.0616* spending_loyalty['spending_score']**

The graph was plotted with the regression line using **plt.scatter()** and **plt.plot()** and the x and y axes limits were set.

This process was repeated for remuneration v loyalty and age v loyalty.

In order to cluster with k-means using Python, the relevant packages and library were loaded.

The dataset was imported, unnecessary columns were dropped and the data explored.

Pairplots were created with seaborn using **sns.pairplot()**

The **Elbow** and **Silhouette** methods were applied to establish optimal number of clusters.

Pairplots were created to test the outcome. 5 clusters was the optimal number and this was then visualised, the number of observations per class established and the predicted K-means outlined.

In order to apply NLP using Python the relevant packages were installed and libraries loaded.

The data was then imported.

For both summary and review all was chanegd to lower case and join with a space. eg. **df_nlp['review'] = df_nlp['review'].apply(lambda x: " ".join(x.lower() for x in x.split()))**

All the punctuation was replaced in both columns. eg. **df_nlp['review'] = df_nlp['review'].str.replace('[^\w\s]','')**

A check for duplicates was carried out using **duplicated().sum()** and duplicates were then dropped **drop_duplicates(subset=['review'])**

Tokenisation was applied to both columns tokenisation to both review columns eg. **df_token_review = [word_tokenize(_) for _ in df_clean2.review]**

String all the text together in a single variable for each column:

```
all_review = ''

for i in range(df_clean2.shape[0]):

   # Add each comment.

   all_review = all_review + df_clean2['review'][i]
```

Wordclouds were plotted:

```
wordcloud = WordCloud(width = 1600, height = 900,

            background_color ='white',

            colormap='plasma',

            min_font_size = 10).generate(all_review)

# Plot the WordCloud image.

plt.figure(figsize = (16, 9), facecolor = None)

plt.imshow(wordcloud)

plt.axis('off')

plt.tight_layout(pad = 0)

plt.show()
```

Frequency distribution was calculated:

```
fdist = FreqDist()

for word in word_tokenize(all_review):

   fdist[word.lower()] += 1
```

All alphanumerics characters were removed from each **column all_review = [word for word in all_review if word.isalnum()]** and stopwords were removed: **fdist1 = [x for x in fdist if x.lower() not in english_stopwords]**

Wordclouds were then created without stopwords.

In order to identify the 15 most common words and polarity in each column **FreqDist()** was used.

Counter was imported and a dataframe was generated from counter:
pd.DataFrame(Counter(fdist).most_common(15),

columns=['Word', 'Frequency']).set_index('Word')

For polarity, a function was created:

def generate_polarity(df_token_review):

'''Extract polarity score (-1 to +1) for each comment'''

return TextBlob(comment).sentiment[0]

Sentiment Intensity Analyser was then used to determine polarity: **df_polarity = {" ".join(_) : sia.polarity_scores(" ".join(_)) for _ in df_token_review}**

A dataframe of sentiment scores (pos, neg, neu and compound) was then produced for each row in each column whichw as then plotted.

sort_values() and **.head()** was then used to establish the 20 most positive/negative entries in each column.

Using R

Data was loaded and explored.

Tidyverse installed and import.

The dataset was imported using **read.csv**

The dataset was sense checked using **head()** and then viewed using **view()**

A new dataframe was created and unnecessary columns were not included.

df2 <- sales[,c('Product','Platform','NA_Sales','EU_Sales','Global_Sales')]

This was then viewed, **view()**

The structure of the dataframe was checked using **str()**

Product was converted to a string using **as.character(df2$Product)**

The dplyr library was loaded and distinct/unique entries were checked. **n_distinct(df2$Product)**

unique(df2$Product)

A **summary()** was queried to view the descriptive statistics.

The dataframe was sorted by Global_Sales using **order()**

A scatterplot, histogram and boxplot were created using **qplot()**

The min, max and mean values were determined using **print(min())**, **print(max())** and **print(mean())** functions.

Again descriptive statistics were checked using **summary()**

The data was grouped based on Product determining the sum of sales per Product.

df3 <- df2 %>% group_by(Product) %>%

summarise(Total_Sales=sum(Global_Sales), Total_NA=sum(NA_Sales), Total_EU=sum(EU_Sales),

.groups='drop')

Additional variables were created summing NA and EU sales, and to establish a figure for the Other category

df3$Total_EU_NA <- df3$Total_NA + df3$Total_EU

df3$Total_Other <- (df3$Total_Sales - df3$Total_EU_NA)

Scatterplots, histograms, boxplots and barplots were grated using **qplot()** and **ggplot()** and bins were used for the histogram.

Subsets were created to analyse the data in groups based on total sales (e.g. >20m)

top_products <- data.frame(subset(df, Total_Sales >20))

view(top_products)

A Q-Q plot was created to determine normality **qqnorm()** with a reference line added **qqline()**

The moments library was imported, **library()**, and Shapiro-Wilk test carried out **shapiro.test(())** to query distribution of data.

Skewness and Kurtosis tests were carried out **skewness()** and **kurtosis()**

A correlation check was carried out, **cor(df3$a, df3$b)**

A new dataframe was created on the original data using only numeric columns.

sales1 = data.frame(sales$Ranking,sales$Product,sales$Year,sales$NA_Sales,sales$EU_Sales,sales$Global_Sales)

Structure was checked **str()**

Correlations were checked **cor()**

A number of simple linear regression plots were created. For example,

plot(sales1$sales.Year, sales1$sales.Global_Sales)

model1 <- lm(sales.Global_Sales~sales.Year,

data=sales1)

abline(coefficients(model1))

The summary regression data was queried, **summary(model1)**

Psych package was installed and the relevant library called.

Correlations were plotted and size specified using **corPlot(cex=2)**

A number of multiple linear regression models were created

model5 = lm(sales.Global_Sales~sales.Year+sales.Ranking+sales.Product, data=sales1)

The regression data was queried for these models, eg. **summary(model5)**

Predictions were prepared based on given values by comparing with observed values for a number of records.

```
predictTest = predict(model6, data=sales1$sales.NA_Sales,
                      interval='confidence')
```

A new dataframe was created from rows with the specified values

```
sales2 = data.frame(sales1[sales1$sales.NA_Sales %in% c('34.02','3.93','2.73','2.26','22.08'),])
```

A second new dataframe was created using the identified row-names from sale2 to extract appropriate rows from predictTest outcome.

These dataframes were then merged:

```
data_frame_merge <- merge(sales2, predictTest2,
                          by = 'row.names', all = TRUE)
view(data_frame_merge)
```

Appendix 2 – Outputs

Table 1 *– Pearson's Correlation Coefficient – spending score v loyalty points*

	spending_score	loyalty_points
spending_score	1.00000	0.67231
loyalty_points	0.67231	1.00000

Figure 1 *– Spending score v loyalty points scatterplot*

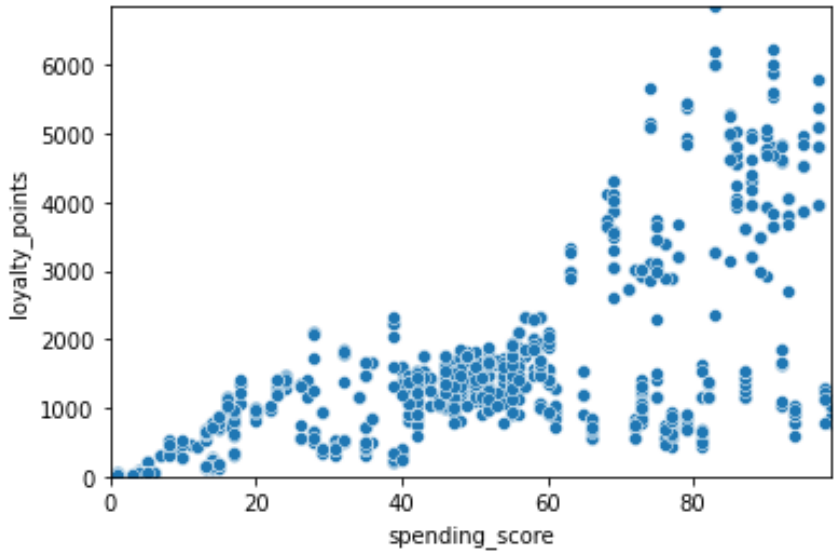

Figure 2 *– Spending score v loyalty points scatterplot with reference line*

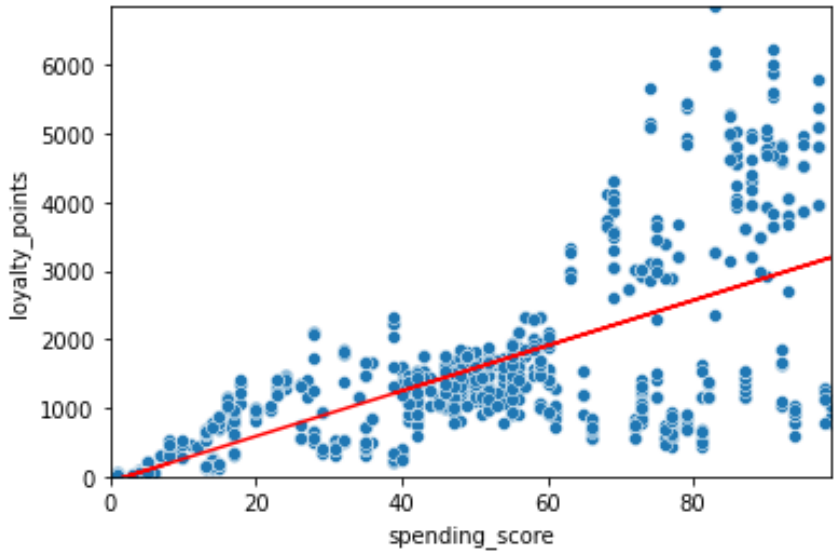

Table 2 *– OLS linear regression spending score v loyalty points*

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.452
Model:	OLS	**Adj. R-squared:**	0.452
Method:	Least Squares	**F-statistic:**	1648.
Date:	Mon, 19 Dec 2022	**Prob (F-statistic):**	2.92e-263
Time:	11:28:52	**Log-Likelihood:**	-16550.
No. Observations:	2000	**AIC:**	3.310e+04
Df Residuals:	1998	**BIC:**	3.312e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-75.0527	45.931	-1.634	0.102	-165.129	15.024
x	33.0617	0.814	40.595	0.000	31.464	34.659

Omnibus:	126.554	**Durbin-Watson:**	1.191
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	260.528
Skew:	0.422	**Prob(JB):**	2.67e-57
Kurtosis:	4.554	**Cond. No.**	122.

