Assessed Practical II: Brexit

Sungyeon Lim (201764876)

Data

The dataset consists of various factors that could have influenced the voting pattern during Brexit. These factors include the proportion of individuals belonging to the ABC1 social classes, the proportion of residents who were born outside the UK, the median income of all residents, the median age of residents, and the proportion of residents with any university-level education. The dataset also contains a variable that indicates whether more than 50% of people in each electoral ward voted to leave or not, which is the output variable in this case. A total of 344 electoral wards are included in the dataset.

```
brexit <- read.csv("brexit.csv", header = TRUE)
head(brexit)
```

```
##        abc1   notBornUK medianIncome medianAge withHigherEd voteBrexit
## 1 0.1336406 0.012605042    0.2525773 0.5000000   0.08552632       TRUE
## 2 0.1290323 0.113445378    0.1082474 0.2727273   0.11184210       TRUE
## 3 0.1612903 0.004201681    0.1288660 0.6363636   0.11842105       TRUE
## 4 0.3225806 0.046218487    0.2268041 0.4545455   0.21710526       TRUE
## 5 0.3456221 0.058823529    0.2010309 0.5454545   0.24342105       TRUE
## 6 0.2211982 0.012605042    0.2319588 0.4545455   0.09210526       TRUE
```

```
dim(brexit)
```

```
## [1] 344   6
```

Task 1

K-means algorithm is an unsupervised learning algorithm that separates data points into several clusters. To compute the k-means algorithm, a suitable number of clusters needs to be determined. There are two methods commonly used to determine the optimal number of clusters, k: the elbow method and the silhouette analysis.

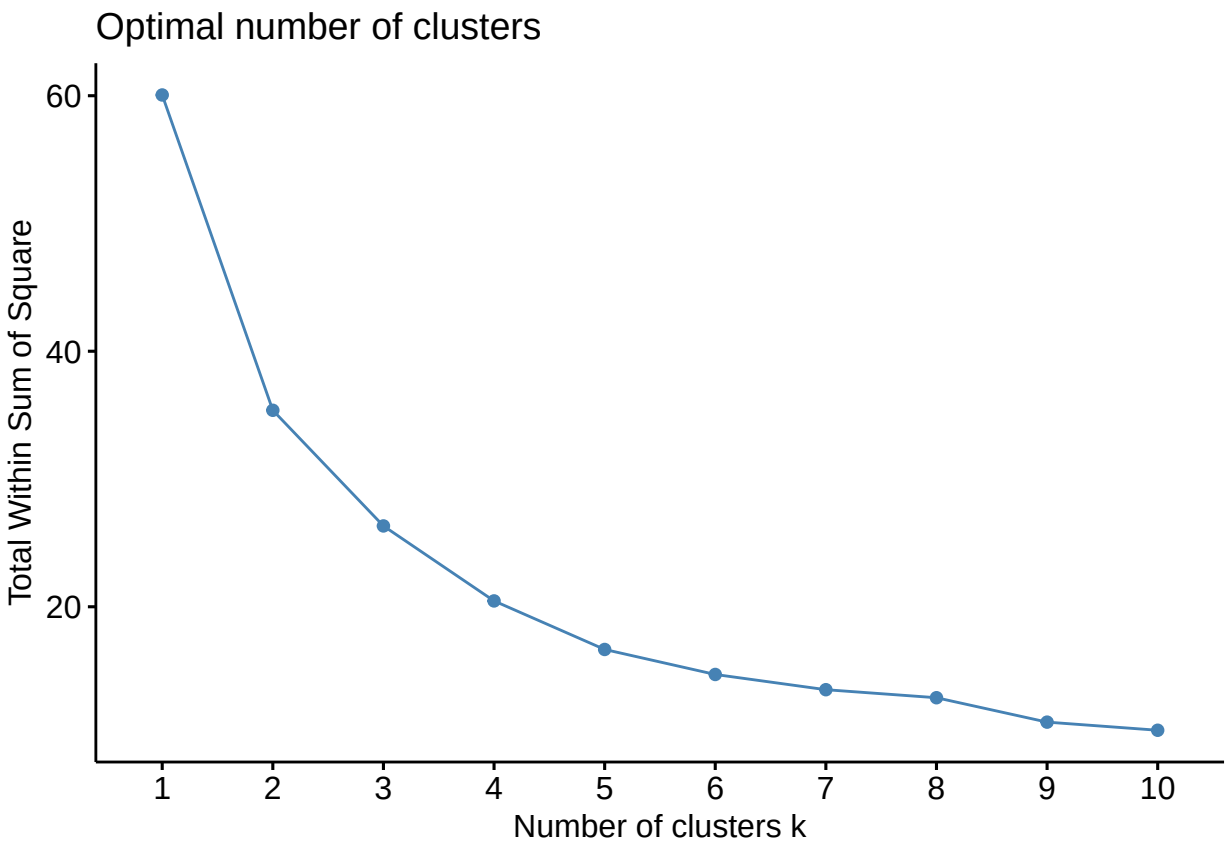
The elbow method involves plotting the total within-cluster sum of squares (WSS) values for each k from 1 to 10 and identifying the point where the graph flattens. This point indicates the optimal k for the model. Here, the WSS represents how well the data points are clustered. The graph below shows that it starts to flatten around k=2 or 3. To make a more confident decision, a further analysis can be performed using the silhouette analysis.

