MATH5743M: Statistical Learning- Assignment 2

Aishwarya Selvaraj

2024-04-16

A Statistical Analysis on Brexit.

INTRODUCTION:

The Brexit referendum, a pivotal moment in UK history, presented an opportunity to delve deeply into the voting dynamics that decided its outcome. This assignment aimed to analyze the demographic and socio-economic factors influencing the Brexit vote using advanced statistical modeling techniques. By employing logistic regression and Lasso regression, we sought to uncover patterns and predictors that could explain the distribution of Leave and Remain votes across different electoral wards. The focus was not only on identifying the factors that significantly influenced the voting decision but also on enhancing the interpretability and reliability of the statistical models used, ensuring that the insights derived were both robust and actionable.

TASK 1:

```r
library(tidyverse)
library(cluster)
library(scatterplot3d)
library(plotly)
```

Using K means algorithm to cluster the inputs:

$set.seed(123)$ ensures that the results of the random processes within K-means are reproducible and $k.max = 10$ limits the number of clusters tested to 10.

```r
brexit_data <- read.csv("brexit.csv")
set.seed(123)
k.max <- 10
```

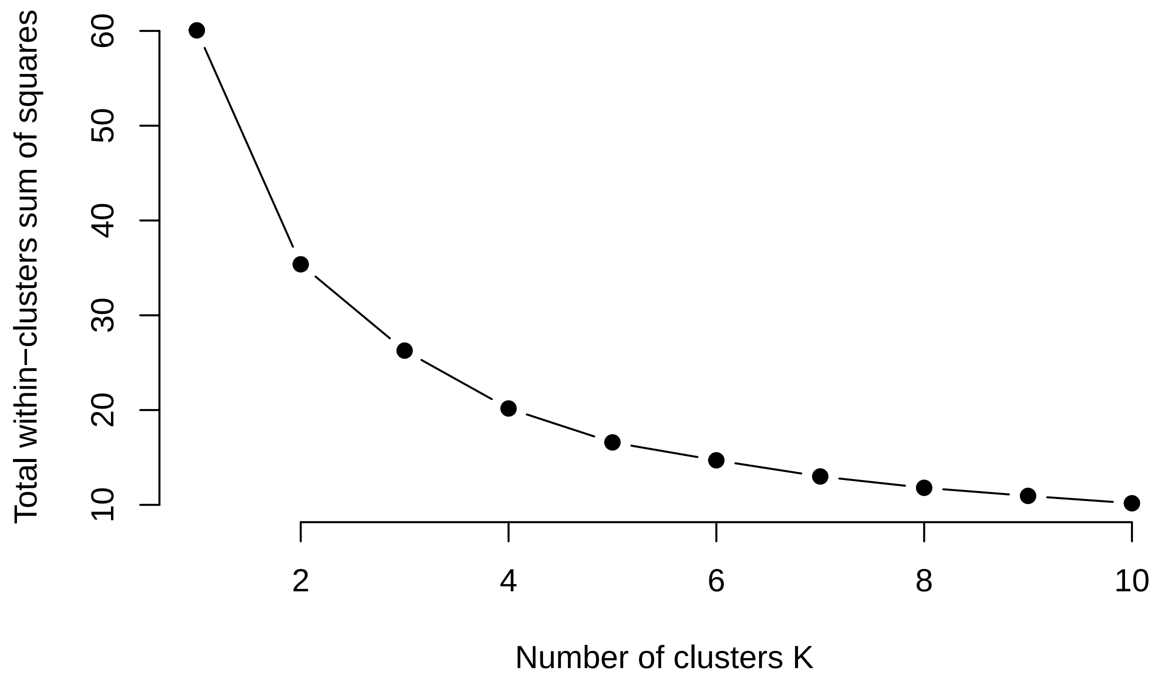
Validating the optimal number of clusters:

Elbow Method:

"Elbow Plot," which is frequently used in K-means clustering to determine the best number of clusters (k). This plot displays the total within-cluster sum of squares (WSS) on the y-axis against the number of clusters k on the x-axis. The WSS measures the compactness of the clusters; lower values indicate that the data points are closer to their respective cluster centroids.

$sapply(1 : k.max, function(k))$ applies the K-means algorithm for k ranging from 1 to k.max and calculates the WSS for each k. $plot(1 : k.max, wss, ...)$ generates the elbow plot for visual inspection to identify the optimal k by locating the elbow point.

```r
wss <- sapply(1:k.max, function(k){
  kmeans(brexit_data[, 1:5], centers = k, nstart = 50)$tot.withinss
})
plot(1:k.max, wss, type = "b", pch = 19, frame = FALSE,
     xlab = "Number of clusters K",
     ylab = "Total within-clusters sum of squares")
```



Upon examining the plot, as the number of clusters increases from 1 to 10, the WSS rapidly decreases until around k=4. After this point, the decrease in WSS becomes more gradual, indicating that the addition of each subsequent cluster is less useful. The "elbow" of the plot, where the rate of decrease sharply changes, indicates the optimal number of clusters. In this instance, the elbow seems to appear around k=4, which could be argued as the best number of clusters for the dataset.

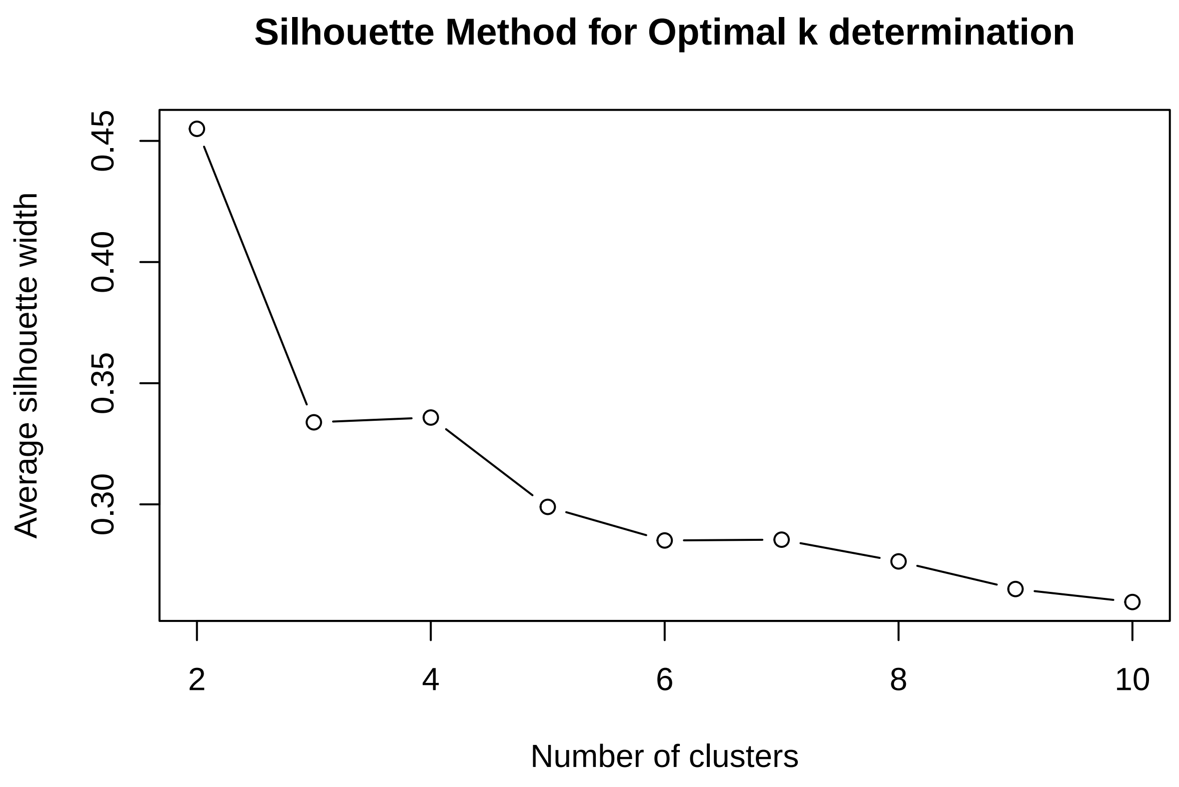
Silhoutte Method:

To calculate the silhouette score and determine the optimal number of clusters for Brexit dataset:

The *cluster* library is used for computing the silhouette widths, and stats is generally preloaded for k-means in R. The loop calculates k-means clusters and their corresponding silhouette scores for each value of k from 2 to 10. *silhouette*() computes the silhouette widths for each point in the dataset, which are then averaged to get the silhouette score for that k. The plot helps visually identify the k with the highest average silhouette score, suggesting it as the optimal number of clusters.

```r
library(cluster)
library(stats)
data_matrix <- as.matrix(brexit_data[, c("abc1", "notBornUK",
                                         "medianIncome", "medianAge", "withHigherEd")])
```

```r
k_values <- 2:10
silhouette_scores <- numeric(length(k_values))
for (k in k_values) {
  set.seed(123)
  km <- kmeans(data_matrix, centers = k, nstart = 25)
  sil_widths <- silhouette(km$cluster, dist(data_matrix))
  silhouette_scores[k - 1] <- mean(sil_widths[, 3])
}
plot(k_values, silhouette_scores, type = 'b', xlab = "Number of clusters",
     ylab = "Average silhouette width",
     main = "Silhouette Method for Optimal k determination")
```



The plot provided showcases the average silhouette widths for different numbers of clusters from 2 to 10. The silhouette score measures how similar an object is to its own cluster compared to other clusters. A higher silhouette width indicates better clustering quality, where objects are more similar to their own cluster and distinct from others.

Analysis of the Silhouette Plot:

Peak Value: The highest average silhouette width occurs at $k = 2$, indicating that two clusters provide the best separation and cohesion among the data points. Decline: After $k = 2$, the silhouette scores decline and then level off, suggesting that increasing the number of clusters beyond 2 does not lead to improvements in clustering quality.

