65/100

BUSN 41201- Big Data

Spring Quarter 2024 – Veronika Rockova

Analysis of Socio-Economic Perceptions in Germany

26 May 2024

By Alec Zhang, Junhan Fu, Coco Qu, Yu Guo

Honor Code

We pledge out honor that we have not violated the Booth Honor Code during this assignment.

BUS 41201 Final Project

Alec Zhang, Junhan Fu, Coco Qu, Yu Guo

May 26, 2024

1. Executive Summary

Between 2005 and 2019, 4,000 individuals were surveyed periodically about various aspects of their lives, ranging from financial conditions to political affiliations and family dynamics. The dataset comprises 17,398 observations and 1,209 variables, encapsulating not only personal and family background details but also broader socio-economic indicators. This rich dataset allows us to delve into multiple dimensions of German society, seeking to understand how elements such as political engagement, health, and social class interrelate.

Our study is structured around three key questions: 1) Which features most significantly influence individuals' self-assessed health status; 2) How the features from aspects political engagement, socioeconomic self-assessment, and perceptions of economic and social trends affects individuals' current economic situation; 3) What factors most significantly influence self-assessed social class. Because self-assessed social class is influenced by long-term factors such as education and family background, while current economic situation reflects immediate financial conditions. This analysis explores how both are shaped by different socio-economic variables, highlighting their unique and shared influences.

To facilitate our analysis, we used a wide range of machine learning models from supervised learning like logistic regression and tree models to unsupervised learning like PCA. By addressing these questions, we aim to uncover the complex interplay factors influencing various aspects of life in Germany.

2. Introduction to Dataset and Analysis

```
##            Do you need all this printed output in a report?
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

## corrplot 0.92 loaded
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: lattice

## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)

## Rows: 17398 Columns: 1209
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (1164): uniqueid, year, personid, x1, x2, x3, x4, x5, x6, x7, x8, x9, x1...
## lgl   (45): x96, x97, x98, x435, x637, x811, x812, x815, x816, x817, x818, x...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Between 2005 and 2019, 4000 individuals were periodically surveyed about their lives, with questions ranging from their financial situation to political affiliation to the number of children in their household. Participants were also asked to rate their current health status on the following scale:

- 1 VERY GOOD
- 2 GOOD
- 3 SATISFACTORY
- 4 NOT THAT GOOD
- 5 BAD

Each row of the data corresponds to a survey response from an individual between 2005 and 2019. The data is sourced from a Kaggle Competition.

Here are some basic properties about the dataset:

```
## Dimensions:
##  17398 1209

## Missing Values by Column (Percentage) for the First 40 Columns:

##  [1] "0%"     "0%"     "0%"     "0.5%"   "0.16%"  "1.76%"  "1.58%"  "84.13%"
##  [9] "75.98%" "7.32%"  "86.16%" "86.3%"  "86.07%" "86.81%" "85.8%"  "86.48%"
## [17] "0.03%"  "84.06%" "91.8%"  "21.03%" "87.72%" "10.65%" "55.35%" "96.56%"
## [25] "85.42%" "85.48%" "85.6%"  "85.49%" "85.49%" "83.85%" "83.85%" "83.85%"
## [33] "83.85%" "83.85%" "83.85%" "83.85%" "83.85%" "83.85%" "83.85%" "83.85%"