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.3.3
```

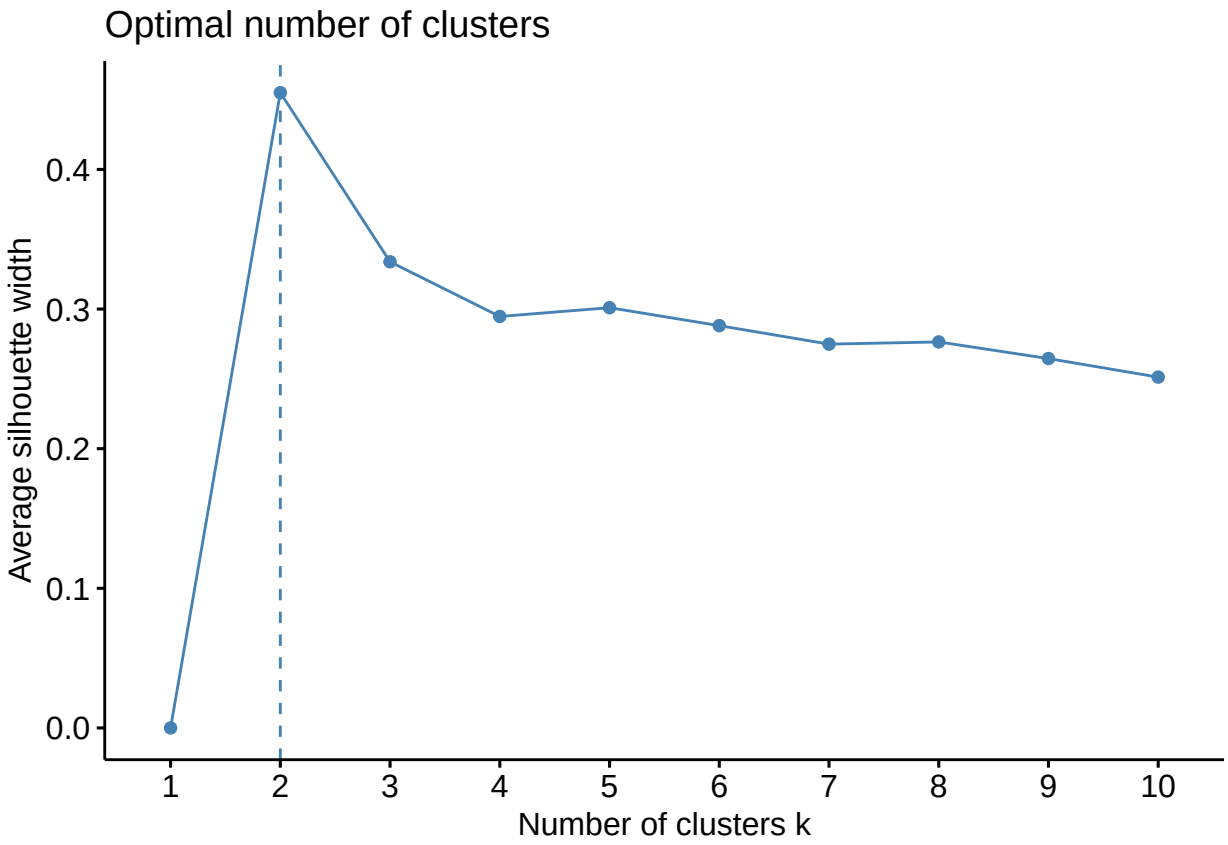
```
## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.3.3

