Turtle Games
Technical Report

By Geoff Thomas Widdowson

Beng(hons) IPS

Introduction/Problem Statement

In this assignment I aim to analyze customer behavior to find trends that could improve sales of the company's game products.

The questions I am looking to answer through this are:

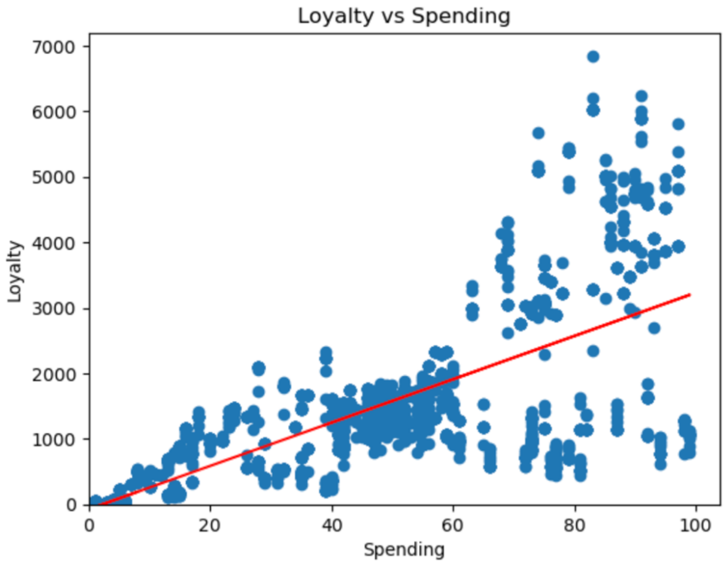
1. How do customers engage with loyalty points and are they important?
2. Can customers be segmented into groups, and be targeted by the marketing department?
3. Is text data from customer reviews useful in informing marketing campaigns and improving the business?
4. Do descriptive statistics provide insights into the suitability of the loyalty points data to create predictive models?

Analytical Approach and Exploration

I began exploring all the available data using Python where I considered the data and how I could break down the questions before conducting further analysis using R.

Initially I cleaned the data in python finding no duplicate data, dropping unnecessary columns and renaming them cleanly. There were 2000 rows in the data set once cleaned and screenshots of example code can be found in the appendix.

Following this I created a model to predict loyalty against spending, salary and age using linear regression modeling in to understand customer loyalty fully.

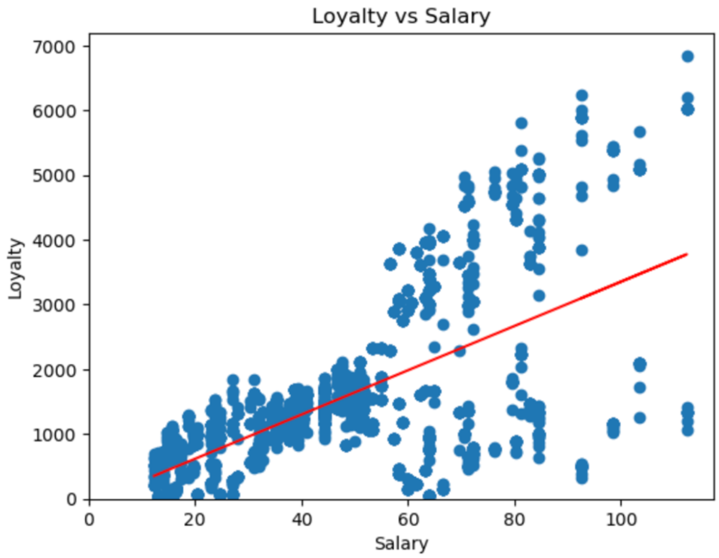


Red = Predicted : Blue = Actual

We can clearly see that in these models loyalty increases with spending and salary showing their importance!

However it is worth noting that around £60K in both the line of best fit does not accurately model customer behaviour with a wider unequal variance of data and *h*eteroscedasticity occurring.

This indicates that Linear regression is a poor model for predicting and further investigation is needed.

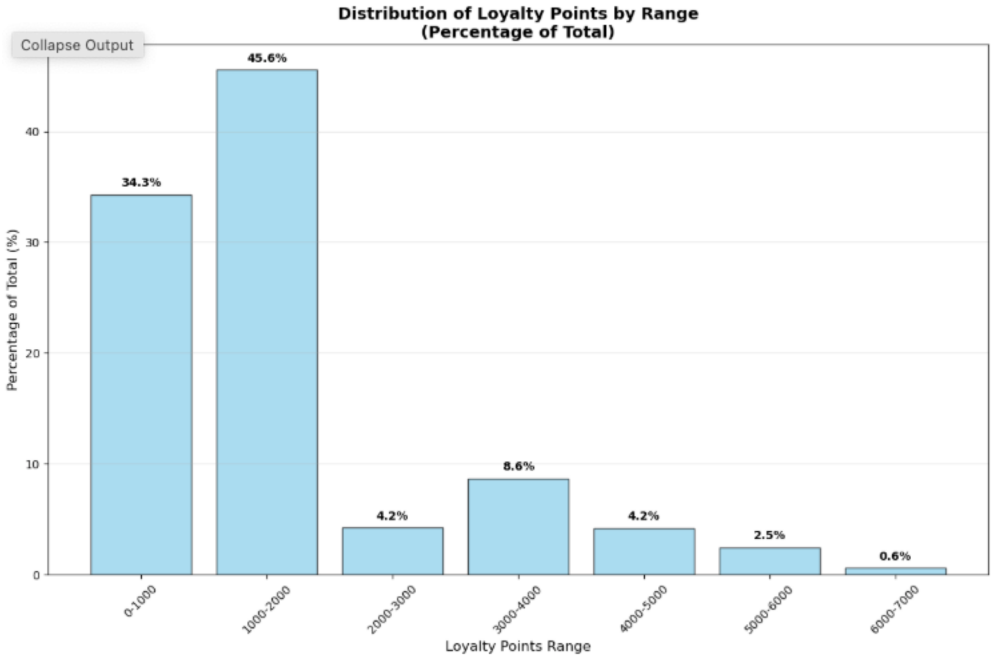


Looking at the correlation with Loyalty and the models R squared values these came out as 0.452 for spending and 0.380 of Salary confirming the model could be better but is ok!

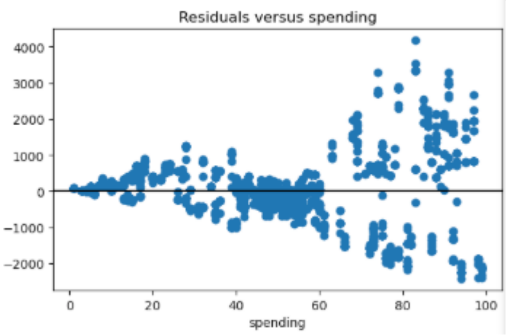
Examining age alongside loyalty there was no strong correlation in the data.