##
## Total Missing Percentage for the Entire Dataset: 72.71%
```

Since the number of columns are too many, we only display the missing value percentage for the first 40 columns.

Our goal is to build models including series of regression models, and tree models to predict participant's health status (health), current economic situation in Germany (x1), and their self-assessed social classes (x163). These models should also provide insight about which aspects of the social-economical, biological, political factors impact the response variables the most.

As we see above, the dataset is very large with more than 17000 records, more than 1200 features with a mixture of categorical and numerical values, and a large percentage of missing values. These increase the difficulty of our analysis and might result in inaccurate predictions.

Some Variables' Descriptions

health

Type: Categorical (Numeric byte)
Description: Respondent's current health status
Missing Values: 0 out of 24,840

x1 These summaries give me no information about actual distributions!

Type: Categorical (Numeric byte)
Description: Respondent's perception of the current economic situation in Germany.
Missing Values: 104 out of 24,840

x111

Type: Categorical (Numeric byte)
Description: Level of trust in the health service.
Missing Values: 14,458 out of 24,840

x112

Type: Categorical (Numeric byte)
Description: Level of trust in the Federal Constitutional Court.
Missing Values: 15,151 out of 24,840

x118

Type: Categorical (Numeric byte)
Description: Level of trust in television.
Missing Values: 14,515 out of 24,840

x119

Type: Categorical (Numeric byte)
Description: Level of trust in newspapers.
Missing Values: 14,654 out of 24,840

x120

Type: Categorical (Numeric byte)
Description: Level of trust in universities and higher education.
Missing Values: 15,287 out of 24,840

x121

Type: Categorical (Numeric byte)
Description: Level of trust in the federal government.
Missing Values: 14,558 out of 24,840

x122

Type: Categorical (Numeric byte)
Description: Level of trust in the police.
Missing Values: 14,478 out of 24,840

x163

Type: Categorical (Numeric byte)
Description: Respondent's self-assessment of their social class.
Missing Values: 655 out of 24,840

x190

Type: Categorical (Numeric byte)
