TURTLE GAMES

APPROACHES AND INSIGHTS

PART 1- BACKGROUND AND CONTEXT

Turtle Games company is a game manufacturer and retailer with a global customer base. The company manufactures and sells its own products, along with sourcing and selling products manufactured by other companies. Its product range includes books, board games, video games, and toys.

This data analysis report will provide better insights on the how turtle games can improve its sales performance. The following sub-set of questions will be addressed in this report

- How customers accumulate loyalty points
- How groups with the customer base can be used to identify the target specific market segments
- how social data (e.g., customer reviews) can be used to inform marketing campaigns
- the impact that each product has on sales
- how reliable the data is (e.g., normal distribution, skewness, or kurtosis)
- what the relationship(s) is/are (if any) between North American, European, and global sales?

PART 2- ANALYTICAL APPROACH

Firstly, imported various python libraries i.e., NumPy, pandas, matpotlib, seaborn and stats model.

To further sense check the data imported the turtles_review.csv and viewed the data sets. Used the isna (). sum () function to identify if there are any missing values in the data set.

To address, the first set of questions around how customers accumulate loyalty points, three possible relationships were investigated between the dependent variable loyalty points and independent variables age, spending and remuneration score. OLS regression results were used to analyse the data to compare the possible relationships between the variables. ***Refer to appendix 1***

To determine how the customer base can be used to identify the target specific market segments, two variables spending score and renumeration were used to further investigate and resolve this sub-set question. A decision was made to import further python libraries were imported such as the StandardScaler,KMeans, Silhouette _score and cdist. Elbow method and Silhouette method used for determining the number of optimal clusters. ***Refer to appendix 2.***

Social media is a key platform for any business to identify what the customer think about the products and as such customer feedback serves very key for any product success in a business. For our data analysis NLP (Natural language Processing) was used to identify the most 15 common words in online product reviews and the top 20 positive and negative reviews from customers. Use of additional libraries such as the wordcloud, word_tokenize, FreqDist, stopwords etc. Data cleaning steps performed isna(). sum () function to identify any missing values and drop unnecessary columns to only perform analysis on two columns i.e., review and summary.

Furthermore, to prepare the data for NLP, use lambda function and join function to change both review and summary column to lower case and join with space, remove punctations and drop duplicates in each column. ***Refer to appendix 3***

Use of tokenize library nltk to tokenize words in each column to assist in creation of word clouds. The first part of the data cleaning and word cloud image included stop words which does not give a clear picture on the actual customer reviews, further filter and cleaning was done to exclude the stop words to project a better output of the reviews. Importation of stop words library and nltk corpus library. For the second part of the subset question around the positive and negative comments through creating dataframes for the positive and negative sentiments for top 20 comments in each column. ***Refer to appendix 4***

Use of R in data analysis we included importation of R library such as tidy verse and imported the turtle_sales.csv file. We further cleaned the data by removing all redundant columns.

To address the question around reliability of data various functions were used to calculate the normality of sales data. A new data frame of created to firstly sub-set the total sales based on product id. Shapiro-Wilk Test used to measure the goodness of fit test. Furthermore, skewness and kurtosis measures used to check the normality of the distribution. ***Refer to appendix 5.***

To determine the relationship between the North American, European and Global Sales we used the correlation function. Furthermore, simple regression model and multiple regression model used to identify if there is any/no relationship noted between the North American, EU and Global Sales. ***Refer to appendix 6.***

PART 3- DASHBOARD AND DEVELOPMENT

The aim of the dashboard is to provide Turtle games better insights on how it can improve its sales performance through addressing the sub-set questions above.

For the first sub-set question, a decision was made to use scatter plots to identify relationships between the dependent variable (Loyalty points) and independent variables (Age, spending and renumeration score).

To address the second sub-set question, a decision was made to present visualizations in both scatter plots and pair plots to identify the target specific segments.

Use of word clouds to help visualize the most common words used. Wordcloud visualizations help stakeholders visualize the most common words used in customer reviews. Game was the most frequently used word followed by great and fun. Histograms were further used to project the sentiment polarity scores.

In R built visualizations using qplot function and 9 visualizations for Global Sales, North American Sales and EU sales exploring three types of chart type i.e., Histogram, boxplot and scatter plots to get a relationship between the product id and sales.

Ggplots used to create better visualizations to identify the impact of product to sales.

