Analysing Customer Trends to Improve the Sales Performance of Turtle Games

Christian Putra Chen

24th July 2023

London School of Economics and Political Science

Course: Data Analytics Career Accelerator
Module: Advanced Analytics for Organisational Impact

1 Business Context and Background

Turtle Games is a game manufacturer and retailer with a global consumer base, aiming to improve their overall sales performance. By utilising powerful data analytics techniques in **Python** and **R**, patterns in customer purchasing behaviour can be identified from historic data, unveiling a treasure trove of insights that can be leveraged to boost sales.

Key questions:

1. How do customers accumulate loyalty points, and can this be predicted?
2. What market segments exist within the customer base?
3. Can social data inform marketing campaigns?
4. What factors affect overall sales?

2 Analytical Approach

The project makes use of the **Python** and **R** programming languages, with scripts written in a **Jupyter Notebook** or **RStudios** respectively. Efficient data analysis requires the adoption of a thorough and systematic work flow, such as the following:

1. Install and import key libraries for **Python** and **R**—**pandas**, **numpy**, **seaborn**, **matplotlib.pyplot**, **tidyverse**, **plotly**.
2. Install and import all relevant libraries for: *linear regression* (Figure A1), *k-means clustering* (Figure A2), *natural language processing* (NLP) and *sentiment analysis* (Figure A3).
3. Import, sense check, and clean the data.
4. Create user-defined functions for ease of use e.g. Figure A4.
5. Upload the code to **GitHub** for version control.
6. Continue with the analysis, looping back whenever necessary.

For extensive cleaning of relatively small data files, I would normally opt for **Excel**; however, upon initial analysis in **Python** and **R**, I determined the files to be relatively clean. These were my initial observations:

- **turtle_reviews.csv**: Analysed with **Python**—contains 2000 individual customer records.
- **turtle_sales.csv**: Analysed with **R**—contains 352 sales records.

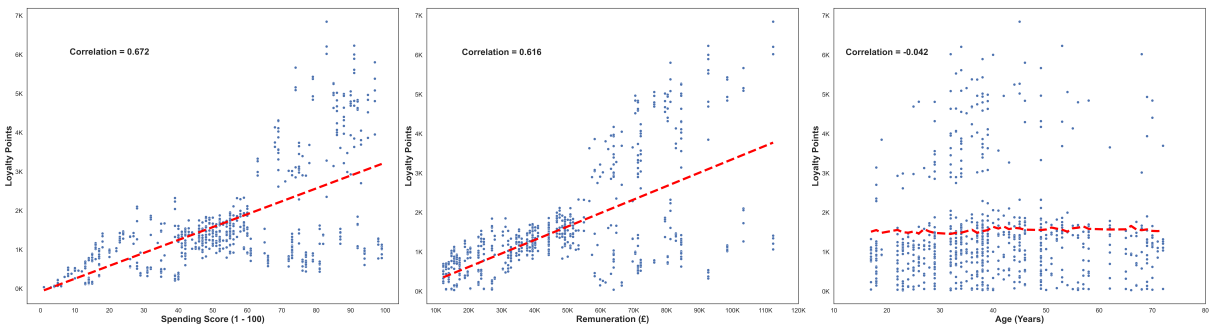

Figure 1: Correlation plots depicting loyalty points against spending score (left), remuneration (middle), and age (right). The first two have positive correlations above 0.6, and the last one is relatively uncorrelated.

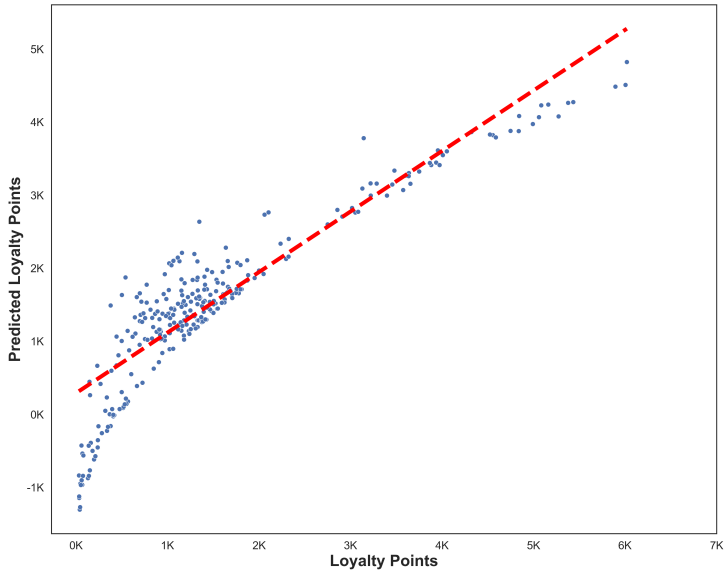

Figure 2: Correlation plot depicting predicted loyalty points against actual loyalty points. Predictions are negative below roughly 0.4K, but predictions above this seem to be quite accurate.