Table 3 *– Regression table spending v loyalty points*

```
0       1214.3498
1       2602.9370
2        123.3170
3       2470.6906
4       1247.4114
          ...
1995    2206.1978
1996     189.4402
1997    2933.5530
1998     453.9330
1999     189.4402
Name: spending_score, Length: 2000, dtype: float64
```

Figure 3 *– OLS method output spending score v loyalty points*

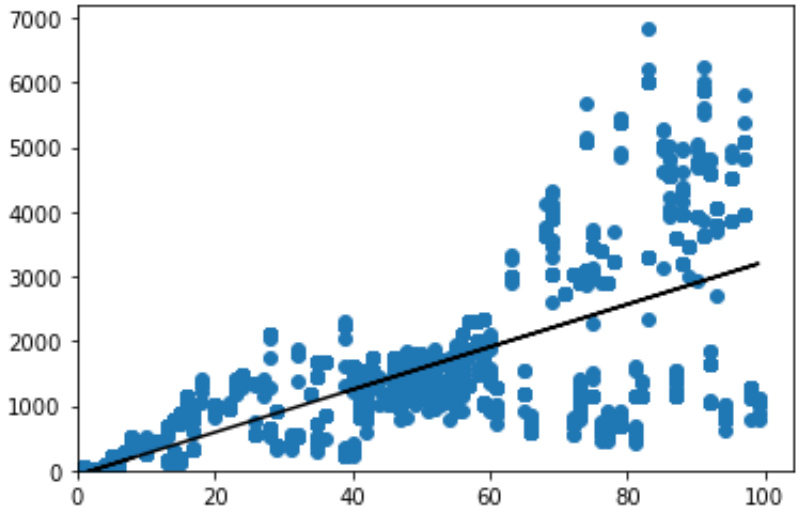

Table 4 *– Pearson's Correlation Coefficient -remuneration v loyalty points*

	loyalty_points	
remuneration		
remuneration	1.000000	0.616065
loyalty_points	0.616065	1.000000

Figure 4 *- Remuneration v loyalty points scatterplot*

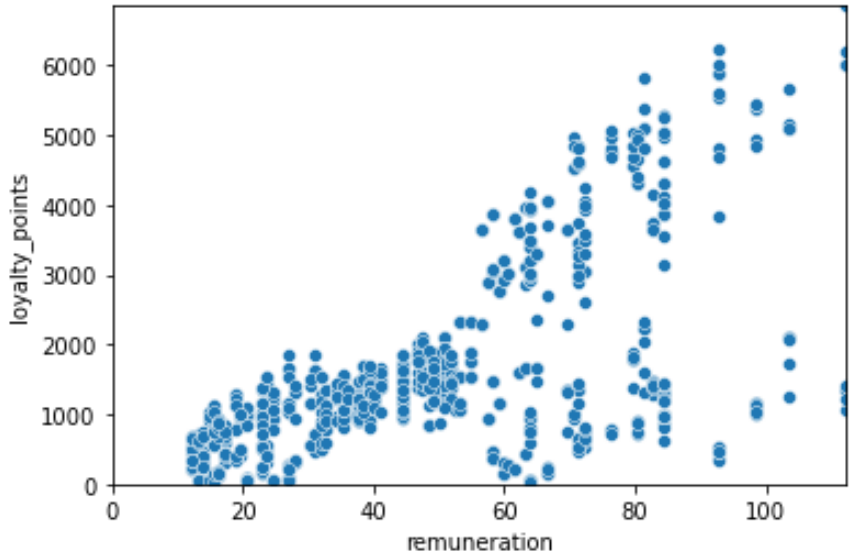

Figure 5 *- Remuneration v loyalty points scatterplot with reference line*

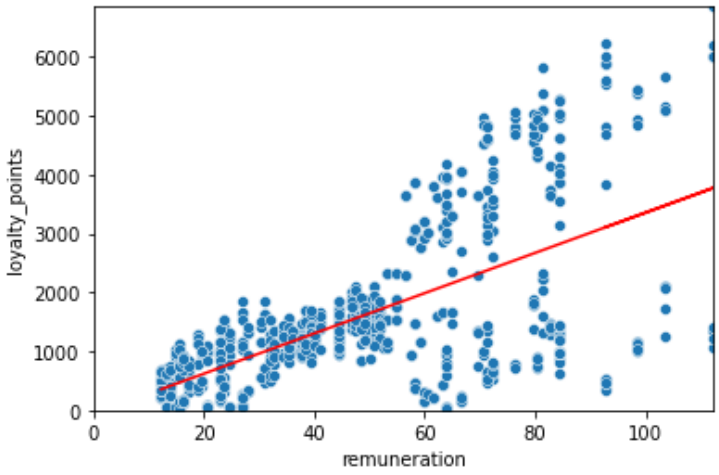

Table 5 *- OLS linear regression remuneration v loyalty points*

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.380
Model:	OLS	**Adj. R-squared:**	0.379
Method:	Least Squares	**F-statistic:**	1222.
Date:	Mon, 19 Dec 2022	**Prob (F-statistic):**	2.43e-209
Time:	11:28:53	**Log-Likelihood:**	-16674.
No. Observations:	2000	**AIC:**	3.335e+04
Df Residuals:	1998	**BIC:**	3.336e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-65.6865	52.171	-1.259	0.208	-168.001	36.628
x	34.1878	0.978	34.960	0.000	32.270	36.106

Omnibus:	21.285	**Durbin-Watson:**	3.622
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	31.715
Skew:	0.089	**Prob(JB):**	1.30e-07
Kurtosis:	3.590	**Cond. No.**	123.

Table 6 *- Regression table remuneration v loyalty points*

```
0         354.823440
1         354.823440
2         382.857436
3         382.857436
4         410.891432
             ...
1995     2821.815088
1996     3102.155048
1997     3102.155048
1998     3298.393020
1999     3102.155048
Name: remuneration, Length: 2000, dtype: float64
```

Figure 6 *- OLS method output remuneration v loyalty points*

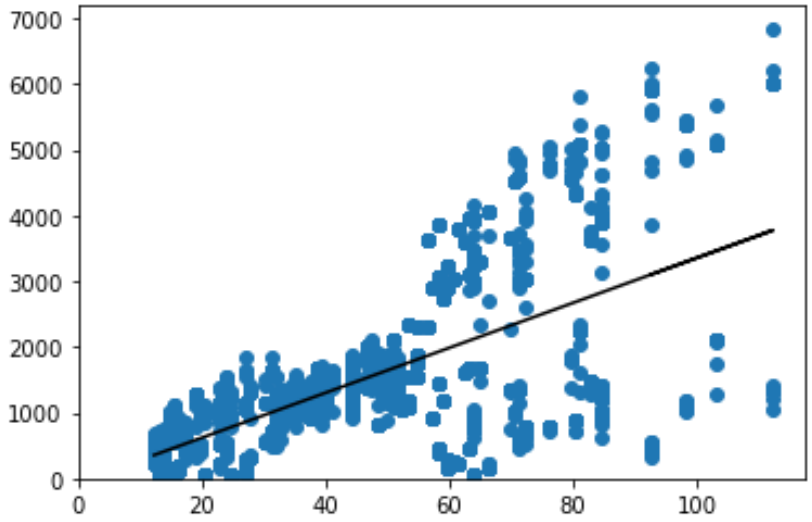

Table 7 *- Pearson's Correlation Coefficient - age v loyalty points*

	age	loyalty_points
age	1.000000	-0.042445
loyalty_points	-0.042445	1.000000

Figure 7 *- Remuneration v loyalty points scatterplot*

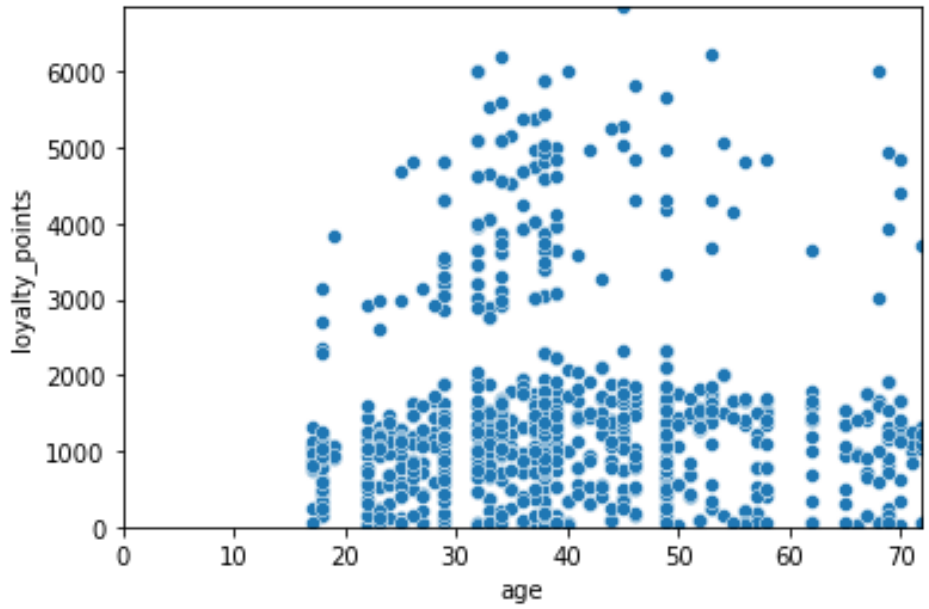

Figure 8 *- Age v loyalty points scatterplot with reference line*

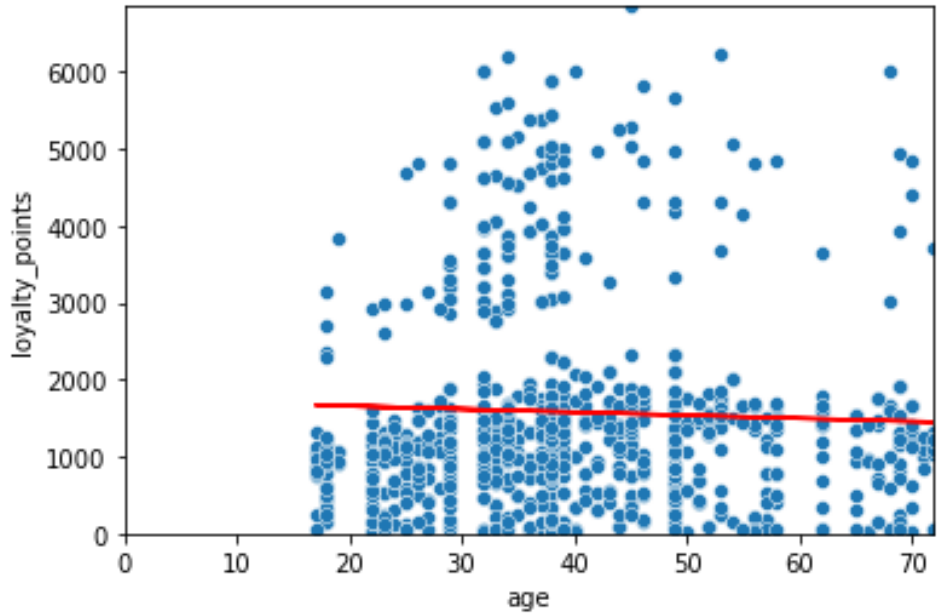

Table 8 *- OLS linear regression age v loyalty points*

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.002
Model:	OLS	**Adj. R-squared:**	0.001
Method:	Least Squares	**F-statistic:**	3.606
Date:	Mon, 19 Dec 2022	**Prob (F-statistic):**	0.0577
Time:	11:28:53	**Log-Likelihood:**	-17150.
No. Observations:	2000	**AIC:**	3.430e+04
Df Residuals:	1998	**BIC:**	3.431e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	1736.5177	88.249	19.678	0.000	1563.449	1909.587
x	-4.0128	2.113	-1.899	0.058	-8.157	0.131

Omnibus:	481.477	**Durbin-Watson:**	2.277
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	937.734
Skew:	1.449	**Prob(JB):**	2.36e-204
Kurtosis:	4.688	**Cond. No.**	129.