A decision was made to plot linear regression charts and created 3 model to identify if there is any relationship between the EU, North American and Global Sales.

PART 4- PATTERNS, TRENDS AND INSIGHTS

Based on the OLS regression results score for each relationship the closest relationship is between the loyalty points and spending score i.e. The r-square score of 0.452 (45.2%) which means that the higher the spending the higher the loyalty points whilst there was slightly lower relationship between the renumeration and loyalty points of 0.380 (38%) which means that the higher the renumeration the higher the spending the higher the loyalty points accumulated by the customers.

On the other hand, based on there was no minimal/no relationship between the age and loyalty points at an r-square score of 0.002 meant there was no direct relationship between age and loyalty points.

Further analysis, both independent variables spending and renumeration had a positive coefficient of spending (33.06) and renumeration (34.18) which indicates that if as the spending increases the loyalty points increases, also the higher the income the higher the customer spending power hence the higher the loyalty points and vice versa. On the other hand, the independent variable age generated a negative coefficient which indicated an if the customer age increased the loyalty points would decrease and vice versa.

To analyse, the sub question around how the customer base can be used to identify the target specific market segments, both Elbow and Silhouette method gave an optimal cluster of 5. Further analysis for these results were that the male category with a renumeration of 12.3 had the most spending score of 81. Turtle games should focus on the male category as such as practically male population would spend more on video games.

Game was the most frequently used word followed by great and fun. Turtle games can assume that customers did have fun using the video games. Based on the results of the frequency of words used with a polarity score of over 0.05 indicates a positive customer sentiment towards the company products. For top 20 positive and negative review comments, the data is mixed such that some customers are happy with the games with a positive sentiment score of 1 whilst other customer customers complaint about the product highlighting its complexity of use.

Based on the visualizations created in R, it can be concluded that product id 107 (WII) had the highest sales amongst the three divisions. Turtle games generated highest revenue from the sporting products.

The normality of data set for all three sales columns using Shapiro-Wilk Test resulted in a p-value of more than 0.05 which indicates a normal distribution. For all sales data the skewness score was greater than 1, NA Sales (3.04), EU Sales (2.88) and Global Sales (3.06) which indicates a substantially skewed distribution. For kurtosis for all sales data columns the score was greater than 1 i.e., NA Sales (15.60), EU Sales (16.22) and Global Sales (17.79) which indicates that the distribution is too peaked.

For EU to NA Sales, based on the results of linear regression model in determining the relationship between the EU Sales and North American Sales the multiple R-Square 38.56%. This point explains some variation between the sales. This result is how-ever not very strong meaning the model is not very fit. Furthermore, based on the results we can see F-statistic score is very large (108.6) and the p-value is very small, which means we should reject the null hypothesis and conclude that there is strong evidence that a relationship does exist between EU sales and NA Sales.

For EU to Global Sales, based on the results of the linear regression model in determining the relationship between the EU and Global Sales the multiple R-Square was 72.01%. This point explains variability in the sales columns. This result is stronger and indicates a good fit. Furthermore, based on the results we can see F-statistic score is very large (445.2) and the p-value is very small, which means we should reject the null hypothesis and conclude that there is strong evidence that a relationship does exist between EU sales and Global Sales.

For NA to Global Sales, based on the results of the linear regression model in determining the relationship between the EU and Global Sales the multiple R-Square was 83.95%. This point explains variability in the sales columns. This result is stronger and indicates a good fit. Furthermore, based on the results we can see F-statistic score is very large (904.7) and the p-value is very small, which means we should reject the null hypothesis and conclude that there is strong evidence that a relationship does exist between NA sales and Global Sales.

Furthermore, based on the results of multiple regression model determined the relationship between the EU, Global and NA sales. The multiple r-square was 96.68%. This point explains strong relationship between the sales data and indicates a good fit. Furthermore, based on the results we can see F-statistic score is very large (2504) and the p-value is very small, which means we should reject the null hypothesis and conclude that there is very strong relationship does exist between NA, EU and Global sales.

PART 5: RECOMMENDATIONS AND NEXT STEPS

Turtle Games could adapt other measures to generate customer reviews such a 0-10 score matrix in its questionnaire per category such as the delivery of the product, use of product, product designing etc. This will help the company get better insights on the customer behaviour of the company products.

Furthermore, the company can use additional data to identify the relationship such as age to sales per customer per territory would enable Turtle games to better target customers in each market. Product ID is still a vague basis to project the actual picture of the customers.