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
fviz_nbclust(brexit[, 1:5], kmeans, method="wss")
```



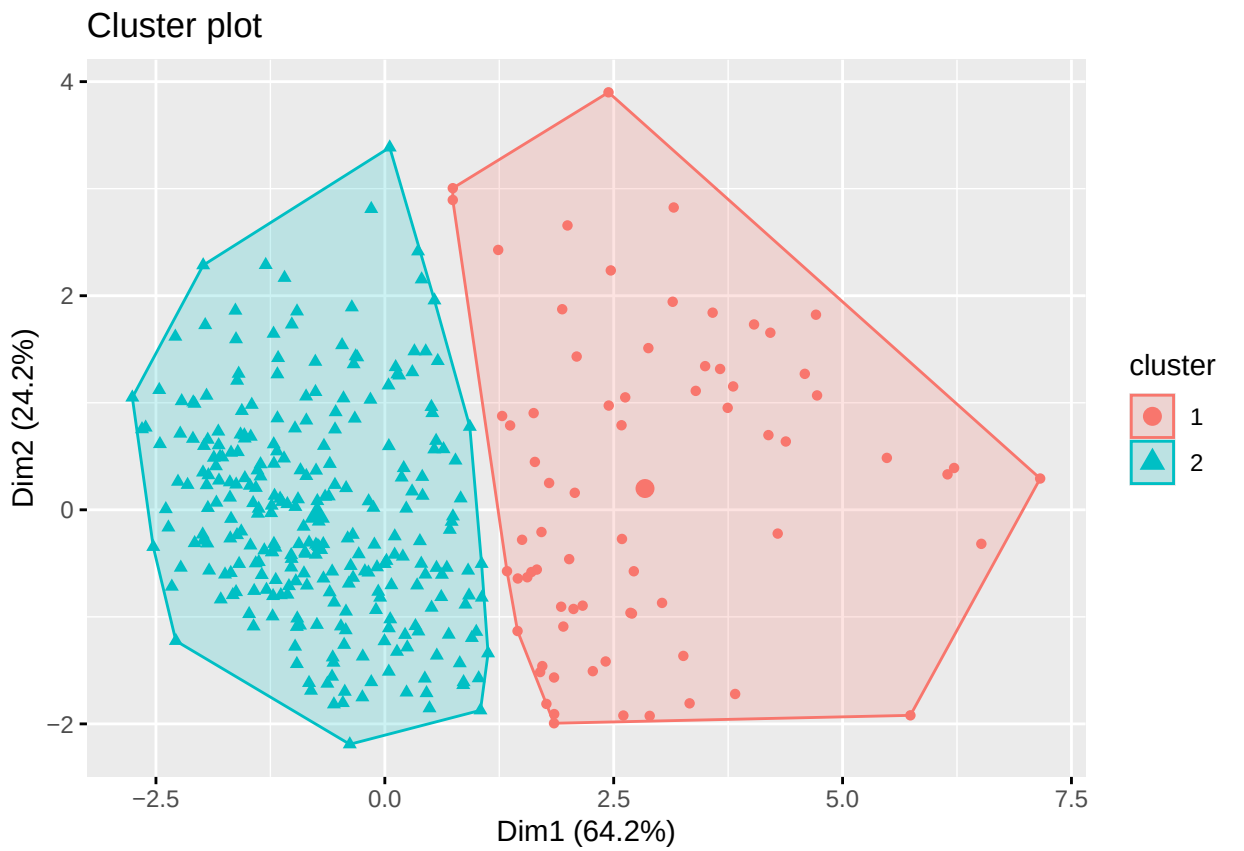
The silhouette analysis is another method to determine the optimal number of clusters. The silhouette score measures how similar a data point is within the cluster compared to other clusters. A graph is plotted to compare the silhouette scores for each k from 1 to 10 to find the point where it maximises. The highest point on the graph indicates the optimal number of clusters, which is 2 in this case. Therefore, it can be concluded that 2 is the ideal number of clusters for this dataset.