Table 9 *- Regression table age v loyalty points*

```
0       1664.287210
1       1644.223185
2       1648.235990
3       1636.197575
4       1604.095135
           ...
1995    1588.043915
1996    1563.967085
1997    1600.082330
1998    1600.082330
1999    1608.107940
Name: age, Length: 2000, dtype: float64
```

Figure 9 *- OLS method output age v loyalty points*

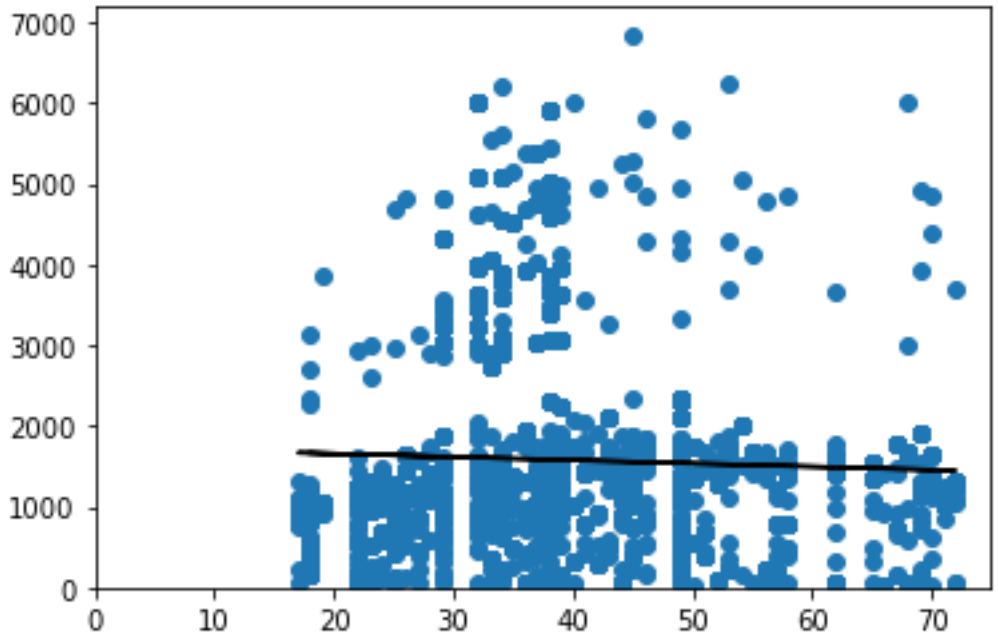

Figure 10 *– Remuneration v spending score*

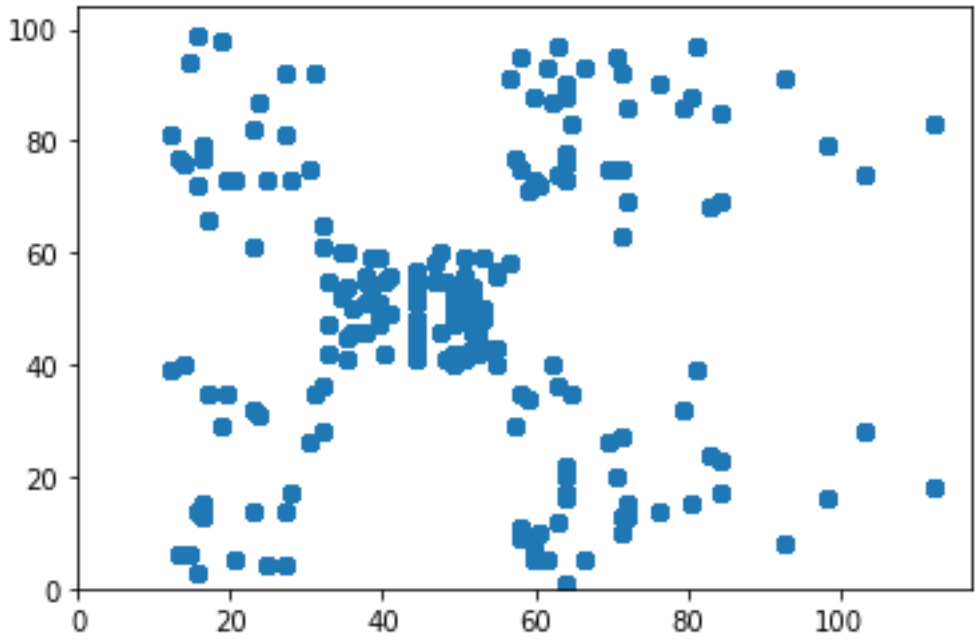

Figure 11 *– Pairplot remuneration v spending score*

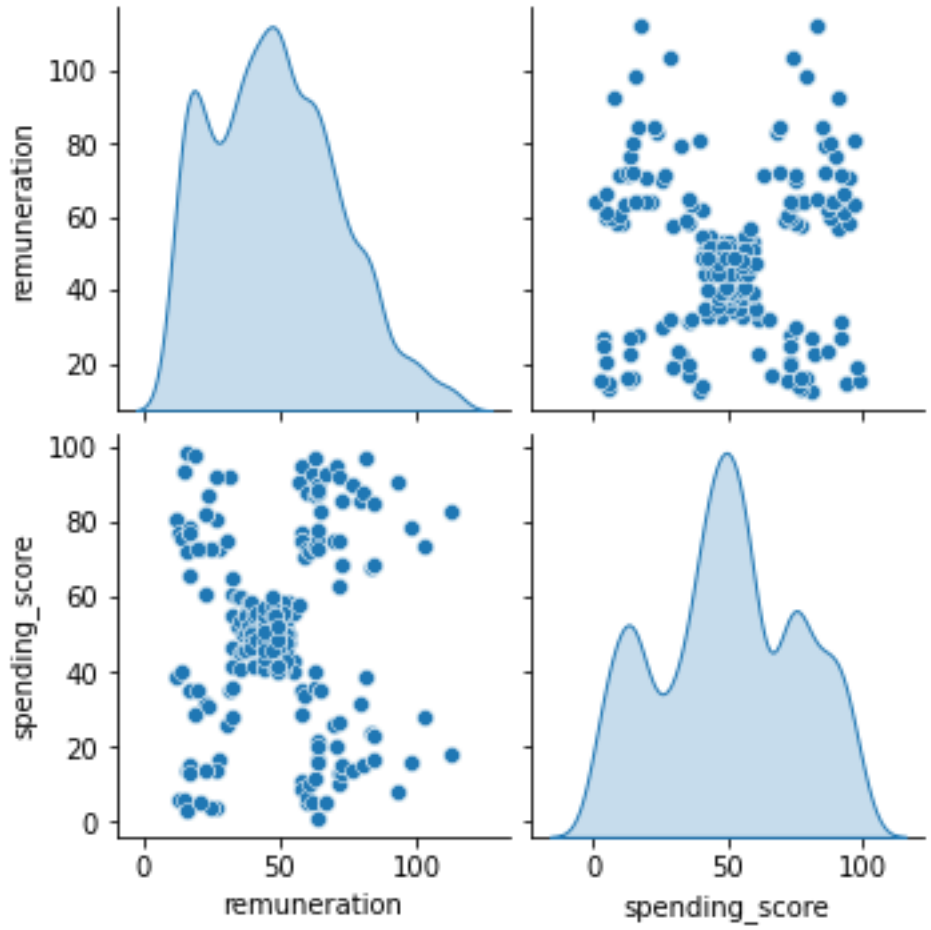

Figure 12 – *The Elbow Method*

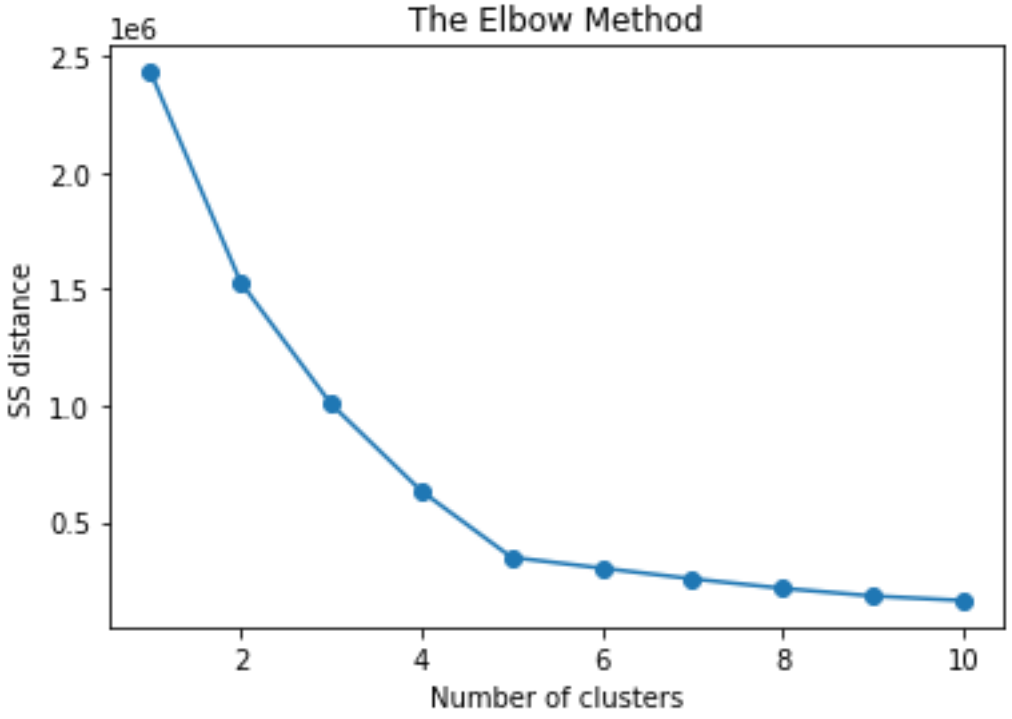

Figure 13 *– The Silhouette Method*

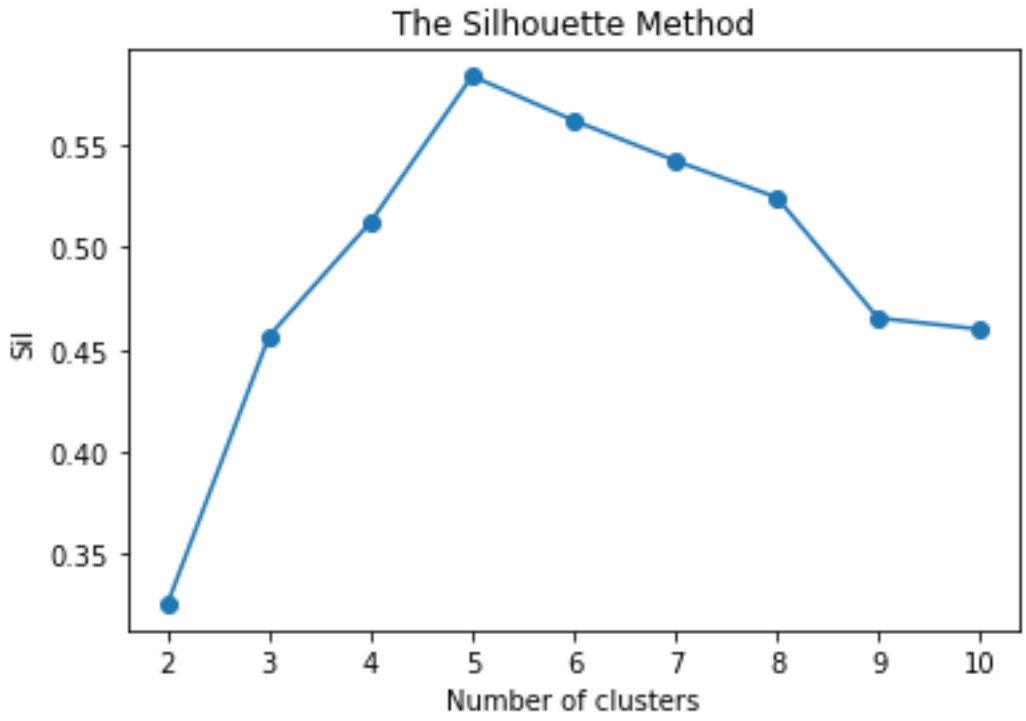

Figure 14 *– Pairplot with k-means = 3*