APPENDIX

1a) SPENDING VS LOYALTY

In [9]:
```
# Create formula and pass through OLS methods.
f = 'y ~ x'
test = ols(f, data = reviews).fit()

# Print the regression table.
test.summary()
```

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.452
Model:	OLS	**Adj. R-squared:**	0.452
Method:	Least Squares	**F-statistic:**	1648.
Date:	Sun, 04 Dec 2022	**Prob (F-statistic):**	2.92e-263
Time:	16:28:37	**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

1b) Loyalty vs Spending

In [14]:
```
# Create formula and pass through OLS methods.
f = 'y ~ x'
test = ols(f, data = reviews).fit()

# Print the regression table.
test.summary()
```

Out[14]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.380
Model:	OLS	**Adj. R-squared:**	0.379
Method:	Least Squares	**F-statistic:**	1222.
Date:	Sun, 04 Dec 2022	**Prob (F-statistic):**	2.43e-209
Time:	16:28:39	**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.

1c) Loyalty and Age

In [19]:
```
# Create formula and pass through OLS methods.
f = 'y ~ x'
test = ols(f, data = reviews).fit()

# Print the regression table.
test.summary()
```

Out[19]:

OLS Regression Results

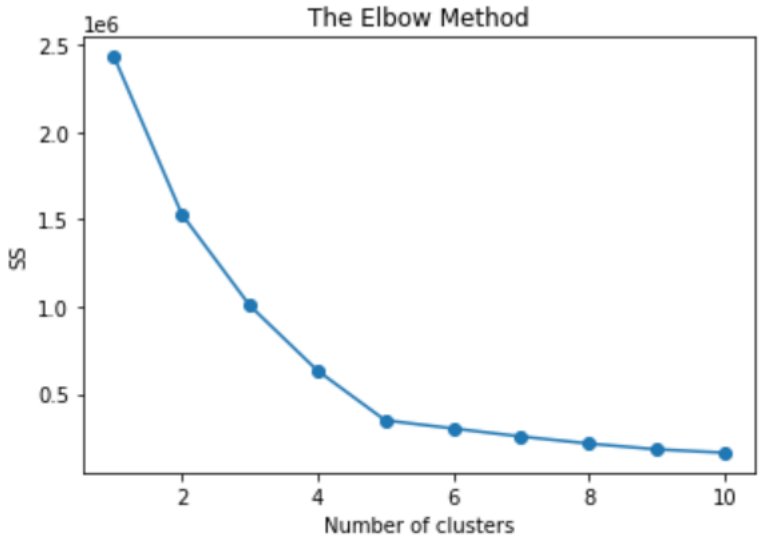
Dep. Variable:	y	**R-squared:**	0.002
Model:	OLS	**Adj. R-squared:**	0.001
Method:	Least Squares	**F-statistic:**	3.606
Date:	Sun, 04 Dec 2022	**Prob (F-statistic):**	0.0577
Time:	16:28:40	**Log-Likelihood:**	-17150.
No. Observations:	2000	**AIC:**	3.430e+04
Df Residuals:	1998	**BIC:**	3.431e+04
Df Model:	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.

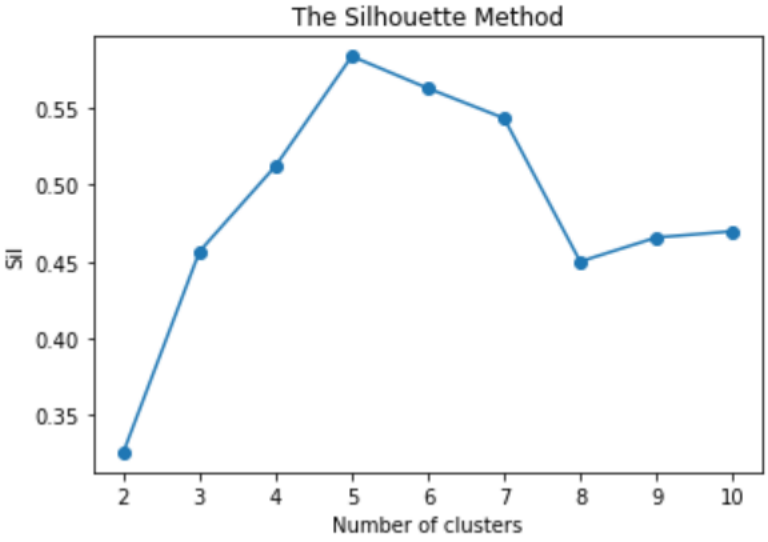
2) Customer target segments -ELBOW METHOD