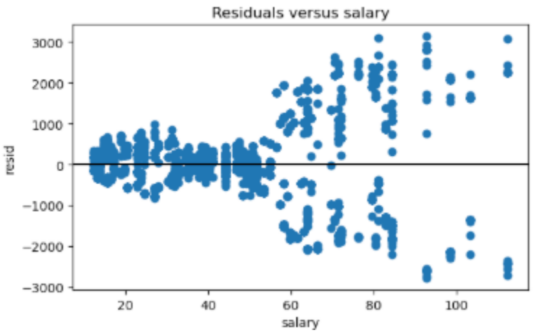
Analytical Approach (Cont)

It is clear from looking at the data that there is a wide variance in the range of loyalty points in customers and this can be seen clearly below.



More data in the whole range would improve modeling!





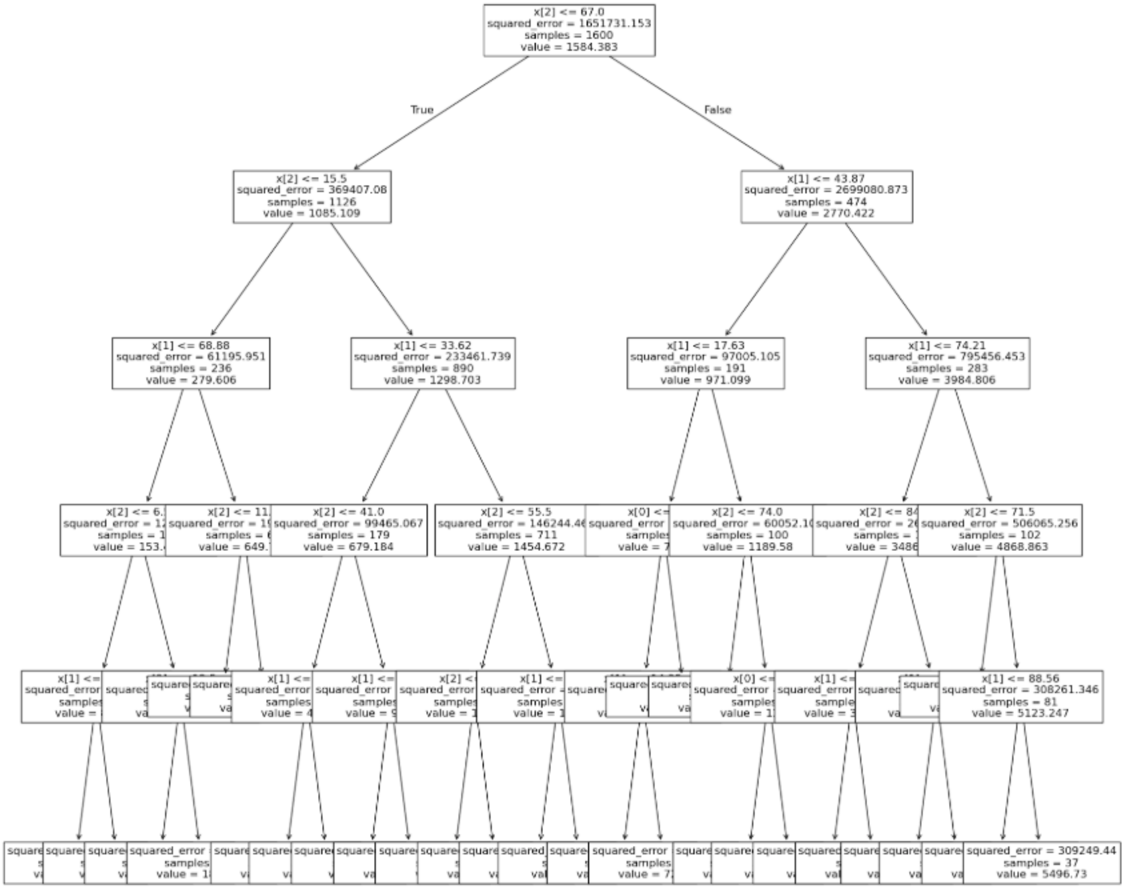
It was also worth noting that there was a broad spread of residuals for high earners and high spenders confirming the lack of accuracy in the models as the numbers increase.

Analytical Approach – Decision Tree Modeling

Following this I looked at the data using decision trees. I believe these models to be better and more accurate so we could look further at using this further.

The Best model I created had an accuracy of 98% however I believe that the existing tree may be over fitted so further examination is needed to ensure it is truly accurate.

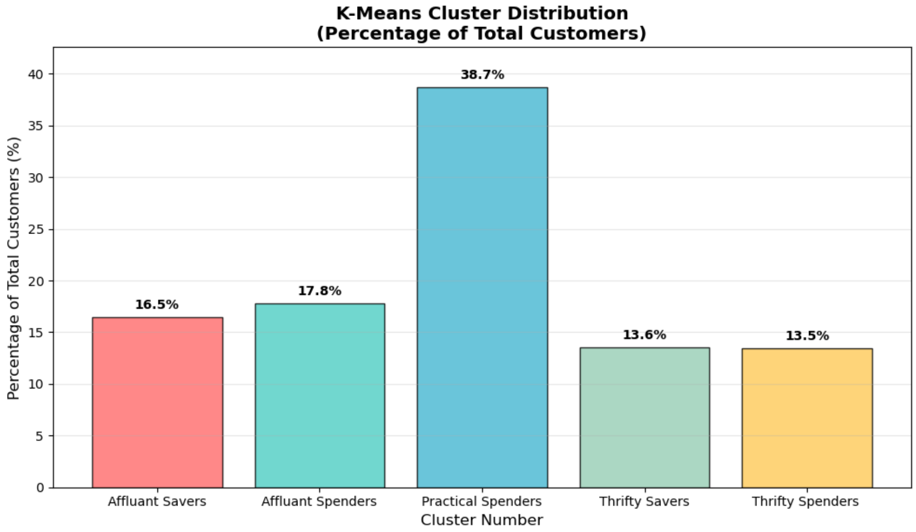
You can see below the best fitting tree created during the analysis.

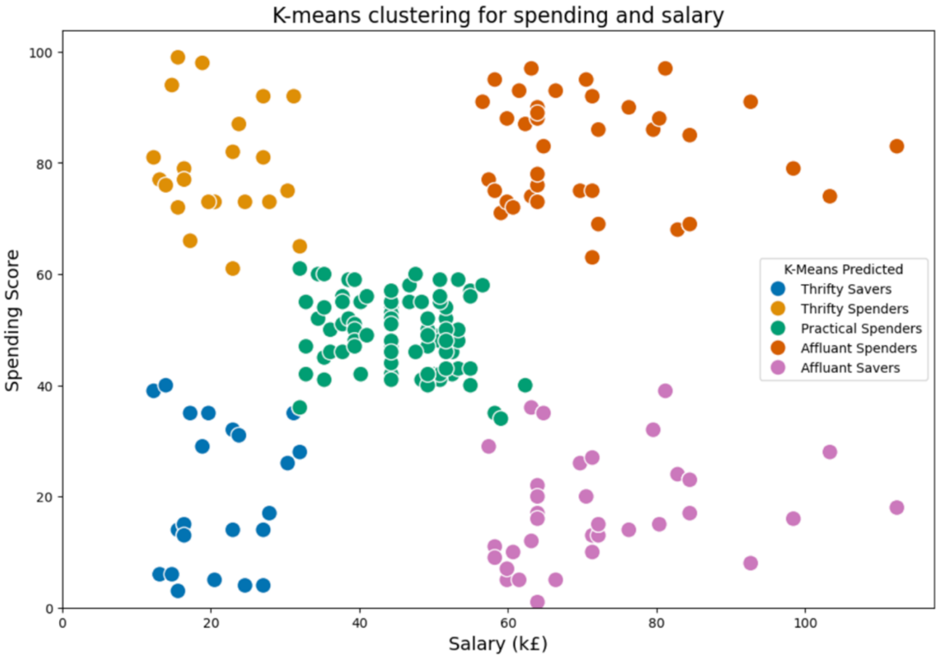


Analytical Approach and clustering

I went onto look at clustering models using K-means testing which showed that customers could be clearly separated into 5 distinct groups.