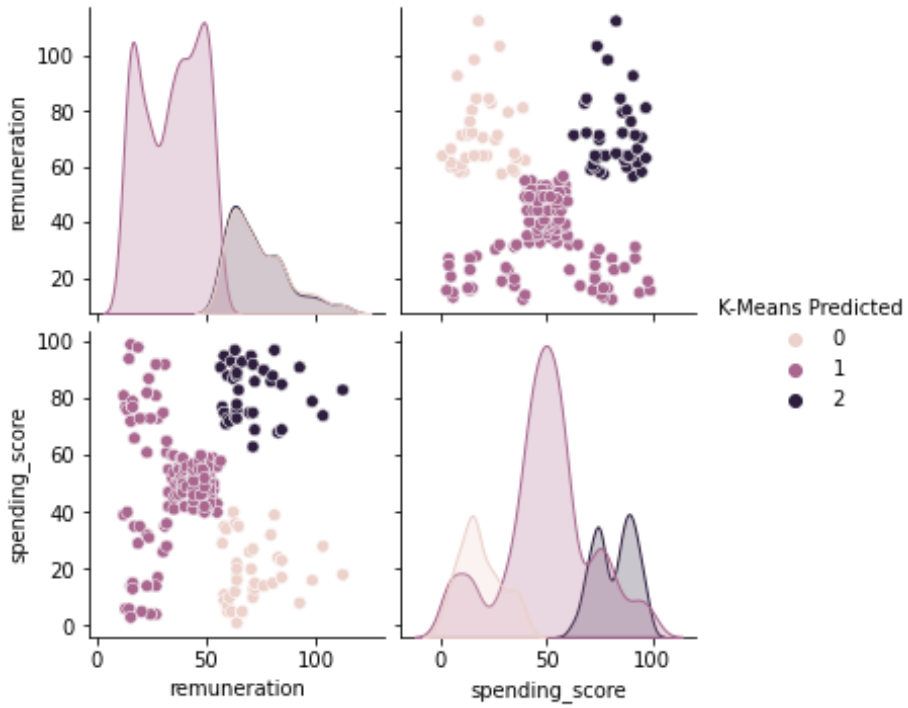

Figure 15 *– Pairplot with k-means = 4*

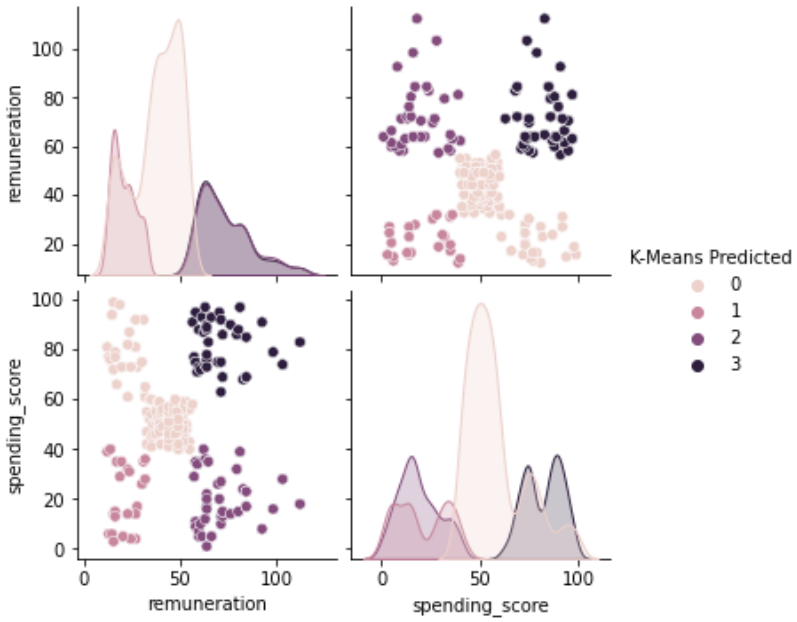

Figure 16 *– Pairplot with k-means = 5*

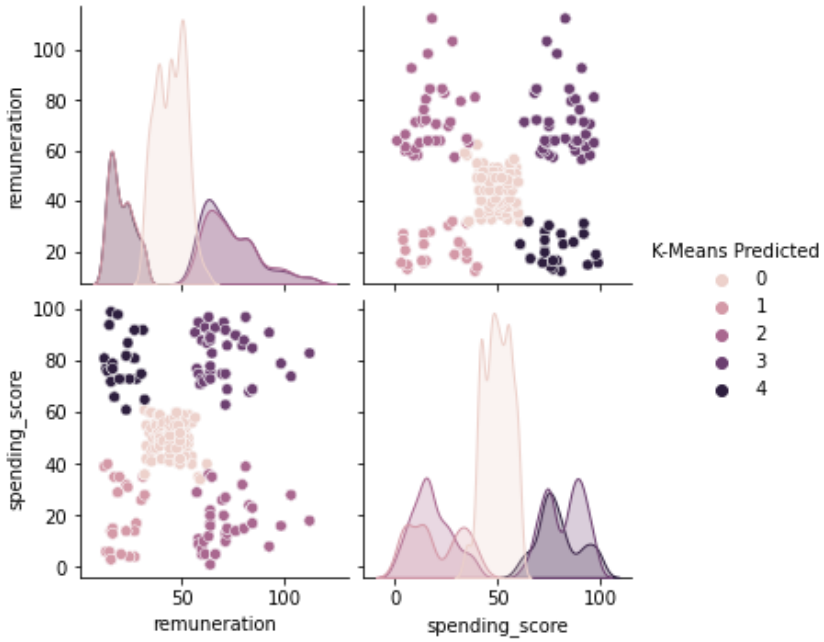

Figure 17 *– Scatterplot with k-means = 5*

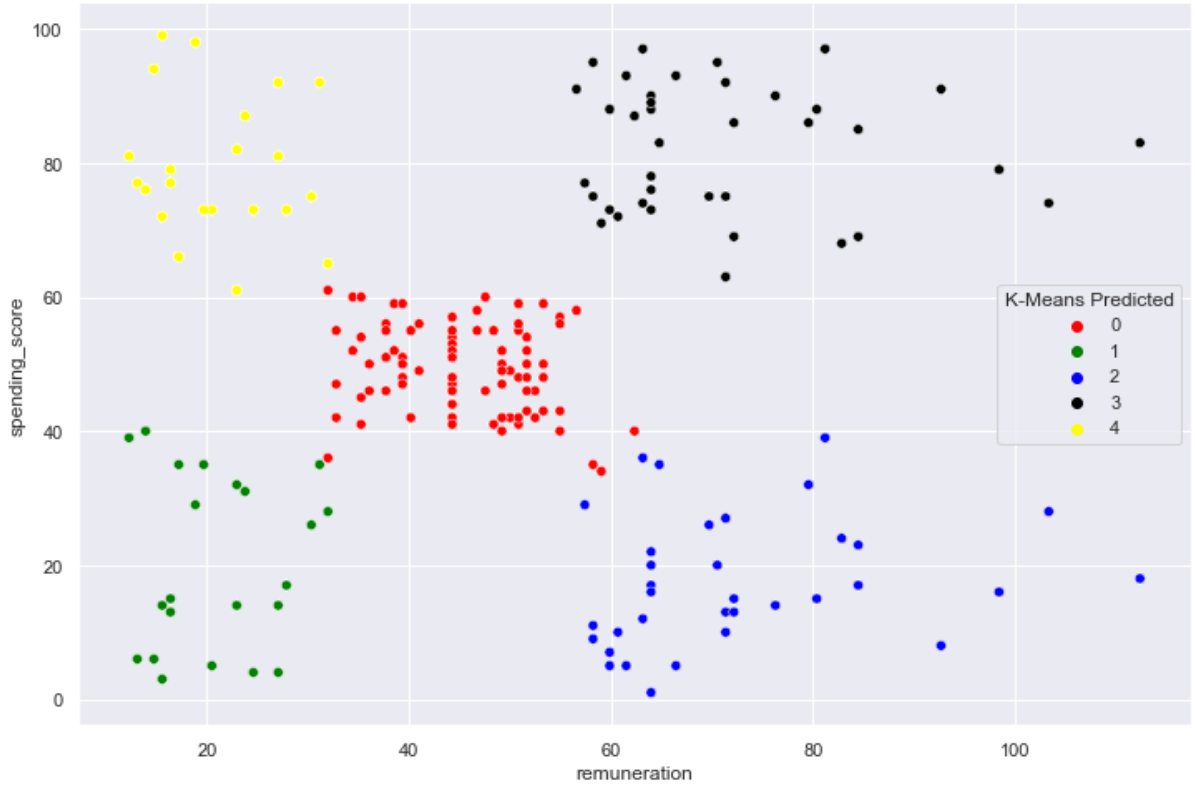

Table 10 *– Number of observations per cluster*

```
0    774
3    356
2    330
1    271
4    269
Name: K-Means Predicted, dtype: int64
```

Table 11 *– Predicted K-means*

```
   remuneration  spending_score  K-Means Predicted
0         12.30              39                  1
1         12.30              81                  4
2         13.12               6                  1
3         13.12              77                  4
4         13.94              40                  1
```