In [26]:
```
# Determine the number of clusters: Elbow method.

# Import the KMeans class.
from sklearn.cluster import KMeans

# Elbow chart for us to decide on the number of optimal clusters.
ss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i,
                    init = 'k-means++',
                    max_iter = 500,
                    n_init = 10,
                    random_state = 42)
    kmeans.fit(x)
    ss.append(kmeans.inertia_)

plt.plot(range(1, 11),
         ss,
         marker='o')

plt.title("The Elbow Method")
plt.xlabel("Number of clusters")
plt.ylabel("SS")

plt.show()
```



2)B) SILHOUETTE METHOD-KMEANS

In [27]:
```
# Determine the number of clusters: Silhouette method.

# Import silhouette_score class from sklearn.
from sklearn.metrics import silhouette_score

# Find the range of clusters to be used using 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')

plt.title("The Silhouette Method")
plt.xlabel("Number of clusters")
plt.ylabel("Sil")
```



3) CHANGE LOWER CASE FOR REVIEW AND SUMMARY COLUMN AND REMOVE PUNCTUATIONS FROM COLUMNS

2a) Change to lower case and join the elements in each of the columns respectively (review and summary)

In [45]:
```
#Change review column to lower case and join with a space

df6['review']=df6['review'].apply(lambda a:" ".join(a.lower()for a in a.split()))
df6['summary']=df6['summary'].apply(lambda b:" ".join(b.lower()for b in b.split()))


#view the dataframe
df6.head()
```

2b) Replace punctuation in each of the columns respectively (review and summary)

In [46]:
```
# Remove punctuation from the review column
df6['review'] = df6['review'].str.replace('[^\w\s]','')

# Preview the result.
df6['review'].head()
```

```
# Remove punctuation from the summary column

df6['summary'] = df6['summary'].str.replace('[^\w\s]','')

# Preview the result.
df6['summary'].head()
```

2c) Drop duplicates in both columns

In [48]:
```
# Check the number of duplicate values in the review column.
df6.review.duplicated().sum()
```

Out[48]: 50

In [49]:
```
# Check the number of duplicate values in the summary column.
df6.summary.duplicated().sum()
```

Out[49]: 649

In [143]:
```
# Drop duplicates(Review)

df6a = df6.drop_duplicates(subset=['review'])
df6b = df6.drop_duplicates(subset=['summary'])

# Preview data.
df6a.reset_index(inplace=True)
df6b.reset_index(inplace=True)

#View the dataframe
df6a.head()
df6b.head()
```

4) USE OF NLP- TOKENIZATION AND WORD CLOUD

3. Tokenise and create wordclouds

In [99]:
```
# Import nltk and download nltk's resources to assist with tokenisation.
import nltk

nltk.download('punkt')
from nltk.tokenize import word_tokenize
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\chand\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

In [144]:
```
# Tokenise the words.
df6a['tokens'] = df6a['review'].apply(word_tokenize)

# Preview data.
df6a['tokens'].head()
```

Out[144]:
```
0    [when, it, comes, to, a, dms, screen, the, spa...
1    [an, open, letter, to, galeforce9, your, unpai...
2    [nice, art, nice, printing, why, two, panels, ...
3    [amazing, buy, bought, it, as, a, gift, for, o...
4    [as, my, review, of, gf9s, previous, screens, ...
Name: tokens, dtype: object
```

In [145]:
```
# Tokenise the words.
df6b['tokens'] = df6b['summary'].apply(word_tokenize)

# Preview data.
df6b['tokens'].head()
```

Out[145]:
```
0    [the, fact, that, 50, of, this, space, is, was...
1    [another, worthless, dungeon, masters, screen,...
2                  [pretty, but, also, pretty, useless]
3                                        [five, stars]
4                                        [money, trap]
Name: tokens, dtype: object
```

In [54]:
```
# You might need to install WordCloud.
!pip install WordCloud
```

In [55]:
```
# Import along with matplotlib and seaborn for visualisation.
from wordcloud import WordCloud
import matplotlib.pyplot as plt
import seaborn as sns
```

In [146]:
```
# String all the comments together in a single variable.
# Create an empty string variable.
all_comments = ''
for i in range(df6a.shape[0]):
    # Add each comment.
    all_comments = all_comments + df6a['review'][i]
```

In [147]:
```
# Set the colour palette.
sns.set(color_codes=True)