No duplicate records were found in either data set (Figures A5 and A6), and the only null values found were two missing years in **turtle_sales.csv** (Figures A7 and A8); however, I opted to keep these rows, as the years column was deemed redundant for this analysis and promptly removed. After cleaning, the first of these data sets was saved as **turtle_reviews_clean.csv**, ready for exploratory analysis in **Python**.

Additionally—in preparation for sentiment analysis via NLP techniques—the review and summary columns of the newly created **turtle_reviews_clean.csv** had to be prepared accordingly, as follows:

1. Change all words to lower case and join elements with a space.
2. Remove all punctuation.
3. Remove all duplicates—50 in review column, 649 in summary column.
4. Tokenise and create wordclouds (Figure A9).
5. Remove non-alphanumeric characters and stopwords.
6. Create new wordclouds (Figure A10).
7. Perform sentiment analysis using the **textblob** library.

Finally, an exploratory analysis was conducted in **R**, consisting primarily of exploring the distribution of sales data through methods such as: quantile-quantile (Q-Q) plots (Figure A11), the Shapiro-Wilk test (Figure A12), and various tests for kurtosis and skewness (Figure A13). Immediately prior to this, sales data was grouped by product to identify key sales characteristics pertaining to particularly successful products—outliers were intentionally kept for this reason.

3 Visualisation and Insights

Analysis began in **Python**, with an attempt to predict customer loyalty points based on spending score, remuneration, and age. This was done, at first, by fitting the data to an *ordinary*

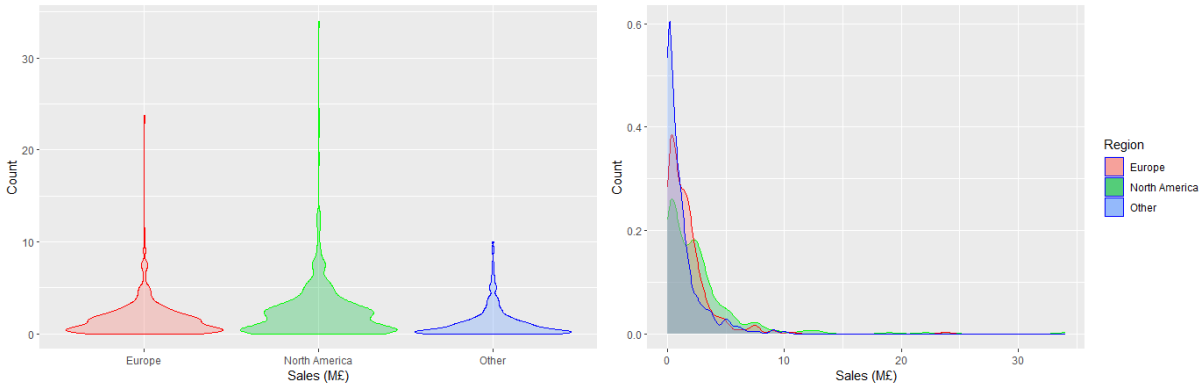

Figure 3: Density plots of regional sales by product count. Both show the same data, but one is a violin plot (left) and the other is an overlapping density plot (right). Most products have sales below roughly £5M, but there are a handful of extraordinarily high-performing outliers e.g. product 107 (the highest performer).

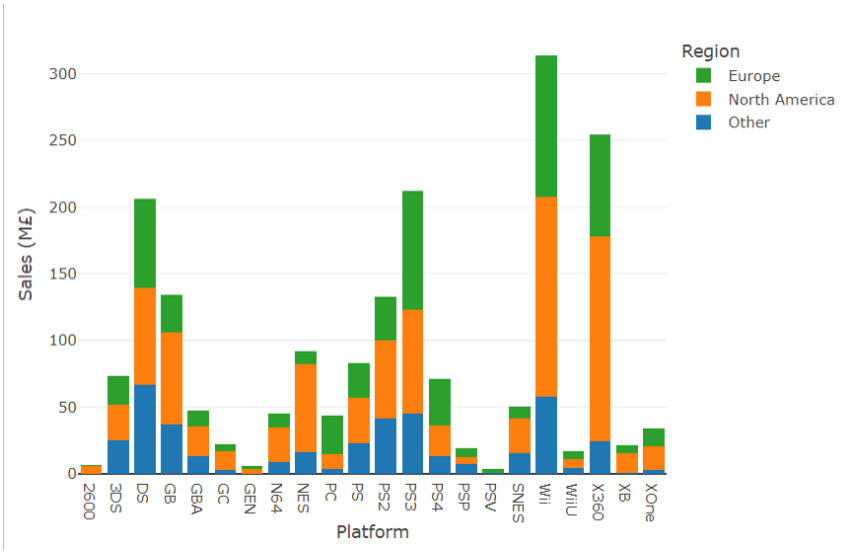

Figure 4: Barplot of regional sales by platform—there are notable differences in product preference by region.

least-squares regression (OLS) model, leading to cross-correlative insights between these variables (Figure 1). This was then extended into the realm of *multiple linear regression* (MLR), requiring splitting of the data into training/testing sets with an 80:20 ratio. The testing set yielded $R^2 = 82.9\%$ (Figure A14) and a *mean-absolute percentage error* (MAPE) of only 1.3% (Figure A15)! Therefore, the model has excellent predictive power, but only above a certain threshold (Figure 2). In terms of data reliability, via *variance inflation factor* (VIF), we found negligible multicollinearity (Figure A16); however, a *Breusch–Pagan test* revealed the presence of *heteroscedasticity* (Figure A17), which makes the data partially unreliable.