Figure 18 *– Wordcloud – customer reviews*

Figure 19 *– Wordcloud - summary*

Figure 20 *– Wordcloud - review without alphanumeric characters or stopwords*

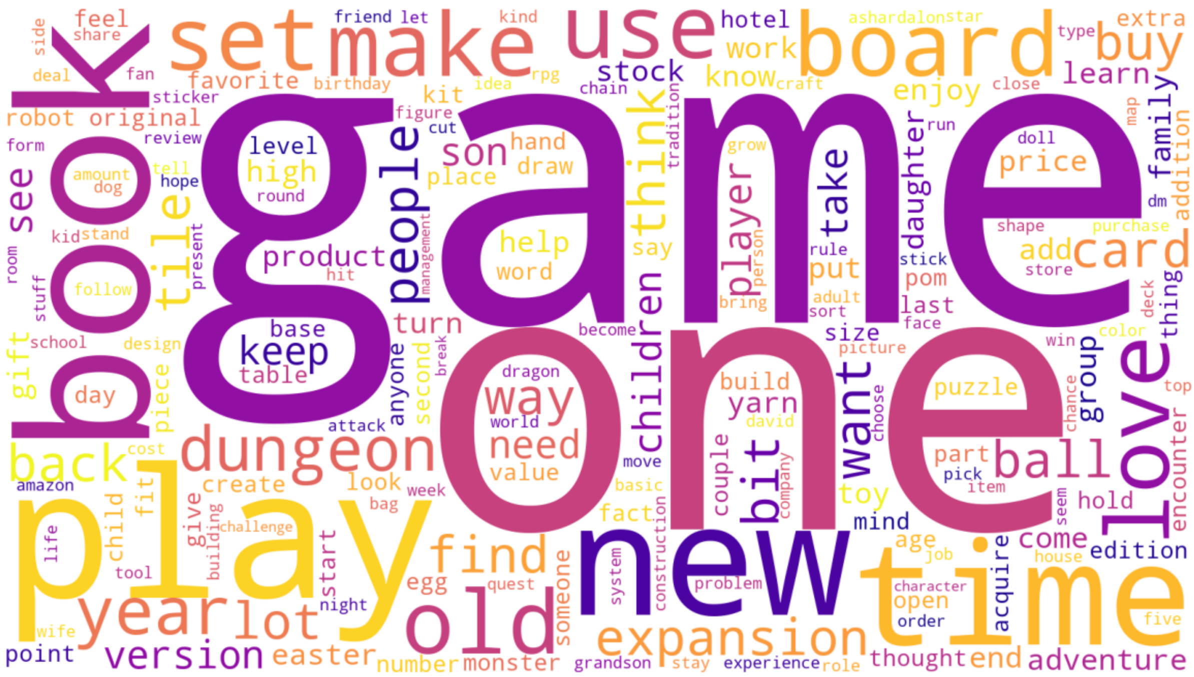

Figure 21 *– Wordcloud - summary without alphanumeric characters or stopwords*

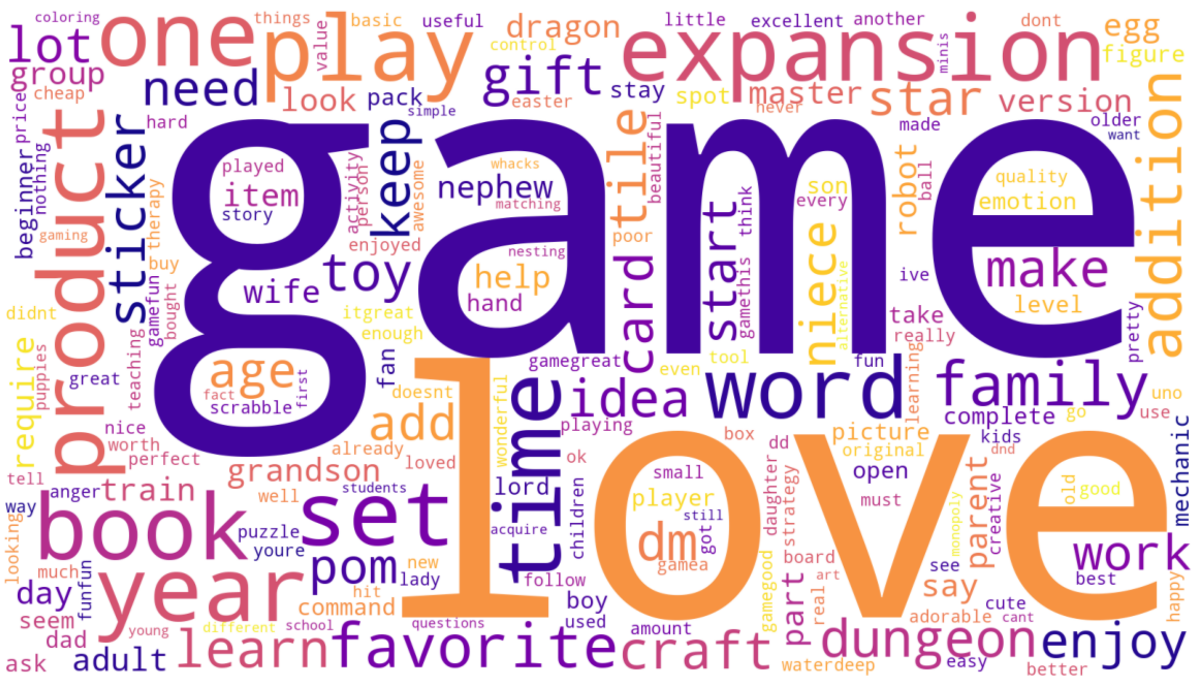

Table 12 *– 15 most common words in review*

Word	Frequency
the	4938
and	2934
to	2843
a	2796
of	2259
it	1748
i	1669
is	1590
this	1357
for	1352
game	1303
with	1116
you	1078
in	1061
that	988

Table 13 *– 15 most common words in summary*

Word	Frequency
for	224
the	222
to	190
a	184
game	162
and	161
of	131
is	101
great	100
it	88
fun	84
this	78
but	73
with	70
i	55