# Create a WordCloud object.
word_cloud = WordCloud(width = 1600, height = 900,
                background_color ='white',
                colormap = 'plasma',
                stopwords = 'none',
                min_font_size = 10).generate(all_comments)
```

In [148]:
```
# Plot the WordCloud image.
plt.figure(figsize = (16, 9), facecolor = None)
plt.imshow(word_cloud)
plt.axis('off')
plt.tight_layout(pad = 0)
plt.show()
```

In [149]:
```
# String all the comments together in a single variable (summary)
# Create an empty string variable.
all_comments = ''
for i in range(df6b.shape[0]):
    # Add each comment.
    all_comments =all_comments+df6b['summary'][i]
```

In [150]:
```
# Set the colour palette.
sns.set(color_codes=True)

# Create a WordCloud object.
word_cloud = WordCloud(width = 1600, height = 900,
                background_color ='white',
                colormap = 'plasma',
                stopwords = 'none',
                min_font_size = 10).generate(all_comments)
```

In [151]:
```
# Plot the WordCloud image.
plt.figure(figsize = (16, 9), facecolor = None)
plt.imshow(word_cloud)
plt.axis('off')
plt.tight_layout(pad = 0)
plt.show()
```

5. RELIABILITY OF DATA

```
#Determine the normality of the data set
#Run a Shapiro-Wilk Test:
shapiro.test(Product_Sales$Total_NA_Sales)
shapiro.test(Product_Sales$Total_EU_Sales)
shapiro.test(Product_Sales$Total_Global_Sales)
#Determining the skewness and Kurtosis for each sales data
#Install the moments package and load library
install.packages("moments")
library(moments)
#Determining the skewness and kurtosis for North American Sales
skewness(Product_Sales$Total_NA_Sales)
kurtosis(Product_Sales$Total_NA_Sales)
#Determining the skewness and kurtosis for European Union Sales
skewness(Product_Sales$Total_EU_Sales)
```

5. IDENTIFY THE RELATIONSHIP BETWEEN THE NORTH AMERICAN, EUROPEAN SALES AND GLOBAL SALES.

```
#Determining the correlation between the sales data
cor(Product_Sales$Total_EU_Sales,Product_Sales$Total_NA_Sales)
cor(Product_Sales$Total_EU_Sales,Product_Sales$Total_Global_Sales)
cor(Product_Sales$Total_NA_Sales,Product_Sales$Total_Global_Sales)
```

6. SYNTAX FOR LINEAR REGRESSION AND MLR

```
#Create a simple linear regression
#Create a model with only one x variable (EU Sales to North American Sale
model1 <- lm(Total_EU_Sales~Total_NA_Sales,
             data=Product_Sales)
#Plot the model
plot(Product_Sales$Total_EU_Sales,Product_Sales$Total_NA_Sales)
# View more outputs for the model - the full regression table.
summary(model1)
plot(model1$residuals)
abline(coefficients(model1))
```

```
#Create a simple linear regression model for EU to Global Sales
model2 <- lm(Total_EU_Sales~Total_Global_Sales,
             data=Product_Sales)
#Plot the model

plot(Product_Sales$Total_EU_Sales,Product_Sales$Total_Global_Sales)
# View more outputs for the model - the full regression table.
summary(model2)
plot(model2$residuals)
```

```
#Create a simple linear regression model for North American to Global Sal
model3 <- lm(Total_NA_Sales~Total_Global_Sales,
             data=Product_Sales)
#Plot the model
plot(Product_Sales$Total_NA_Sales,Product_Sales$Total_Global_Sales)
# View more outputs for the model - the full regression table.
summary(model3)
plot(model3$residuals)
abline(coefficients(model3))
```

```
#Create a MLR model
#Select only numeric columns from the original data frame
Product_Sales2=subset(Product_Sales, Select=-c(Product))
str(Product_Sales2)
View(Product_Sales2)
#Multiple linear regression model
Sales_Model= lm(Total_Global_Sales~Total_NA_Sales+Total_EU_Sales,data=Pro
Sales_Model
#Predictions based on given values
#Create a new object with prediction function
Predict_Sales=predict(Sales_Model,newdata = Product_Sales2,interval='conf
#View the object
View(Predict_Sales)
```