Exploring The UK Weather Stations

Murotiwamambo Mudziviri

Contents

INTRODUCTION

The following report was generated using R Markdown. It is accompanied by a file named "UK_Weather.zip" that contains all the data that was obtained, generated and used data in this report. It also contains a file named "UK_weather.Rmd" which is the actual R Markdown file used to generate this report. The R Markdown file contains fully annotated code for all the parts.

PART 0 : Preparing the Weather Stations Data

Data from 37 weather stations was imported from the UK Met Office website (Historic Station). The raw data presented several challenges that had to be rectified before commencing any form of analysis. Some of the challenges were:

1. A lot of missing data. Some stations were missing a full column or a significant portion of multiple columns.
2. The starting and ending years of the data was not consistend from station to station. Some stations started recording data in the 1800s. Others were even closed decades ago.
3. There were special characters included in the data to communicate different things.

Steps taken to clean the weather stations data

All the data was imported from the same website and saved in a common folder as "stationname_raw.txt". All comments and special characters were removed to remain with column names and numeric values only. The data marked as "provisional" was also removed because it had not gone through a full network quality control yet. Such data could lead to wrong conclusions if riddled with errors that have not been caught yet.

Other missing values within columns were replaced by "NA", a place holder that is acceptable in R. A value of 0 could have been used, but it affects the results of computations performed on the data. Therefore, it may result in misleading findings.

The values marked as "NA" were ignored during computations such as finding the mean, e.t.c. For example, if a data set with 10 values including a single "NA" is used to find the mean, the result will be the mean calculated from the remaining 9 non "NA" values.

Upon completion of the data cleaning as described above, the resulting file was exported as a "station.csv" document for use in other parts of the analysis. An example of the first few rows of clean data is shown in Table 1 below.

Table 1: The first 5 rows of cleaned Nairn station data.

```
head(Nairn, 5)
```

```
## # A tibble: 5 x 7
##    yyyy    mm tmax_degC tmin_degC af_days rain_mm sun_hours
##   <dbl> <dbl>     <dbl>     <dbl>   <dbl>   <dbl>     <dbl>
## 1  1931     1       5         0.6      11    78.4      43.4
## 2  1931     2       6.7       0.7       7    48.9      63.6
## 3  1931     3       6.2      -1.5      19    37.6     145.
## 4  1931     4      10.4       3.1       3    44.6     110.
## 5  1931     5      13.2       6.1       1    63.7     167.
```

The data from weather stations consists of 7 variables as seen in Table 1. These are year(yyyy), month(mm), mean daily maximum temperature (tmax_degC), mean daily minimum temperature(tmin_degC), days of air frost (af_days), total rainfall(rain_mm) and total sunshine duration (sun_hours).

PART 1 : Clustering The Stations Based On Similar Weather

Aim

The aim of part 1 is to use a clustering algorithm to see if the weather stations can be clustered into groups that have "similar" weather. The k-means clustering algorithm was used in this case. This algorithm was chosen because of its simplicity and potential for scalability.

A summary of the K-mean clustering algorithm

The k-means clustering algorithm is a type of unsupervised learning. That means, it is used to find clusters of similar data points within unlabeled datasets. To generate those clusters, on has to select the value for k. This represents the number of clusters that can be expected from the dataset. K is known as the number of centroids. Once k is selected, it is assigned random values/coordinates within the limits of the data. The eucledian distance between each centroid and each data point is calculated. Each data point is assigned to the closest centroid. The value/cordinates for each centroid are updated by averaging the data points assigened to each, and the whole process is iterated until the values of the centroids no longer change (have stabilized).

k-mean clustering makes 3 assumptions that may affect its performance if they are not met. These are:

1. Each variable's variance has a spherical distribution.
2. Equal variance for all variables.
3. The probability of each data point to end up any one of the k clusters is the same.

While our data may not have the ideal size and stracture that is required to develop a high perfoming clustering model, k-means clustering performed well.

Approach

The majority of clean weather station data ended in 2018. An average of the most recent 5 years for each variable was used in k-means clustering. Upon examining the data, it was observed that most weather stations were missing the total sunshine duration data. As such, that variable was not used in the analysis. Table 2 shows the first 5 rows of the data prepared for use in the k-means clustering algorithm. It contains 5 variables : average maximum temperature (tmax_av_values), average minimum temperature(tmin_av_values), average air frost days(af_days_av_values) and average rainfall(rain_mm_av_values). The station name was included as the row name. This was done to eventually track which stations were grouped together.

Table 2: The first 5 rows of data prepared for the k-means clustering algorithm

```
head(data_for_part1, 5)
```

```
##              tmax_av_values tmin_av_values af_days_av_values
## aberporth          13.16833       7.925000         0.5333333
## armagh             13.82667       6.180000         2.8333333
## ballypatrick       11.90000       6.195000         1.6000000
## bradford           13.52000       6.586667         2.6333333
## braemar            11.21667       3.148333         8.6333333
##              rain_mm_av_values
## aberporth             81.98000
## armagh                69.89000
## ballypatrick         112.09167
## bradford              73.48333
## braemar               79.77000
```

The data displayed on table 2 shows 4 features with different units of measurement. Therefore, they are not comparable in their orginal state. The data was scaled to give equal weight to each variable during clustering.

Table 3: The first 5 rows of scaled data for k-means clustering.