```
fviz_nbclust(brexit[, 1:5], kmeans, method="silhouette")
```



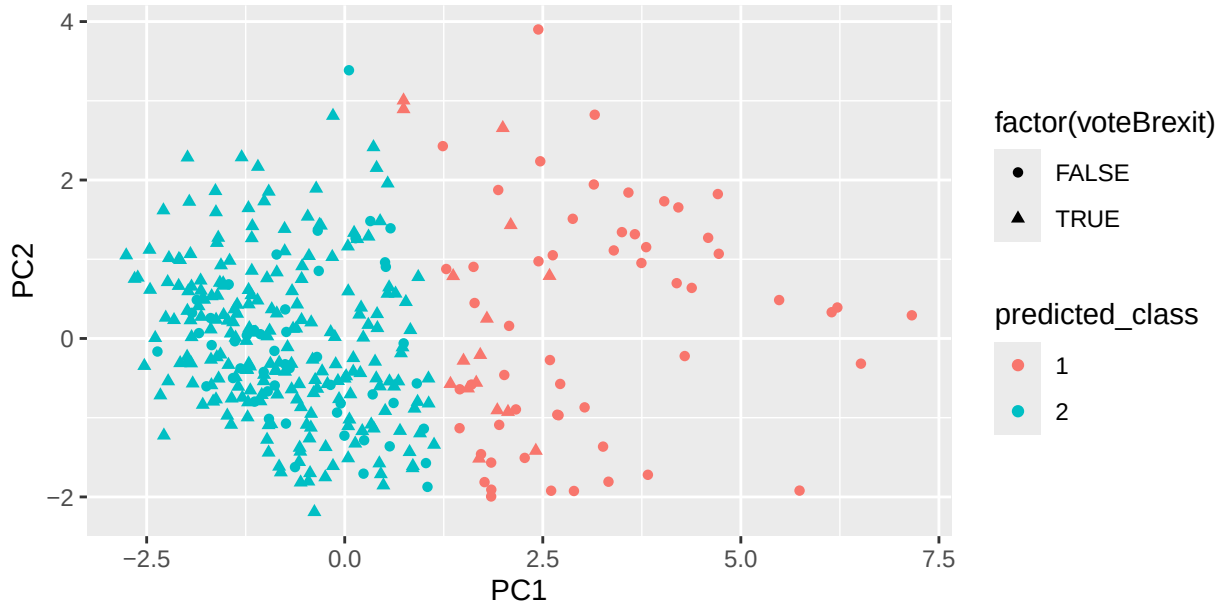
Then, the given dataset is partitioned into two clusters using the k-means clustering algorithm. The resulting clusters are visualised in a 2-dimensional graph.

```
final <- kmeans(brexit[, 1:5], 2, nstart=20)
fviz_cluster(final, data=brexit[, 1:5], geom="point")
```



In order to determine if the model is predictive, a graph is visualised that displays the real labels as two different shapes of data points and the predicted labels as two different colours. By analysing the graph, it is observed that the data points in red are predominantly circular shapes, while the data points in blue are mostly triangular shapes. Based on this observation, it can be concluded that the model has effectively clustered the data points into two distinct clusters.

```
pca_result <- prcomp(brexit[, 1:5], scale.=TRUE)
pca_values <- as.data.frame(pca_result$x)
kmeans_res <- cbind(brexit, predicted_class=as.factor(final$cluster), pca_values)

ggplot(kmeans_res, aes(x=PC1, y=PC2, shape=factor(voteBrexit), color=predicted_class))+
  geom_point() +
  coord_fixed() +
  scale_shape(solid=TRUE)
```



Task 2

A logistic regression model is defined with all inputs and the output, voteBrexit, and trained using the given dataset.

```
model1 <- glm(voteBrexit ~ ., data=brexit, family=binomial)
summary(model1)
```

```
##
## Call:
## glm(formula = voteBrexit ~ ., family = binomial, data = brexit)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.1386     0.8477  -0.164 0.870122
## abc1          17.5780     2.9114   6.038 1.56e-09 ***
## notBornUK      5.6861     1.8033   3.153 0.001615 **
## medianIncome  -6.3857     1.9217  -3.323 0.000891 ***
## medianAge      5.9209     1.4066   4.209 2.56e-05 ***
## withHigherEd -26.7443     3.5762  -7.478 7.52e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 426.52  on 343  degrees of freedom
## Residual deviance: 247.39  on 338  degrees of freedom
## AIC: 259.39
##
## Number of Fisher Scoring iterations: 6
```

Based on the coefficient estimates, the model can be described by the following linear equation.

```
summary(model1)$coef
```

```
##                  Estimate Std. Error    z value     Pr(>|z|)
## (Intercept)   -0.1385963  0.8476664 -0.1635033 8.701222e-01
## abc1          17.5779980  2.9114177  6.0376077 1.564157e-09
## notBornUK      5.6861383  1.8033339  3.1531256 1.615323e-03
## medianIncome  -6.3857396  1.9217008 -3.3229625 8.906690e-04
## medianAge      5.9208767  1.4065616  4.2094684 2.559722e-05
## withHigherEd -26.7442592  3.5761868 -7.4784291 7.521625e-14
```

$$log(\frac{P}{1-P}) = -0.139 + 17.578X_{abc1} + 5.686X_{notBornUK} - 6.386X_{medianIncome} + 5.921X_{medianAge} - 26.744X_{withHigherEd} + \epsilon$$

To determine which input variables are relevant for explaining the output, the magnitude and direction of coefficients for each input variable can be considered. However, before that, confidence intervals need to be examined to check whether the coefficients are statistically significant. In logistic regression, as the standard errors are estimated, the confidence intervals are also approximated, leading to lower precision calculations. Therefore, instead of defining the interval based on the number of degrees of freedom, the appropriate value from the normal distribution can be used. In this case, only the confidence interval for the intercept includes 0, which means that all the coefficients of the input variables are statistically significant.