Interpretation:

The highest silhouette score at $k = 2$ suggests that the dataset is optimally partitioned into two distinct groups. This indicates that two main patterns or groups can be observed in the Brexit voting data, which could represent differing demographics, economic conditions, or educational backgrounds influencing the voting behavior.

Recommendations:

1. Cluster Number: Based on the silhouette analysis, should consider using two clusters for further analysis of the dataset to ensure that the clusters are meaningful and distinct.
2. Further Analysis: Analyze the characteristics of these two clusters to understand the defining features of each group. This can provide insights into demographic or other factors that distinguish between the two voting behaviors.

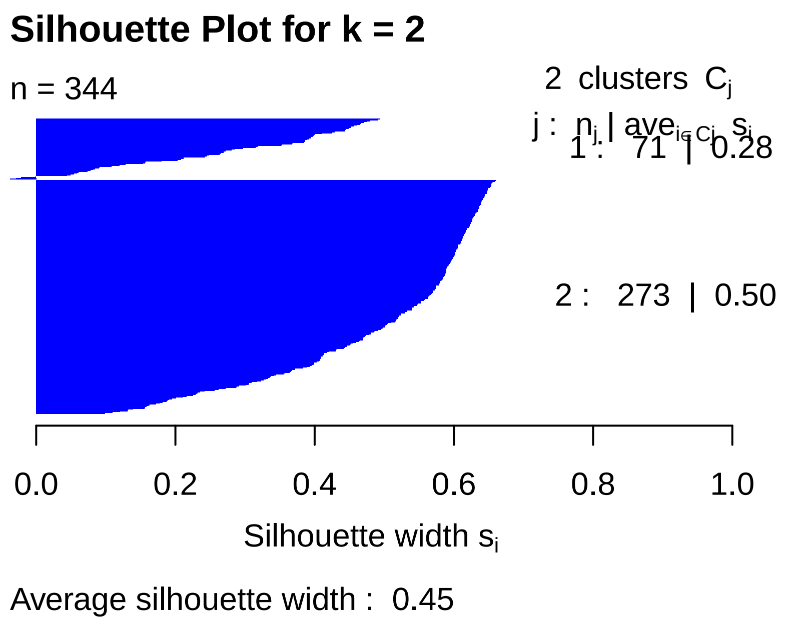
This method provides a clear, quantitative measure to back the choice of the number of clusters, ensuring that the analysis is based on data-driven insights.

Justifying my choice:

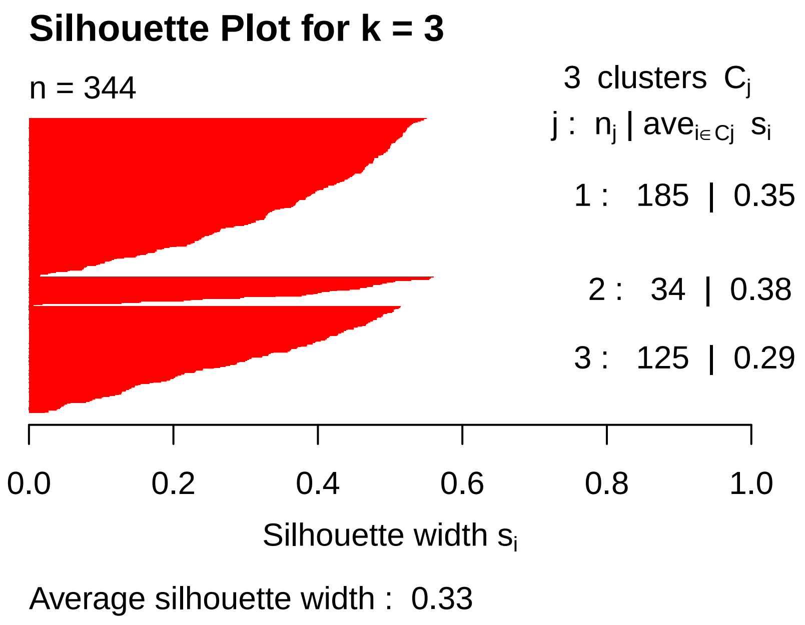
The below code performs K-means clustering on the dataset with 2, 3, and 4 clusters using the kmeans function. This function is correctly applied to the dataset excluding the outcome variable (assumed to be in the 6th column based on brexit_data[, -6]).

1. **Silhouette Analysis:** For each clustering result, it calculates the silhouette widths using the silhouette function from the cluster package. This helps in assessing the quality of the clustering by measuring how similar an object is to its own cluster compared to other clusters.
2. **Plotting Silhouette Analysis:** The silhouette analysis for each k value (2, 3, and 4) is visualized using the plot function. These plots should display the silhouette widths for each data point in the clusters, helping determine the optimal number of clusters based on which configuration gives the highest average silhouette width.
3. **Cluster Visualization:** It visualizes the clusters using the fviz_cluster function from the factoextra package. This visualization will show how data points are grouped in each cluster configuration, with different colors representing different clusters.

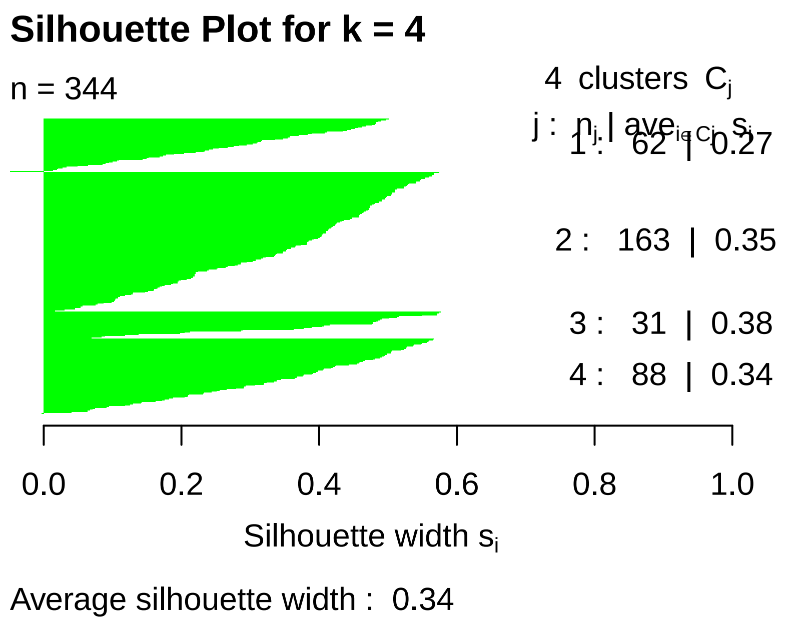
```r
library(cluster)
library(ggplot2)
library(factoextra)
set.seed(123)
k2 <- kmeans(brexit_data[, -6], centers = 2, nstart = 25)
k3 <- kmeans(brexit_data[, -6], centers = 3, nstart = 25)
k4 <- kmeans(brexit_data[, -6], centers = 4, nstart = 25)
sil2 <- silhouette(k2$cluster, dist(brexit_data[, -6]))
sil3 <- silhouette(k3$cluster, dist(brexit_data[, -6]))
sil4 <- silhouette(k4$cluster, dist(brexit_data[, -6]))
plot(sil2, col = "blue", main = "Silhouette Plot for k = 2")
```



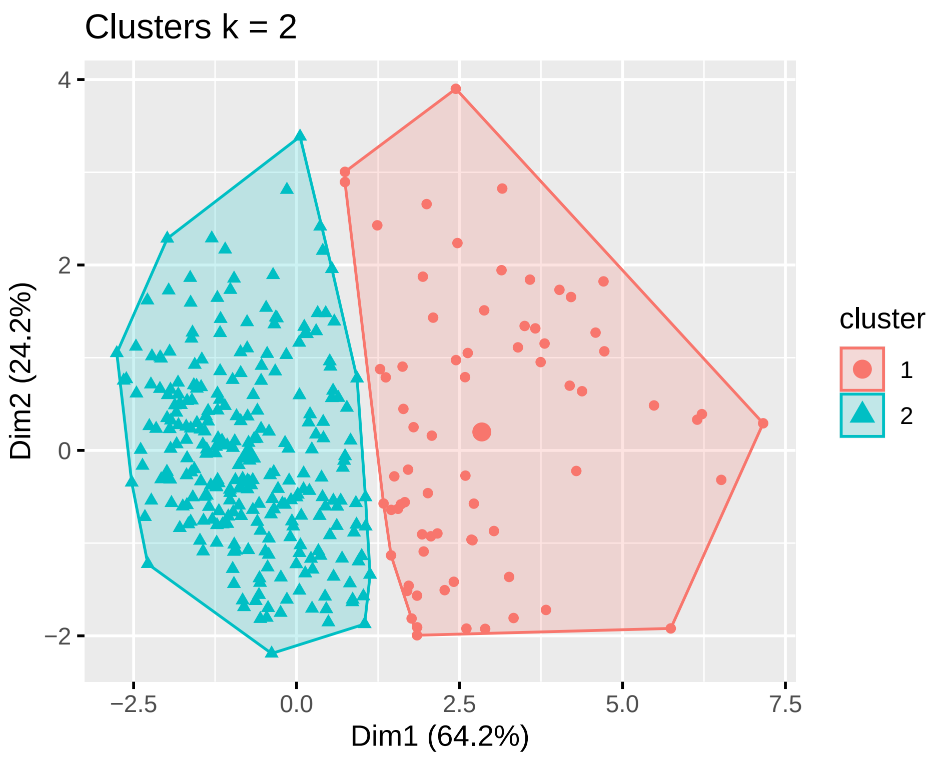
```r
plot(sil3, col = "red", main = "Silhouette Plot for k = 3")
```