Figure 22 *– Boxplot of sentiment score - reviews*

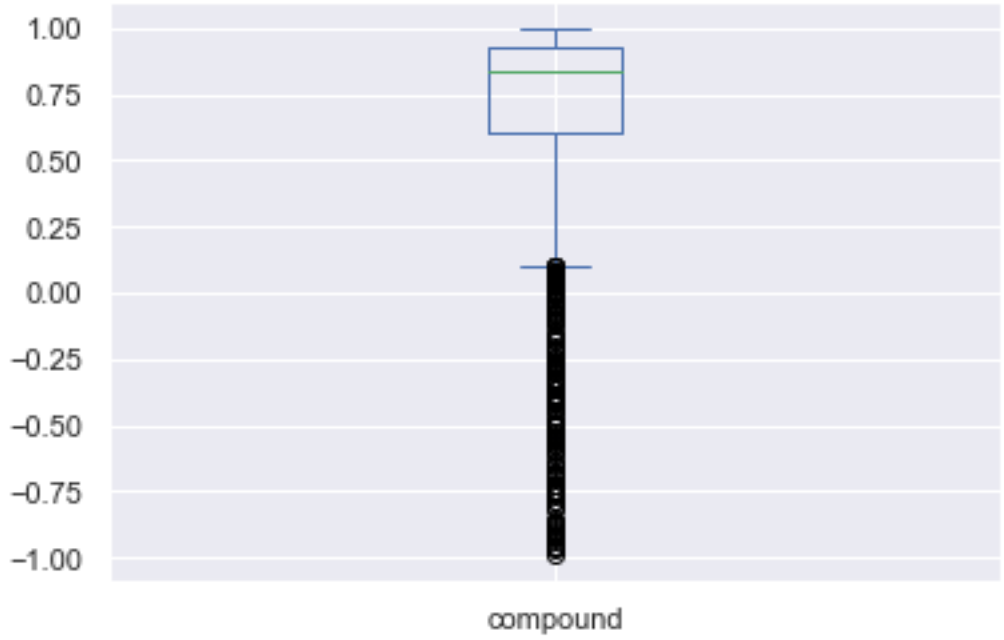

Figure 23 *– Polarity of sentiment score - reviews*

Figure 24 *– Histogram of sentiment scores - reviews*

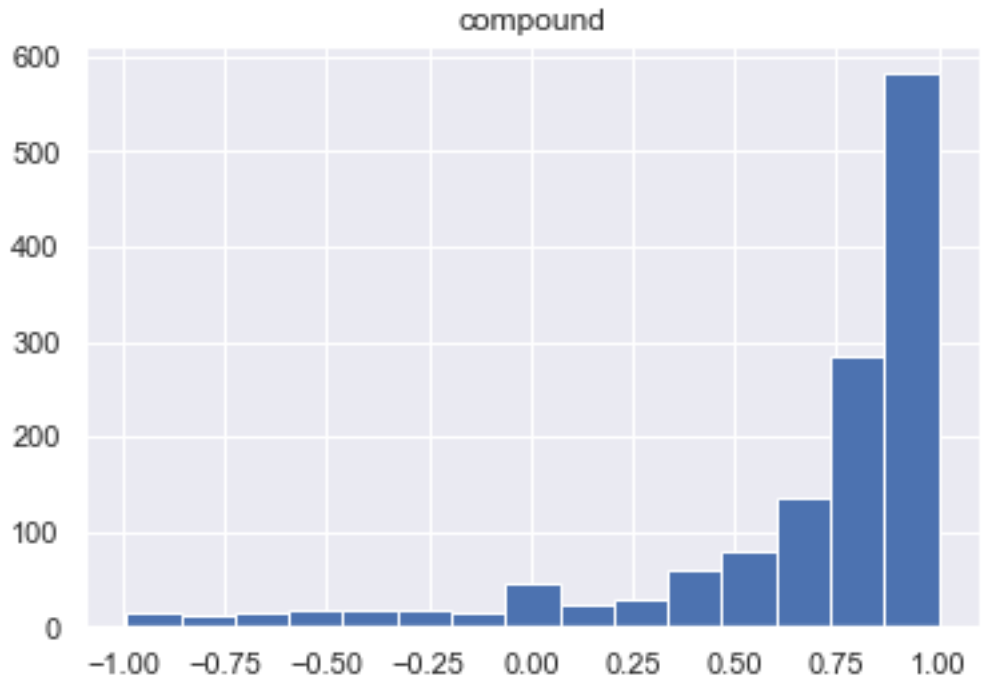

Figure 25 *– Boxplot of sentiment scores - summary*

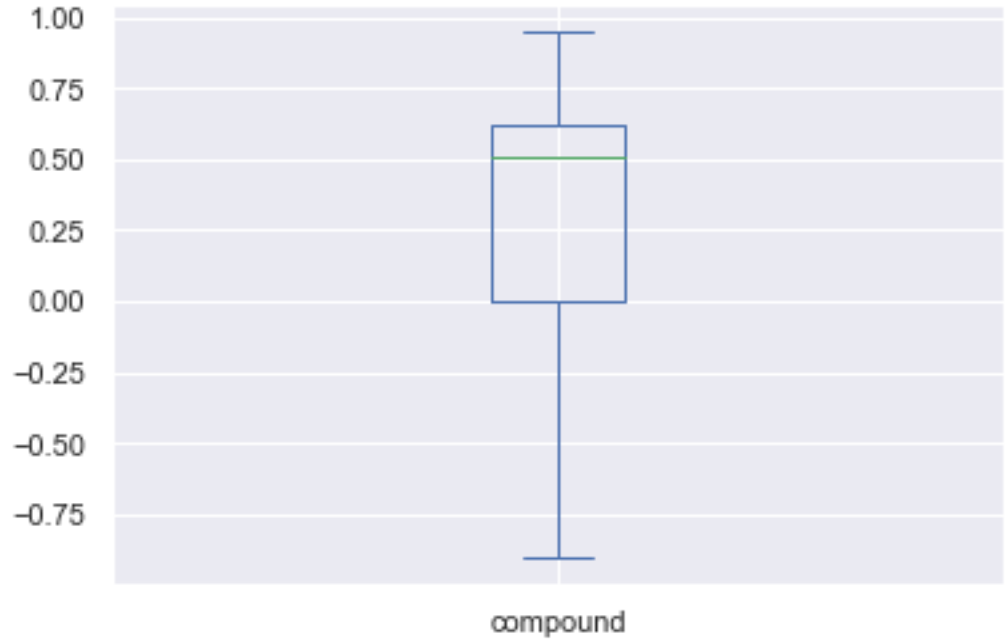

Figure 26 *– Polarity of sentiment scores - summary*

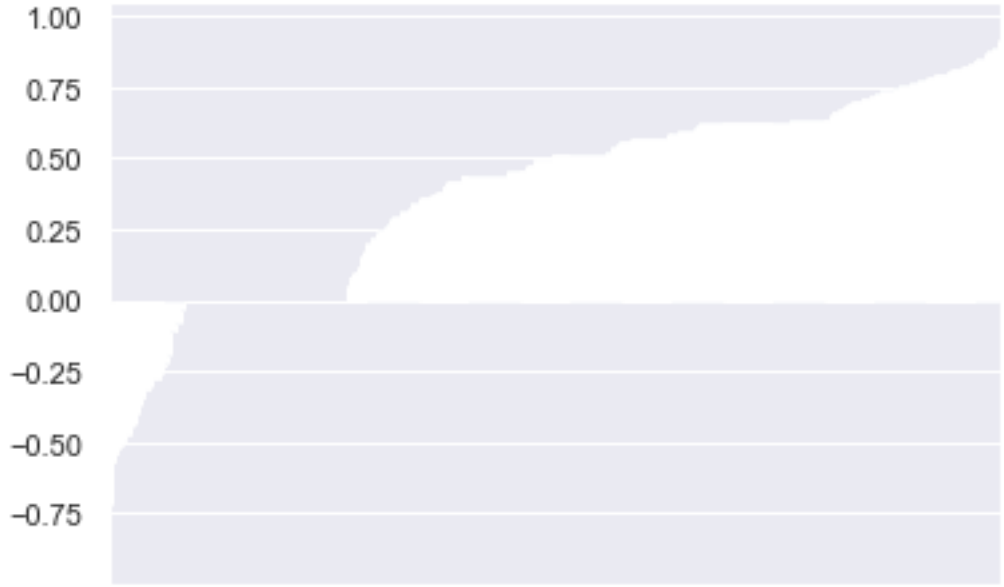

Figure 27 *– Histogram of sentiment scores - summary*

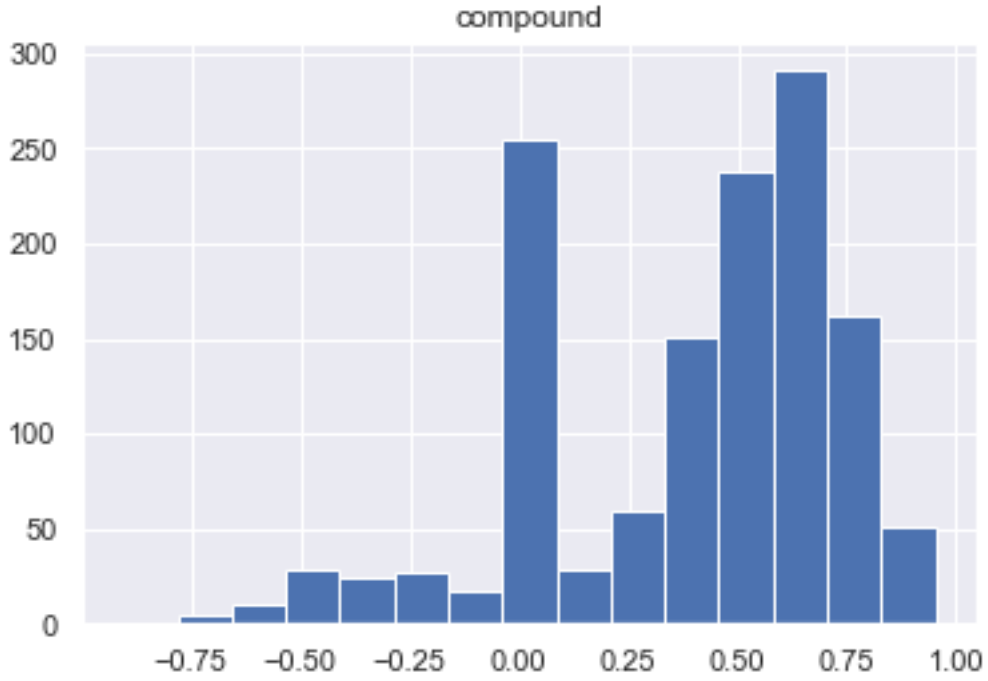

Table 14 *– Top 20 negative reviews*

neg	neu	pos	compound	
incomplete kit very disappointing	0.538	0.462	0.000	-0.5413
a crappy cardboard ghost of the original hard to believe they did this but they did shame on hasbro disgusting	0.487	0.455	0.058	-0.9052
got the product in damaged condition	0.367	0.633	0.000	-0.4404
i bought this thinking it would be really fun but i was disappointed its really messy and it isnt nearly as easy as it seems also the glue is useless for a 9 year old the instructions are very difficult	0.362	0.592	0.045	-0.9520
not as easy as it looks	0.325	0.675	0.000	-0.3412
we really did not enjoy this game	0.325	0.675	0.000	-0.4389
hard to put together	0.318	0.682	0.000	-0.1027
my 8 yearold granddaughter and i were very frustrated and discouraged attempting this craft it is definitely not for a young child i too had difficulty understanding the directions we were very disappointed	0.318	0.613	0.069	-0.8674
easytouse great for anger management groups	0.314	0.339	0.347	0.1027
its ok but loses its luster quickly	0.309	0.524	0.168	-0.3291
smaller than we thought kind of disappointed in it	0.298	0.702	0.000	-0.5256
i really like this game it helps kids recognize anger and talk about difficult emotions	0.287	0.463	0.250	-0.2040
its uno with questions about anger its an okay way to discuss anger but it gets repetitive and the students start to get bored after about half a round	0.287	0.671	0.042	-0.8126
its really uno type game but anger control stuffi dont like it due to younger children really dont respond well to anger management techniquesthey do better with expectation management instead these are ok but i wouldnt buy them again	0.285	0.592	0.123	-0.8668
they were ok but not really considered it a book really small disappointed	0.283	0.619	0.099	-0.6082
horrible nothing more to say would give zero stars if possible	0.259	0.741	0.000	-0.5423
cute idea horrible execution if you want your child in tears then this is your book my seven year old got very frustrated with this whole thing	0.257	0.622	0.121	-0.6997
very fun game to use with kids working on handling anger you play like uno but have to answer questions about anger	0.245	0.564	0.191	-0.5652
these are nice enough but probably not worth the price i didnt love how its missing certain letters not one p so my child could spell her name also missing other basic letters while giving a few too many qs	0.244	0.602	0.154	-0.6706
this is horrible the directions are very hard for a child to read and comprehend themselves the yarn made a mess my daughter was so excited to get this and cried when she couldnt understand how to make them i would not recommend this to anyone	0.236	0.705	0.059	-0.8067

Table 15 *– Top 20 positive reviews*

neg	neu	pos	compound	
cute	0.0	0.000	1.000	0.4588
perfect	0.0	0.000	1.000	0.5719
fun gift	0.0	0.000	1.000	0.7351
entertaining	0.0	0.000	1.000	0.4404
fun good service	0.0	0.139	0.861	0.7351
its fun	0.0	0.233	0.767	0.5106
very cute	0.0	0.233	0.767	0.5095
liked it	0.0	0.263	0.737	0.4215
a fun game we enjoy it a great deal	0.0	0.274	0.726	0.8910
i like pie	0.0	0.286	0.714	0.3612
great easter gift for kids	0.0	0.300	0.700	0.7906
a great creation tool it helps me concentrate	0.0	0.312	0.688	0.8316
good pricegame is fun cant really complain	0.0	0.319	0.681	0.8189
yes quick wonderful and accurate	0.0	0.319	0.681	0.7506
kids love it	0.0	0.323	0.677	0.6369
excellent stickers my grand daughter loves peppa pig	0.0	0.325	0.675	0.8860
these are great	0.0	0.328	0.672	0.6249
gorgeous i love the book and the pictures are beautiful	0.0	0.331	0.669	0.9201
my favorite game made better	0.0	0.337	0.663	0.7096
recipient loved this and it looked like fun	0.0	0.340	0.660	0.8658

Table 16 *– Top 20 negative summaries*

neg	neu	pos	compound	
disappointing	1.000	0.000	0.0	-0.4939
meh	1.000	0.000	0.0	-0.0772
boring	1.000	0.000	0.0	-0.3182
disappointed	1.000	0.000	0.0	-0.4767
frustrating	1.000	0.000	0.0	-0.4404
defective poor qc	0.857	0.143	0.0	-0.7184
not great	0.767	0.233	0.0	-0.5096
mad dragon	0.762	0.238	0.0	-0.4939
no 20 sided die	0.753	0.247	0.0	-0.7269
damaged product	0.744	0.256	0.0	-0.4404
money trap	0.697	0.303	0.0	-0.3182
faulty product	0.697	0.303	0.0	-0.3182
nothing special	0.693	0.307	0.0	-0.3089
wimpy magnets	0.655	0.345	0.0	-0.2263
anger control game	0.649	0.351	0.0	-0.5719
box totally destroyed	0.636	0.364	0.0	-0.5413
really small disappointed	0.628	0.372	0.0	-0.5233
da bomb game	0.615	0.385	0.0	-0.4939
a disappointing coop game	0.615	0.385	0.0	-0.4939
very weak game	0.615	0.385	0.0	-0.4927

Table 17 *– Top 20 positive summaries*

	neu	pos	compound	neg
awesome	0.0	0.0	1.0	0.6249
great gift	0.0	0.0	1.0	0.7906
precious	0.0	0.0	1.0	0.5719
pretty cool	0.0	0.0	1.0	0.6705
wow	0.0	0.0	1.0	0.5859
ok ok	0.0	0.0	1.0	0.5267
beautiful	0.0	0.0	1.0	0.5994
perfect	0.0	0.0	1.0	0.5719
great	0.0	0.0	1.0	0.6249
ok	0.0	0.0	1.0	0.2960
nifty	0.0	0.0	1.0	0.4019
entertaining	0.0	0.0	1.0	0.4404
super fun	0.0	0.0	1.0	0.8020
good	0.0	0.0	1.0	0.4404
nice	0.0	0.0	1.0	0.4215
cute	0.0	0.0	1.0	0.4588
wonderful	0.0	0.0	1.0	0.5719
fantastic	0.0	0.0	1.0	0.5574
great helper	0.0	0.0	1.0	0.7579
brilliant	0.0	0.0	1.0	0.5859