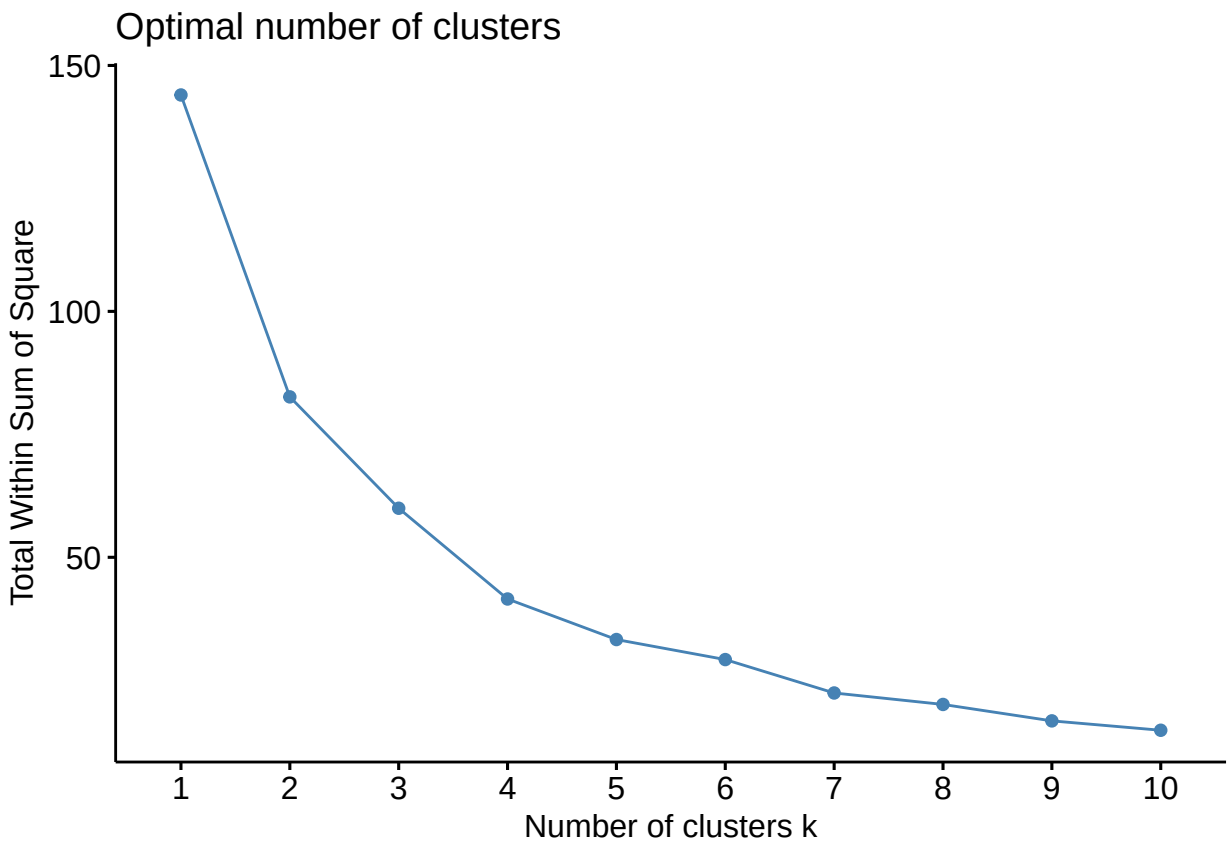
```
data_for_part1_scaled <- scale(data_for_part1) #scaling the data
head(data_for_part1_scaled, 6)
```

```
##              tmax_av_values tmin_av_values af_days_av_values
## aberporth       -0.36382230      0.9521774       -1.24596381
## armagh           0.07603218     -0.4423513        0.10393178
## ballypatrick    -1.21123816     -0.4303639       -0.61992528
## bradford        -0.12886206     -0.1173608       -0.01345045
## braemar         -1.66779598     -2.8651284        3.50801631
## camborne         0.18070641      1.7726451       -1.40247344
##              rain_mm_av_values
## aberporth           0.10477770
## armagh             -0.30097079
## ballypatrick        1.11534539
## bradford           -0.18037612
## braemar             0.03060862
## camborne            0.48350985
```

Selecting the value for k (number of clusters) using the Elbow method

Figure 2: Total Within Sum of Squares (wws) vs the Number of Clusters (k)

```
set.seed(123)
fviz_nbclust(data_for_part1_scaled, kmeans, method = "wss")
```



There are various ways to find the optimum k for k-means clustering. The Elbow Method was deemed

adequate for this analysis. It is implemented by calculating and ploting Total Within Sum of Squares (wws) against the Number of Clusters (k). The plot resembles an arm bend at the elbow. The aim is to find the value of k where the graph bends. In this case, k=4 as seen in fig.2.

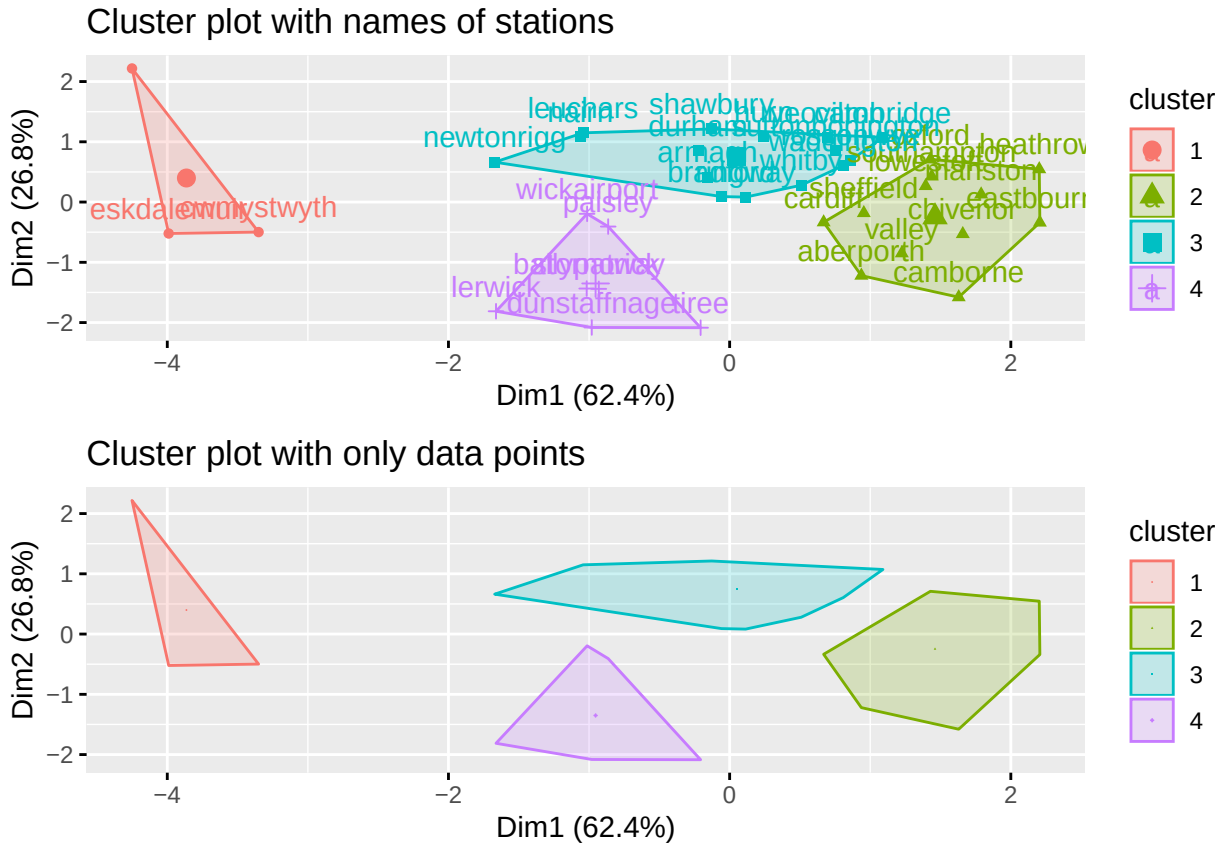
Implementing the k-means clustering algorithm

