Technical Report

1. Cleaning the data

No null values were found. I removed ‘language’ and ‘platform’ columns from the dataset because they added no information. Renamed ‘remuneration (k£)’ to ‘remuneration’ and ‘spending\_score (1-100)’ to ‘spending score’. Saved the cleaned data to a new csv file to preserve the original file.

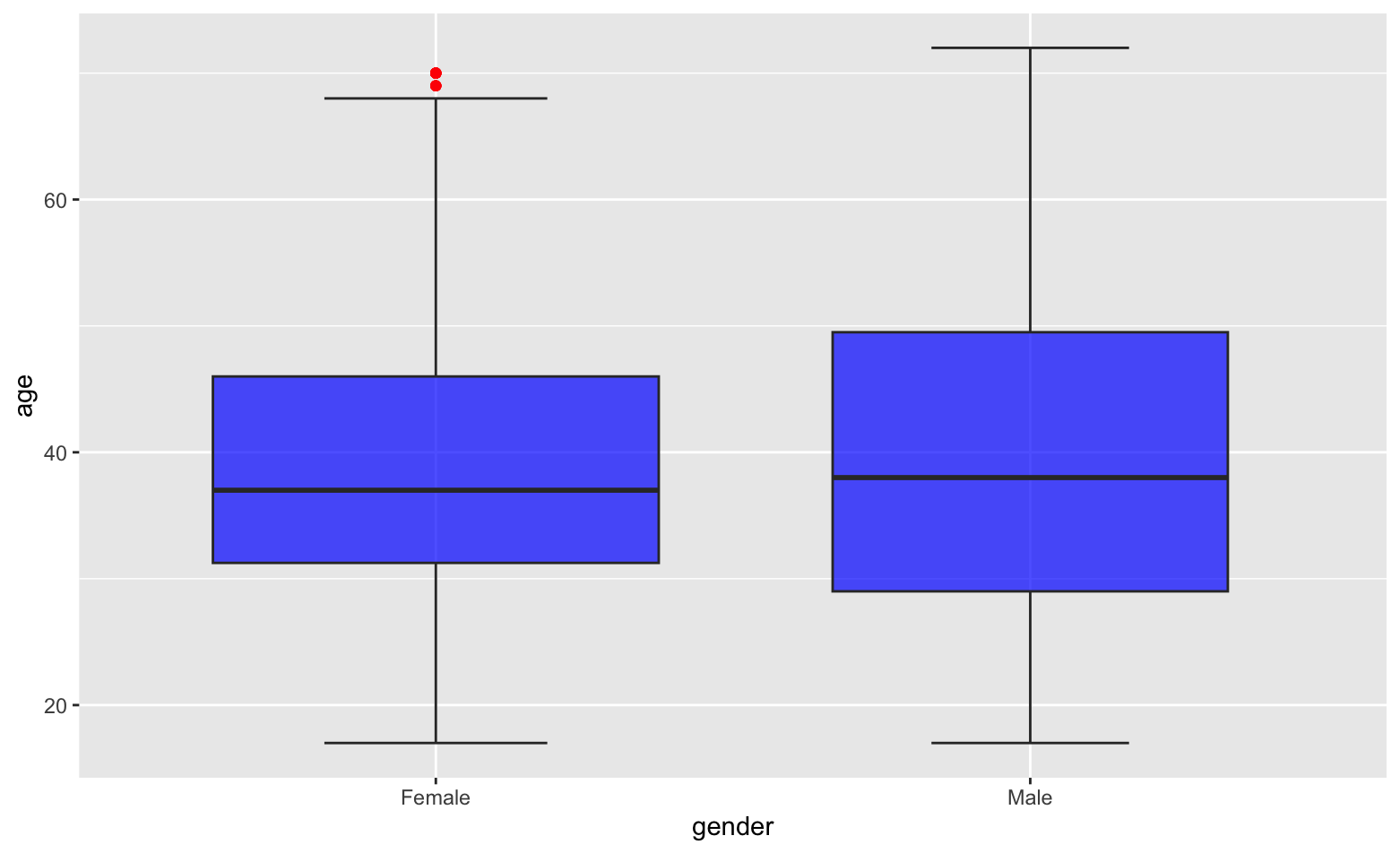
1. Linear Regression
2. Python

In Python, I checked OLS assumptions and found the data to be poorly suited for OLS regression, meaning that results would likely be misleading. The independent variables, or features, of focus are the ‘age’, ‘remuneration’, and ‘spending\_score’ variables. Firstly, there is no clear linear relationship between any of the independent variables and the dependent variable, loyalty\_points. Secondly, using the variance inflation factor (VIF), I found there to be little multicollinearity between the features. Thirdly, the variance of the residuals exhibited heteroscedasticity. Fourthly, the residuals were not normally distributed. Finally, errors appeared to be autocorrelated. There