Figure 28 *– Scatterplot of Global Sales using R*

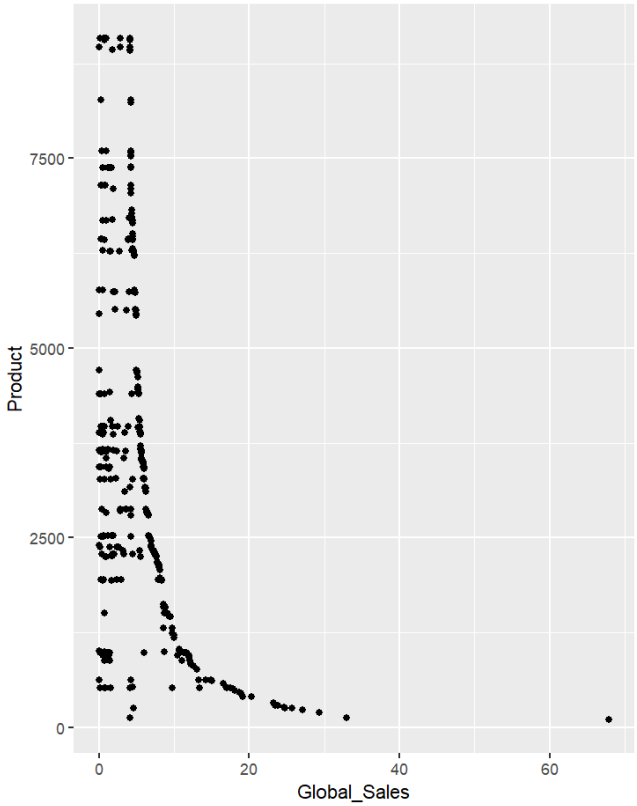

Figure 29 *– Histogram of Global Sales using R*

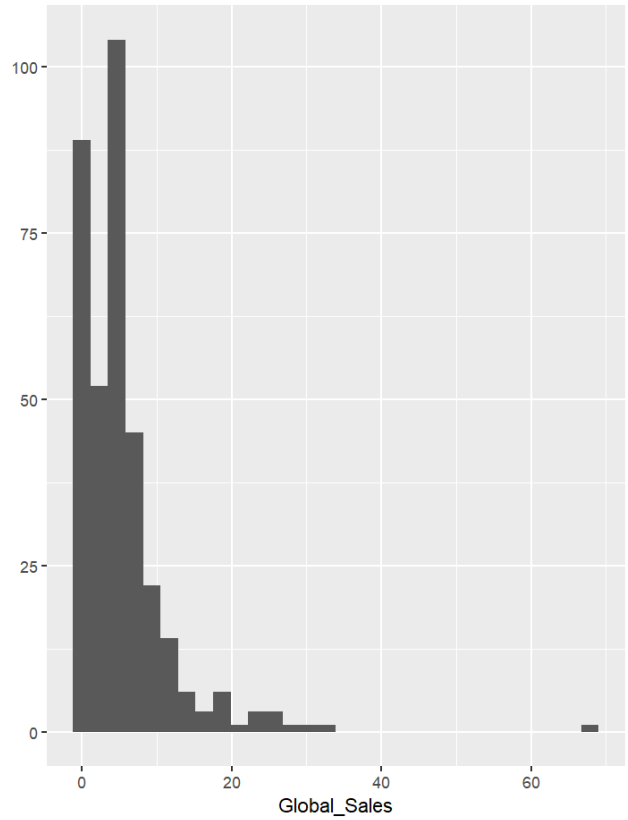

Figure 30 *– Boxplot of Global Sales using R*

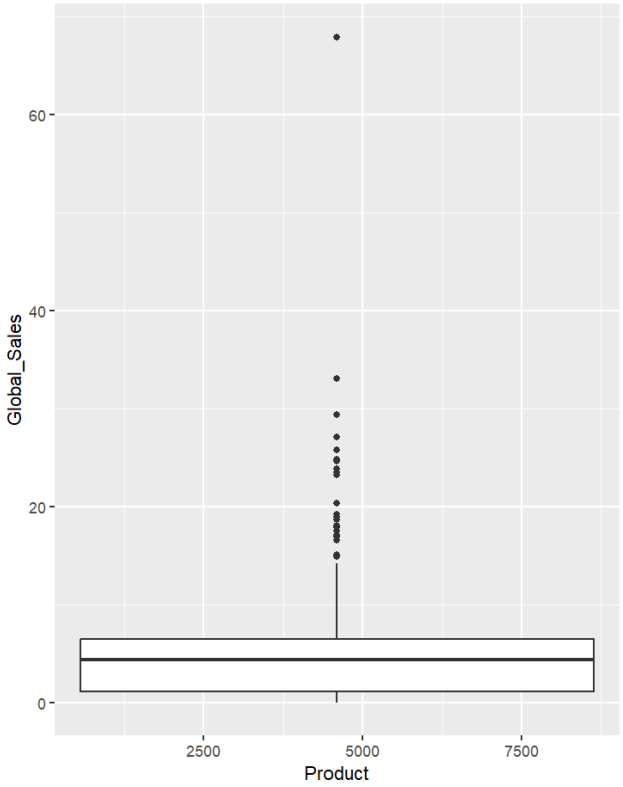

Table 18 *– Descriptive statistics of sales using R*

Product	Platform	NA_Sales	EU_Sales	Global_Sales
Min. : 107	Length:352	Min. : 0.0000	Min. : 0.000	Min. : 0.010
1st Qu.:1945	Class :character	1st Qu.: 0.4775	1st Qu.: 0.390	1st Qu.: 1.115
Median :3340	Mode :character	Median : 1.8200	Median : 1.170	Median : 4.320
Mean :3607		Mean : 2.5160	Mean : 1.644	Mean : 5.335
3rd Qu.:5436		3rd Qu.: 3.1250	3rd Qu.: 2.160	3rd Qu.: 6.435
Max. :9080		Max. :34.0200	Max. :23.800	Max. :67.850

Figure 31 *- Scatterplot on grouped products*

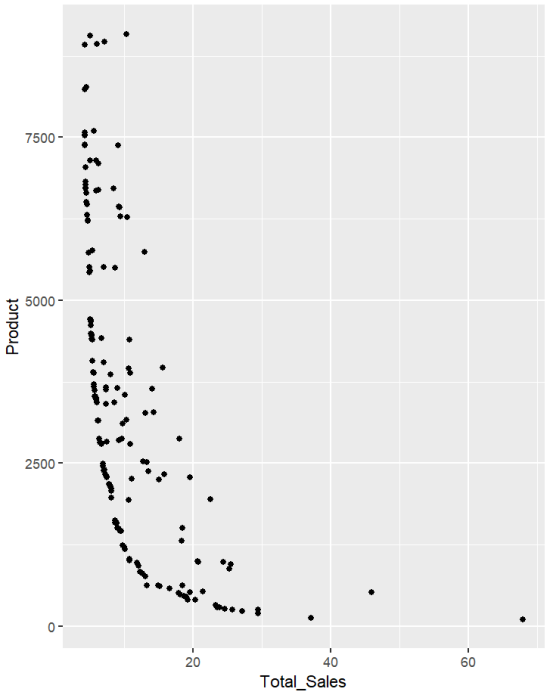

Figure 32 *– Histogram of grouped products – 4 bins*

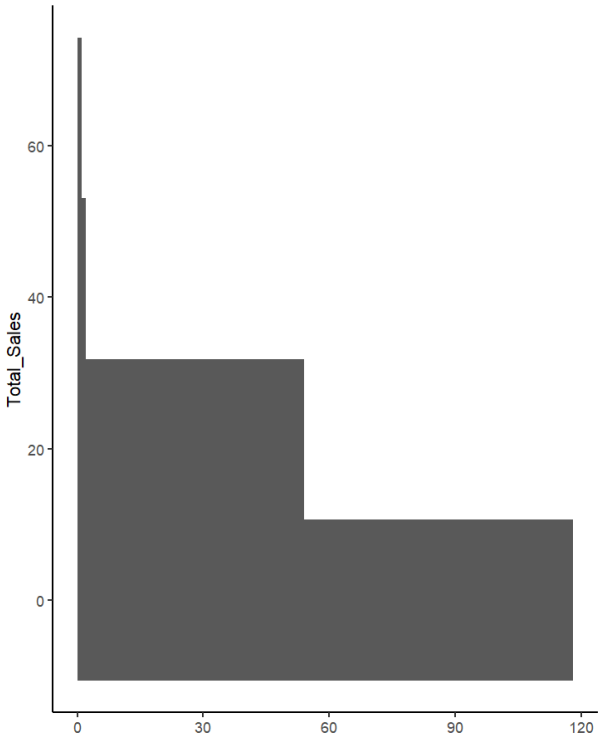

Figure 33 *– Boxplot of grouped products*

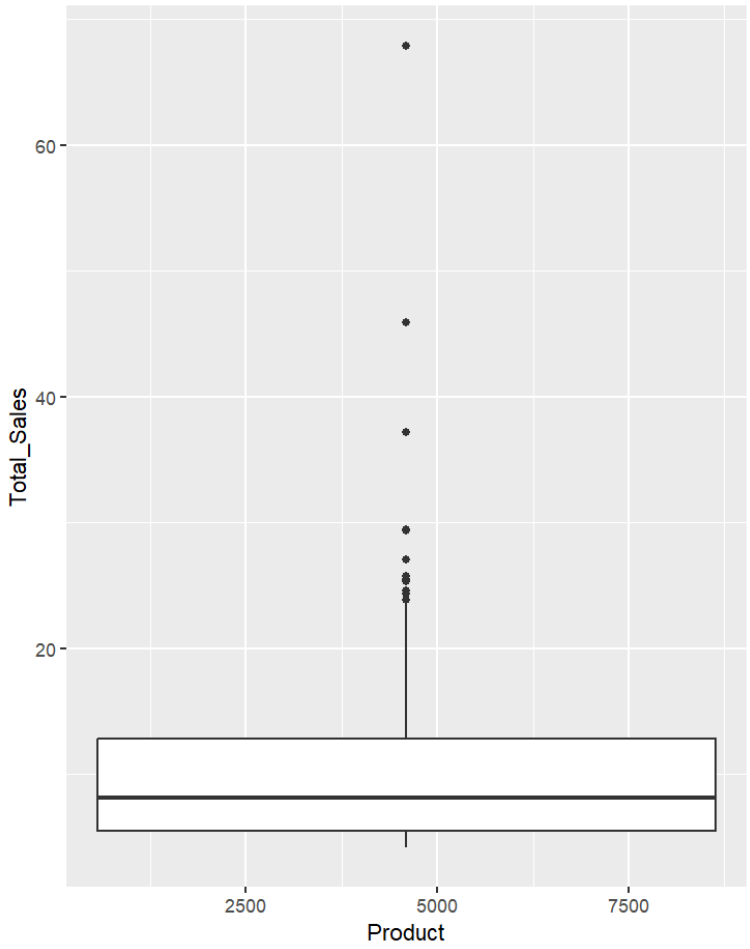

Table 19 *- Products contributions more than £20m to revenue*

	Product	Total_Sales	Total_NA	Total_EU	Total_EU_NA	Total_Other
1	399	20.30	7.44	9.02	16.46	3.84
2	999	20.58	11.09	6.66	17.75	2.83
3	978	20.77	9.75	7.83	17.58	3.19
4	535	21.38	13.11	3.99	17.10	4.28
5	1945	22.46	12.23	7.42	19.65	2.81
6	326	23.21	22.08	0.52	22.60	0.61
7	291	23.47	11.96	5.79	17.75	5.72
8	283	23.80	11.50	7.54	19.04	4.76
9	979	24.36	11.55	9.07	20.62	3.74
10	263	24.61	9.33	7.57	16.90	7.71
11	876	25.28	12.77	9.25	22.02	3.26
12	948	25.45	14.42	7.79	22.21	3.24
13	249	25.72	9.24	7.29	16.53	9.19
14	231	27.06	12.92	9.03	21.95	5.11
15	195	29.37	13.00	10.56	23.56	5.81
16	254	29.39	21.46	2.42	23.88	5.51
17	123	37.16	26.64	4.01	30.65	6.51
18	515	45.86	19.25	18.88	38.13	7.73
19	107	67.85	34.02	23.80	57.82	10.03