The k-means algorithm was implemented using a k value of 4 and was iterated until the centroids were no longer changing. The results are presented in Figure 3.

Figure 3: Results of k-means clustering

```
k4<- kmeans(data_for_part1_scaled, centers = 4, nstart = 100)
```

```
grid.arrange(plot1, plot2, nrow = 2)
```



The clusters plotted above were examined to find the stations that were grouped together. The first polot in Figure 3 was useful at determining were each station was assigned.

a. cluster 1 had 5 stations : lerwick, tiree, dunstaffnage, ballypatrick,stornoway

b. cluster 2 had 18 stations : aberpoth, camborne, valley, chivenor, eastbourne, cardiff, sheffield, manston, lowestoft, whitby,southampton, heathrow, waddington, rossonwye, oxford, yeovilton, cambridge, sutton-bonington

c. cluster 3 had 11 stations : paisley, wickariport, newtonrigg, armagh, nairn, leauchars, shawbury, durham, bradford, harn,ringway

d. cluster 4 had 3 stations: braemar, cwmystwyth,eskdalemuir

The stations in each cluster were marked on the map to see if there are any observable patterns produced by the clustering algorithm. Figure 1 below depicts the UK map with the clusters super imposed to it. The colors of the culsters follows the color code in Figure 3.

The positions of the 4 centroids computed by averaging the data from stations assigned to them is presented in table 4.

Table 4: The data for each centroids (k)

```
data_for_part1 %>% mutate(Cluster = k4$cluster) %>% group_by(Cluster) %>%
summarise_all("mean")
```

```
## # A tibble: 4 x 5
##   Cluster tmax_av_values tmin_av_values af_days_av_values rain_mm_av_values
##     <int>          <dbl>          <dbl>             <dbl>             <dbl>
## 1       1           11.6           4.03              6.81             130.
## 2       2           14.8           8.05              1.42              69.3
## 3       3           14.1           6.40              3.23              62.3
## 4       4           11.9           6.35              1.76             109.
```

Conclusion

The k-means clustering of the weather stations based on the similarity of their weather performed fairly well. The cluster plots in Figure 2 show clear clusters that are not overlapping. Futhermore, the location of members of each cluster on the Uk map shows a pattern (Figure 1). Cluster 4 members are predominantely locates on the right bottom corner of the map while cluster 3 members are found at the top of the map. However, figure 1 shows that cluster 3 stations are burried within regions dominated by other clusters. Table 4 shows that the cluster 3 centroid was different from each of the other clusters on at least 2 of the feaures. Thus, it is possible that although these stations are located in regions that a predominantly populated by members of other clusters, they still have their unique set of features shared amoung themselves only.

From the results and visualizations generated above, I believe the weather stations can and were clustered well, based on the similarities of their weather.