The MLR analysis in **R**—global sales, as predicted by NA and EU sales—performed similarly well, with an MAPE of 10.5% (Figure A18), which is very good; however, its residuals are not normally distributed (Figure A19), again, making the data partially unreliable. For more information on sales trends discovered in **R**, refer to Figures 3 and 4.

Continuing in **Python**, a *k-means clustering* analysis of remuneration against spending score was performed. First, the number of clusters was determined via the *elbow* and *silhouette* methods

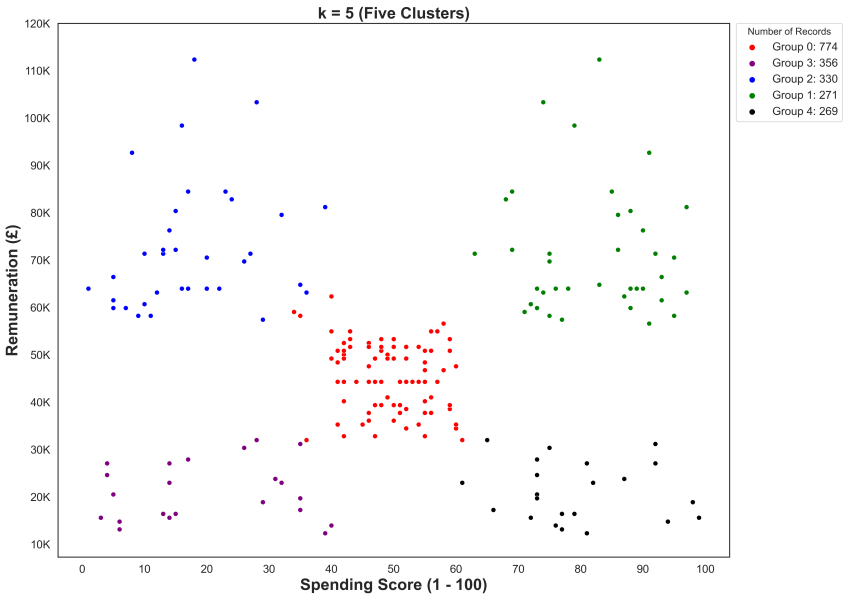

Figure 5: Ideal k-means clustering for customers based on remuneration and spending score.

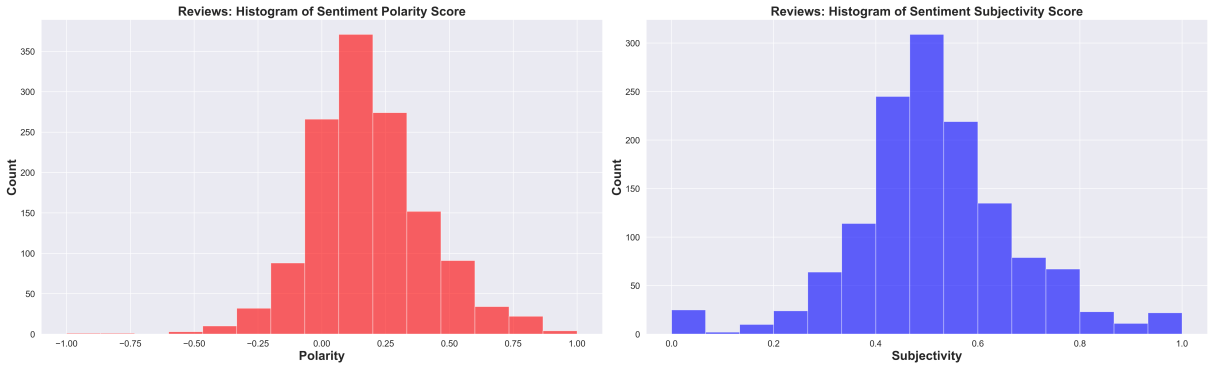

Figure 6: Review polarity and subjectivity are reminiscent of a normal distribution, with the former having a slightly positive skew.