I labeled these according to spending/ saving patterns it was obvious that over 50% customers appear to be earning around £40K or more!

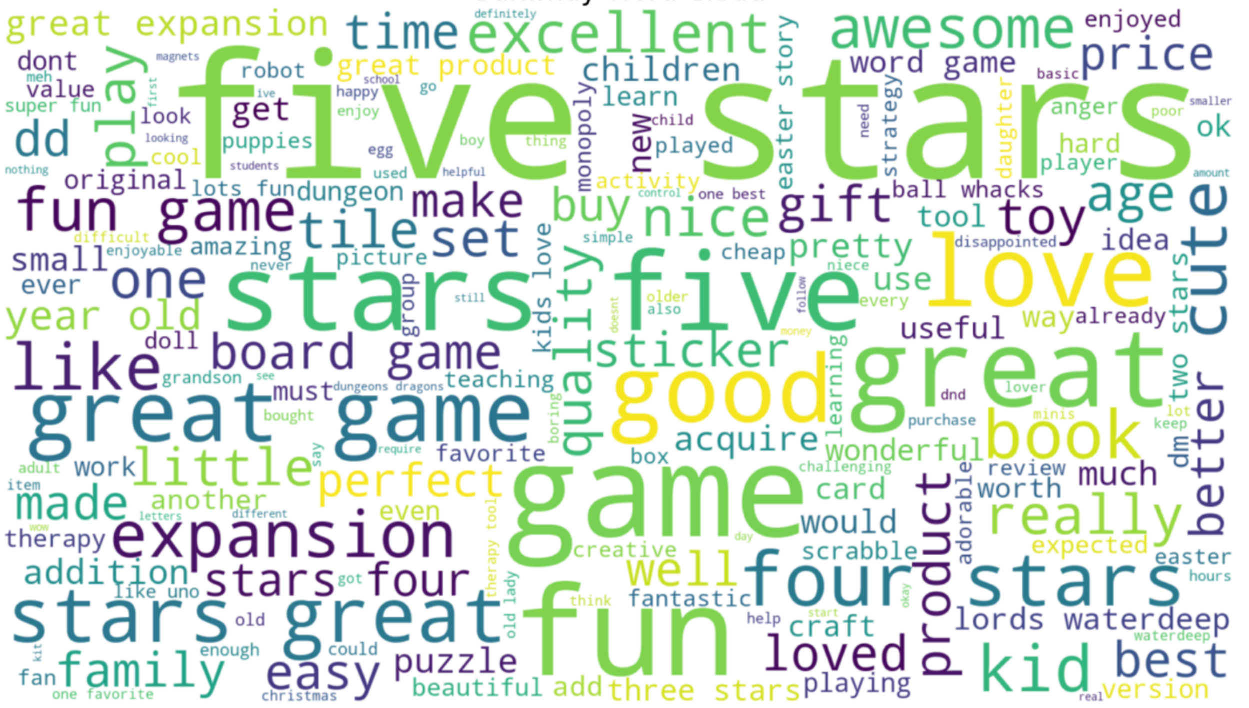




It is clear that customers can be segmented and thus could be targeted by the marketing department.

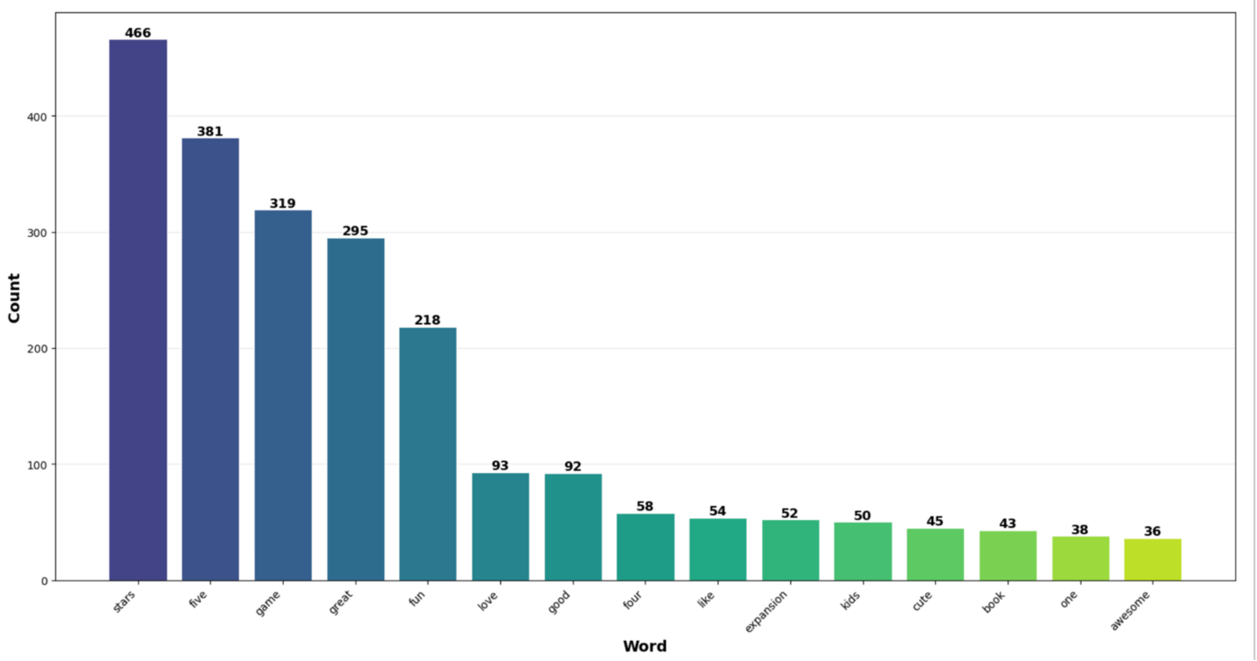
I used Natural Language Processing (NLP) in my analysis in order to understand text data form customer reviews and how useful in could be.

In doing this I initially created word clouds to illustrate the most popular words being used in the reviews and their summaries.