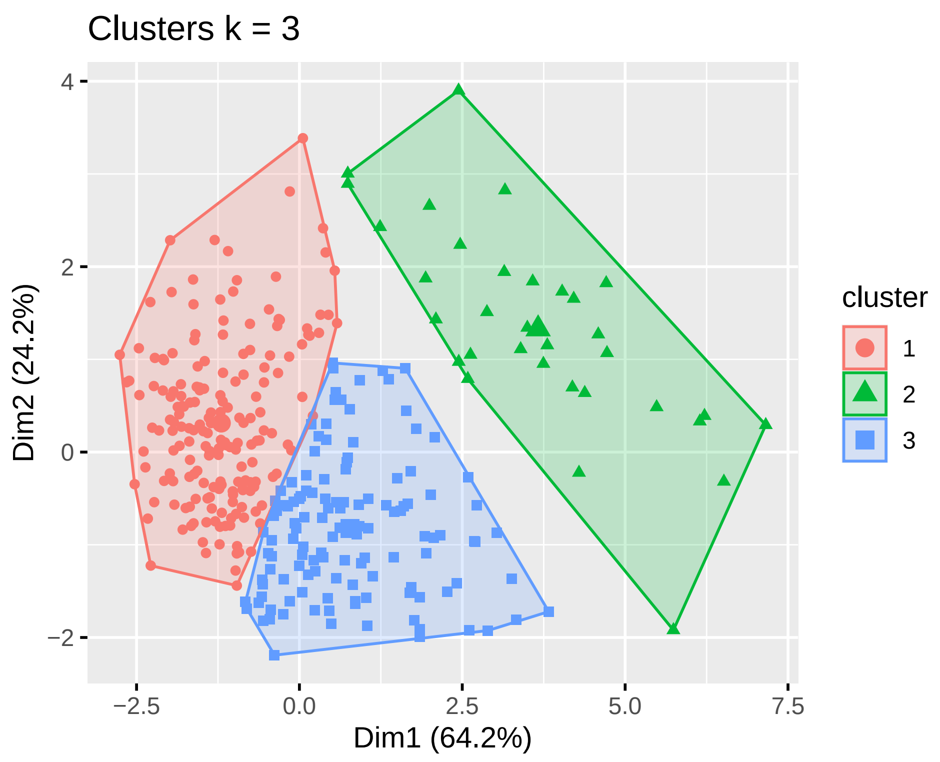
```r
plot(sil4, col = "green", main = "Silhouette Plot for k = 4")
```



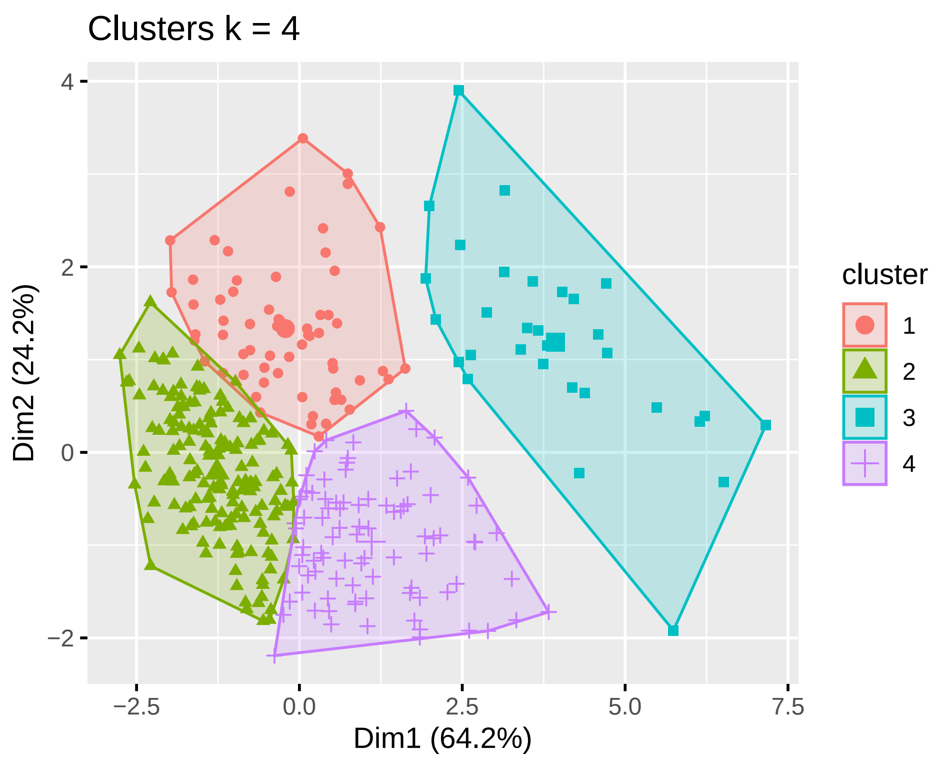
```r
fviz_cluster(k2, data = brexit_data[, -6], geom = "point", main = "Clusters k = 2")
```



```r
fviz_cluster(k3, data = brexit_data[, -6], geom = "point", main = "Clusters k = 3")
```



```r
fviz_cluster(k4, data = brexit_data[, -6], geom = "point", main = "Clusters k = 4")
```



Silhouette Plots:

- Each plot corresponds to a different number of clusters $k = 2, 3, and 4$.
- The silhouette width values range from -1 to $+1$, where values near $+1$ indicate that the point is well-matched to its own cluster and poorly matched to neighboring clusters.
- For $k = 2$: The average silhouette width is 0.45. One cluster has a lower average silhouette score 0.58, suggesting moderate cohesion and separation.
- For $k = 3$: The average silhouette width decreases to 0.33. This indicates less cohesion and separation compared to $k = 2$, as shown by the lower scores and more variation in the silhouette widths.
- For $k = 4$: The average silhouette width is slightly higher at 0.34 compared to $k = 3$, but still lower than $k = 2$, suggesting that adding more clusters does not necessarily improve cluster quality.

Cluster Visualization:

- Clusters $k = 2$: Two distinct groups are visualized, with reasonable separation between them, although the cluster boundaries overlap slightly.
- Clusters $k = 3$: Introduces a third cluster, showing more overlap and less clear separation among the clusters compared to $k = 2$.
- Clusters $k = 4$: Shows four clusters, but with even more overlap, especially between some of the clusters, which could indicate less distinct groupings in the data.

Interpretation:

The Silhouette analysis for $k = 2$ shows a higher average silhouette width, suggesting that two clusters might provide a better balance of cohesion and separation than higher numbers of clusters. However, this needs to be considered along with the context of the data and the specific research or application needs.

The visual plots align with the silhouette scores, showing clearer separation at lower numbers of clusters $k = 2$, and increasing overlap and ambiguity as the number of clusters increases.

The decision on the optimal number of clusters depends on both the statistical metrics (like silhouette scores) and practical considerations of the dataset's context and the specific questions being addressed. If the goal is to maximize the distinctiveness of the clusters with clear separation, $k = 2$ might be the most appropriate choice based on this analysis. However, further analysis and domain knowledge might justify different numbers of clusters depending on additional data insights or application-specific requirements.

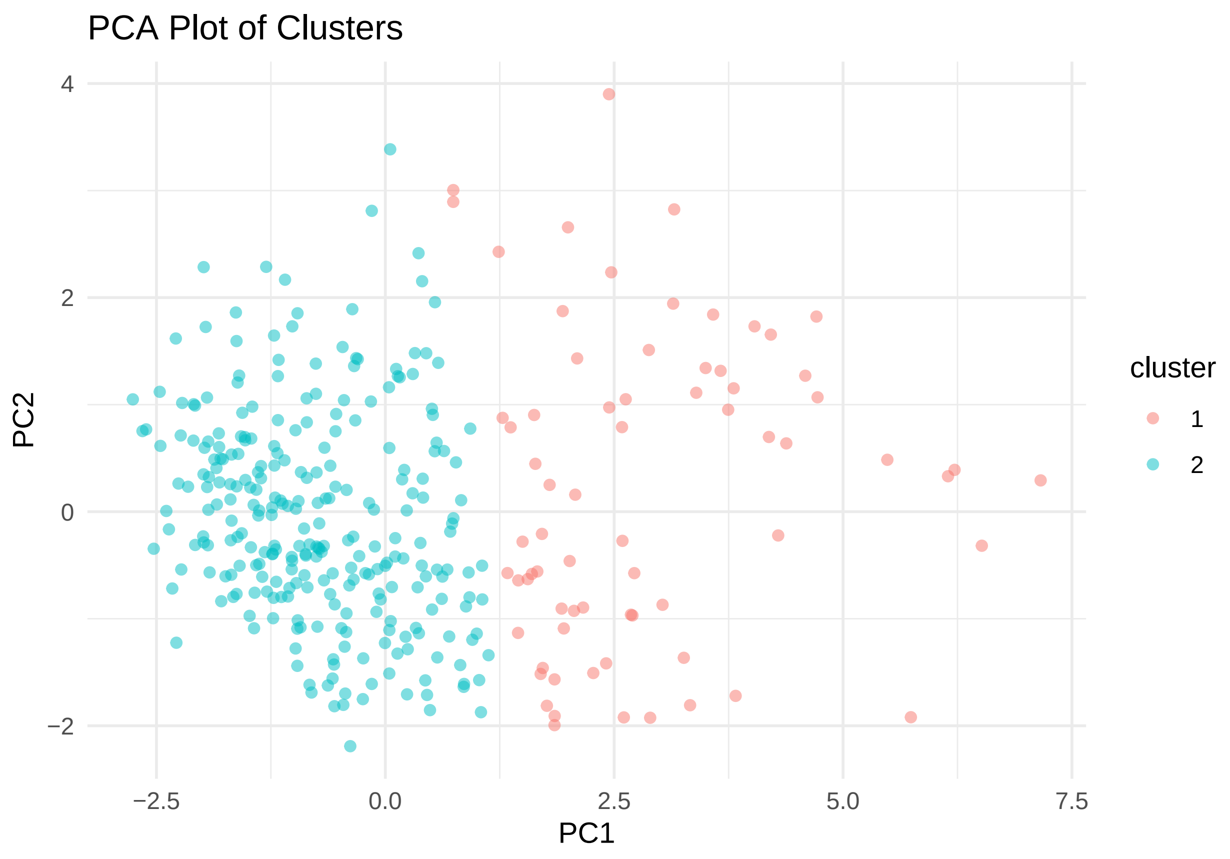
```r
opt.k <- 2
km <- kmeans(brexit_data[, 1:5], centers = opt.k, nstart = 50)
brexit_data$cluster <- km$cluster
```

Visualising the clusters:

The scatter plot shows the clusters formed by the K-means algorithm, with the data points plotted on the first two principal components (PC1 and PC2) obtained from a PCA transformation. Each color represents one of the two clusters. The plot is titled "PCA Plot of Clusters," indicating that PCA was used for dimensionality reduction before visualization.