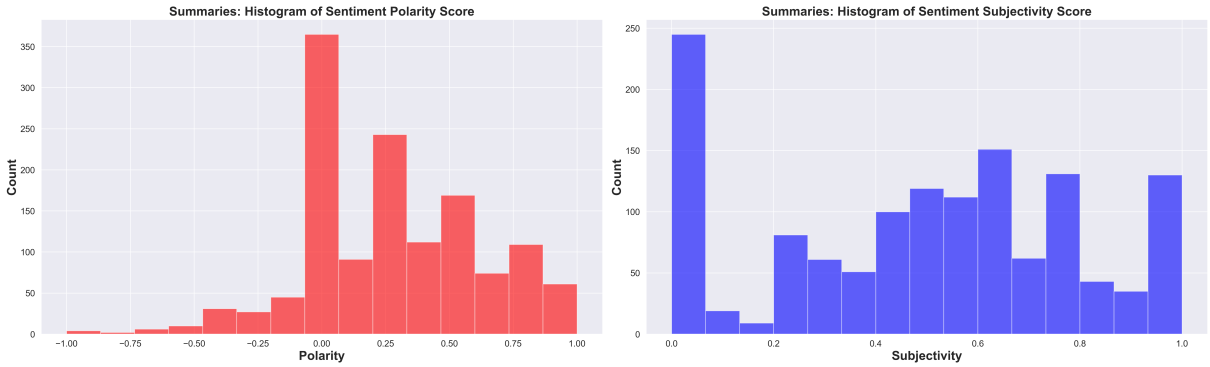

Figure 7: Summary polarity and subjectivity are somewhat sporadically distributed, with the former having a central peak and a substantially positive skew, and the latter having a left-most peak and a weakly negative skew.

(figs. A20 and A21); subsequently, the clustering algorithm was performed on the sample for various values of k to intuit its validity (Figure A22). The results? The current market is best split into $k = 5$ clusters based on remuneration and spending score (Figure 5). The cluster with

Most-Negative Reviews:

	review	review_polarity	review_subjectivity
165	booo unles you are patient know how to measure...	-1.000000	1.000000
147	incomplete kit very disappointing	-0.780000	0.910000
267	one of my staff will be using this game soon s...	-0.550000	0.300000
90	i bought this as a christmas gift for my grand...	-0.500000	0.900000
141	i sent this product to my granddaughter the po...	-0.491667	0.433333
251	my 8 yearold granddaughter and i were very fru...	-0.446250	0.533750
382	i purchased this on the recommendation of two ...	-0.440741	0.485185
312	this game although it appears to be like uno a...	-0.400000	0.400000
355	my son loves playing this game it was recommen...	-0.400000	0.400000
713	if you like me used to play dd but now you and...	-0.400000	0.400000
1011	you can play the expansions one at a time or a...	-0.400000	0.400000
723	if you play dungeons and dragons then you will...	-0.393750	0.550000
600	i was a bit disappointed in the quality of the...	-0.365625	0.709375
331	very fun game to use with kids working on hand...	-0.352500	0.265000
297	i really like this game it helps kids recogniz...	-0.350000	0.450000
389	i am a therapist for children and this game is...	-0.333333	0.300000
338	confusing instructions and its not for 6 year ...	-0.325000	0.531250
4	as my review of gf9s previous screens these we...	-0.316667	0.316667
784	the adventures are tough but you can get throu...	-0.314815	0.507407
631	a crappy cardboard ghost of the original hard ...	-0.305556	0.763889

Most-Negative Summaries:

	summary	summary_polarity	summary_subjectivity
17	the worst value ive ever seen	-1.000000	1.000000
165	boring unless you are a craft person which i am	-1.000000	1.000000
587	boring	-1.000000	1.000000
837	before this i hated running any rpg campaign d...	-0.900000	0.700000
1	another worthless dungeon masters screen from ...	-0.800000	0.900000
116	disappointed	-0.750000	0.750000
266	promotes anger instead of teaching calming met...	-0.700000	0.200000
634	too bad this is not what i was expecting	-0.700000	0.666667
637	bad qualityall made of paper	-0.700000	0.666667
144	at age 31 i found these very difficult to make	-0.650000	1.000000
75	small and boring	-0.625000	0.700000
368	mad dragon	-0.625000	1.000000
575	disappointing	-0.600000	0.700000
723	then you will find this board game to be dumb ...	-0.591667	0.633333
267	anger control game	-0.550000	0.300000
55	really small disappointed	-0.500000	0.575000
360	its uno for the angry	-0.500000	1.000000
646	50th anniversary is a sad day for acquire	-0.500000	1.000000
808	a disappointing coop game	-0.500000	0.550000
1116	its also really lame that the doll didnt come ...	-0.500000	0.750000

Figure 8: Disappointment is a common theme in negative reviews.

Most-Positive Reviews:

	review	review_polarity	review_subjectivity
564	perfect	1.000000	1.000000
1080	my daughter loves her stickers awesome seller ...	1.000000	1.000000
1334	perfect for tutoring my grandson in spelling	1.000000	1.000000
890	the best part i see is the box what a wonderfu...	0.880000	0.860000
498	great quality very cute and perfect for my tod...	0.816667	0.916667
31	the pictures are great ive done one and gave ...	0.800000	0.750000
336	great seller happy with my purchase 5 starrr	0.800000	0.875000
439	great easter gift for kids	0.800000	0.750000
491	these are great	0.800000	0.750000
692	bought this because i wanted it all these dd g...	0.800000	0.750000
824	husband seems happy with it	0.800000	1.000000
826	great accessory to use with the playing mat	0.800000	0.750000
828	great price arrived on time with no damage wil...	0.800000	0.750000
893	this is a great accessory to the starter set i...	0.800000	0.750000
1075	my granddaughter loves these so happy to find ...	0.800000	1.000000
1113	great doll to go with the book animals cant w...	0.800000	0.750000
1187	a great creation tool it helps me concentrate	0.800000	0.750000
1287	prompt service and a great product	0.800000	0.750000
1333	this is a great tool to have at hand when play...	0.800000	0.750000
325	this is a great product i use it as a therapeu...	0.790000	0.875000