are many measures that could be taken to improve the suitability of the data for OLS regression, but because of the extent to the problems, I decided it would be more appropriate to focus more on methods that are less limited by heteroscedasticity in the residuals and non-normal distribution, such as decision trees.

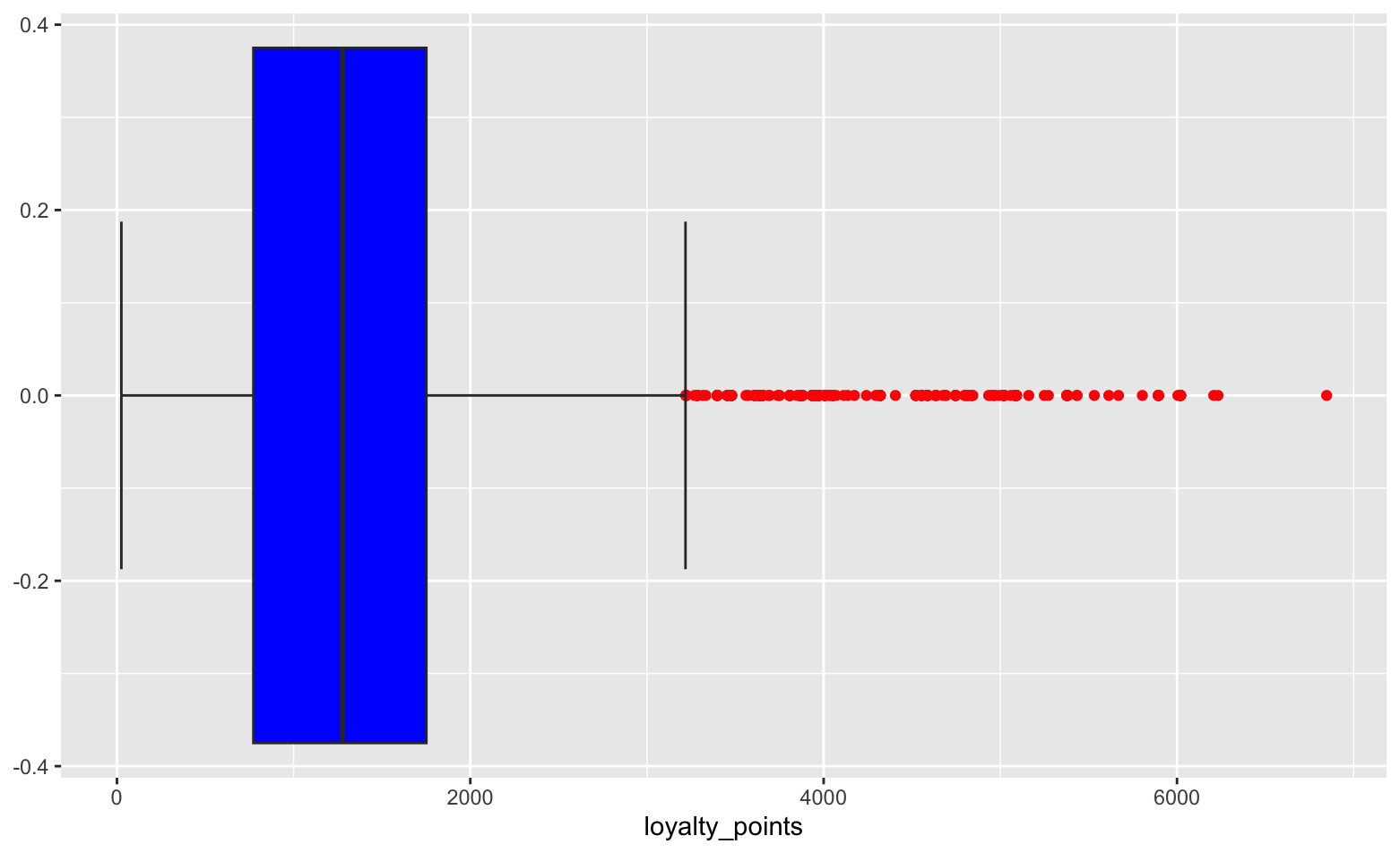
I ran the multiple linear regression with the aforementioned variables and found the R-squared to perform very well, with a value of 0.84. Remuneration and spending\_score variables showed to be very significant, with coefficients of 34 against age’s coefficient of 11. Next, using train\_test\_split() from sklearn.model\_selection, I split the data into training and test sets and fit a regression model using the training data. Finally, I ran the regression using the X (independent variables) test data and compared predicted Y (dependent variable) output against the actual Y test data. The R-squared and adjusted R-squared were 0.84 and 0.83, respectively, which implies high explanatory power and no evidence of overfitting.