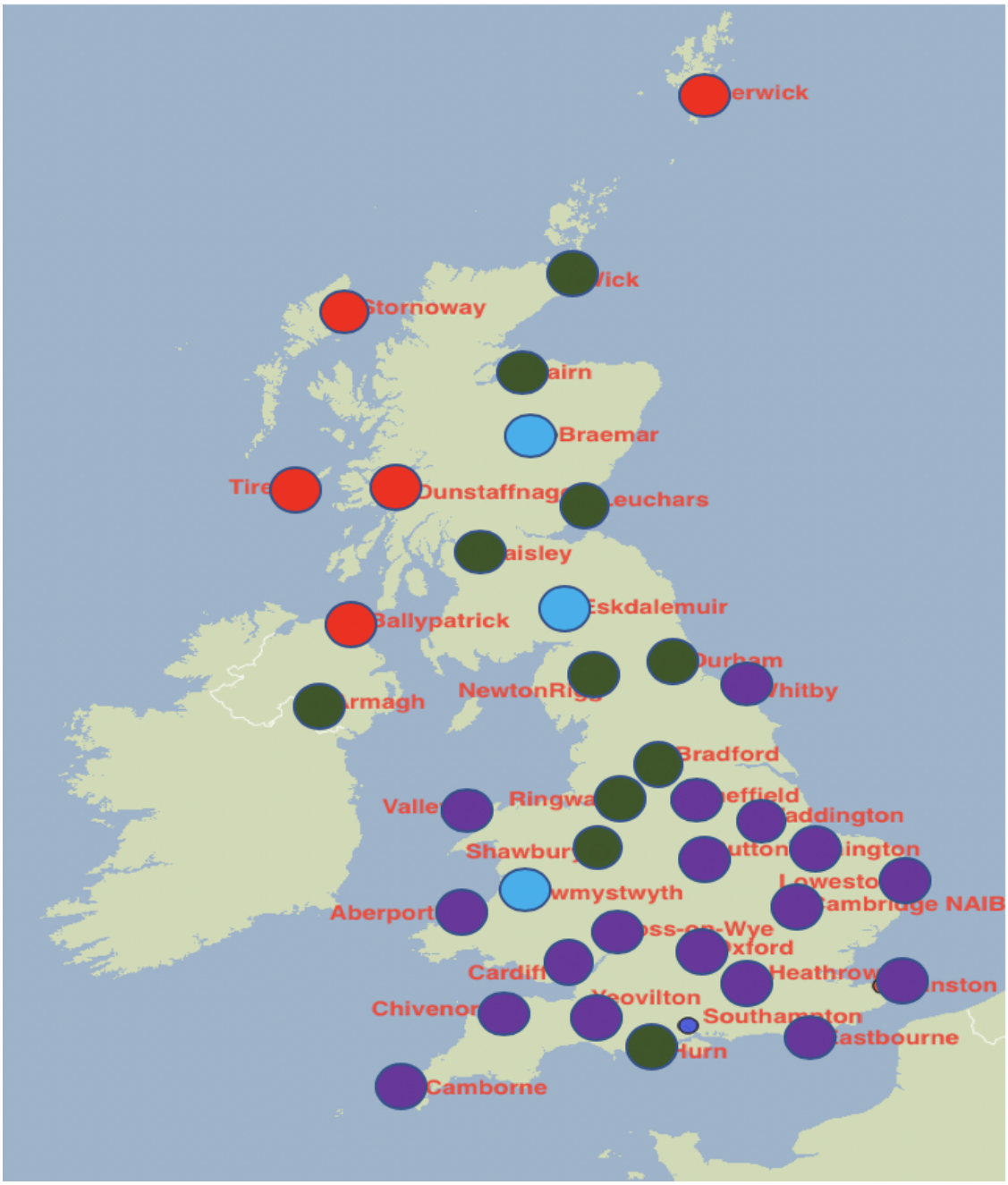


Figure 1: Map of the UK and the clustering results.

PART 2 : Classifying the Stations Based on Weather Information

Aim

The aim of part 2 is to build a k Nearest Neighbour (kNN) classifier that can predict the region where each station belongs based on its weather infomation. There are three regions: north, central, and south. These regions are not standard, they were developed purely for this analysis only.

K Nearest Neighbour (KNN) Summary

The kNN algorithm is a supervised learning algorithm. Thus, it uses labeled data to establish relationships between data points based other features. When given a new set of features that were not seen in training, the algorithm will use what it "learned" to correctly predict the class of the data points represented by those features.

kNN is a relatively simple algorithm. It stores the labeled examples provided during training. When a new set of variables is provided that need to be labeled, kNN will assign this new case to the class where there are most of its closest neighbors (most similar data from the training examples). The number neighbors are consider when predicting the class of a new case is known as k.

Approach

Firstly, the UK was divided into 3 regions using latitude. The most northern and most southern latitudes were provided. Using that information, three ranges of latitude were developed: north, central, and south. Each weather station was then labeled as either north or central or south, based on its latitude's position, relative to the formulated ranges.

It is widely known that latitude is inversely related to temperature. Therefore, I believed that the weather variable in this data set that is most likely to develop a high performing kNN classifier is temperature data. Consequesntly, only the maximum and minimum temperature variables were used in this analysis. Table 5 below shows the first few rows of the data prepared for this analysis.

Table 5: The first 5 rows of the data prepare for the kNN classifer

```
head(data_for_part2, 5)
```

```
##              location tmax_av_values tmin_av_values
## aberporth       south       13.16833       7.925000
## armagh         middle       13.82667       6.180000
## ballypatrick   middle       11.90000       6.195000
## bradford       middle       13.52000       6.586667
## braemar        middle       11.21667       3.148333
```

The features (predictors) used in this analysis were average maximum temperature (tmin_av_values) and average minimum temperature (tmin_av_values). The target variable is the location. Since both predictor variables are recorded using the same units, and possibly measured using the same instruments, the data will not be scaled.

Implementing the KNN classifier

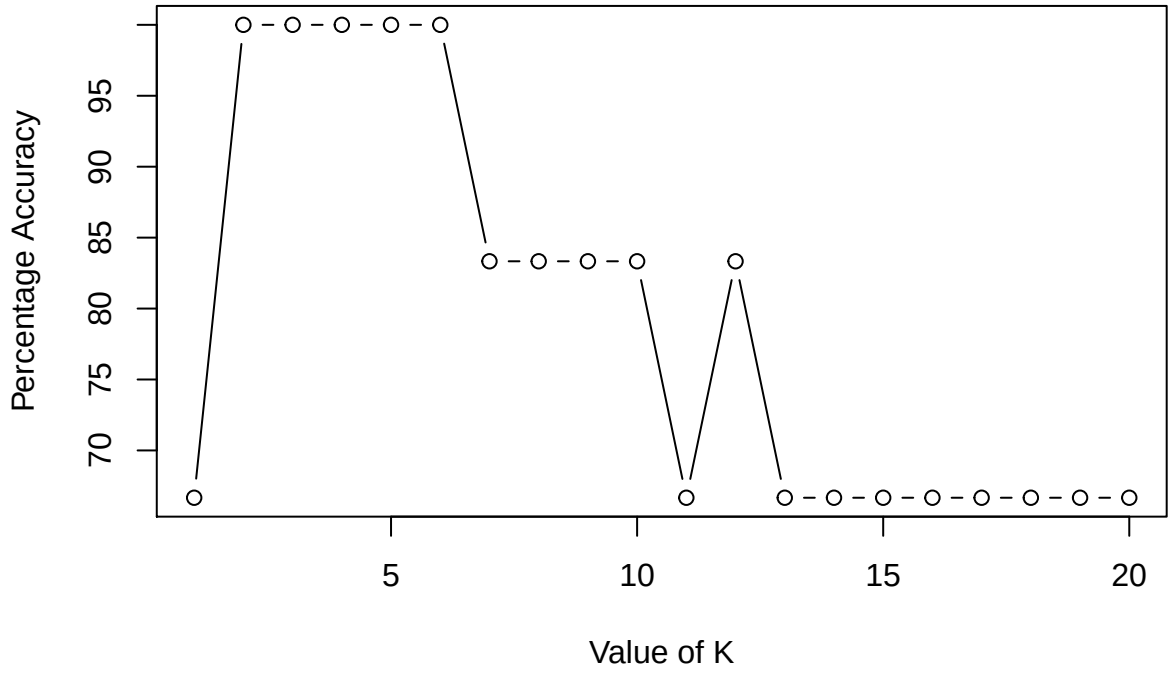
Firstly, the data was sperated between a training set (31 data point) and a test set (6 datapoint). The goal was to train the kNN algorithm using the training set and then evaluate its perfomance using the test set.