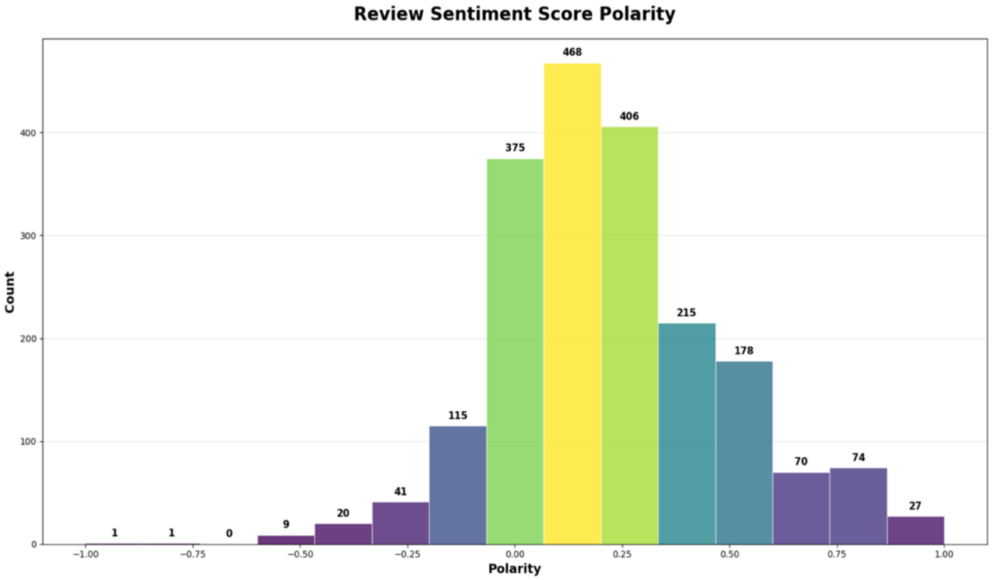
Summay Word Cloud

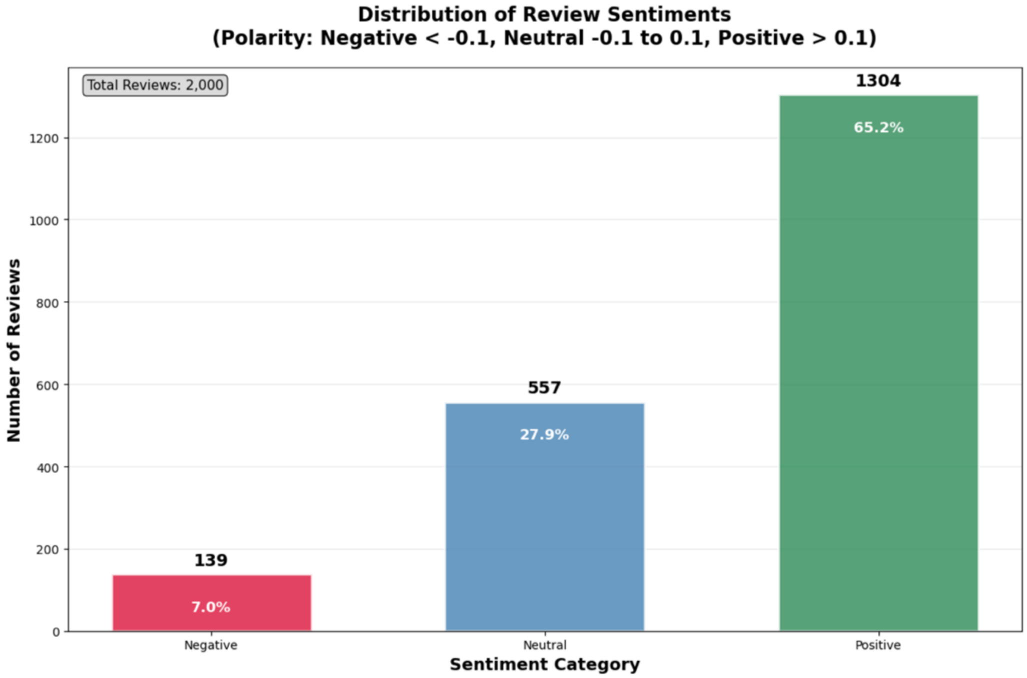
Most popular words are clearly Five Stars as well as fun and great positivity abounds!



Following this I did a sentiment analysis of both reviews and summaries in order understand the popularity expressed in the text further.

In looking at the Reviews we can see examples of the analysis below:

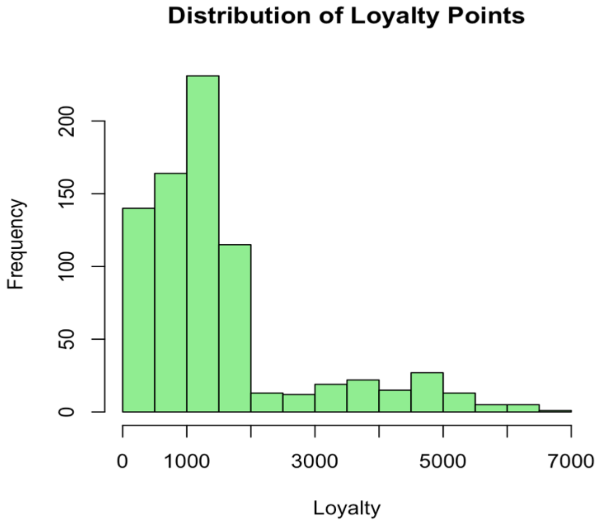


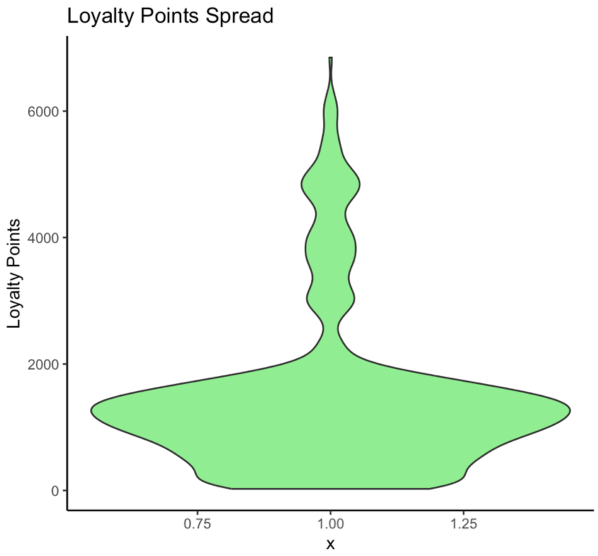


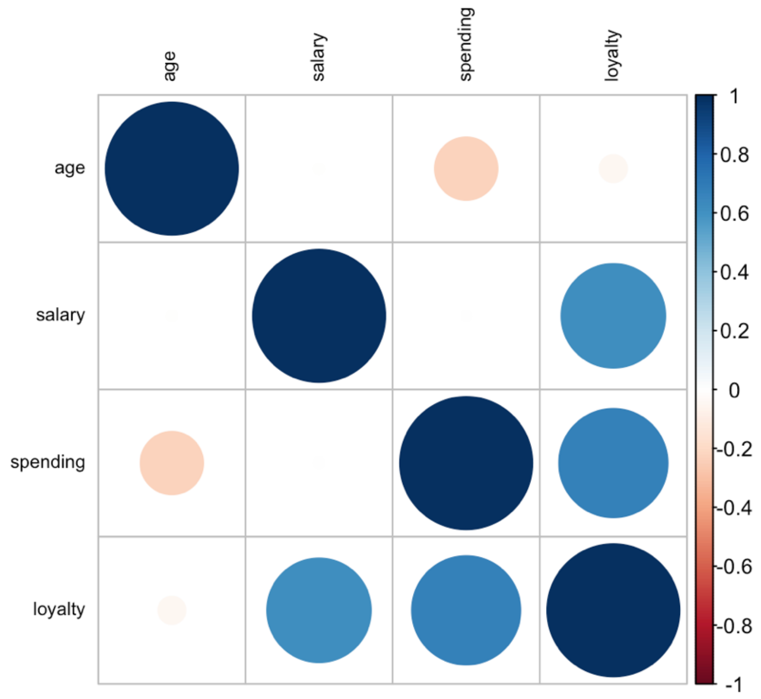
It clear from looking at this text analysis we get overall positive sentiments that, alongside other data, could be used to target customers by the marketing team based on sentiment.

I chose not look at the customer demographics such as gender and education as the data could not be used to gain any meaningful insight alongside spending scores and loyalty points. This could be considered further in the future with more specific data.

I did consider the distribution of loyalty points and spending scores as can be seen from my analysis work in R. It confirms my earlier investigations showing most loyalty points are under 2000.





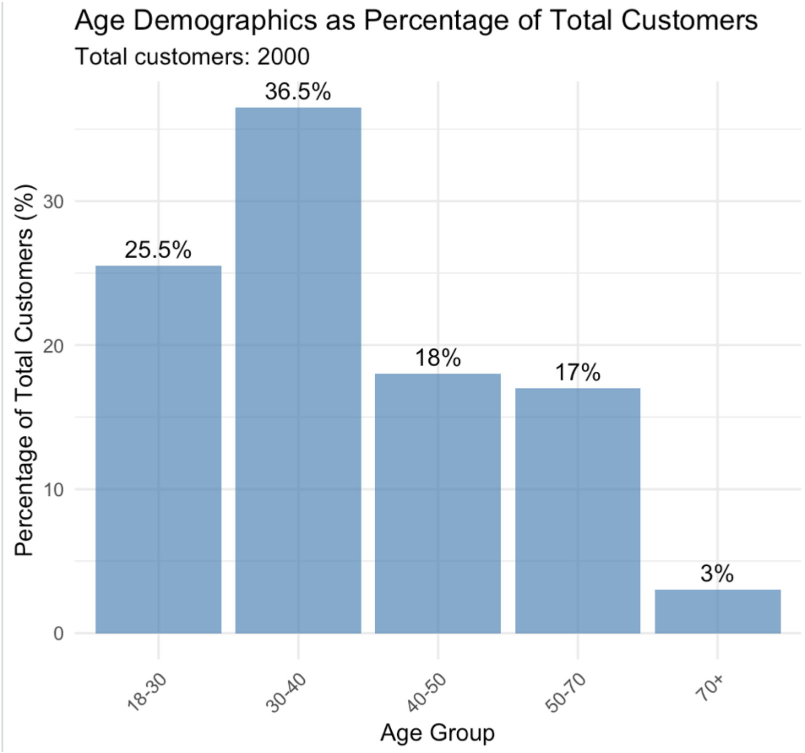


I confirmed my initial results from Python with regards to linear regression in R.

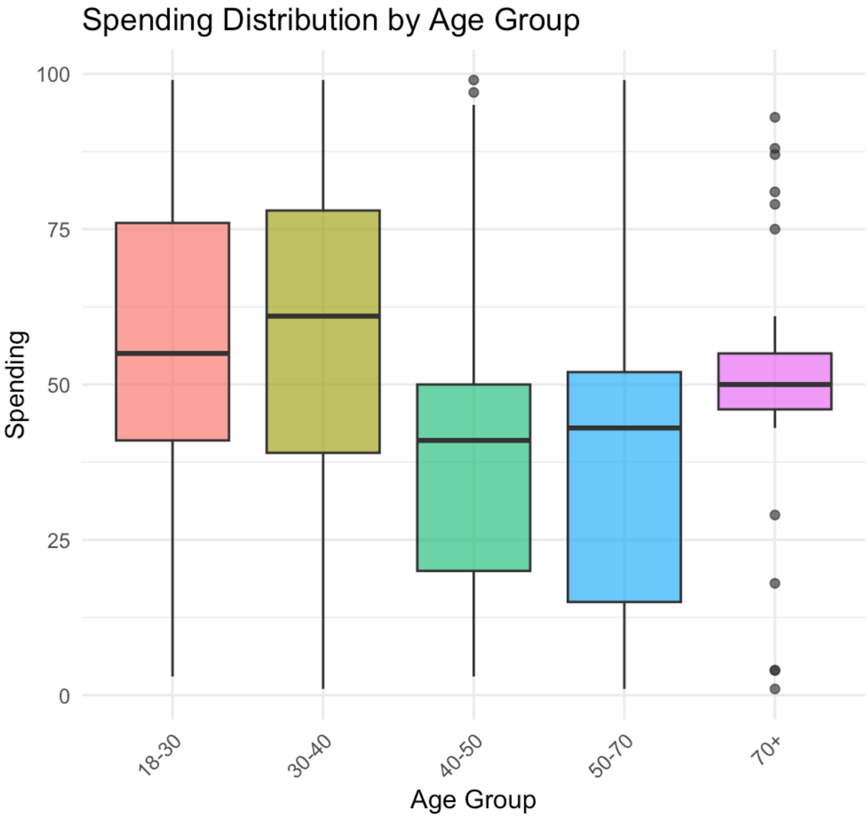
The creation of a correlation matrix confirmed the positive relationship with loyalty points, salary and spending while making me consider looking at age further.

The correlation matrix is shown to the left here.

Looking further at age I explored the customers and spending score distribution.