1. R

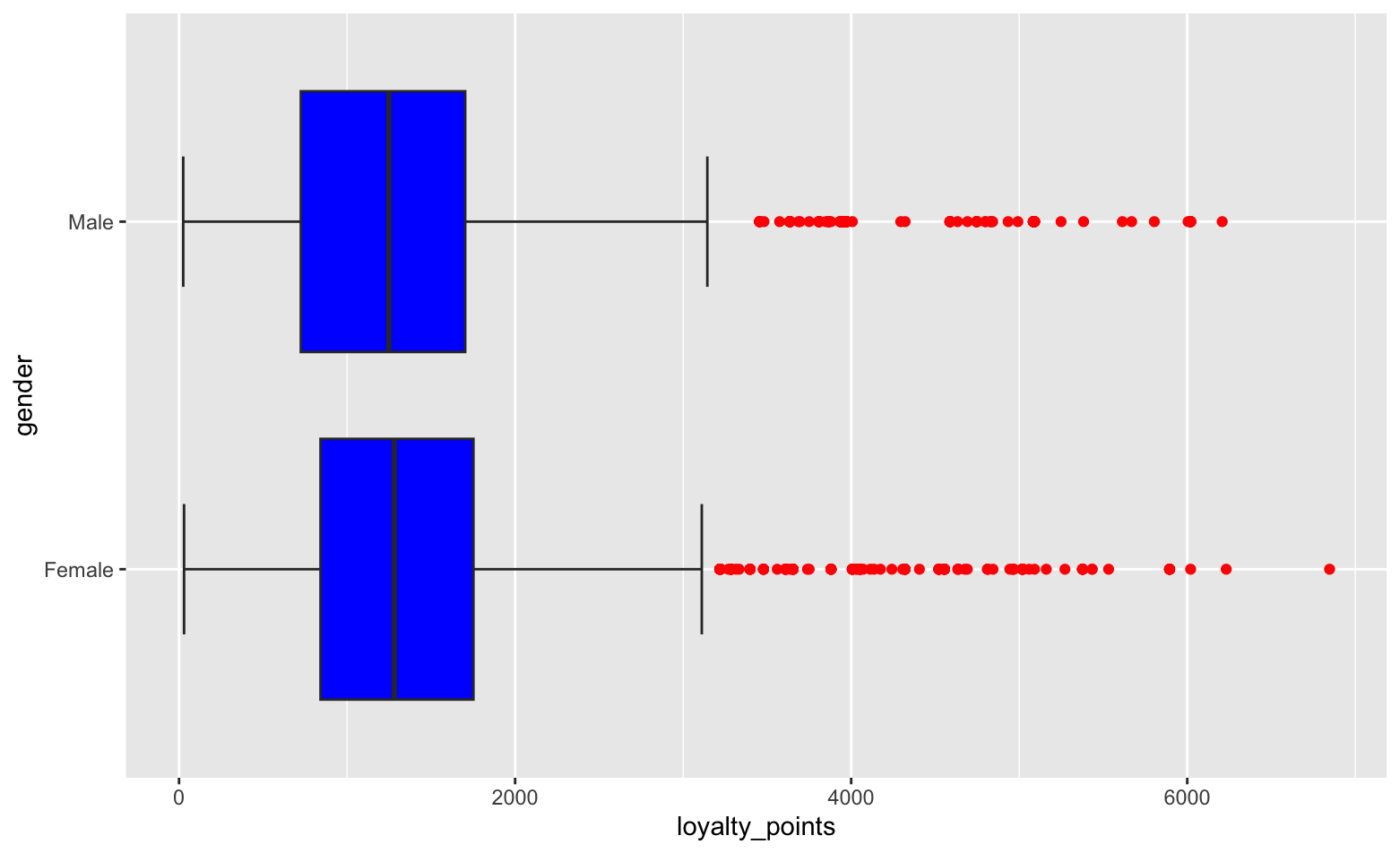
In R, I use the tidyverse package to carry out exploratory data analysis and the same regression analysis. The regression results were the same. The exploratory analysis reaffirmed my decision to include the chosen X variables and exclude the rest. All three variables, age, remuneration, and spending\_score have clear relationships with loyalty\_points. The only other variable I seriously considered adding was gender. If gender proved to be a relevant variable, advertising could be tailored to the different gender groups.