To commence the building of a kNN classifier, the value for K has to be defined. There are many ways to find k. The simplest method is to find the square root of the data set size. The optimum k is often close, if not equal to that value.

Using that method, k was found to be equal to 6. This value for k was also supported by plotting the percentage accuracy of a kNN classifier vs the k value. Figure 4 below shows that k=6 achieves an accuracy percentage of up to 100%.

Figure 4: Percentage Accuracy vs The Value of K

```
plot(optimum_k, type="b", xlab="Value of K",ylab="Percentage Accuracy")
```



Training and Evaluting the KNN classifier

The kNN classifier was trained using k = 6. It was then used to predict the classes for 6 new examples. The classifier performed well. It accurately predicted the classes for all of the new examples. It also had a specificity and sensitivity value of 1.

Table 6: A confussion matrix of the predictions made by the kNN classifier.

```
# training the classifier
KNN_6 <- knn(train=data_train, test=data_test, cl=data_train_lables, k=6)

#using the trained classifier to predict the results for the test sample
CrossTable(x = data_test_lables, y = KNN_6, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
```

```
##
## Total Observations in Table:  6
##
##
##                  | KNN_6
## data_test_lables |    middle |     north |     south | Row Total |
## -----------------|-----------|-----------|-----------|-----------|
##           middle |         2 |         0 |         0 |         2 |
##                  |     1.000 |     0.000 |     0.000 |     0.333 |
##                  |     1.000 |     0.000 |     0.000 |           |
##                  |     0.333 |     0.000 |     0.000 |           |
## -----------------|-----------|-----------|-----------|-----------|
##            north |         0 |         1 |         0 |         1 |
##                  |     0.000 |     1.000 |     0.000 |     0.167 |
##                  |     0.000 |     1.000 |     0.000 |           |
##                  |     0.000 |     0.167 |     0.000 |           |
## -----------------|-----------|-----------|-----------|-----------|
##            south |         0 |         0 |         3 |         3 |
##                  |     0.000 |     0.000 |     1.000 |     0.500 |
##                  |     0.000 |     0.000 |     1.000 |           |
##                  |     0.000 |     0.000 |     0.500 |           |
## -----------------|-----------|-----------|-----------|-----------|
##     Column Total |         2 |         1 |         3 |         6 |
##                  |     0.333 |     0.167 |     0.500 |           |
## -----------------|-----------|-----------|-----------|-----------|
##
##
```

The result observed from this classifier was perfect, perhaps too perfect to exist in the real world. Therefore, it is important to note that the data sample is very small.

PART 3 : Does weather affects people's happiness ratings

Aim

This part aims to determine if the weather affects the average ratings of people's happiness.

Approach

In this analysis, data prepared in part 1 - that consist of tmax, tmin, af days, and rainfall - was be combined with the average happiness ratings. The happiness data has information down to the county level. However, it was decided to proceed with the analysis at a regional level becuase of the distribution the weather station. There are 12 regions in the UK. Each weather station was assigned to a single region based on its latitude and longitude. The variables of the stations that were assigned a particular region were averaged to produce a new data set whose first few rows are shown in Table 7.

Table 7: The data prepared for part 3 analysis

```
data_for_part3
```

```
##                           region_happiness     tmax     tmin  af_days
## NORTH_EAST                            7.34 12.24889 5.538889 3.616667
## NORTH_WEST                            7.39 13.75583 6.479167 2.425000
## YORKSHIRE_AND_THE_HUMBER              7.41 13.52000 6.586667 2.633333
## EAST_MIDLANDS                         7.51 14.01167 7.350000 1.858333
## WEST_MIDLANDS                         7.43 14.57167 6.911667 2.794444
## EAST                                  7.51 13.42917 6.552540 2.953175
## LONDON                                7.38 15.66278 7.580000 2.200000
## SOUTH_EAST                            7.54 14.94000 8.116667 1.366667
## SOUTH_WEST                            7.50 15.11339 7.907674 2.003625
## WALES                                 7.44 15.66278 7.580000 2.200000
## SCOTLAND                              7.45 11.90687 5.753750 3.106250
## NORTHERN_IRELAND                      7.75 13.21167 5.640000 3.966667
##                               rain
## NORTH_EAST                125.71000
## NORTH_WEST                 62.35750
## YORKSHIRE_AND_THE_HUMBER   73.48333
## EAST_MIDLANDS              67.08667
## WEST_MIDLANDS              53.51889
## EAST                       92.08944
## LONDON                     56.12833
## SOUTH_EAST                 52.42000
## SOUTH_WEST                 71.91521
## WALES                      56.12833
## SCOTLAND                   92.80958
## NORTHERN_IRELAND           70.30750
```

Exporing the the data

When searching for a correlation between variables in a data set, it is important to ensure that there are no outliers. There were some outliers identified in the happiness ratings and the rain fall variables. Outliers can lead to misleading results, hence, it is essential to remove them early on.

Figure 5 : The Initial Box plots of the variables with outliers

```r
# Box plots to see if there are any outliers in the data
# They are all plotted seperated because they are on different sclaes.
par(mfrow =c(2,5))
boxplot(data_for_part3$region_happiness, main = "happiness")
boxplot(data_for_part3$tmax, main = "tmax")
boxplot(data_for_part3$tmin, main ="tmin")
boxplot(data_for_part3$af_days, main = "af_days")
boxplot(data_for_part3$rain, main = "rain")
```