```
coeff1 <- summary(model1)$coefficients[, 1:2]
zc <- qnorm(0.975)

b0_min1 <- coeff1[1, 1]-zc*coeff1[1, 2]
b0_max1 <- coeff1[1, 1]+zc*coeff1[1, 2]

b1_min1 <- coeff1[2, 1]-zc*coeff1[2, 2]
b1_max1 <- coeff1[2, 1]+zc*coeff1[2, 2]

b2_min1 <- coeff1[3, 1]-zc*coeff1[3, 2]
b2_max1 <- coeff1[3, 1]+zc*coeff1[3, 2]

b3_min1 <- coeff1[4, 1]-zc*coeff1[4, 2]
b3_max1 <- coeff1[4, 1]+zc*coeff1[4, 2]

b4_min1 <- coeff1[5, 1]-zc*coeff1[5, 2]
b4_max1 <- coeff1[5, 1]+zc*coeff1[5, 2]

cat("Confidence interval for b0: ", c(b0_min1, b0_max1), "\n")
```

```
## Confidence interval for b0:  -1.799992 1.522799
```

```r
cat("Confidence interval for b1: ", c(b1_min1, b1_max1), "\n")
```

```
## Confidence interval for b1:  11.87172 23.28427
```

```r
cat("Confidence interval for b2: ", c(b2_min1, b2_max1), "\n")
```

```
## Confidence interval for b2:  2.151669 9.220608
```

```r
cat("Confidence interval for b3: ", c(b3_min1, b3_max1), "\n")
```

```
## Confidence interval for b3:  -10.1522 -2.619275
```

```r
cat("Confidence interval for b4: ", c(b4_min1, b4_max1))
```

```
## Confidence interval for b4:  3.164067 8.677687
```

After determining statistical significance, the magnitude and direction of the coefficients can be considered. In logistic regression, the magnitude of coefficients indicates the strength of the association between each input variable and the log odds, the probability of the event occurring, which in this case is whether the electoral ward votes to leave for Brexit. The larger the magnitude of the coefficients, the stronger the effects of the variables are. Furthermore, the direction of coefficients refers to whether the effect of the input variable on the probability of the output is positive or negative. A positive coefficient indicates that an increase in the value of that variable leads to an increase in the probability of the output, meaning that the event is likely to occur. In this case, a positive coefficient means that as the value of the variable increases, the electoral ward is more likely to vote to leave. On the other hand, a negative coefficient indicates that an increase in the value of that variable leads to a decrease in the probability of the output, indicating that the event is less likely to happen. In this case, negative coefficients indicate that as the value of the variable decreases, the electoral ward is more likely to vote to remain.

Here are the absolute values of the coefficients listed in decreasing order: withHigherEd, abc1, medianIncome, medianAge, and notBornUK. This implies a stronger effect on the output by the variables in the order they are listed.

```r
sort(abs(coef(model1)[-1]), decreasing=TRUE)
```

```
## withHigherEd         abc1 medianIncome    medianAge    notBornUK
##    26.744259    17.577998     6.385740     5.920877     5.686138
```

The coefficients are listed below. The coefficients of abc1, notBornUK, and medianAge are positive, while those of medianIncome and withHigherEd are negative. This implies that an increase in values for the variables abc1, notBornUK, and medianAge leads to an increase in log odds, while an increase in values of medianIncome and withHigherEd leads to a decrease in log odds. This also indicates the correlations between each variable and the log odds of the output. In this case, there are positive correlations between variables abc1, not BornUK, and medianAge and log odds, which implies that the electoral wards are more likely to vote to leave. On the other hand, there are negative correlations between variables medianIncome and withHigherEd and log odds, which indicates that the electoral wards are more likely to vote to remain.