Most-Positive Summaries:

	summary	summary_polarity	summary_subjectivity
5	best gm screen ever	1.0	0.3
23	wonderful designs	1.0	1.0
27	perfect	1.0	1.0
61	theyre the perfect size to keep in the car or ...	1.0	1.0
107	perfect for preschooler	1.0	1.0
112	awesome sticker activity for the price	1.0	1.0
132	awesome book	1.0	1.0
133	he was very happy with his gift	1.0	1.0
150	awesome	1.0	1.0
166	awesome and welldesigned for 9 year olds	1.0	1.0
337	excellent	1.0	1.0
389	excellent therapy tool	1.0	1.0
407	the pigeon is the perfect addition to a school...	1.0	1.0
423	best easter teaching tool	1.0	0.3
462	wonderful	1.0	1.0
466	all f the mudpuppy toys are wonderful	1.0	1.0
471	awesome puzzle	1.0	1.0
476	not the best quality	1.0	0.3
514	excellent puzzle	1.0	1.0
521	the best feedback i can have	1.0	0.3

Figure 9: Gift giving is a common theme in positive reviews.

by far the most customers is that of middle spenders/earners—these should perhaps be the focus of future marketing campaigns.

Natural language processing was also utilised to capture the sentiment of customer reviews and summaries. The key results are depicted in Figures 6 and 7; but in particular, we note the theme of the most-positive and most-negative reviews and summaries (Figures 8 and 9). At a glance, we can see that many of the negative comments indicate disappointment, and many positive comments indicate that the customer bought an item as a gift.

4 Patterns and Predictions

There are many patterns and trends that have been uncovered by thorough analysis, such as the tendency for a small number of the products to garner most of the sales. This is akin to the *Matthew Principle*—once the ball gets rolling for a product, it can speed up exponentially. With that key observation in mind, we can proceed with our conclusions:

- MLR models are highly effective at predicting both loyalty points and global sales via complementary information, with MAPEs of 1.3% and 10.5% respectively. However, the loyalty model possesses heteroscedasticity, and the sales model is non-normally distributed, reducing data reliability.
- Cluster analysis shows that there are $k = 5$ fairly distinct customer groups based on earnings and spendings, the largest of which is middle spenders/earners with 774 members. Marketing efforts could be targeted towards all groups with high spenders.
- Variations in regional product preference could be explored further to enhance targeted marketing.
- Sentiment analysis is a useful tool for uncovering common complaints and likes, but it is of limited value unless a high-quality NLP model is used. However, based on the high frequency of disappointment-related words, product quality should be prioritised; also, based on the gift-giving theme of many positive reviews, special deals could be offered that incentivise gift purchases.

Appendix A: Supplementary Figures

```
# Import additional useful libraries.
import statsmodels.api as sm
from statsmodels.formula.api import ols
import random
import statsmodels.stats.api as sms
from sklearn import datasets
from sklearn import linear_model
from statsmodels.stats.outliers_influence import variance_inflation_factor
import sklearn
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.linear_model import LinearRegression
```

Figure A1: Importing libraries for linear regression, with best-practice aliases, and adjusting relevant settings.

```
# Import additional useful libraries.
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.metrics import accuracy_score
from scipy.spatial.distance import cdist
```

Figure A2: Importing libraries for k-means clustering, with best-practice aliases, and adjusting relevant settings.

```
# Import additional useful libraries.
from wordcloud import WordCloud
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist
from nltk.corpus import stopwords
from textblob import TextBlob
from scipy.stats import norm
import itertools  # Has tools that are faster than looping manually.
from nltk.stem.snowball import SnowballStemmer
from nltk.stem.wordnet import WordNetLemmatizer
```

```
In [70]: # Download relevant NLTK additional data.
         # nltk.download('punkt')  # Tokenisation model.
         # nltk.download('stopwords')  # Stopwords e.g. "a", "the" etc.
         # nltk.download('wordnet')  # Stemming model.
         # nltk.download('omw-1.4')  # Lemma corpus.
```

Figure A3: Importing libraries for NLP, with best-practice aliases, and adjusting relevant settings.