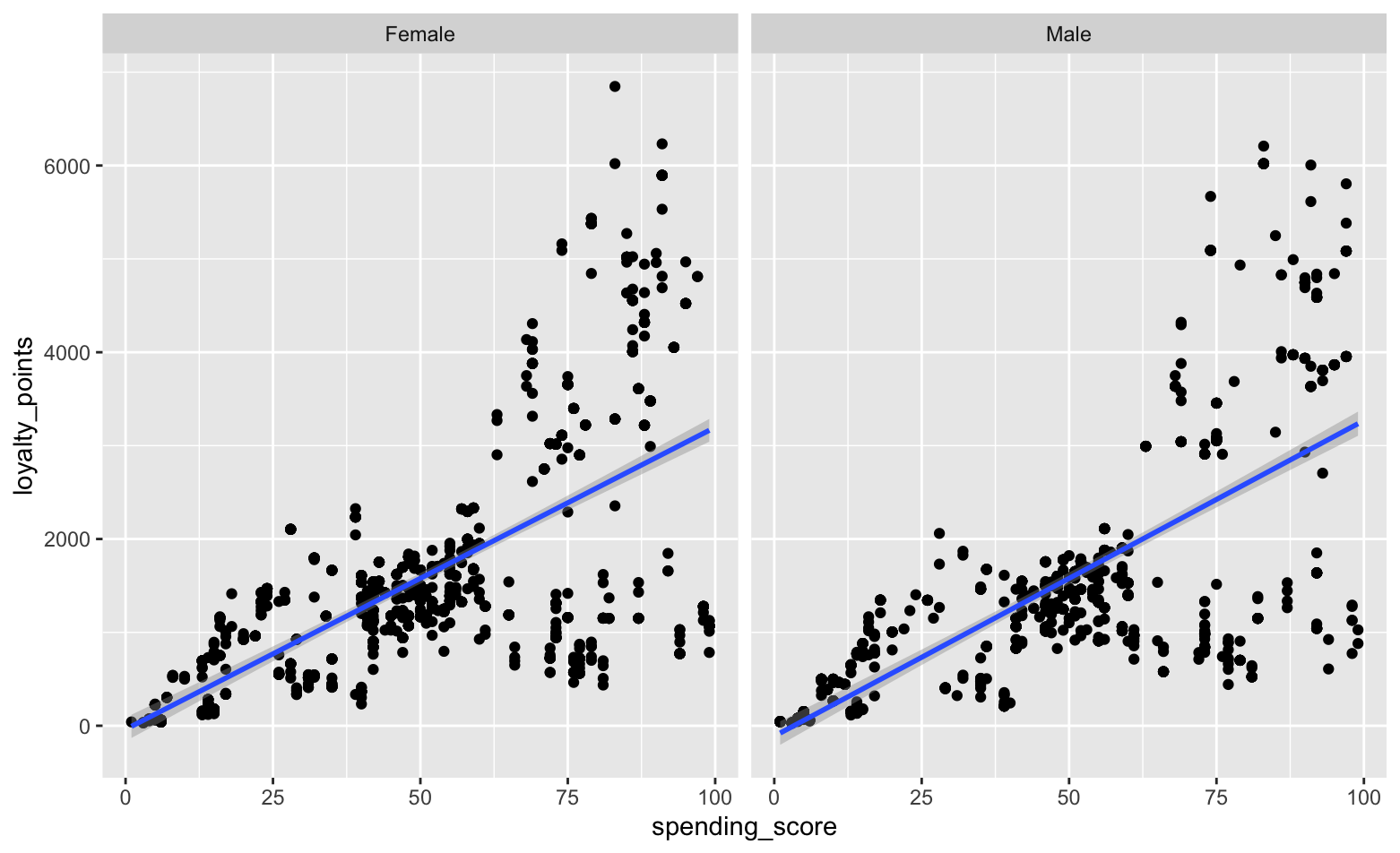
The figure above shows the mean age is similar for men and women, but the age range and interquartile range is larger in the male group, implying that it is more common for men across all age groups to be interested in games.