```r
coef(model1)[-1]
```

```
##         abc1    notBornUK medianIncome    medianAge withHigherEd
##    17.577998     5.686138    -6.385740     5.920877   -26.744259
```

The Guardian conducted an analysis of relationships between variables and the output. Scatter graphs for each variable and the log odds of the output are examined to identify relationships. The analysis finds that as the medianAge values increase, the log odds decrease, suggesting that electoral wards are more likely to vote to leave. On the other hand, the graphs reveal that as variables including abc1, notBornUK, medianIncome, and wighHigherEd increased, the log odds increased as well, indicating that electoral wards are more likely to vote to remain. The graphs also suggest that the variables have a stronger effect on the output in the following order: withHigherEd, medianIncome, abc1, medianAge, and notBornUK. This can be seen by clear correlations in the graphs, indicating stronger effects.

Comparing the findings from the logistic regression of the given dataset, it can be seen that the order of strong effect variables is similar, except medianIncome and abc1 are swapped. Furthermore, the model shows positive correlations between variables abc1 and notBornUK and the output. In contrast, the graphs from the Guardian indicate opposite correlations for these two variables. In conclusion, it can be inferred that the findings from the model are valid and reasonable.

Task 3

There are several factors that can affect the interpretability of regression coefficients, and it is important to take them into account. Firstly, multicollinearity can make it difficult to interpret coefficients when input variables are highly correlated with each other. This can lead to imprecise estimates and make it challenging to determine the unique contribution of each input variable. Secondly, the sample size needs to be large enough. A small sample size can result in unstable estimates of the coefficients. Another factor is outliers. Outliers in the data can influence coefficient estimates, leading to potentially biased results. Lastly, the scale of variables is considered. Variables with larger scales may have larger coefficient magnitudes, but this does not necessarily mean they have a stronger effect on the output.

Now, the reliability of the logistic model's coefficients can be evaluated by considering the previously mentioned factors. To begin with, multicollinearity can be examined using the cor function to calculate the correlations between the input variables. The correlation matrix below shows that some pairs of variables are highly correlated with each other.

```
cor(brexit[, c("abc1", "notBornUK", "medianIncome", "medianAge", "withHigherEd")])
```

```
##                    abc1  notBornUK medianIncome  medianAge withHigherEd
## abc1          1.0000000  0.3947856    0.7835105 -0.1662686    0.8863129
## notBornUK     0.3947856  1.0000000    0.5584333 -0.7314555    0.5501448
## medianIncome  0.7835105  0.5584333    1.0000000 -0.3467963    0.7412892
## medianAge    -0.1662686 -0.7314555   -0.3467963  1.0000000   -0.2356651
## withHigherEd  0.8863129  0.5501448    0.7412892 -0.2356651    1.0000000
```

Another way to examine multicollinearity is the Variance Inflation Factor (VIF). VIF measures the increase in the variance of the regression coefficients caused by multicollinearity. The VIF values calculated below reveal that the variables abc1 and withHigherEd are correlated to one or more other input variables in the model.

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.3
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.3.3
```

```
vif(model1)
```

```
##          abc1    notBornUK medianIncome    medianAge withHigherEd
##      9.994053     3.406493     2.683698     3.111688     8.558226
```

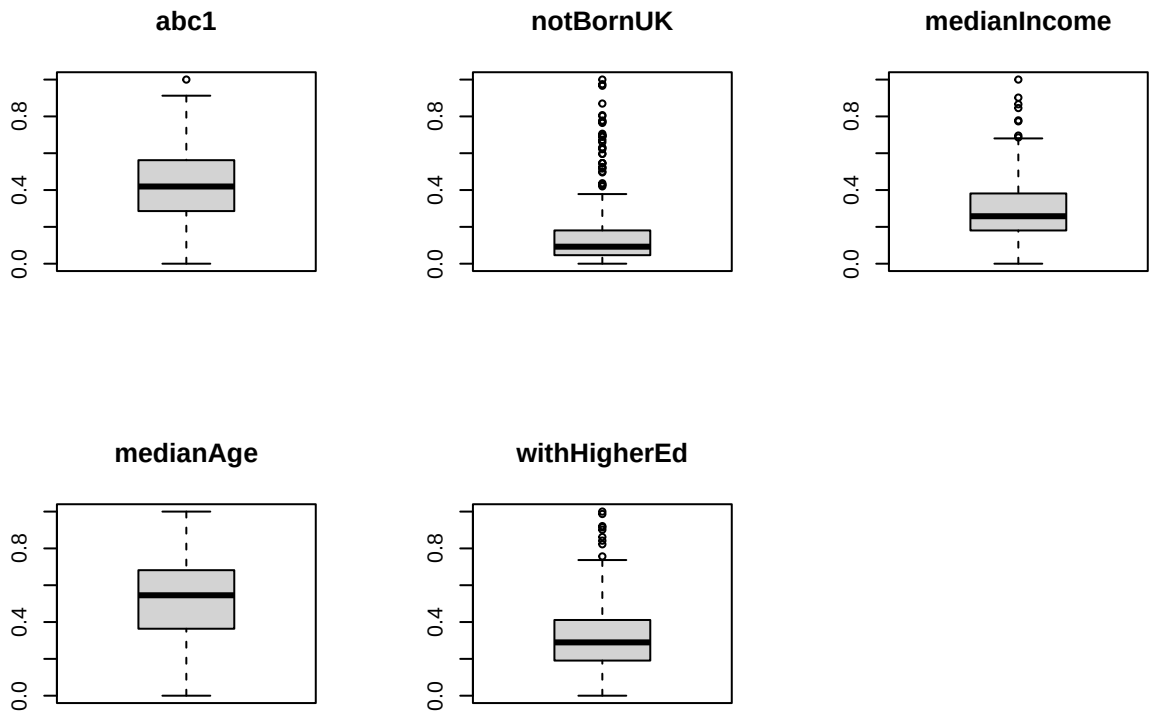
Secondly, the sample size is considered. There are 344 instances in total, which may be enough data for modelling.

```
dim(brexit)
```

```
## [1] 344   6
```