2.3 Checking for missing values

Functions:

```
In [8]: # Build function that checks for missing values in DataFrame, and returns either:
        # 1. The DataFrame containing only rows with null values,
        # 2. Said DataFrame with null values highlighted in red (for visual checks).
        def df_null(df, **kwargs):
            # Create a DataFrame subset containing missing values using isna().
            df_na = df[df.isna().any(axis=1)]

            # Provide optional keyword argument to highlight null values in red (if any exist).
            highlight = kwargs.get('highlight', None)

            # Return output DataFrame (or styler, if highlight=1)
            if highlight == 1:
                return df_na.style.highlight_null('red')
            else:
                return df_na
```

Figure A4: Creating a user-defined function that returns a DataFrame with only rows that contain at least one null value.

```
In [9]: # Check for duplicate rows.
        # Add docstring for clarity.
        print("The number of duplicate rows:", reviews_na.shape[0])

        The number of duplicate rows: 0
```

Figure A5: Checking for duplicate values in **Python**.

```
> # Check for duplicates of Product-Platform pair
> sum(duplicated(subset(sales, select=c(Product, Platform))))
[1] 0
```

Figure A6: Checking for duplicate values in **R**.

```
In [8]: # Any missing values?
        # Determine whether there are missing values in turtle_reviews.csv.
        reviews_na = df_null(reviews)

        # Print the number of rows that have a NaN value.
        # Add docstring for clarity.
        print("The number of rows that have missing values:", reviews_na.shape[0])

        # View the DataFrame containing only rows with null values.
        if reviews_na.shape[0] != 0:
            display(reviews_na)

        The number of rows that have missing values: 0
```

Figure A7: Checking for null values in **Python**.

```
> # Number of missing values.
> sum(is.na(sales))
[1] 2
>
> # View rows with missing values.
> sales[!complete.cases(sales),]
    Ranking Product Platform Year   Genre       Publisher NA_Sales EU_Sales Global_Sales
180     180    7141      PS2   NA  Sports Electronic Arts      3.49      0.21         4.29
258    1128     948       PC   NA Shooter      Activision      0.48      0.66         1.34
> # Only the Year column has missing values, so we can keep these rows,
> # as we will be removing this column prior to analysis.
```

Figure A8: Checking for null values in **R**.

Figure A9: Word clouds for reviews (left) and summaries (right) prior to word filtering.

Figure A10: Word clouds for reviews (left) and summaries (right) after word filtering.