Using a boxplot to map out the loyalty\_points distribution, we se that the mean amount of loyalty\_points is less that 1500, with many outliers beyond the upper limit.



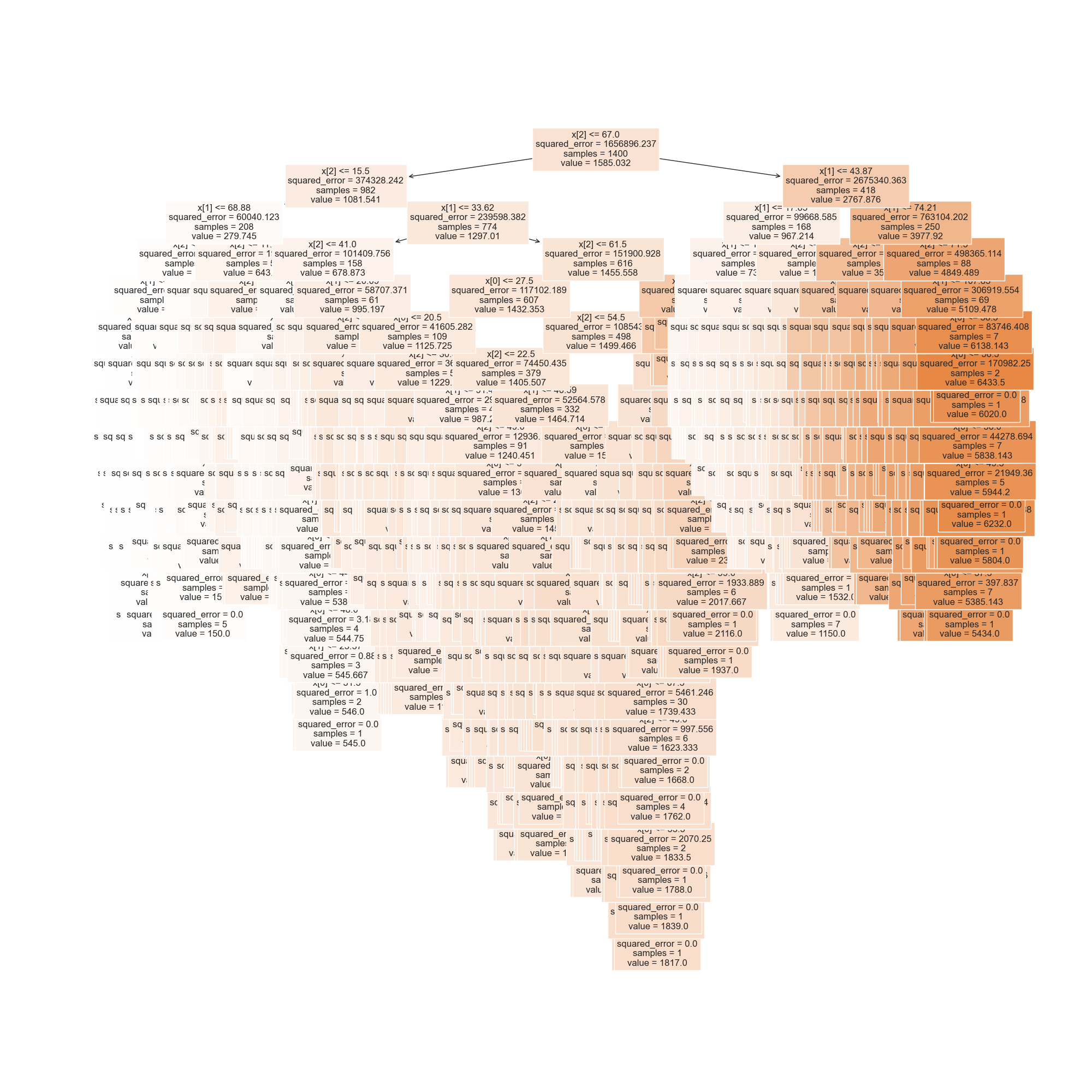
Comparing the loyalty\_points by gender (above), I see little variation.



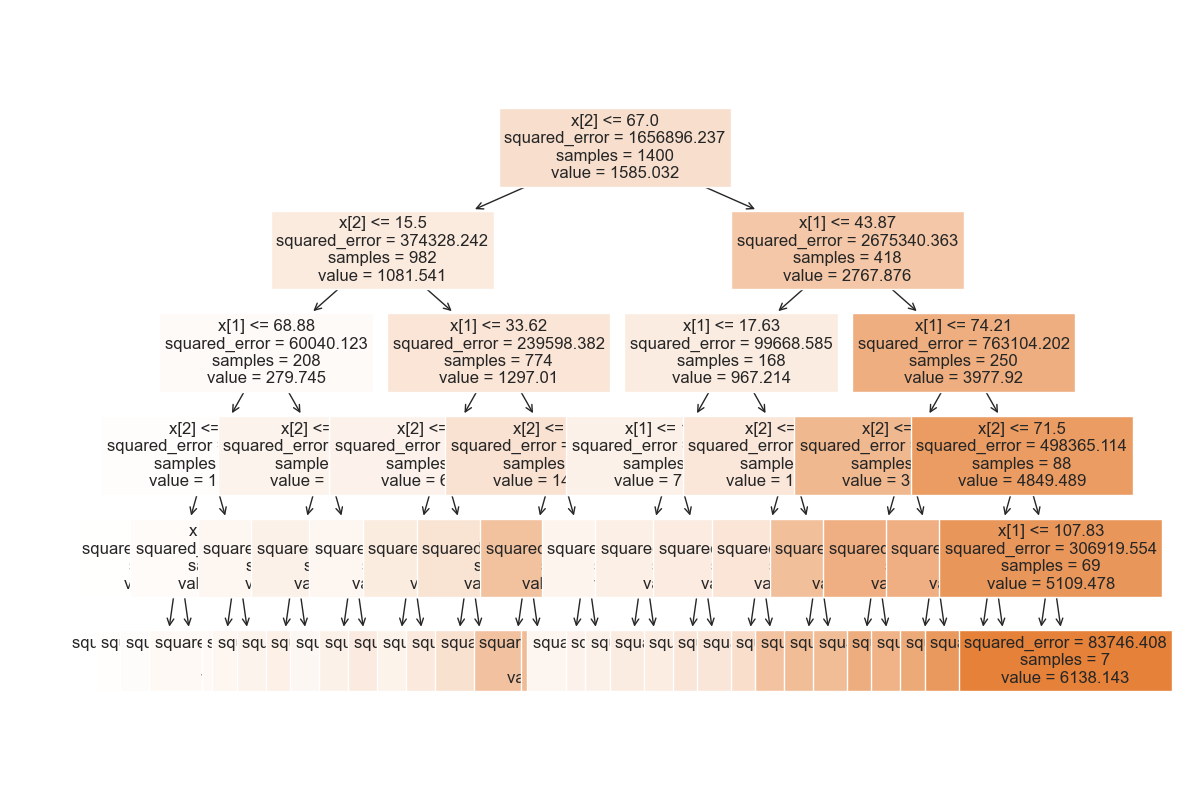
Similarly, when I plotted loyalty\_points against spending\_score, the ouput was very similar in both gender groups. Note the blue regression lines: they are almost identical. In conclusion, because the results across the gender groups are invariably similar, I saw no benefit in including gender as a feature in the regression model, or in any of the models.