Next, box plots are used to identify outliers for each input variable. The plots indicate that there are some outliers for notBornUK, medianIncome, and withHigherEd, which should be removed for a better model. It is essential to preprocess outliers to build a robust model.

```
par(mfrow = c(2, 3))
boxplot(brexit$abc1, main="abc1")
boxplot(brexit$notBornUK, main="notBornUK")
boxplot(brexit$medianIncome, main="medianIncome")
boxplot(brexit$medianAge, main="medianAge")
boxplot(brexit$withHigherEd, main="withHigherEd")
```



Lastly, the scales of variables are examined by applying the summary function. It is evident that all input variables are already scaled between 0 and 1.

```r
summary(brexit)
```

```
##       abc1          notBornUK        medianIncome      medianAge
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.2857   1st Qu.:0.04622   1st Qu.:0.1804   1st Qu.:0.3636
##  Median :0.4194   Median :0.09244   Median :0.2577   Median :0.5455
##  Mean   :0.4315   Mean   :0.16068   Mean   :0.2923   Mean   :0.5103
##  3rd Qu.:0.5622   3rd Qu.:0.18067   3rd Qu.:0.3814   3rd Qu.:0.6818
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000
##   withHigherEd    voteBrexit
##  Min.   :0.0000   Mode :logical
##  1st Qu.:0.1908   FALSE:107
##  Median :0.2895   TRUE :237
##  Mean   :0.3162
##  3rd Qu.:0.4095
##  Max.   :1.0000
```

It can be concluded that the findings from the analysis of factors affecting the interpretability of the coefficient are evident.

Then, the logistic regression model is analysed further using BAGGING to determine the variability of the coefficients. The coefficient of variation (CV) is derived using BAGGING. It is calculated by the ratio of the standard deviation to the mean, expressed as a percentage. The coefficient of variation indicates the variability of coefficients across the model. Higher values refer to greater dispersion around the mean, which means greater instability in the coefficient estimates. This makes the coefficient values less consistent and more sensitive to changes in the data. Lower coefficient variability indicates more stable and reliable coefficient estimates, less influenced by the specific sample of data.

The following code implements BAGGING for logistic regression 1000 times to evaluate the coefficient of variation for each variable. The analysis suggests that variables can be ordered by their increasing absolute values of coefficient of variation, as follows: withHigherEd, abc1, medianAge, medianIncome, and notBornUK. Compared with the analysis of the magnitude of coefficients, the only difference is that medianAge and medianIncome swap positions. Therefore, it can be concluded that the previous findings are reliable and well-supported by the coefficient of variation.

```r
coefficients_matrix <- matrix(NA, nrow = 1000, ncol = length(coef(model1)))

# Perform bagging for logistic regression
for (i in 1:1000) {
  bs_data <- brexit[sample(nrow(brexit), replace = TRUE), ]
  bs_model <- glm(voteBrexit ~ ., data = bs_data, family = binomial)
  coefficients_matrix[i, ] <- coef(bs_model)
}

coefficients_mean <- apply(coefficients_matrix, 2, mean)
coefficients_sd <- apply(coefficients_matrix, 2, sd)
coefficient_variability <- (coefficients_sd/coefficients_mean)*100
print("intercept     abc1    notBornUK medianIncome medainAge withHigherEd")
```

```
## [1] "intercept     abc1    notBornUK medianIncome medainAge withHigherEd"
```

```r
print(coefficient_variability)
```

```
## [1] -594.49293   16.01856   31.31869  -30.18627   22.39168  -14.13288
```

Task 4

Based on the analysis conducted for Task 3, it has been observed that some of the input variables exhibit a high degree of correlation with each other, which leads to multicollinearity. This has been demonstrated by the correlations calculated between input variables and their VIF values. To address this issue and improve the model, an alternative logistic regression approach called ridge logistic regression is used. Also known as logistic regression with ridge or L2 regularisation, this technique has several advantages and disadvantages compared to standard logistic regression without regularisation.

Firstly, ridge regression is useful when dealing with multicollinearity. The technique shrinks the coefficients, mitigating the impact of multicollinearity and providing more stable estimates. Moreover, ridge regression reduces overfitting by penalising large coefficients, improving generalised performance on unseen data. Additionally, ridge regression produces stable coefficient estimates by the penalty term, ensuring the estimates are more stable and less sensitive to small changes in the training data. This makes the model generalise better to outliers and noisy data. Lastly, it performs well when dealing with large multivariate data where the number of input variables is larger than the number of observations.