Description: Belief in whether success depends on one's own education.
Missing Values: 20,248 out of 24,840

x453

Type: Numeric (byte)
Description: Respondent's body height in centimeters.
Missing Values: 16,842 out of 24,840

x454

Type: Numeric (byte)
Description: Respondent's body weight in kilograms.
Missing Values: 17,020 out of 24,840

x760

Type: Numeric (byte)
Description: Age of the respondent's spouse.
Missing Values: 10,845 out of 24,840

x1180

Type: Categorical (Numeric byte)
Description: Social class of the respondent's household.
Missing Values: 16,130 out of 24,840

x1035

Type: Numeric (byte)
Description: Number of biological children the respondent has.
Missing Values: 316 out of 24,840

x633

Type: Numeric (byte)
Description: Age of the respondent.
Missing Values: 19,446 out of 24,840

x631

Type: Numeric (byte)
Description: Year of birth of the respondent.
Missing Values: 48 out of 24,840

x969

Type: Numeric (byte)
Description: Age of the second person in the household.
Missing Values: 5,458 out of 24,840

x639

Type: Categorical (Numeric byte)
Description: Respondent's general school leaving certificate.
Missing Values: 35 out of 24,840

x896

Type: Categorical (Numeric byte)
Description: General school leaving certificate of the respondent's father.
Missing Values: 2,609 out of 24,840

x943

Type: Categorical (Numeric byte)
Description: Educational level of the respondent's mother according to ISCED 1997.
Missing Values: 1,226 out of 24,840

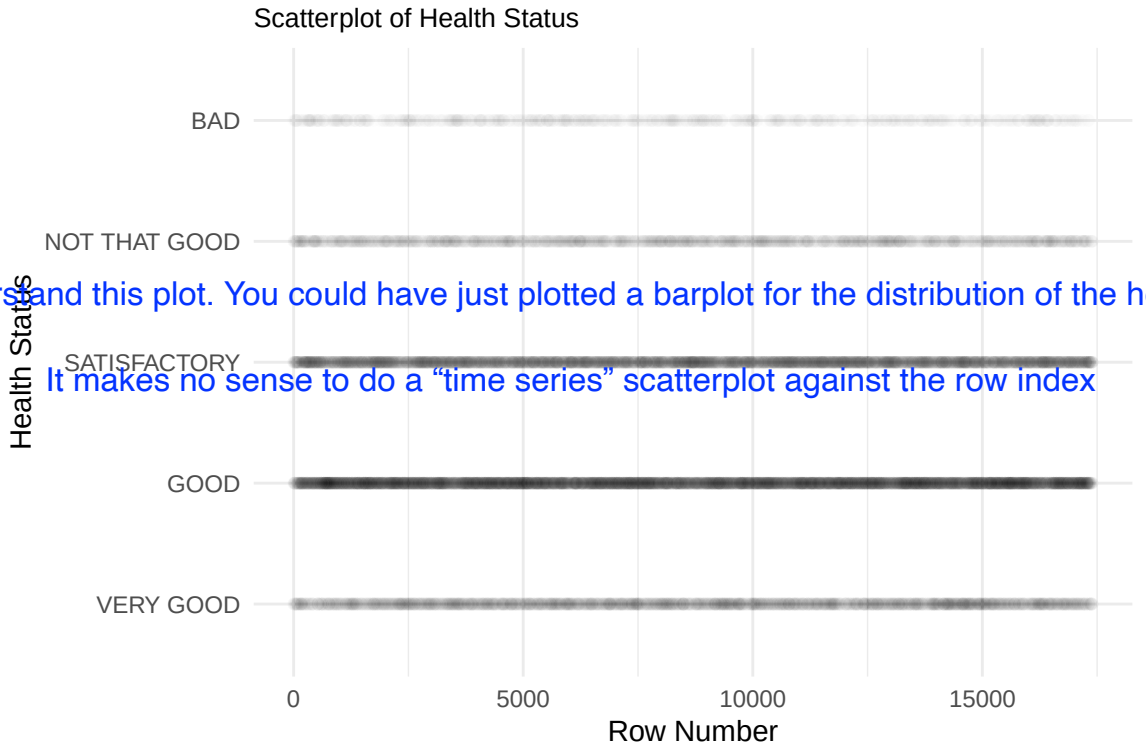
3. Exploratory Data Analysis (EDA)

3.1 EDA of potential Y variables

A. First potential Y: health status

The scatterplot displays individual data points where the x-axis represents the row number (essentially the order of the observations in the dataset) and the y-axis represents the health status. Each point on the scatterplot corresponds to a unique observation in the dataset, with the transparency set to 50% (alpha = 0.5) to help visualize overlapping points.

The scatterplot helps us see the overall distribution of health status values. We can identify how health statuses are spread throughout the dataset and if there are any clusters or gaps.

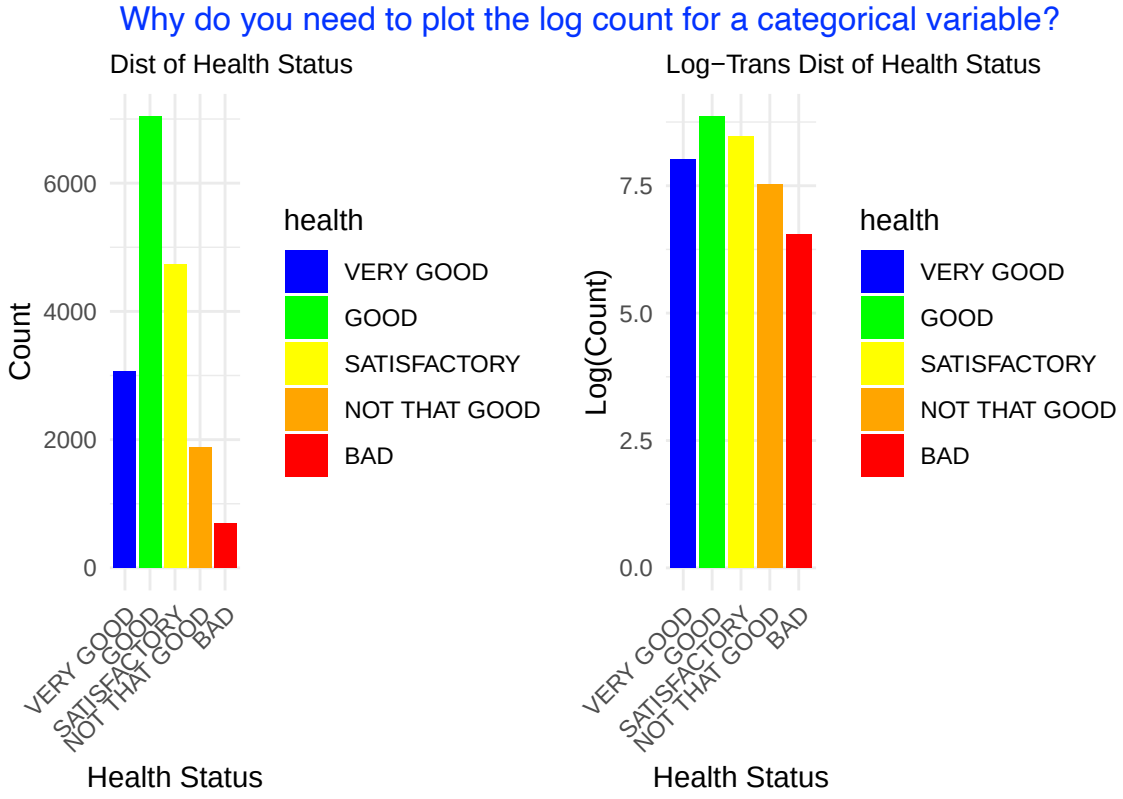