$prcomp(brexit\_data[, 1 : 5], center = TRUE, scale. = TRUE)$ performs PCA on the input variables for dimensionality reduction. $ggplot2$ to create a scatter plot of the first two principal components, coloring the points by the cluster they belong to.

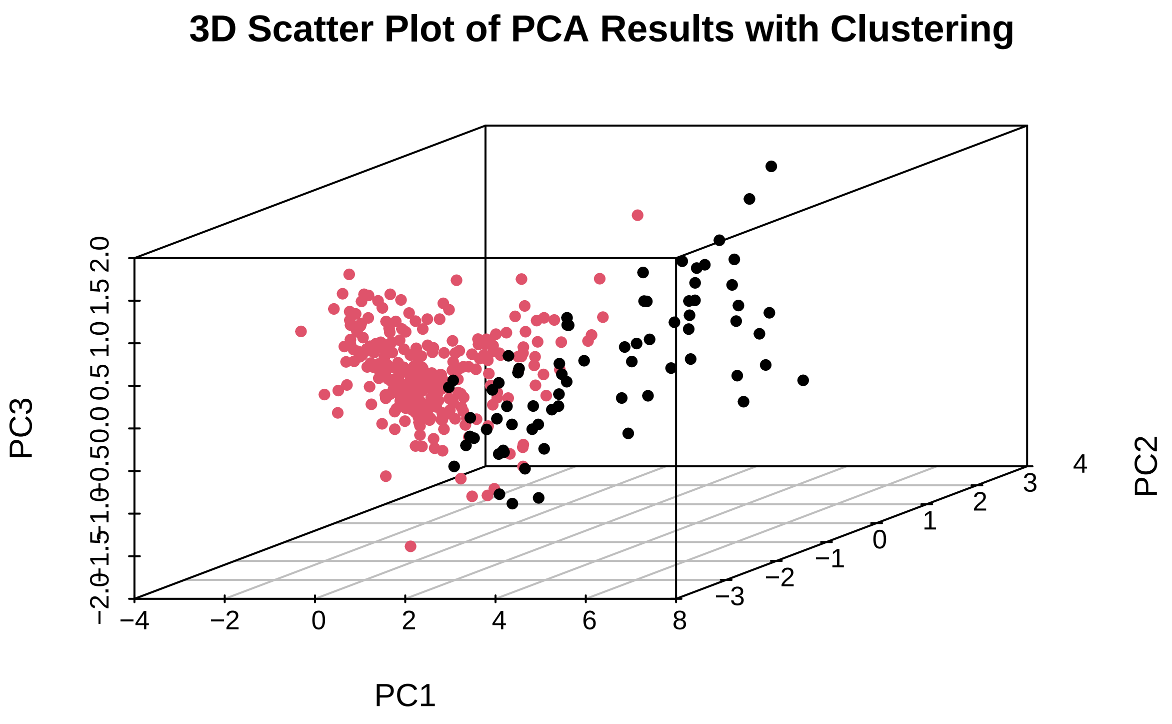
```r
library(ggplot2)
pca <- prcomp(brexit_data[, 1:5], center = TRUE, scale. = TRUE)
pca_data <- data.frame(pca$x, cluster = as.factor(brexit_data$cluster))
ggplot(pca_data, aes(PC1, PC2, color = cluster)) +
  geom_point(alpha = 0.5) +
  labs(title = "PCA Plot of Clusters") +
  theme_minimal()
```



- Cluster 1: This cluster is mostly concentrated towards the right side of the plot, indicating similarity in the data points within this cluster in terms of the principal components.
- Cluster 2: Occupying mostly the left and central parts of the plot, these points show a denser grouping compared to Cluster 1, suggesting tighter cohesion within this cluster. Overlap and Spread: There's some overlap between the clusters around the center of the plot, indicating that while the clusters are distinct, there are some areas where their characteristics converge.
- Practical Implications: If these clusters represent different voting patterns (Leave or Remain), the PCA plot suggests that there are two discernible groups with different demographic or socioeconomic characteristics that might influence their voting behavior.
- Cluster Significance: The PCA components might relate to the most significant factors (like median age, median income, etc.) that explain the variance in voting patterns, as PCA tends to align the principal components with the directions of maximum variance.

The PCA plot provides a clear visual confirmation of the cluster analysis results. The clusters appear to be well-defined, though the exact meaning of each principal component would need further investigation to understand what demographic characteristics are most influential in differentiating these clusters. This type of visualization is essential for interpreting complex multivariate data and can support further statistical analysis or reporting.

```r
scatterplot3d(x = pca$x[, 1], y = pca$x[, 2], z = pca$x[, 3],
              color = brexit_data$cluster, pch = 20,
              xlab = "PC1", ylab = "PC2", zlab = "PC3",
              main = "3D Scatter Plot of PCA Results with Clustering")
```



This visualization aids in understanding the spatial distribution of clusters in the multidimensional space reduced to three dimensions. The degree of separation between clusters can give an intuition about how distinct the groups are, which could be indicative of the different voting behaviors captured by the model. Clusters that are closer together might represent electoral wards with similar demographic profiles, while widely separated clusters might represent wards with more distinct characteristics.

Validating K-means model:

Each row represents the centroid of a cluster, which is the mean value of each feature within that cluster. The centroids serve as a profile for each cluster, giving insight into the common characteristics of the wards grouped together by the K-means algorithm.

```r
centroids <- km$centers
print(centroids)
```

```
##        abc1  notBornUK medianIncome medianAge withHigherEd
## 1 0.6618420 0.42289028    0.5168433 0.3072983    0.5660675
## 2 0.3715501 0.09248315    0.2339224 0.5631036    0.2512531
```

Cluster 1:

- Higher proportion of individuals in ABC1 social classes (middle to upper class) compared to Cluster 2.
- Higher proportion of residents born outside the UK compared to Cluster 2.
- Higher median income compared to Cluster 2.
- Lower median age compared to Cluster 2.
- Higher proportion of residents with higher education compared to Cluster 2.

Cluster 2:

- Lower proportion of individuals in ABC1 social classes compared to Cluster 1.
- Lower proportion of residents born outside the UK compared to Cluster 1.
- Lower median income compared to Cluster 1.
- Higher median age compared to Cluster 1.
- Lower proportion of residents with higher education compared to Cluster 1.

The below output helps lay the groundwork for a more comprehensive justification and by displaying the mean values of demographic and socio-economic attributes for each cluster formed by the K-means algorithm, along with the mean proportion of the Brexit vote outcome within each cluster.

```
brexit_data_with_clusters <- brexit_data
brexit_data_with_clusters$cluster <- as.factor(brexit_data_with_clusters$cluster)
cluster_summary <- brexit_data_with_clusters %>% group_by(cluster) %>% summarise_all(mean)
print(cluster_summary)
```

```
## # A tibble: 2 x 7
##   cluster  abc1 notBornUK medianIncome medianAge withHigherEd voteBrexit
##   <fct>   <dbl>     <dbl>        <dbl>     <dbl>        <dbl>      <dbl>
## 1 1       0.662    0.423         0.517     0.307        0.566      0.225
## 2 2       0.372    0.0925        0.234     0.563        0.251      0.810
```

From these summaries, we can see clear differences between the two clusters in terms of socio-economic and demographic characteristics. Cluster 1 appears to represent areas with higher socio-economic status, lower median age, and a lower likelihood of voting for Brexit, while Cluster 2 represents areas with lower socio-economic status, higher median age, and a higher likelihood of voting for Brexit.

Analyzing the Relationship Between Clusters and Brexit Vote Outcomes:

The contingency table shows the distribution of Brexit vote outcomes: FALSE for Remain, TRUE for Leave across the two clusters identified by the K-means algorithm.

```
brexit_table <- table(brexit_data$voteBrexit, brexit_data$cluster)
print(brexit_table)
```

```
##
##          1   2
##   FALSE  55  52
##   TRUE   16 221
```

From above results, we can observe the following:

- In Cluster 1, there are 55 wards where people voted *FALSE* (Remain) and 16 wards where people voted TRUE (Leave).
- In Cluster 2, there are 52 wards where people voted *FALSE* (Remain) and 221 wards where people voted TRUE (Leave).

These numbers suggest a clear association between cluster membership and the Brexit vote outcomes. Cluster 2, which represents areas with lower socio-economic status, higher median age, and lower education levels, has a significantly higher proportion of wards where people voted for Brexit (TRUE) compared to Cluster 1.

This analysis reinforces the earlier observations that certain demographic and socio-economic characteristics captured by the K-means clustering algorithm are indeed associated with the likelihood of voting for Brexit.

Proportion Table:

The proportion table normalizes these counts to show the percentage of wards within each cluster that voted Remain or Leave.

The *prop.table*() function with $margin = 2$ computes column-wise proportions, allowing us to compare the vote outcomes within each cluster, irrespective of the cluster's size.

- In Cluster 1, approximately 77.46% of wards voted to Remain, while 22.54% voted to Leave.
- In Cluster 2, approximately 19.05% of wards voted to Remain, while 80.95% voted to Leave.

```
print(prop.table(brexit_table, margin = 2))
```

```
##
##                   1         2
##   FALSE 0.7746479 0.1904762
##   TRUE  0.2253521 0.8095238
```

This normalized view allows for a comparison of the vote outcomes within each cluster, irrespective of the cluster's size. It reaffirms the earlier observation that Cluster 2, characterized by lower socio-economic status and higher median age, has a significantly higher proportion of wards where people voted for Brexit compared to Cluster 1.