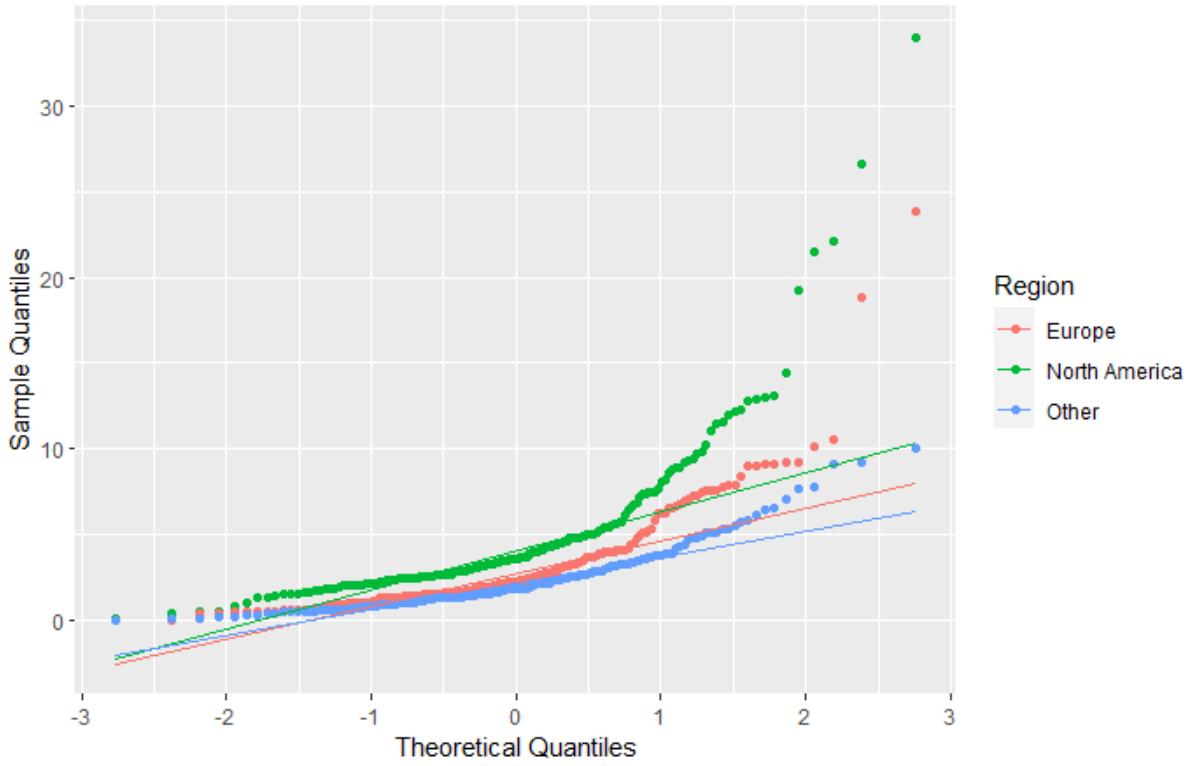

Figure A11: Q-Q plot of regional sales after grouping by product.

```
> shapiro.test(product_sales_total$Global_Sales_Total)

        Shapiro-Wilk normality test

data:  product_sales_total$Global_Sales_Total
W = 0.70955, p-value < 2.2e-16

> # The p-value is <0.05, so the data is not normally distributed.
>
> # Shapiro-Wilk test for North-American sales.
> shapiro.test(product_sales_total$NA_Sales_Total)

        Shapiro-Wilk normality test

data:  product_sales_total$NA_Sales_Total
W = 0.69813, p-value < 2.2e-16

> # The p-value is <0.05, so the data is not normally distributed.
>
> # Shapiro-Wilk test for European sales.
> shapiro.test(product_sales_total$EU_Sales_Total)

        Shapiro-Wilk normality test

data:  product_sales_total$EU_Sales_Total
W = 0.74058, p-value = 2.987e-16

> # The p-value is <0.05, so the data is not normally distributed.
>
> # Shapiro-Wilk test for Other sales.
> shapiro.test(product_sales_total$Other_Sales_Total)

        Shapiro-Wilk normality test

data:  product_sales_total$Other_Sales_Total
W = 0.85808, p-value = 9.64e-12

> # The p-value is <0.05, so the data is not normally distributed.
```

Figure A12: Shapiro-Wilk test of global and regional sales after grouping by product.

```
> # Skewness of global sales.
> skewness(product_sales_total$Global_Sales_Total)
[1] 3.066769
> # Positive skewness of roughly 3.1.
>
> # Kurtosis of global sales.
> kurtosis(product_sales_total$Global_Sales_Total)
[1] 17.79072
> # Kurtosis of roughly 17.8, much greater than 3, suggesting a heavy tail
> # and asymmetry.
>
> # Skewness of North-American sales.
> skewness(product_sales_total$NA_Sales_Total)
[1] 3.048198
> # Positive skewness of roughly 3.0.
>
> # Kurtosis of North-American sales.
> kurtosis(product_sales_total$NA_Sales_Total)
[1] 15.6026
> # Kurtosis of roughly 15.6, much greater than 3, suggesting a heavy tail
> # and asymmetry.
>
> # Skewness of European sales.
> skewness(product_sales_total$EU_Sales_Total)
[1] 2.886029
> # Positive skewness of roughly 2.9.
>
> # Kurtosis of European sales.
> kurtosis(product_sales_total$EU_Sales_Total)
[1] 16.22554
> # Kurtosis of roughly 16.2, much greater than 3, suggesting a heavy tail
> # and asymmetry.
>
> # Skewness of Other sales.
> skewness(product_sales_total$Other_Sales_Total)
[1] 1.625362
> # Positive skewness of roughly 1.6.
>
> # Kurtosis of Other sales.
> kurtosis(product_sales_total$Other_Sales_Total)
[1] 6.078014
> # Kurtosis of roughly 6.1, much greater than 3, suggesting a heavy tail
> # and asymmetry. However, this is lower than NA and EU by a significant margin,
> # thus suggesting better symmetry than the aforementioned regions.
```

Figure A13: Skewness and kurtosis of global and regional sales after grouping by product.

```
In [42]: # Print the R-squared value.
         print("The R^2 value is:", mlr.score(x_test, y_test)*100, "%")

         The R^2 value is: 82.90723396315805 %

         Close to the adjusted R^2 of the training data set: 84.2%
```

Figure A14: The R^2 score for our MLR model. It is very close to the original adjusted-R^2 score from the training data set.

```
In [46]: # Evaluate the model.

         # Call the 'metrics.mean_squared_error' function.
         print('Mean Square Error (MSE):', metrics.mean_squared_error(y_test, y_pred))

         # Call the 'metrics.mean_absolute_error' function.
         print('Mean Absolute Error (MAE):', metrics.mean_absolute_error(y_test, y_pred))

         # Call the 'metrics.mean_absolute_error' function.
         print('Mean Absolute Percentage Error (MAPE):', metrics.mean_absolute_percentage_error(y_test, y_pred), "%")

         Mean Square Error (MSE): 277188.70233220584
         Mean Absolute Error (MAE): 402.23503056376904
         Mean Absolute Percentage Error (MAPE): 1.278257724464776 %

         An MAPE of 1.3% < 10% is excellent!
```

Figure A15: The MAPE for our MLR model in **Python**. It is $\ll 10\%$, which is excellent!

	VIF Factor	features
0	20.73	const
1	1.06	spending_score
2	1.00	remuneration
3	1.06	age

Figure A16: VIF lies within the range $[1, \infty]$, so all VIFs being exceedingly close to one means there is minimum multicollinearity.

```
Results of Breusch-Pagan test:
{'LM stat': 39.20687709402344, 'LM Test p-value': 1.56905186811619e-08, 'F-stat': 13.363756098044375, 'F-test p-value': 1.28991
282169574e-08}
```

Figure A17: A Breusch-Pagan value $\ll 0.05$ implies heteroscedasticity.

```
> # Calculate mean-absolute-percentage error (MAPE).
> mape(sales_sample$Global_Sales, predict_123)
[1] 0.1053323
> # MAPE: 10.5%. Less than 10% is excellent, so this prediction is very good.
```

Figure A18: The MAPE for our MLR model in **R**. It is almost 10%, which is very good.