I don't understand this plot. You could have just plotted a barplot for the distribution of the health status.

It makes no sense to do a "time series" scatterplot against the row index

- Density and Overlap: There is a high density of points for "GOOD" and "SATISFACTORY" health statuses. This suggests that a large portion of the dataset reports these health status levels. The points for "VERY GOOD", "NOT THAT GOOD", and "BAD" are less dense, indicating fewer observations in these categories.
- Lack of Clear Trend: Since the row number is not an intrinsic variable but rather an ordering of the data, there is no discernible trend in health status over row numbers. The health statuses are distributed fairly uniformly across the row numbers. This uniform distribution implies that there is no inherent ordering or time-related pattern in the health status data based on row numbers.
- Potential Data Imbalance: The higher concentration of points in the "GOOD" and "VERY GOOD" categories compared to the other categories might indicate a potential imbalance in the dataset. This could be important for further analysis, especially if health status is used as a target variable in predictive modeling.

We now plot the distribution of health status in the dataset using two different visualizations: a regular histogram and a log-transformed histogram. These visualizations help us understand the distribution and skewness of the health status variable.



Distribution of Health Status (Regular Histogram)

- Right Skew: The original histogram shows a right-skewed distribution, with the majority of observations falling into the "GOOD" and "SATISFACTORY" categories. This skewness indicates that fewer individuals reported their health status as "BAD" or "NOT THAT GOOD", while a large number of individuals reported "GOOD" health status.
- Imbalance in Categories: The "GOOD" category has the highest count, followed by "SATISFACTORY". The "VERY GOOD" category also has a significant number of observations. The "NOT THAT GOOD" and "BAD" categories have much lower counts, indicating an imbalance in the dataset.

Log-Transformed Distribution of Health Status

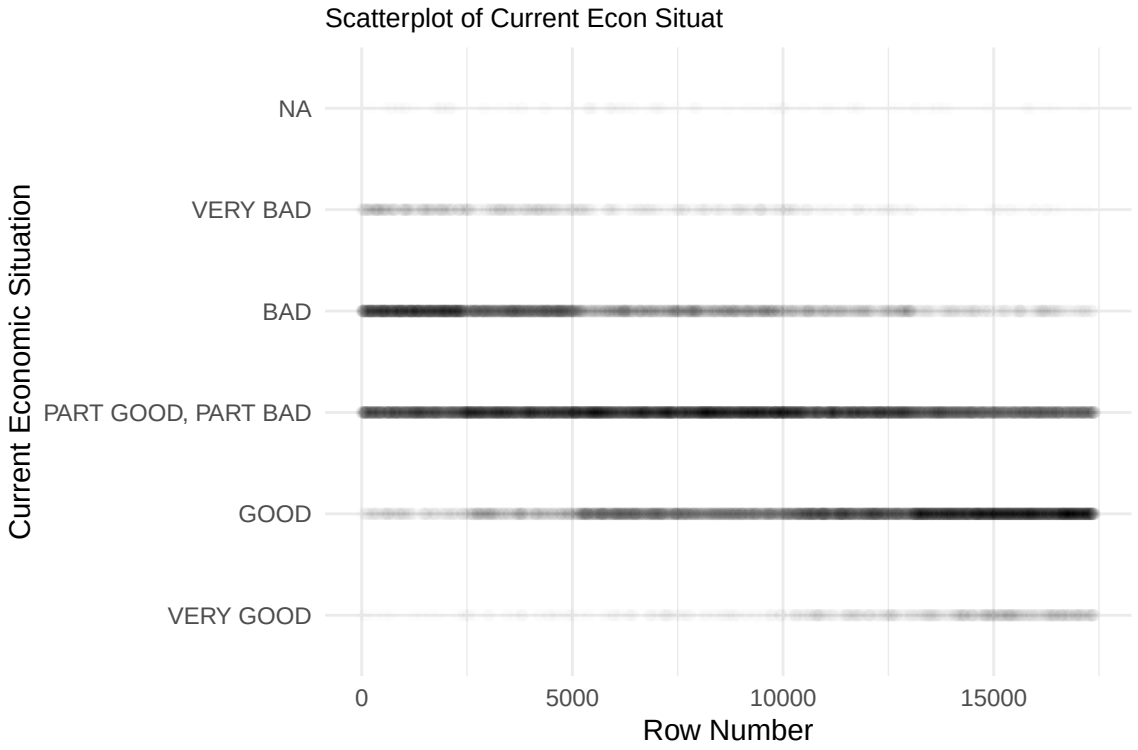