Statistically, these proportions can reveal significant associations between clusters and voting behavior, which could be tested for independence using a **Chi-squared test**. The results would inform whether the clustering successfully captures differences in Brexit voting tendencies, potentially reflecting underlying demographic differences.

Visualize the Contingency Table:

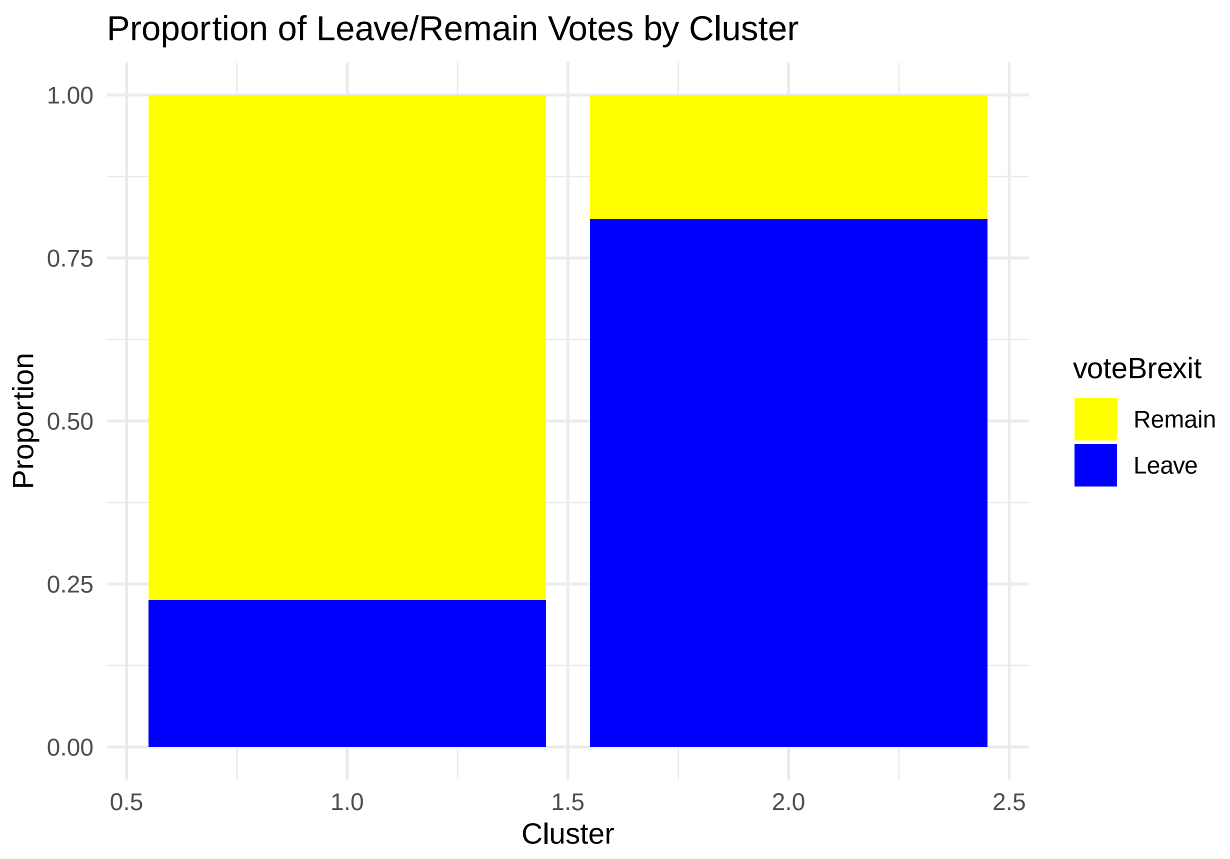
Visualizing data can often reveal patterns that are not obvious from raw tables.

```
 total_counts <- colSums(brexit_table)
```

```
library(reshape2)
```

```
brexit_melted <- melt(brexit_table)
colnames(brexit_melted) <- c("voteBrexit", "Cluster", "Count")
brexit_melted$TotalCount <- total_counts[as.character(brexit_melted$Cluster)]
brexit_melted$Proportion <- with(brexit_melted, Count / TotalCount)
brexit_melted$voteBrexit <- factor(brexit_melted$voteBrexit, labels = c("Remain", "Leave"))
```

```r
library(ggplot2)
ggplot(brexit_melted, aes(x = Cluster, y = Proportion, fill = voteBrexit)) +
  geom_bar(stat = "identity", position = "stack") +
  labs(x = "Cluster", y = "Proportion", title = "Proportion of Leave/Remain Votes by Cluster") +
  scale_fill_manual(values = c("Remain" = "yellow", "Leave" = "blue")) +
  theme_minimal()
```



The stacked bar chart justifies the clusters validity by illustrating that they represent distinct voting behaviors. The extent of the separation between Remain and Leave within each cluster supports the idea that the clusters are not arbitrary but are meaningfully associated with the Brexit vote outcomes.

Comparing visual patterns with the results of statistical test:

Chi-squared test:

The Chi-squared test statistic, $X\text{-}squared = 87.023$, with degrees of freedom $df = 1$, has yielded a $p - value$ below $2.2e - 16$, which is virtually zero. This remarkably low p-value indicates a highly significant association between the clusters and the Brexit vote outcome. It suggests that cluster membership and the voting behavior in the Brexit referendum are not independent. Instead, there is a strong relationship between cluster characteristics and how the wards voted, underscoring the socio-economic and demographic factors' influence on the Brexit vote.

```r
chi_results <- chisq.test(brexit_table)
print(chi_results)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
```

```
##
## data:  brexit_table
## X-squared = 87.023, df = 1, p-value < 2.2e-16
```

Logistic Regression to assess the predictive power:

A logistic regression analysis is conducted to examine the influence of cluster membership on Brexit vote outcomes.

Coefficients Interpretation: Intercept: The intercept -1.2347 represents the log odds of voting for Brexit when the cluster factor is at its reference level (Cluster 1).

Cluster Factor (Cluster 2): The coefficient for Cluster 2 2.6817 represents the change in log odds of voting for Brexit when comparing Cluster 2 to Cluster 1. This coefficient is statistically significant $p < 0.001$, indicating that Cluster 2 significantly influences the likelihood of voting for Brexit compared to Cluster 1.

Significance Levels: Both the intercept and the coefficient for Cluster 2 are highly significant, with p-values much smaller than the conventional significance level of 0.05. This suggests strong evidence that cluster membership affects the likelihood of voting for Brexit.

Model Fit: The model's fit is assessed using the null deviance, residual deviance, and AIC value.

Null Deviance: The null deviance 426.52 represents the difference in log likelihood between a model with only an intercept and a model with predictors.

Residual Deviance: The residual deviance 341.63 measures the difference in log likelihood between the fitted model and the saturated model (a model with perfect fit). A lower residual deviance indicates a better fit of the model.

AIC (Akaike Information Criterion): The AIC value 345.63 provides a measure of the model's goodness of fit while penalizing for model complexity. A lower AIC value indicates a better balance between goodness of fit and model complexity.

Overall Interpretation: The logistic regression analysis demonstrates that cluster membership, particularly belonging to Cluster 2, significantly influences the likelihood of voting for Brexit. The model provides valuable insights into how demographic and socio-economic characteristics captured by the clusters predict voting patterns in the Brexit referendum.

```r
brexit_data$cluster_factor <- as.factor(brexit_data$cluster)
logit_model <- glm(voteBrexit ~ cluster_factor, family = binomial(link = "logit"), data = brexit_data)
summary(logit_model)
```

```
##
## Call:
## glm(formula = voteBrexit ~ cluster_factor, family = binomial(link = "logit"),
##     data = brexit_data)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.2347     0.2840  -4.347 1.38e-05 ***
## cluster_factor2   2.6817     0.3232   8.298  < 2e-16 ***
## ---
## 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: 341.63  on 342  degrees of freedom
## AIC: 345.63
##
## Number of Fisher Scoring iterations: 4
```

The statistical analyses provided by Chi-squared test and logistic regression offer a strong justification for the clustering results. The Chi-squared test confirms a significant association between clusters and Brexit voting, while the logistic regression provides insights into the predictiveness of cluster membership on voting behavior. Some clusters have emerged as significant predictors, which supports the clusters' relevance and validates the K-means model. It suggests that the cluster memberships derived from socio-demographic and economic features can indeed serve as indicators of Brexit voting patterns.

TASK 2:

Building Logistic Regression Model:

Using Logistic Regression to find which input variable is relevant for explaining the output:

Creating a logistic regression model using all input variables abc1, notBornUK, medianIncome, medianAge, withHigherEd. The conversion of voteBrexit from logical values to binary 0 and 1 is done to proceed with the model.

```
unique_values <- unique(brexit_data$voteBrexit)
print(unique_values)
brexit_data$voteBrexit <- ifelse(brexit_data$voteBrexit == "TRUE", 1, ifelse(brexit_data$voteBrexit ==
table(brexit_data$voteBrexit)
```

```
brexit_data <- na.omit(brexit_data)
logit_model <- glm(voteBrexit ~ abc1 + notBornUK + medianIncome +
                      medianAge + withHigherEd,
family = binomial(link = "logit"), data = brexit_data)
summary(logit_model)
```

```
##
## Call:
## glm(formula = voteBrexit ~ abc1 + notBornUK + medianIncome +
##     medianAge + withHigherEd, family = binomial(link = "logit"),
##     data = brexit_data)
##
## 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
```

- The model coefficients suggest significant relationships between all predictor variables and the vote-Brexit outcome.
- *withHigherEd* has the strongest **negative effect**, indicating a higher level of higher education is strongly associated with voting to **Remain**.
- *abc*1 has a significant **positive effect**, implying that higher values more affluent social classes correlate with a higher likelihood of voting to **Leave**.