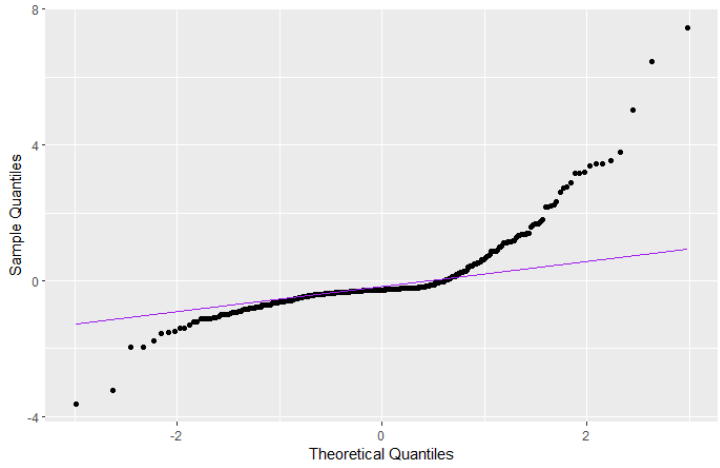

Figure A19: Q-Q plot of our MLR model in **R**, used to predict global sales using NA and EU sales.

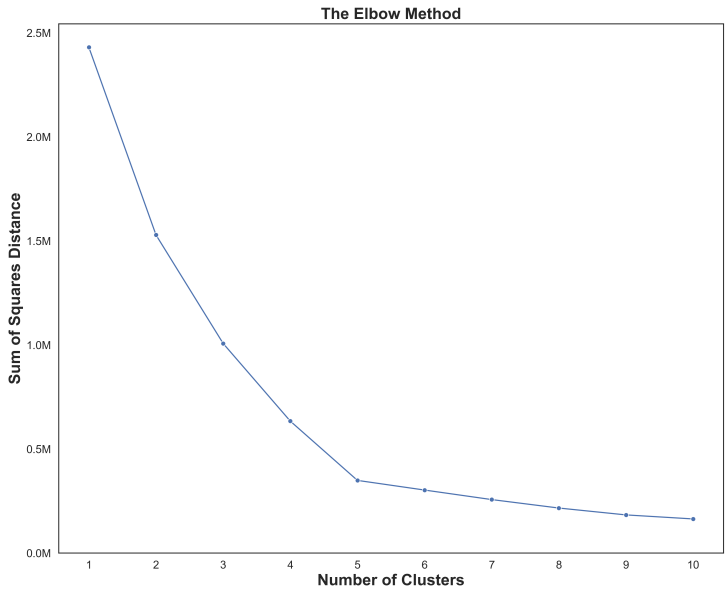

Figure A20: The elbow test starts levelling off at around $k = 5$.

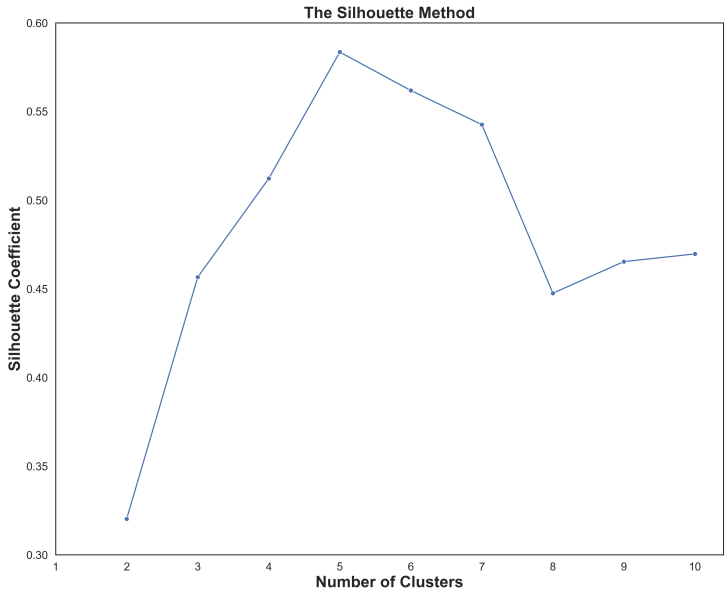

Figure A21: The silhouette test peaks at around $k = 5$.

```
K-Means Prediction for Four Clusters (k = 4):

0    1013
3     356
2     351
1     280
Name: K-Means Prediction (k = 4), dtype: int64


K-Means Prediction for Five Clusters (k = 5):

0    774
3    356
2    330
1    271
4    269
Name: K-Means Prediction (k = 5), dtype: int64


K-Means Prediction for Six Clusters (k = 6):

0    767
1    356
4    271
3    269
2    214
5    123
Name: K-Means Prediction (k = 6), dtype: int64
```

Figure A22: Of these different k values, $k = 5$ produces the most evenly-distributed clusters.