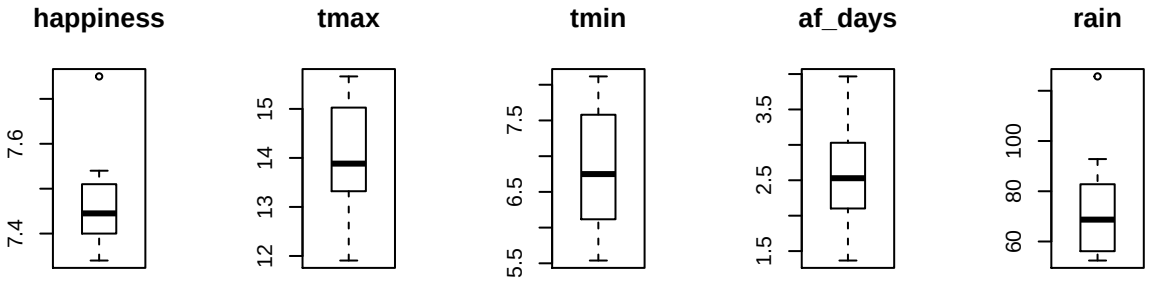
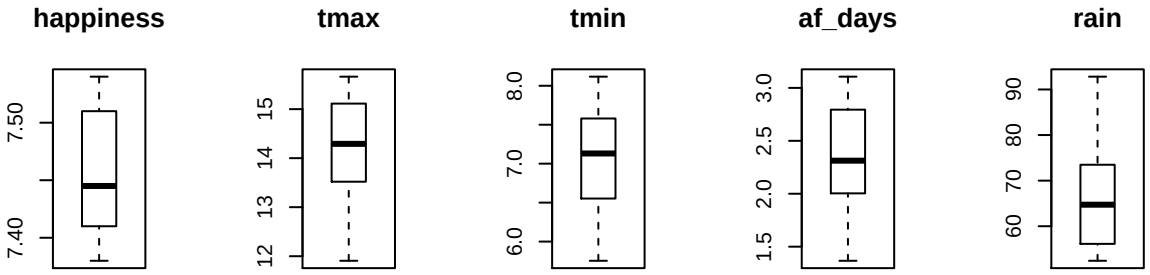


Figure 6 : The Final Box plots of the variables without outliers

```r
# Box plots to see if there are any more outliers in the data
par(mfrow =c(2,5))
boxplot(data_for_part3$region_happiness, main = "happiness")# the first plot
boxplot(data_for_part3$tmax, main = "tmax")
boxplot(data_for_part3$tmin, main ="tmin")
boxplot(data_for_part3$af_days, main = "af_days")
boxplot(data_for_part3$rain, main = "rain")# the plast plot
```

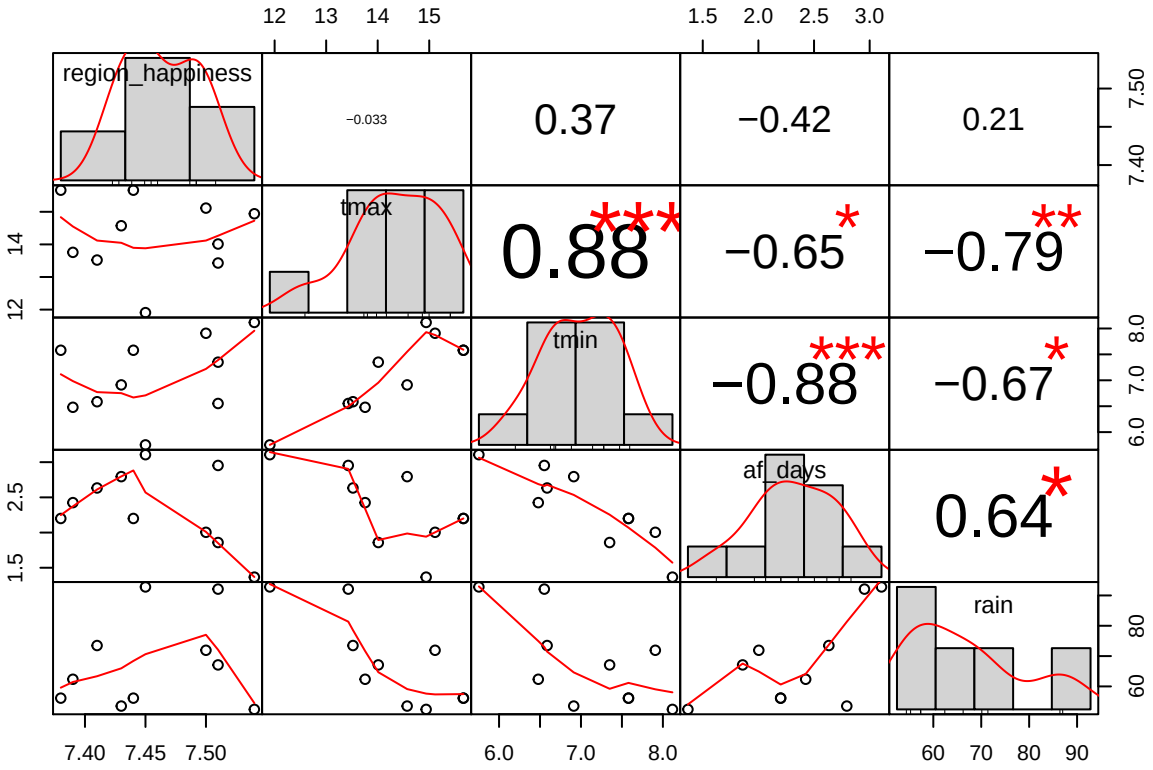


A correlation Matrix

After cleaning the data, it is important to visualize the relationships between variables in the data set. A pairwise plot that also included a correlation matrix was select for that purpose. A correlation matrix is often used to explore the strength of a relationship between variables before a major analysis is done. As such, this step was done to inform decisions made during the implementation of a multiple regression model that was build to further investigate the objective of part 3: to determine if the weather affects the average ratings of people's happiness.

Figure 7 : A pair wise plot and correlation matrix

```r
# Exploring the pair wise correlations between the data variables
chart.Correlation(data_for_part3)
```



The plots in figure 6 show a range of relationships between various weather variables and the happiness rankings. It seems that people's happiness ratings have a weak inverse relationship with maximum temperature and the number of air frost days, their correlation coefficiencies are -0.033 and -0.42, respectively. Tha happiness ratings also have a weak positive relationship with the minimum temperature and total rainfall. Their respective correlation coefficiencies are 0.37 and 0.21.