Interpreting Model Coefficients:

Identifying the direction and magnitude of each input:

The coefficients show how each predictor variable influences the likelihood of voting for Brexit (1). The Estimate column gives the log-odds impact of each predictor, where:

- Positive values increase the log-odds of the outcome being 1 (voting to Leave).
- Negative values decrease the log-odds of the outcome being 1.

```r
model_summary <- summary(logit_model)
print(model_summary$coefficients)
```

```
##                 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
```

Intercept:

- Estimate: -0.1386

This coefficient represents the log-odds of voting for Brexit when all other predictor variables are zero. Its practical significance is limited without the context of other variables.

abc1:

- Magnitude: 17.5780
- p-value: 1.56e-09
- Direction: Positive

This coefficient is positive and very large, indicating a strong association between being in a higher social class and the likelihood of voting for Brexit. The p-value is extremely low, signifying that this effect is statistically significant. This suggests that as the proportion of middle to upper-class individuals in an area increases, so does the likelihood of the area voting for Brexit.

notBornUK:

- Magnitude: 5.6861
- p-value: 0.001615
- Direction: Positive

The positive coefficient indicates that areas with a higher proportion of non-UK-born residents are more likely to vote for Brexit. The relationship is statistically significant and suggests a possible perception or reality of economic or cultural impacts influencing voting behavior.

medianIncome:

- Magnitude: -6.3857
- p-value: 0.000891
- Direction: Negative

The negative coefficient suggests that higher median incomes are associated with a lower likelihood of voting for Brexit. This result is statistically significant, indicating that wealthier areas might favor remaining in the EU, possibly due to economic ties or perceived benefits of EU membership.

medianAge:

- Magnitude: 5.9209
- p-value: 2.56e-05
- Direction: Positive

The positive and significant coefficient for median age suggests that older populations were more likely to vote for Brexit. This aligns with many analyses suggesting that older voters felt more positively about Brexit due to nostalgic or sovereignty-related reasons.

withHigherEd:

- Magnitude: -26.7443
- p-value: 7.52e-14
- Direction: Negative

This is the strongest coefficient in magnitude but negative, indicating a very strong tendency for areas with higher proportions of university-educated residents to vote against Brexit. This effect is highly significant statistically and supports the narrative that more educated populations were less inclined towards Brexit, perhaps due to differing views on the economic and cultural impacts.

Higher education has the most significant mitigating effect against leaving the EU, while higher socio-economic status (abc1) shows a strong propensity towards voting to Leave.

Effect Strength and Order:

Based on the coefficients and their significance:

People with withHigherEd have stronger effect and below is the order of decreasing effect:

withHigherEd: Most substantial negative effect on the likelihood of voting to Leave.

abc1: Strong positive effect, indicating a higher likelihood of Leave with higher socio-economic status.

medianAge: Also a significant positive effect.

medianIncome: Significant negative effect.

notBornUK: Significant positive effect.

```r
effect_strength <- abs(model_summary$coefficients[, "Estimate"])
effect_order <- sort(effect_strength, decreasing = TRUE)
print(effect_order)
```

```
## withHigherEd         abc1 medianIncome    medianAge    notBornUK  (Intercept)
##   26.7442592   17.5779980    6.3857396    5.9208767    5.6861383    0.1385963
```

The order of influence based on the absolute value of the coefficients would be:

1. withHigherEd
2. abc1
3. medianAge
4. medianIncome
5. notBornUK

Thus, $withHigherEd$ not only has the highest absolute coefficient value but also carries a significant weight in its influence on voting behavior, underscoring the potent role education plays in political and economic decisions such as Brexit.

Justification of Input Relevance:

The $p - values$ indicate statistical significance < 0.05 means statistically significant. This suggests that the model has found reliable patterns in the data regarding what factors influence voting behavior.The direction $positive/negative$ of each coefficient tells us how each demographic factor influences the decision to Leave or Remain.

```r
significant_inputs <- model_summary$coefficients[, "Pr(>|z|)"] < 0.05
print(significant_inputs)
```

```
##  (Intercept)         abc1    notBornUK medianIncome    medianAge withHigherEd
##        FALSE         TRUE         TRUE         TRUE         TRUE         TRUE
```

This indicates that these predictors have statistically significant relationships with the probability of an area voting for Brexit, under the typical significance level of 0.05. The significant p-values for these variables affirm that the logistic regression model has effectively captured meaningful and statistically significant patterns in the data, which are reliable indicators of voting behavior concerning Brexit. These insights are crucial for understanding the demographic underpinnings of political outcomes and can be used for more nuanced policy making and political analysis.

Comparison with Guardian Data and commenting:

Higher Education:

Guardian Data: Areas with higher percentages of residents with higher education seem more likely to vote to Remain, as depicted by the concentration of yellow points (Remain) at higher percentages.

Regression Model: The strong negative coefficient for withHigherEd aligns with this, as it indicates that higher education levels correlate with a lower likelihood of voting to Leave.

Consistency: The regression model supports the Guardian's visual data.

ABC1 Social Grade:

Guardian Data: The distribution appears to be mixed, but there may be a slight concentration of Leave votes (blue points) as the percentage of ABC1 residents increases.

Regression Model: The positive coefficient for abc1 also suggests that a higher social grade correlates with an increased likelihood of voting to Leave.

Consistency: There is some consistency, although the Guardian's data visualization is less clear on this trend.

Median Income:

Guardian Data: There does not seem to be a clear pattern in the scatter plot relating median income to Leave votes.

Regression Model: A significant negative effect of medianIncome suggests that higher incomes are associated with a tendency to Remain.

Consistency: The model's findings may seem counterintuitive compared to the Guardian's visualization, which does not show a strong visible correlation.

Median Age:

Guardian Data: Older median age areas show a trend towards voting to Leave, as indicated by the blue points skewed to the right.

Regression Model: The positive effect of medianAge agrees with this observation, indicating that areas with an older population were more likely to vote to Leave.

Consistency: The regression model is consistent with the visual trend observed in the Guardian's data.

Residents Not Born in the UK:

Guardian Data: The scatter plot seems to show that areas with more residents not born in the UK have a mixture of Remain and Leave votes without a clear distinction.

Regression Model: The positive coefficient for notBornUK is somewhat consistent with the Guardian's plot, suggesting these areas might lean towards Leave, although the visual data does not present a strong correlation.

Consistency: The consistency is ambiguous; the model indicates a clear effect, while the Guardian's visualization does not show a strong trend.

When comparing the coefficients and their associated significance from the logistic regression model with the plots from the Guardian, there is general agreement on the influence of higher education and median age on the tendency to vote Remain and Leave, respectively. The direction and magnitude of the effects identified in the model align well with these visual trends.

However, for median income and the proportion of non-UK-born residents, the Guardian's plots do not show as clear a trend as the logistic regression model suggests. This discrepancy might arise from the complexity of the scatter plots and the density of data points that can obscure clear trends. It may also reflect limitations in visual data interpretation compared to the mathematical precision of regression modeling.

In summary, the logistic regression analysis provides a statistically grounded confirmation of some trends visible in the Guardian's data, while also highlighting relationships that are not immediately apparent from the visualizations alone. This demonstrates the **added value of a logistic regression** approach in understanding the underlying factors that influenced the Brexit vote.

TASK 3:

Factors affecting interpretability of the regression coefficients of the fitted model :

There are many factors that could affect the interpretability of the regression coefficients in a model like Mulitiicollinearity, Data Scaling, Non-Linearity, Outliers, Overfitting and so on.

Considering the Brexit dataset below are the factors affecting the fitted model:

- **Demographic Correlation:**

In sociopolitical datasets like the one used for Brexit analysis, demographic variables such as age, income, education level, and class (represented by variables like abc1, medianIncome, medianAge, withHigherEd, etc.) are often correlated. For example, higher education levels might correlate with higher income, which can affect the interpretability of their individual coefficients in predicting voting behavior.

- **External Socio-Political Factors:**

The impact of unmeasured socio-political factors that could influence voting behavior (such as political campaigning, regional political sentiment, or recent economic events) might not be captured in the model, potentially confounding the effects of the included variables.

- **Voting Behavior Complexity:**

The decision to vote for or against Brexit is complex and may not be fully explained by the variables included. There may be nonlinear relationships or higher-order interactions between variables that the logistic regression model cannot capture.

- **Data Representation:**

If the data does not accurately represent the broader population due to sampling bias or if there are missing data points for certain demographics, this could skew the regression coefficients and affect their interpretability.

- **Causal Inference:**

While logistic regression can indicate associations, it does not imply causation. Interpreting the coefficients as causal effects without proper causal inference methods could lead to incorrect conclusions.

- **Temporal Dynamics:**

The data is cross-sectional, related to a single event in time. The interpretation of coefficients does not take into account how demographic influences on voting behavior might evolve over time or in response to future events.

- **Economic Indicators:**

The variable medianIncome is a broad economic indicator and may not capture the full economic context or sentiment that could influence voting decisions, such as economic inequality or perception of economic stability.

- **Cultural and Local Context:**

The variable notBornUK accounts for the birthplace but may not fully encapsulate cultural differences, integration levels, or local community effects, which could be significant in the context of Brexit.

Reliability of Input Relevance:

Based on the discussion above an approach is planned to assess reliability:

Data is to be standardized to ensure that the scale of the variables does not affect interpretation. Multicollinearity will be checked by calculating the Variance Inflation Factor (VIF).

```r
brexit_data_standardized <- as.data.frame(scale(brexit_data[, c("abc1", "notBornUK",
                                                           "medianIncome","medianAge","withHigherE
brexit_data_standardized$voteBrexit <- brexit_data$voteBrexit
library(car)
vif_result <- vif(glm(voteBrexit ~ ., data = brexit_data_standardized, family = binomial))
print(vif_result)
```

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

The VIF provides a measure of the increase in the variance of the coefficient estimates due to collinearity, with higher values indicating a greater degree of multicollinearity.