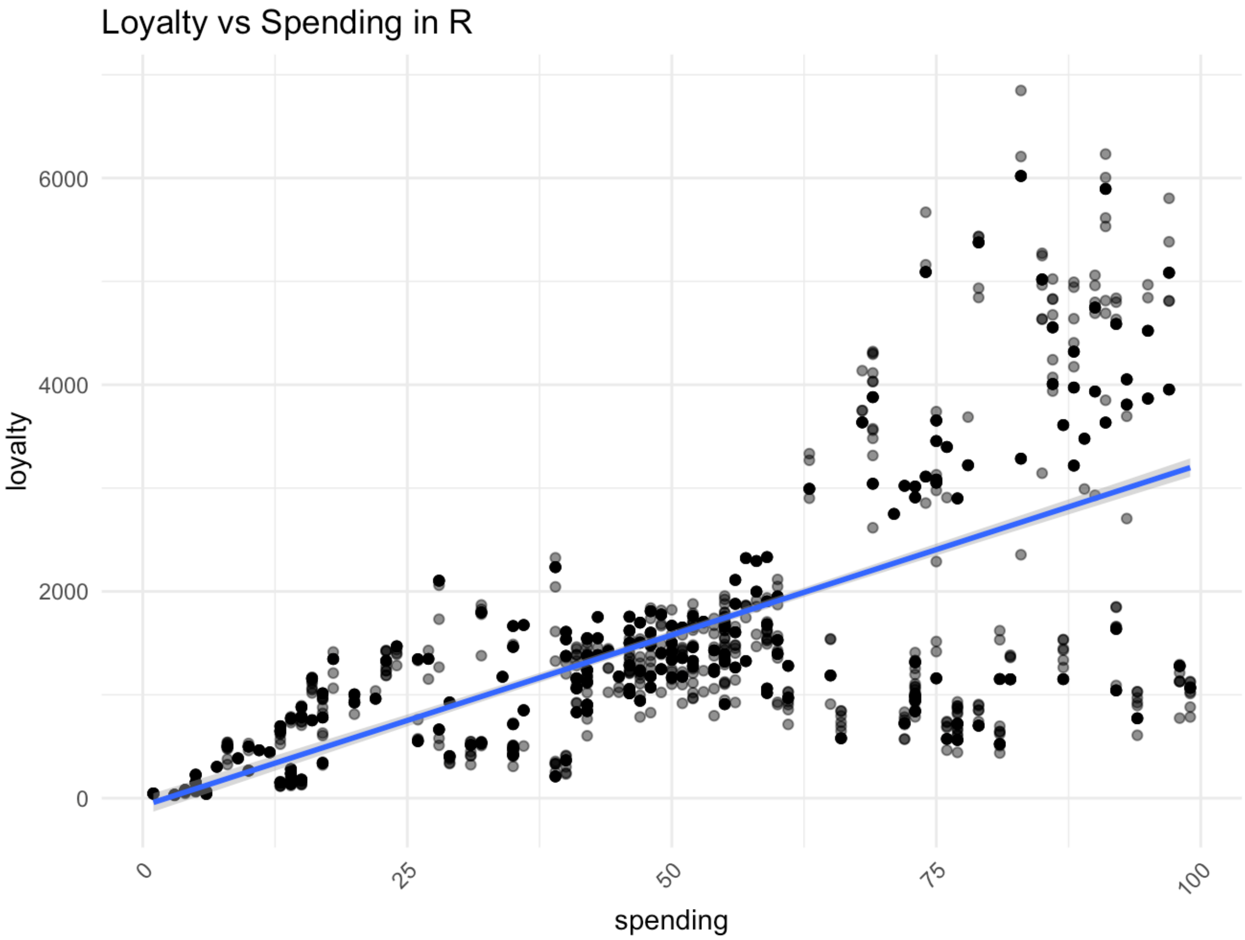


We can clearly see over 62% of customers are under the age of 40!



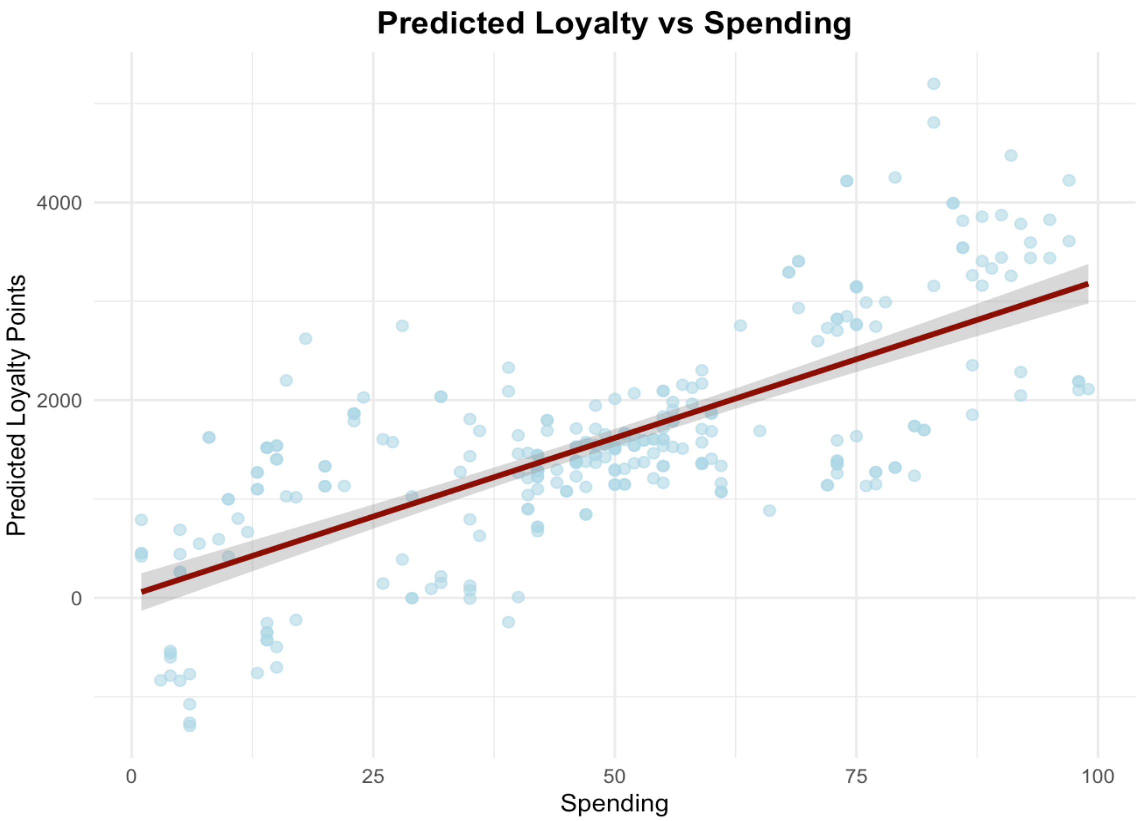
It is clear that younger customers have a higher spending score!