1. Exploring the structure using decision trees.

After splitting the data into training and test sets, I fitted a decision tree regression model with the training data and ran the model using the X\_test data to predict the y\_test data. The model performed extremely well, with an R-squared of 0.996. However, the tree had many branches, which hurt its readability. See the plot below.

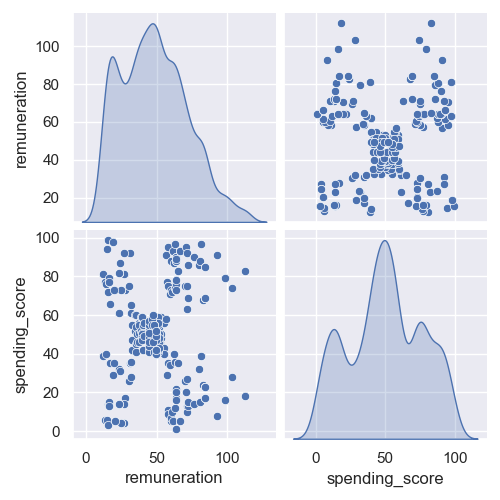


To make a model that was more readable, a simple pruning technique by limiting the tree depth was employed. Using a for loop, I ran a decision tree regression with a maximum depth ranging from one to ten and compared the accuracy results across them all. Although much at this point was subjective, I chose to limit maximum depth to five. The fewer branches included, the greater the mean squared error, but the more readable the model. Five seemed like a good balance. Below is the pruned tree plot.