- **abc1:**

The VIF is approximately 9.99, which is close to or above the common threshold of 10 which is used to indicate serious multicollinearity. This suggests that abc1 may be linearly dependent on other variables, which could affect the interpretability and stability of its coefficient.

- **notBornUK, medianIncome, medianAge:**

The VIF's for these variables are above 3 but well below 10, which typically suggests moderate multicollinearity. It may not be severe enough to necessitate corrective measures, but it's something to be mindful of.

- **withHigherEd:**

The VIF is approximately 8.56, indicating a significant but not critical level of multicollinearity. This suggests some caution should be taken when interpreting the coefficient for this variable as well.

The high VIF for $abc1$ and $withHigherEd$ suggests that these variables have a linear relationship with other variables, cautioning against over-interpreting their coefficients. However, the standardized data allows for better comparison of coefficient magnitudes. While $abc1$ and $withHigherEd$ are important predictors, their coefficients may be inflated due to multicollinearity. Other variables seem to have more reliable coefficients. The ordering of variables should consider both the VIF and the magnitude of coefficients, but a high VIF may indicate a lower ranking due to reliability issues.

ORDER

To improve model interpretability either removing or combining variables to reduce multicollinearity can be done. Use penalized regression methods like ridge regression, LASSO. It's essential to validate the model with a holdout set or through cross-validation to ensure that the predictors perform well on unseen data. The VIF results indicate the presence of multicollinearity, which is a vital factor to consider in the interpretability and reliability of regression coefficients.

BAGGING for Logistic Regression:

For BAGGING (Bootstrap Aggregating), below steps will be performed:

creating bootstrap samples of our dataset and fitting a logistic regression model to each sample. Aggregating the results to understand the variability of our coefficients.

```r
library(boot)
logit_bagging <- function(data, indices) {
  d <- data[indices, ]
  fit <- glm(voteBrexit ~ abc1 + notBornUK + medianIncome + medianAge + withHigherEd,
             data = d, family = binomial)
  return(coef(fit))
}
set.seed(123)
boot_results <- boot(data = brexit_data, statistic = logit_bagging, R = 1000)
print(boot_results)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = brexit_data, statistic = logit_bagging, R = 1000)
##
##
## Bootstrap Statistics :
##        original      bias    std. error
## t1*  -0.1385963  0.01674530   0.7515388
## t2*  17.5779980  0.54171076   2.8493133
## t3*   5.6861383  0.09986501   1.8003584
## t4*  -6.3857396 -0.33180358   1.9668244
## t5*   5.9208767  0.08720239   1.3532105
## t6* -26.7442592 -0.62791549   3.8949848
```

The use of BAGGING (Bootstrap Aggregating) in logistic regression is a robust technique to assess the stability and variability of the regression coefficients. By fitting the model 1000 times with bootstrap dataset, the model's sensitivity to variation in sample data can be analysed.

Analysis of Bootstrap Results:

Each coefficient denoted as $t1$ through $t6$ has an "original" value the coefficient estimate from the full data, a *bias* the average difference between the bootstrap estimates and the original estimate, and a *std.error* the standard deviation of the bootstrap estimates.

- **Intercept (t1):**

$Original$: -0.1386, $Bias$: 0.0167, $Std.Error$: 0.7515, The bias is relatively small compared to the standard error, indicating the intercept's variability is quite large relative to its magnitude, which suggests low reliability.

- **abc1 (t2):**

$Original$: 17.5780, $Bias$: 0.5417, $Std.Error$: 2.8493, Despite the sizable standard error, the relative magnitude of the coefficient compared to its standard error and bias suggests that the effect of abc1 is robust and reliable.

- **notBornUK (t3):**

$Original$: 5.6861, $Bias$: 0.09997, $Std.Error$: 1.8004, The standard error is relatively small compared to the coefficient's magnitude, indicating moderate reliability and stability.

- **medianIncome (t4):**

$Original$: -6.3857, $Bias$: -0.3318, $Std.Error$: 1.9668, The negative bias and the significant size of the standard error relative to the coefficient value indicate some variability, suggesting that this coefficient's influence might be sensitive to data changes.

- **medianAge (t5):**

$Original$: 5.9209, $Bias$: 0.0872, $Std.Error$: 1.3532, The relatively small standard error compared to the coefficient's magnitude suggests that the age effect is quite stable and reliable.

- **withHigherEd (t6):**

$Original$: -26.7443, $Bias$: -0.6279, $Std.Error$: 3.8949, Despite the large negative coefficient, the substantial standard error indicates significant variability, yet the effect remains clearly strong and impactful, confirming its relevance but suggesting caution due to variability.

Ordering:

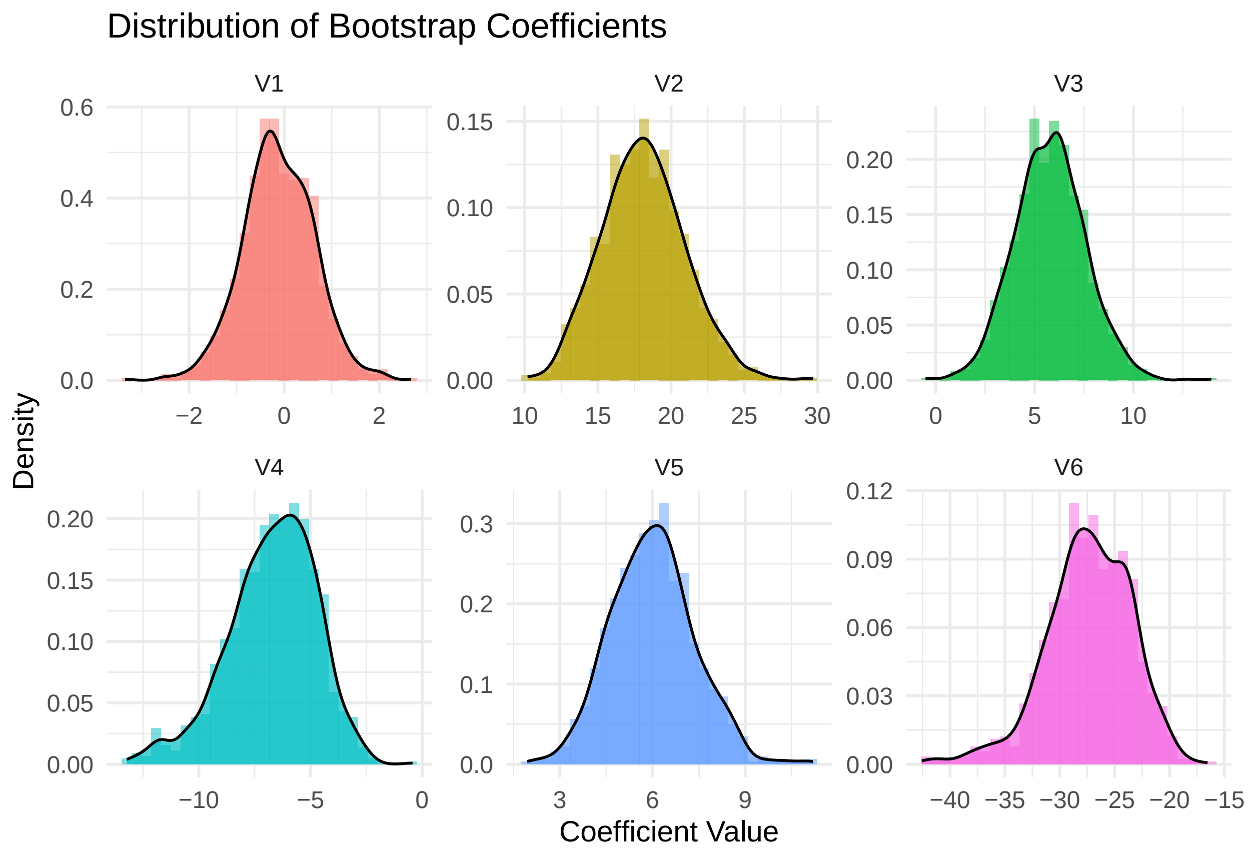
```r
boot_coefs <- boot_results$t
std_errors <- apply(boot_coefs, 2, sd)
coef_data <- data.frame(
  Variable = c("Intercept", "abc1", "notBornUK", "medianIncome", "medianAge", "withHigherEd"),
  OriginalCoefficient = boot_results$t0,
  StdError = std_errors
)
coef_data$InfluenceScore = abs(coef_data$OriginalCoefficient) / coef_data$StdError
coef_data <- coef_data[order(coef_data$InfluenceScore, decreasing = TRUE), ]
print(coef_data[, c("Variable", "OriginalCoefficient", "StdError", "InfluenceScore")])
```

```
##                  Variable OriginalCoefficient  StdError InfluenceScore
## withHigherEd withHigherEd         -26.7442592 3.8949848      6.8663321
## abc1                 abc1          17.5779980 2.8493133      6.1692051
## medianAge       medianAge           5.9208767 1.3532105      4.3754292
## medianIncome medianIncome          -6.3857396 1.9668244      3.2467259
## notBornUK       notBornUK           5.6861383 1.8003584      3.1583369
## (Intercept)     Intercept          -0.1385963 0.7515388      0.1844166
```

The calculation of influence scores helps in determining not only the direction but also the relative magnitude of the impact that each predictor has on the outcome, adjusting for the variability observed across different bootstrap samples. This approach enhances the reliability of the interpretation by ensuring that the effects are consistent across various subsamples of the data, thus mitigating some of the concerns around overfitting or specific sample biases

```r
boot_coefs <- boot_results$t
library(ggplot2)
library(tidyr)
coef_df <- as.data.frame(boot_coefs)
coef_df$sample <- seq_len(nrow(coef_df))
long_coef_df <- gather(coef_df, key = "coefficient", value = "value", -sample)

ggplot(long_coef_df, aes(x = value, fill = coefficient)) +
  geom_histogram(aes(y = ..density..), alpha = 0.5, bins = 30) +
  geom_density(alpha = 0.7) +
  facet_wrap(~ coefficient, scales = "free") +
  theme_minimal() +
  labs(title = "Distribution of Bootstrap Coefficients", x = "Coefficient Value", y = "Density") +
  theme(legend.position = "none")
```



Supporting my answer:

The BAGGING analysis successfully addresses Task 3 by:

Demonstrating Variability:

It provides a quantitative measure of how much each coefficient might vary if the model were fitted to different samples from the same population. This is crucial for understanding the reliability of the model's predictions.