However, some disadvantages of ridge regression need to be considered. Firstly, unlike lasso, ridge regression does not perform variable selection and only shrinks the coefficients toward zero. Moreover, ridge regression requires cross-validation or other model selection techniques to choose the optimal value of the regularisation parameter, lambda. Selecting an appropriate value of lambda is crucial for the optimal performance of the model.

A logistic regression model is defined with all inputs and the output, voteBrexit, and trained using the given dataset. However, this time, the model is trained on ridge regularisation. The plot shows the cross-validation error based on the log values of lambda. The vertical line on the plot indicates the log of the optimal value of lambda, which is the value that minimises the prediction error. This lambda value will give the most accurate model.

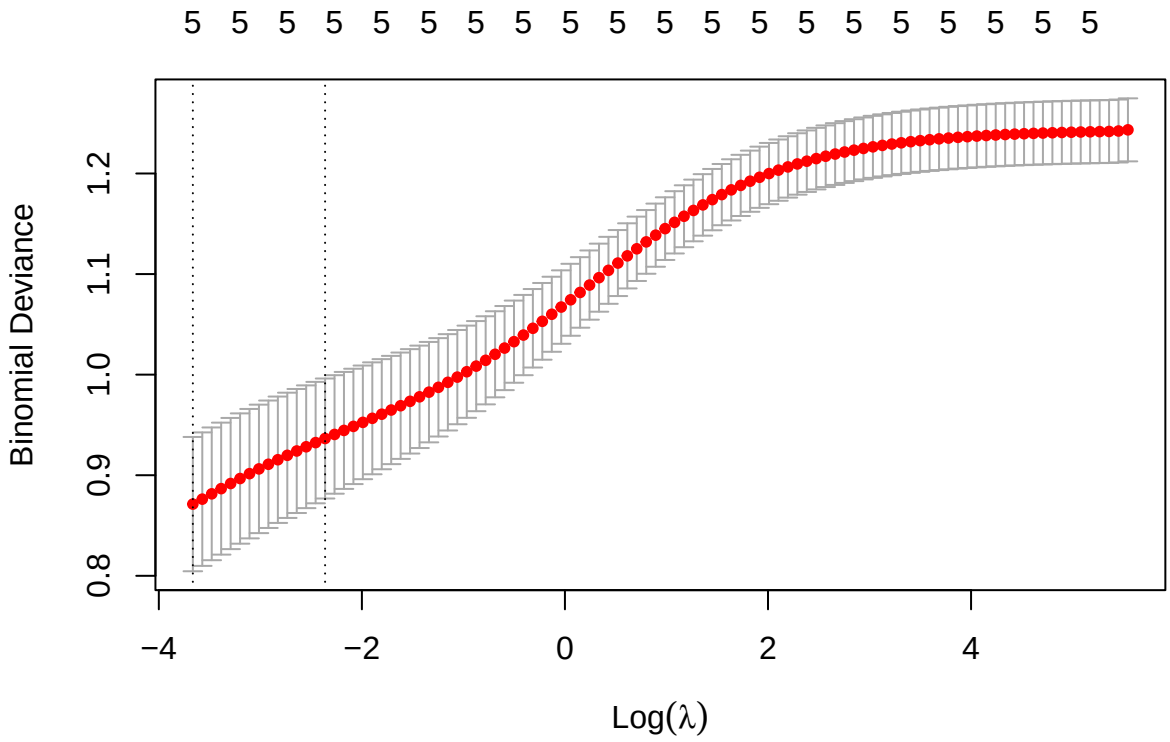
```r
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.3.3
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```r
cv.ridge <- cv.glmnet(data.matrix(brexit[1:5]), brexit$voteBrexit, family="binomial", alpha=0)
plot(cv.ridge)
```



The lambda value obtained from the cross-validation is 0.0256.

```
cv.ridge$lambda.min
```

```
## [1] 0.02560518
```

The coefficients of the model with the optimal lambda are displayed below. The coefficients can be listed in decreasing order of the absolute values as follows: withHigherEd, abc1, medianIncome, medianAge, and notBornUK. These coefficients match the order of coefficient magnitudes from Task 2, indicating that the findings are again reliable and well-supported by the ridge regression model.

```
coef(cv.ridge, cv.ridge$lambda.min)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept)   2.0972767
## abc1          2.0131294
## notBornUK    -0.4345913
## medianIncome -1.6002499
## medianAge     1.5990217
## withHigherEd -7.3774923
```