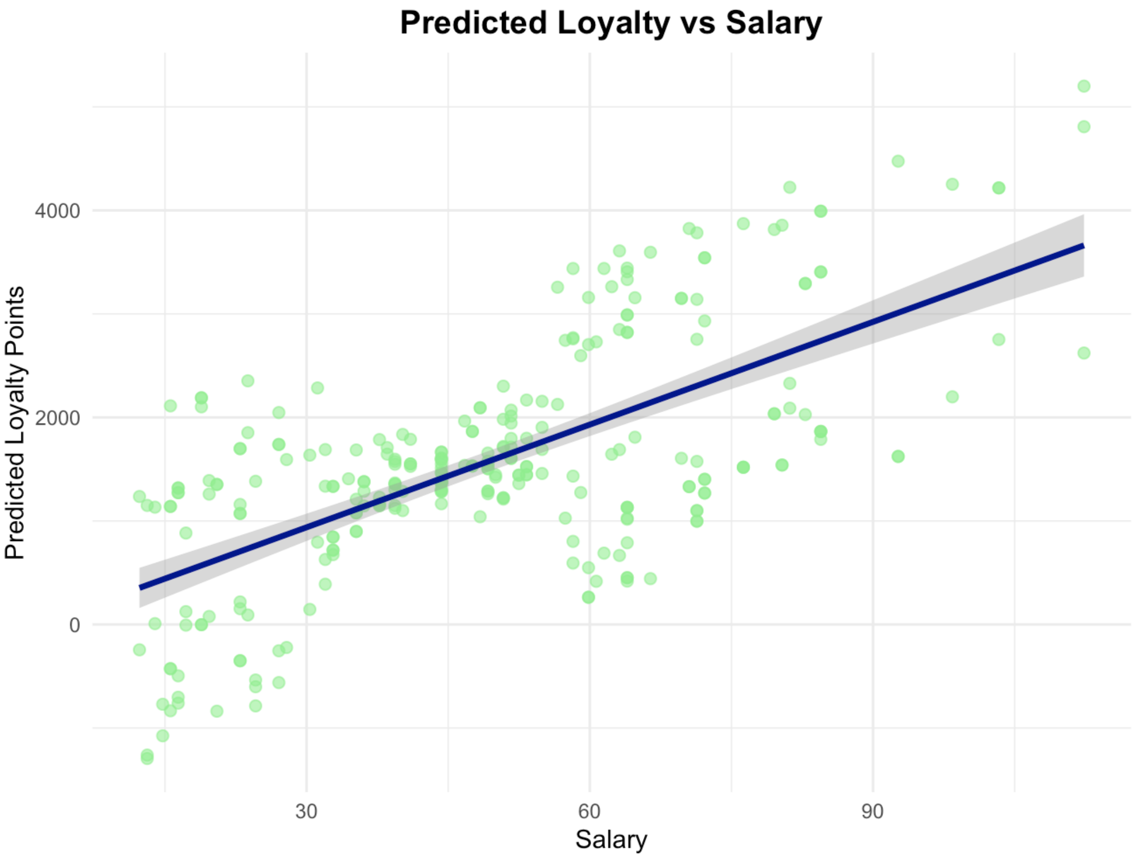
I went on to do various plots in R as we can see below with a linear plot:



Using Multiple Linear regression, I created a new model to see if I could improve on things.

Training the model I was able the accuracy was much higher and an r-squared of 0.838 meaning at almost 84% the percentage of variance makes this an excellent model! You can see over the page the plots I created that show the data points more evenly spread along the line of best fit for predicted loyalty against spending and salary.


Predicted Loyalty vs Spending
Predicted Loyalty Points
Spending
4000
2000
0
0
25
50
75
100


Predicted Loyalty vs Salary
Predicted Loyalty Points
Salary
4000
2000
0
30
60
90

Key Insights:

Question 1 – Loyalty Engagement:

- The Majority of customers have less than 2000 loyalty points and are under 40!

- Salary and Spending can be used to predict loyalty.

- Loyalty appears to increase with salary and spending.

Question 2 – Customer segmentation

- With clustering we were able to identify 5 distinct groups based upon salary and spending

Question 3 – Text Reviews

- Looking at words and sentiment we were able to clearly show that there is very little negativity towards gaming products less than 8% in reviews and under 6% in the summaries!

Question 4 – Descriptive stats and modeling

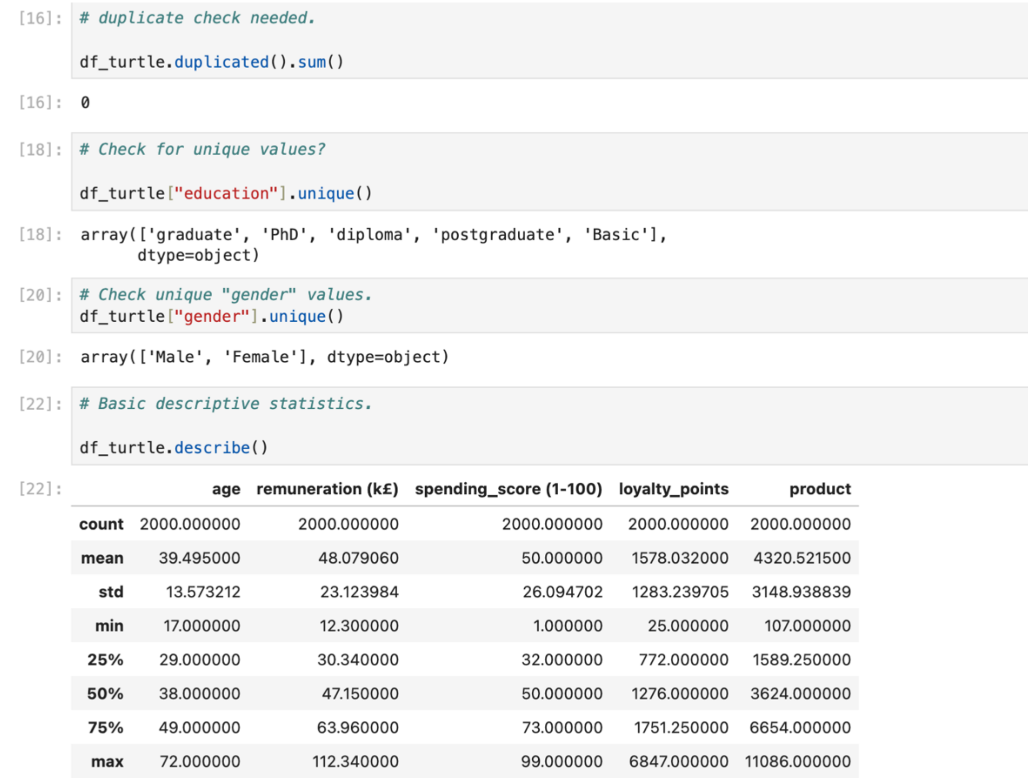
- Modeling was useful with lower salary and spending but less reliable with higher numbers and a lack of actual data means further investigation and creating of new models would be useful.

Recommendations:

- Focus on targeting customers with loyalty of the 1000-2000
- Investigate further loyalty data above 2000 and try to improve this.
- Try to attract further affluent spenders while nurturing existing customers with 2000+ points!
- Look to target 5 clusters of customers specifically based on Spending and Salary to improve sales and customer loyalty.
- Target customers with high loyalty to gain more positive reviews
- Consider how to improve engagement in loyalty for thrifty and affluent savers.
- Is there further information on spending specifics that could be analysed to gain greater insight
- Collect more specific customer and product data to understand loyalty to the brand and what is being bought.
- Consider spending further time on modelling to improve predictions!
- Look at how specifically loyalty points are earned and how this can be tailored and targeted with specific products to specific customer groups.
- Would a rewards program help drive engagement and spending further or is the existing system with loyalty points compelling enough to help drive growth?
- Does a sharp drop-off above 2000 loyalty points indicate issues in the system? If so what are these and how can they be resolved?