Assessing Coefficient Stability:

The standard errors and biases provide insights into which variables are consistently influential across different samples and which might be more dependent on specific data configurations.

Supporting Relevance Determination:

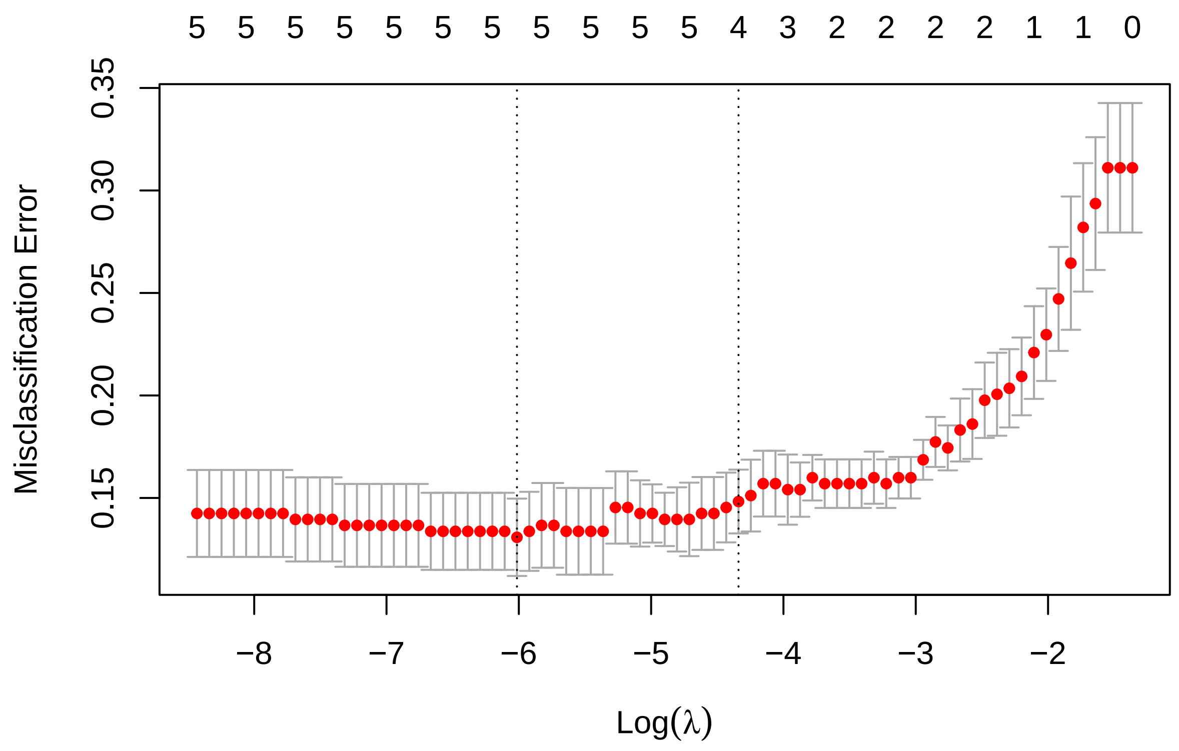
The results help in confidently ranking the inputs based on their stability and effect magnitude. Variables like $abc1$ and $withHigherEd$, despite their variability, show strong effects, suggesting they are highly relevant. Meanwhile, the intercept shows high variability relative to its effect size, indicating less reliability.

In conclusion, the bootstrap analysis using BAGGING provides robust support for understanding and validating the logistic regression model's coefficients. It ensures that conclusions about the model are not only based on a single data set but are likely to hold across similar datasets, enhancing the reliability of the findings.

Task 4:

Lasso (Least Absolute Shrinkage and Selection Operator) regression is particularly useful in scenarios where simplification and variable selection are necessary. It adds a penalty equal to the absolute value of the magnitude of coefficients to the loss function, which helps in reducing overfitting and selecting more relevant predictors by shrinking coefficients for less important variables exactly to zero.

```r
if (!require(glmnet)) {
  install.packages("glmnet")
  library(glmnet)
}
x <- as.matrix(brexit_data[, c("abc1", "notBornUK", "medianIncome", "medianAge", "withHigherEd")])
y <- brexit_data$voteBrexit
lasso_model <- glmnet(x, y, family = "binomial", alpha = 1)
cv_lasso <- cv.glmnet(x, y, family = "binomial", type.measure = "class", alpha = 1)
plot(cv_lasso)
```



```r
best_lambda <- cv_lasso$lambda.min
best_lasso_model <- glmnet(x, y, family = "binomial", alpha = 1, lambda = best_lambda)
coef(best_lasso_model)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                     s0
## (Intercept)    0.516346
## abc1          14.041619
## notBornUK      3.835728
## medianIncome  -4.872059
## medianAge      4.595613
## withHigherEd -22.481409
```

The plot typically helps identify the lambda value that minimizes the misclassification error. This is often where you see a transition from a flat, low-error region to a rising error as lambda increases. In the plot, the optimal lambda appears to be around -5, where the misclassification error is at its minimum before it starts to increase with smaller lambda values.

- Model Complexity: As lambda decreases (moving left on the x-axis), the model becomes more complex because the regularization becomes weaker, allowing more variables to remain in the model. Conversely, as lambda increases, the penalty is stronger, leading to a simpler model with potentially fewer variables.
- Trade-off: The leftmost part of the plot shows a stable, low misclassification error, indicating that there might not be significant overfitting at these levels, despite the increased complexity. However, as lambda decreases further and the model includes more variables (more complex), the misclassification error begins to rise, signaling overfitting.

This graph is instrumental in tuning the regularization strength of logistic regression model. By selecting the lambda at the turning point of the curve -5,by balancing model complexity with predictive accuracy, avoiding both overfitting and underfitting. This lambda value can then be used to finalize my model, potentially leading to better generalization on unseen data.

Benefits:

1. Variable Selection:

Lasso helps in reducing the model complexity by performing automatic variable selection where insignificant input variables are penalized to zero, enhancing model interpretability.

2. Reduction of Overfitting:

By introducing a penalty term, Lasso can reduce overfitting, making the model more generalizable.

3. Handling Multicollinearity:

Lasso can manage multicollinearity better by including only one variable from a group of highly correlated variables, thus minimizing the variance inflation problem.

Disadvantages:

1. Bias:

Adding a penalty can introduce bias into the estimates, particularly if the penalty is large, which might lead to underfitting.

2. Selection of Lambda:

The choice of the regularization parameter, lambda, is crucial and can impact the model's performance significantly. Incorrect selection of lambda might lead to over or under penalization.

3. Scalability:

As the number of predictors increases, the computational complexity also increases, which might be computationally intensive for very large datasets.

As discussed in Task 3 firstly I thought ridge regression will be more suitable but, for the Brexit voting analysis, where understanding the influence of specific variables is crucial, Lasso's ability to eliminate non-influential variables provides clear, interpretable results that are valuable for decision-making and policy formulation. The choice between Lasso and Ridge often boils down to whether the primary goal is prediction accuracy where both might be suitable, and Ridge might edge out if all predictors are believed relevant or model simplicity and variable selection where Lasso has the clear advantage. Here's why Lasso might be more suitable for this context:

- Variable Selection:

Lasso regression is well-known for its ability to perform variable selection by shrinking less important coefficients to exactly zero. This feature of Lasso makes it particularly valuable when we want to identify which predictor variables are the most impactful in influencing the outcome, such as voting for Brexit. This property helps in reducing the model complexity and enhances interpretability, which is crucial for analyzing political or social data where explaining the influence of variables is as important as the prediction itself.

- Sparsity:

The ability of Lasso to produce sparse models, a model with fewer predictor variables is particularly useful in scenarios where we are dealing with datasets that might have variables that do not contribute much to the predictive power of the model. By eliminating these variables, Lasso helps focus on the most relevant predictors, improving model efficiency and clarity.

Conclusion:

Through the comprehensive analysis conducted in this assignment, successfully identified key demographic factors that influenced the Brexit vote, such as education level, income, age, and social class. The use of Lasso regression, in particular, allowed for a refined model that emphasized the most impactful predictors by penalizing less relevant ones, thereby simplifying the complexity inherent in the dataset. This approach not only clarified which variables were most influential but also provided a clear, interpretable model that could be easily communicated and understood. The findings not only aligned with intuitive expectations, highlighting the role of education and economic status in political decisions. But also offered a nuanced view of how these factors interacted in different regions. Ultimately, this assignment not only shed light on the Brexit voting patterns but also demonstrated the power of advanced statistical techniques in dissecting complex socio-political datasets, offering valuable insights for policymakers, researchers, and the general public interested in understanding and addressing the underlying currents of political behavior.

References:

1. Lecture and Practicals notes.
2. https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/
3. https://www.linkedin.com/advice/0/how-can-you-interpret-coefficients-logistic-regression-t53gf#:~:text=The%20sign
4. https://medium.com/@mystery0116/interpret-the-impact-size-with-logistic-regression-coefficients-5eec21baaac8
5. https://www.analyticsvidhya.com/blog/2016/01/ridge-lasso-regression-python-complete-tutorial/
6. http://www.sthda.com/english/articles/36-classification-methods-essentials/149-penalized-logistic-regression-essentials-in-r-ridge-lasso-and-elastic-net/
7. https://www.datacamp.com/tutorial/tutorial-ridge-lasso-elastic-net
8. https://towardsdatascience.com/silhouette-method-better-than-elbow-method-to-find-optimal-clusters-378d62ff6891