Showing 1 to 19 of 19 entries, 6 total columns

Figure 34 *– What proportion of products contribute more than £20m in revenue*

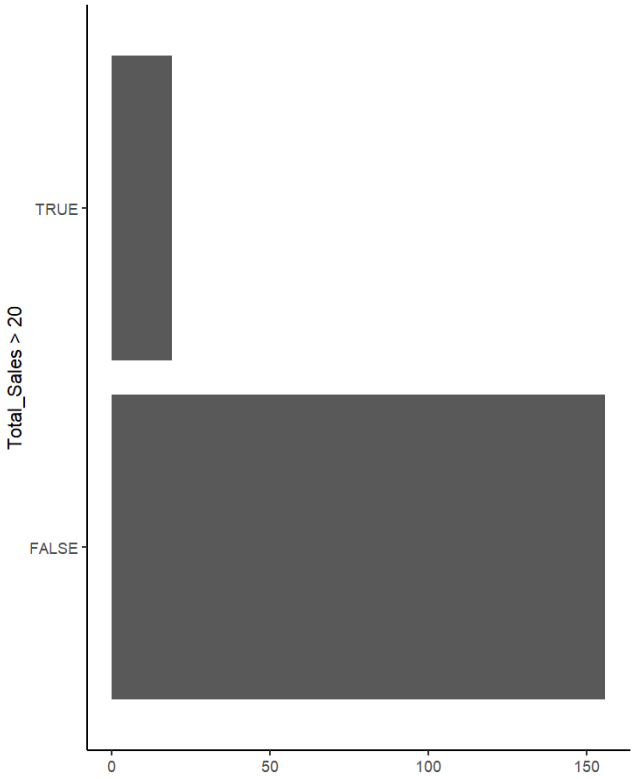

Figure 35 *– QQ Plot Total_Sales*

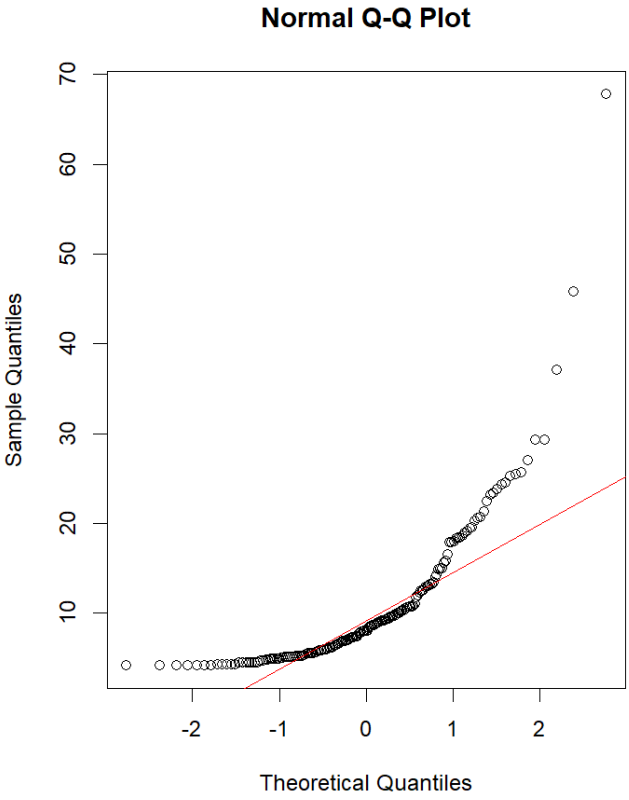

Table 20 *- Shapiro-Wilk Test – Total_Sales*

```
        Shapiro-Wilk normality test

data:  (df3$Total_Sales)
W = 0.70955, p-value < 2.2e-16
```

Table 21 *- Skewness – Total_Sales*

```
[1] 3.066769
```

Table 22 - *Kurtosis – Total_Sales*

```
[1] 17.79072
```

Table 23 *– Pearson's Correlation Coefficient – Product v Revenue*

```
[1] -0.6061376
```

Table 24 *– Correlation between numeric columns in original dataset*

	sales.Ranking	sales.Product	sales.Year	sales.NA_Sales	sales.EU_Sales	sales.Global_Sales
sales.Ranking	1.00000000	0.08060714	NA	-0.3438232	-0.3574656	-0.3910281
sales.Product	0.08060714	1.00000000	NA	-0.4047865	-0.3894246	-0.4409046
sales.Year	NA	NA	1	NA	NA	NA
sales.NA_Sales	-0.34382320	-0.40478645	NA	1.0000000	0.7055236	0.9349455
sales.EU_Sales	-0.35746561	-0.38942459	NA	0.7055236	1.0000000	0.8775575
sales.Global_Sales	-0.39102814	-0.44090460	NA	0.9349455	0.8775575	1.0000000

Figure 36 *– Simple Linear Regression - Year v Revenue*

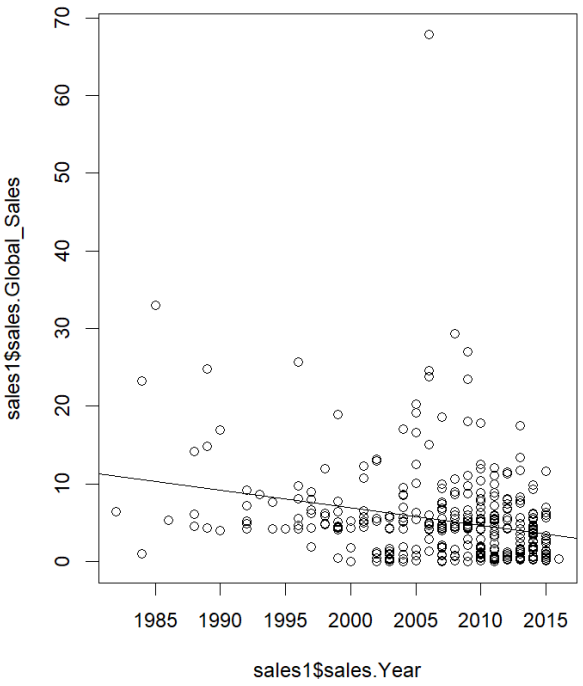

Figure 37 *- Simple Linear Regression – North American sales v Global sales*

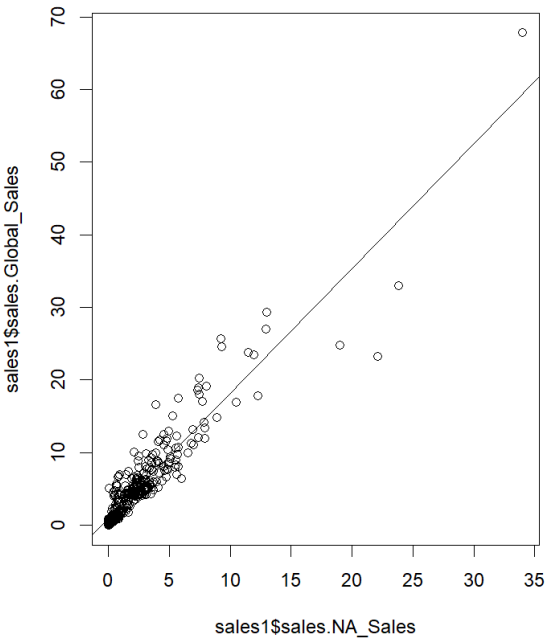

Figure 38 *- Simple Linear Regression – European sales v Global sales*

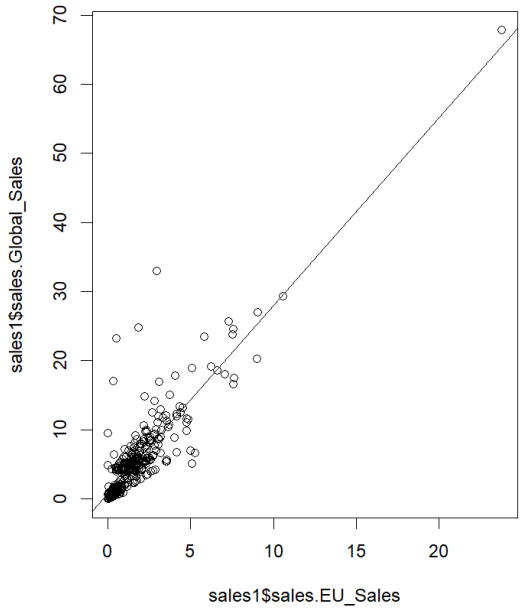

Figure 39 *- Simple Linear Regression – North American sales v European sales*

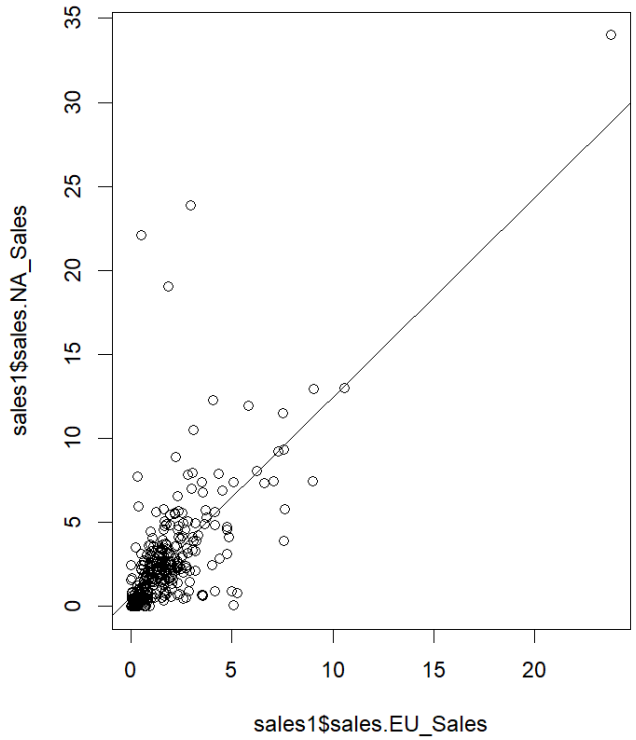

Figure 40 *- Correlation Plot of original data*

Table 25 *– Predictions*

Row.names	sales.Product	sales.NA_Sales	sales.EU_Sales	sales.Global_Sales	fit	lwr	upr
1	107	34.02	23.80	67.85	71.468572	70.162421	72.774723
10	326	22.08	0.52	23.21	26.431567	25.413344	27.449791
176	6815	2.73	0.65	4.32	4.248367	4.102094	4.394639
211	2877	2.26	0.97	3.53	4.134744	4.009122	4.260365
99	3267	3.93	1.56	6.04	6.856083	6.718420	6.993745

Showing 1 to 5 of 5 entries, 10 total columns