Apart from the correlation values, Figure 6 also depicts pairwise plots between the happiness ratings data and all the other variables. It can be observed that af_days and rain variables have a complex - if any - relationship with the happiness ratings. It takes a significant amount of time to establish the nature of that relation, and the data is too small to rigorously follow such pursuits. As such, af_day and rain were not condered in the regression model, because their relationship with the target variabe is obviously non-linear.

Multiple regression

Multiple regression was used in this case to determine the strength of the relationship between multiple predictor variables and a target variable. In this case, the two temperature data are the predictor variables. Happiness ratings are the target variable. It is assumed that all variables are independent.

Table 8: The multilpe regression results.

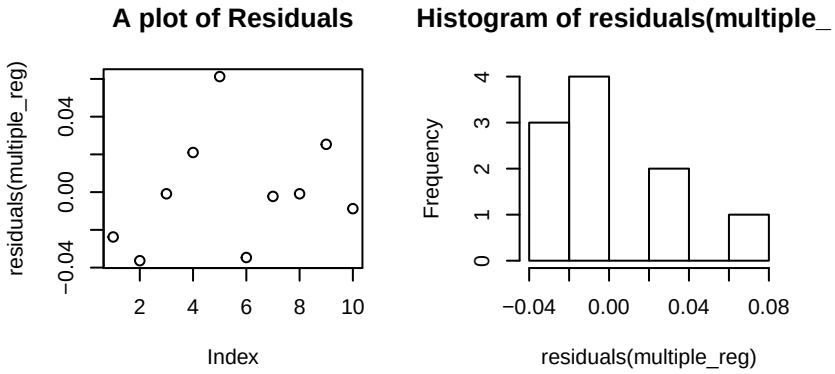
```
summary(multiple_reg)
```

```
##
## Call:
## lm(formula = region_happiness ~ tmax + tmin, data = data_for_part3)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.036339 -0.020020 -0.001583  0.015534  0.061248
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.60339    0.14102  53.919 1.98e-10 ***
## tmax        -0.07696    0.02044  -3.765  0.00703 **
## tmin         0.13413    0.03207   4.182  0.00412 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03382 on 7 degrees of freedom
## Multiple R-squared:  0.7145, Adjusted R-squared:  0.6329
## F-statistic: 8.759 on 2 and 7 DF,  p-value: 0.01243
```

Figure 8: Multiple regreesion residuals plots

```
# Inspecting the residuals to see if they are randomly distributed. This usually is a good indication t
par(mfrow =c(2,3))
plot(residuals(multiple_reg), main = "A plot of Residuals") # a plot to of the residuals
hist(residuals(multiple_reg)) # a histogram of the residuals
```



The adjusted R-squared value of the multiple regression depicted by Table 8 is 0.6329. Therefore, the multiple regression model/line is fairly a good fit, assuming that a value above 5 is acceptable to some degree. Table 8 also reveals a statistically significant relationship between the weather variables considered and the happiness ratings. All their p-values were below the significance thresh-hold of 0.05 by a 10 fold order of magnitude. However, the residuals are not completely randomly distributed as seen in Figure 8. Thus, the regression model may not be as good.

Maximum temperature appears to be inversly related to the happiness ratings with a coefficient of - 0.07547. Minimum temperature seems to be positively related to the happiness ratings with a correlation coefficient of 0.13020. The overall relationship described by the model can be summarised as: happiness ratings = 7.61174 - 0.07547 * tmax + 0.13020 * tmin

It is important to note that the data set used in this anaylsis was very small. Therefore, these findings cannot be generalized.

Conclusion

It seems that there is a correlation between temperature (both maximum and minimum) and happiness ratings in the data set used for this analysis. However, correlation does not imply causation. It is unclear if people's happiness is in any way caused by the weather. The observed results may also be attributed to chance. More data for analysis is needed to reliably determine the nature of the relationship between temperature and the happiness ratings.

PART 4 : Does Money Bring Happiness?

Aim

In this part, I decided to pursue option (4a) and explore the happiness data further. There is a common statement that says "money does not bring happiness." All the data that was used in this study was obtained from a publicly available source: the UK Office for Natinal Statistics.

The data on GDHI per head was used as a measure of the amount of disposable income that people have on averge. GDHI per head is defined as the "amount of money that all of the individuals in the household sector have available for spending or saving after they have paid direct and indirect taxes and received any direct benefits." Although other income data could have been suitable for this analysis, GDHI per head is a better reflection of an indivisual's financial standing becuase it takes into account the prices of commodities within the area they live (Gross).

Approach

To examine the relationship between happiness ratings and GDHI per head, I sought for data that at the county level so that there will be more data points to consider in the analysis. The two data sets that were used, were merged by county name. Only counties whose names appeared the same in both data sets were used. This was done to avoid misundertanding the labeling and contaminating the results by adding wrongly labeled data. Data from 53 data counties was collect this way.

Table 9: The first 5 rows of the data prepared for part 4.

```
head (gdhi_happiness_final)
```

```
##                          mean_gdhi average_rating
## Antrim and Newtownabbey  15637.00           7.99
## Barnet                   23820.00           7.46
## Belfast                  14704.67           7.35
## Birmingham               13097.33           7.40
## Bradford                 13270.33           7.47
## Brent                    20721.00           7.36
```

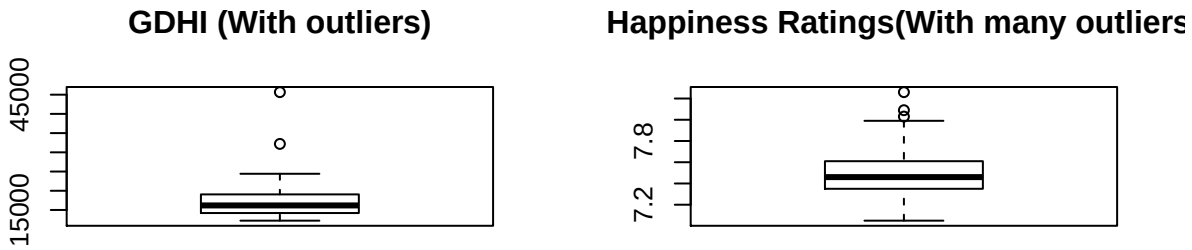
Data Expoloration

It is important to check for, and remove outliers before proceeding to the main analysis. Outliers can negatively affect the quality of conclusions deduced from a particular data set. They can exaggerate or undermine relationships and distributions.