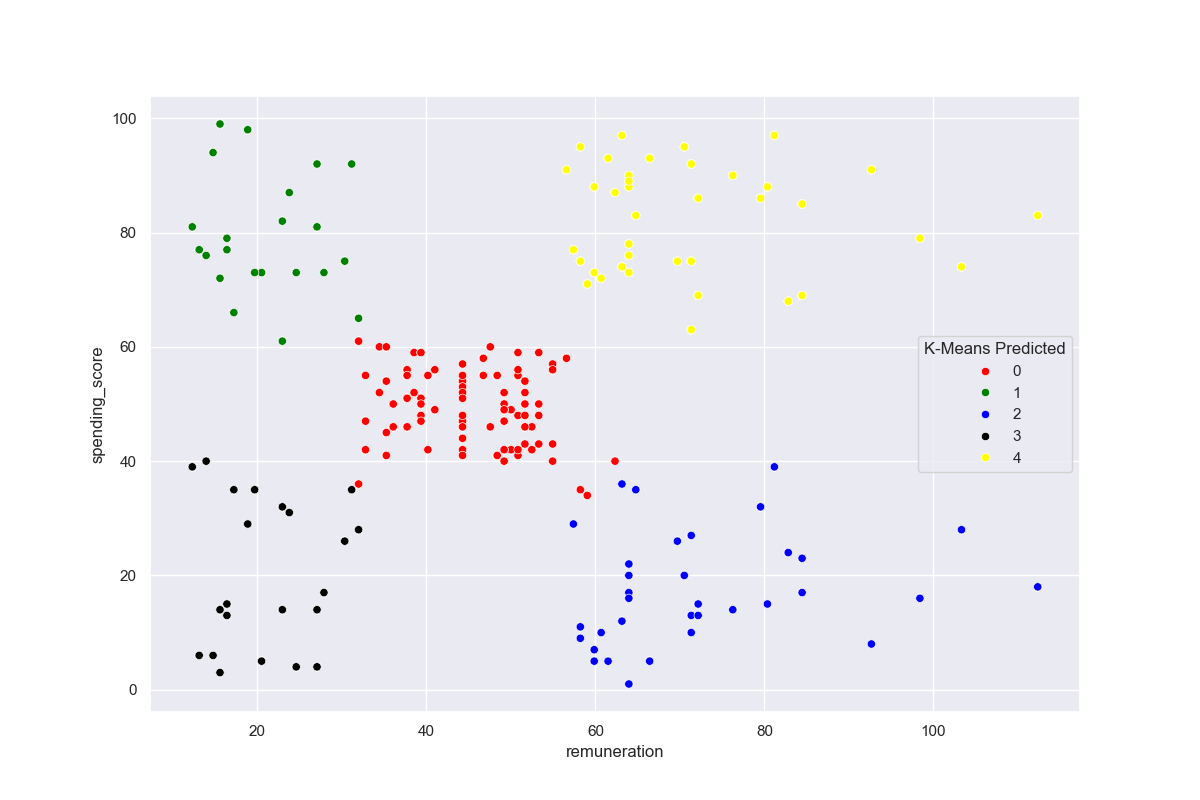


1. Clustering with k-means using Python

By plotting a pairplot of remuneration vs spending\_score, five clusters can be seen to form in the scatterplots.



To determine the number of clusters mathematically, I used the Silhouette and Elbow methods. Both methods pointed to number of clusters, k, equals 5 as the optimal, consistent with my visual interpretation. To consolidate my finding, I evaluated the performance of K-means models for k equals 4 through 7. The outputs reaffirmed my decision to choose five clusters because the data looked most logically segmented. Below is the plot of the final model:



The clusters can be considered categories of customer types and this can inform marketing strategies, as well as production decisions.

1. Natural Language Processing with Python

The goal of this section was to uncover insights into customer sentiment in relation to products of Turtle Games.

I created a subset using the ‘review’, ‘summary’, and ‘product’ columns. I changed the data in the ‘review’ and ‘summary’ columns to lower case and then removed all punctuation. Finally, I dropped duplicate rows in the columns using the drop\_duplicates() function and thereby created two subset data frames, reviews and summaries.

I generated word clouds using the WordCloud package to see which words appear the most in each column. The initial results were not very useful; words like ‘the’ and ‘and’ add no useful information relating to sentiment. Therefore, I tokenised the words using word\_tokenize from the nltk package, created a set of English stop words, and removed all stop words from the subsets. When generating the word clouds of the subsets after, the results were much more insightful. Below is the word cloud for the reviews subset:



Next, using the textblob package, I determined the polarity of each review and summary in the subsets and assigned them sentiment scores, ranging from -1 to 1. Then, I grouped the subsets by product and sorted the products by average sentiment score to determine which were the 20 products with the best reviews and which were the 20 worst products by reviews. The results provide a starting point for decision making relating to these products. With the most positively reviewed products, management might consider increasing production and marketing. As for the poorly reviewed products, management might consider replacing these products with improved versions and use the negative reviews as a rough guide on how improvements could be made.