Appendix:

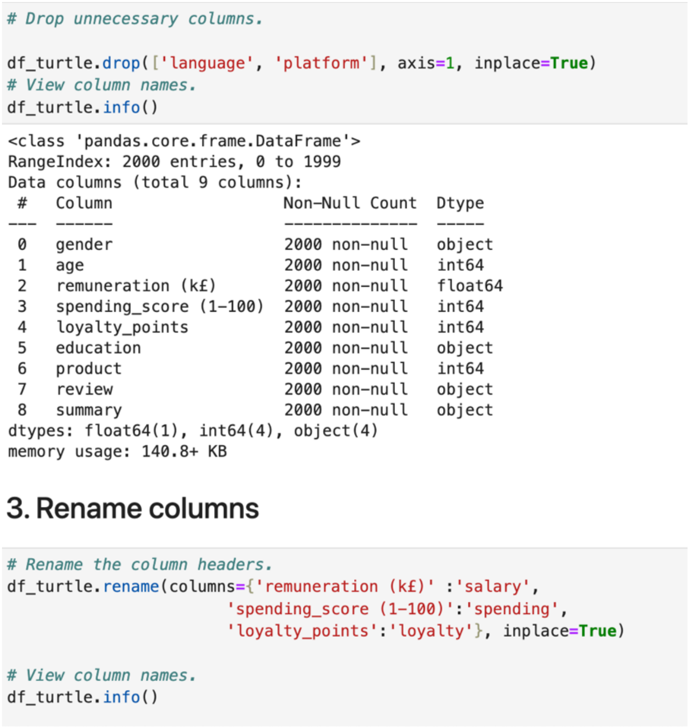
Data Cleaning and checks in Python:



```
[16]: # duplicate check needed.

      df_turtle.duplicated().sum()

[16]: 0

[18]: # Check for unique values?

      df_turtle["education"].unique()

[18]: array(['graduate', 'PhD', 'diploma', 'postgraduate', 'Basic'],
            dtype=object)

[20]: # Check unique "gender" values.
      df_turtle["gender"].unique()

[20]: array(['Male', 'Female'], dtype=object)

[22]: # Basic descriptive statistics.

      df_turtle.describe()
```

[22]:

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

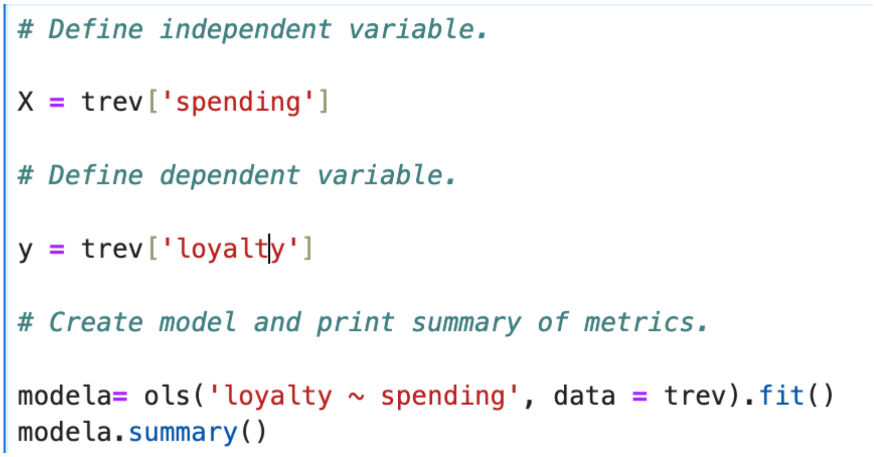


```
# Drop unnecessary columns.

df_turtle.drop(['language', 'platform'], axis=1, inplace=True)
# View column names.
df_turtle.info()
```

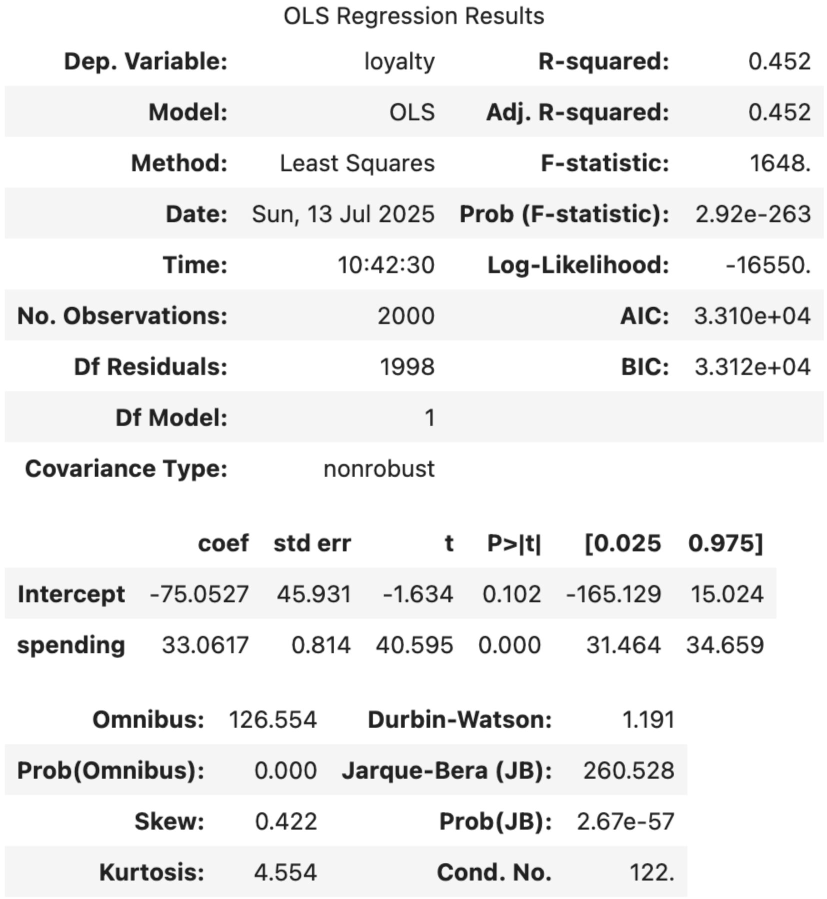
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 9 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   gender                  2000 non-null   object
 1   age                     2000 non-null   int64
 2   remuneration (k£)       2000 non-null   float64
 3   spending_score (1-100)  2000 non-null   int64
 4   loyalty_points          2000 non-null   int64
 5   education               2000 non-null   object
 6   product                 2000 non-null   int64
 7   review                  2000 non-null   object
 8   summary                 2000 non-null   object
dtypes: float64(1), int64(4), object(4)
memory usage: 140.8+ KB
```

3. Rename columns

```
# Rename the column headers.
df_turtle.rename(columns={'remuneration (k£)' :'salary',
                          'spending_score (1-100)':'spending',
                          'loyalty_points':'loyalty'}, inplace=True)

# View column names.
df_turtle.info()
```



```python
# Define independent variable.

X = trev['spending']

# Define dependent variable.

y = trev['loyalty']

# Create model and print summary of metrics.

modela= ols('loyalty ~ spending', data = trev).fit()
modela.summary()
```



```
                         OLS Regression Results
     Dep. Variable:           loyalty          R-squared:      0.452
             Model:               OLS      Adj. R-squared:      0.452
            Method:     Least Squares         F-statistic:      1648.
              Date:  Sun, 13 Jul 2025  Prob (F-statistic):  2.92e-263
              Time:          10:42:30      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
  spending     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.
```

Further cleaning and code data can be found in my Python and R scripts as well as further plots used in exploring the data.