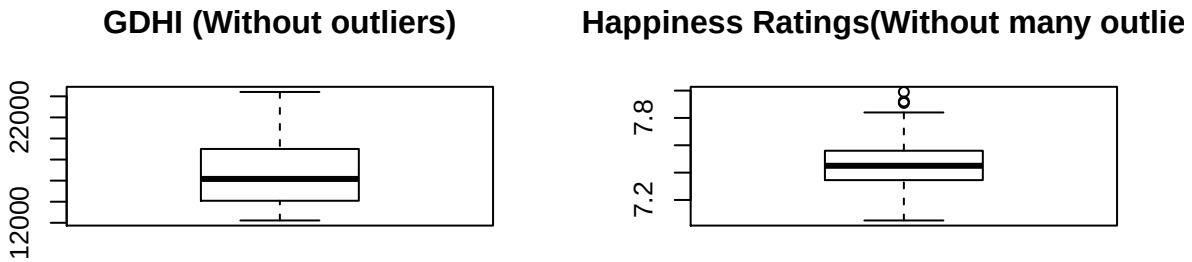
The combined data was plotted on a box plot to identify and eliminate outliers.

Figure 9 : The boxplots shows data with and without outliers

```
par(mfrow =c(2,2))
#searching for outliers
outliers_gdhi <- boxplot(df$mean_gdhi, main = "GDHI (With outliers)")$out
outliers_ratings <- boxplot(df$average_rating, main = "Happiness Ratings(With many outliers)")$out
```



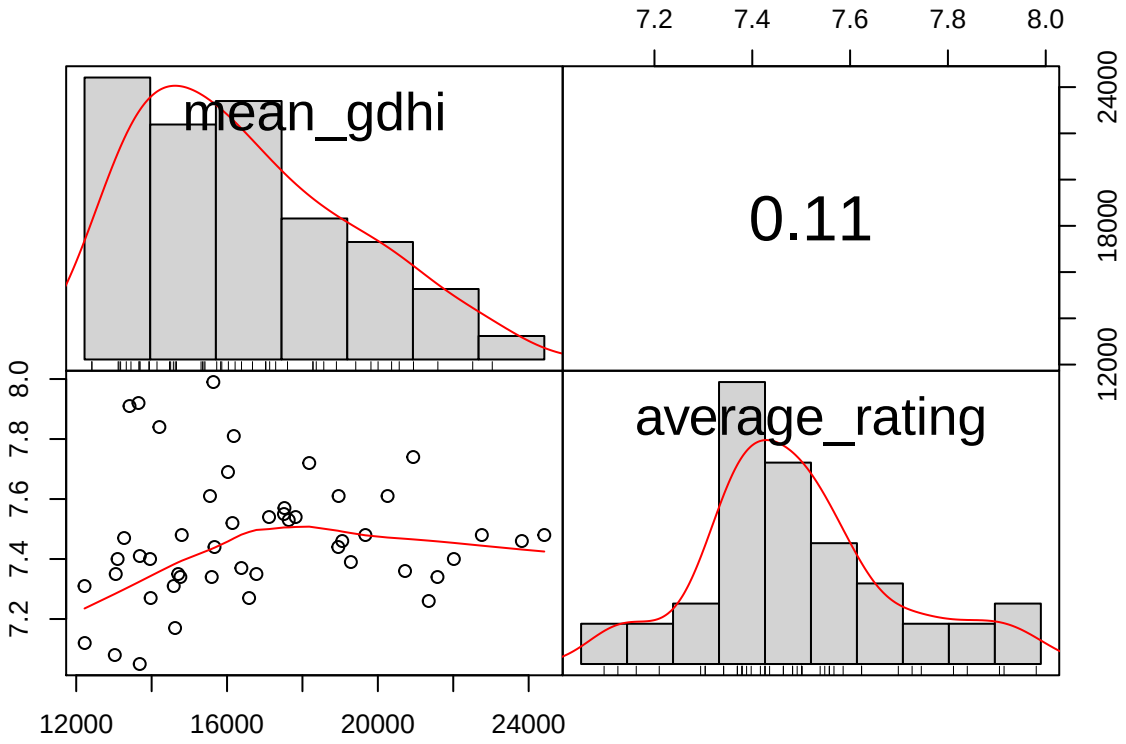

```
par(mfrow =c(2,2))
#Plotting the new data that does not have outliers
boxplot(df$mean_gdhi, main = "GDHI (Without outliers)")
boxplot(df$average_rating, main = "Happiness Ratings(Without many outliers)")
```



The data without outliers was plotted on a correlation matrix to see the general trends and the correlation coefficiencies between the variables (Figure 9). A correlation coefficient of 0.11 was observed. Therefore, there is a weak but positive correlation between the GDHI per head and the happiness ratings in the counties that were considered in this analysis.

Figure 10 : A Correlation for Mean GDHI per Head vs Average Happiness Ratings

```
chart.Correlation(df)
```



However, the plot of the 2 variables depicted in Figure 10 on the left bottom corner reveals an interesting pattern. It seems that happiness ratings increase as GDHI per head increases until the GDHI per head reaches approximately £17 500. Thereafter, the happiness ratings either remain constant or they start to decline.

The data sample considered in this case is not adequate to make generalizations. Nonetheless, for the populations whose data was used in this analysis, it can be argued that money does bring happiness, but only until a certain point. I believe that this finding makes sense. People need enough disposable income to save for partaking in joyful social activities, maintain a good quality of life, and not worry about financial struggles. Once they have achieved to meet most of these needs and wants, it becomes difficult for more income to increase their happiness as it would someone from a more humble economic background.

Citations

Gross Disposable Household Income (GDHI) per head. Retrieved from: https://www.surreyi.gov.uk/dataset/vq187/gross-disposable-household-income-gdhi-per-head

Grolemund, Garrett, and Hadley Wickham. "R For Data Science." R For Data Science, r4ds.had.co.nz/.

"Historic Station Data." Met Office, www.metoffice.gov.uk/research/climate/maps-and-data/historic-station-data.