- Gaussian-like Distribution: Applying a log transformation to the counts helps normalize the distribution, making it look more Gaussian (bell-shaped). This transformation reduces the impact of the right skew and provides a clearer view of the relative differences between categories.
- Better Visualization of Counts: The log-transformed histogram provides a better visualization of the lower counts in the "NOT THAT GOOD" and "BAD" categories, which are less pronounced in the regular histogram due to the skewness. The relative differences between the categories become more apparent after the log transformation.

The regular histogram reveals a right-skewed distribution of health status, with a large number of observations reporting "GOOD" and "SATISFACTORY" health statuses, and fewer observations in the "NOT THAT GOOD" and "BAD" categories. The log-transformed histogram normalizes the distribution, providing a Gaussian-like shape and making the differences between categories clearer.

B. Second potential Y: CURRENT ECONOMIC SITUATION IN GERMANY

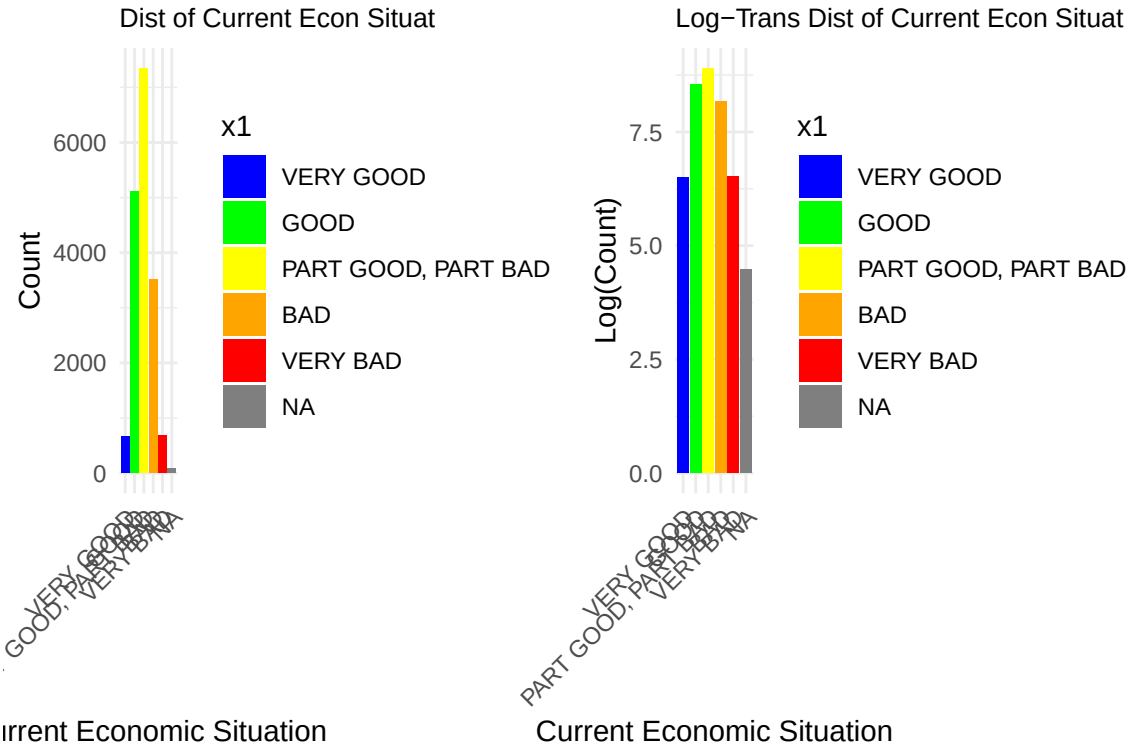
The scatterplot displays individual data points where the x-axis represents the row number (essentially the order of the observations in the dataset) and the y-axis represents the current economic situation in Germany. Each point on the scatterplot corresponds to a unique observation in the dataset, with the transparency set to 0.9% (alpha = 0.009) to help visualize overlapping points.

The scatterplot helps us see the overall distribution of the current economic situation values. We can identify how economic situation statuses are spread throughout the dataset and if there are any clusters or gaps.



- The categories "VERY GOOD," "GOOD," "PART GOOD, PART BAD," and "BAD" have the highest frequency, indicated by the dense clustering of points.
- There are fewer observations in the "VERY BAD" and "NA" categories.
- The spread of points within each category appears consistent, indicating that these assessments are distributed relatively evenly across the dataset without significant clustering in specific segments.

We now plot the distribution of the current economic situation in Germany in the dataset using two different visualizations: a regular histogram and a log-transformed histogram. These visualizations help us understand the distribution and skewness of the current economic situation variable.

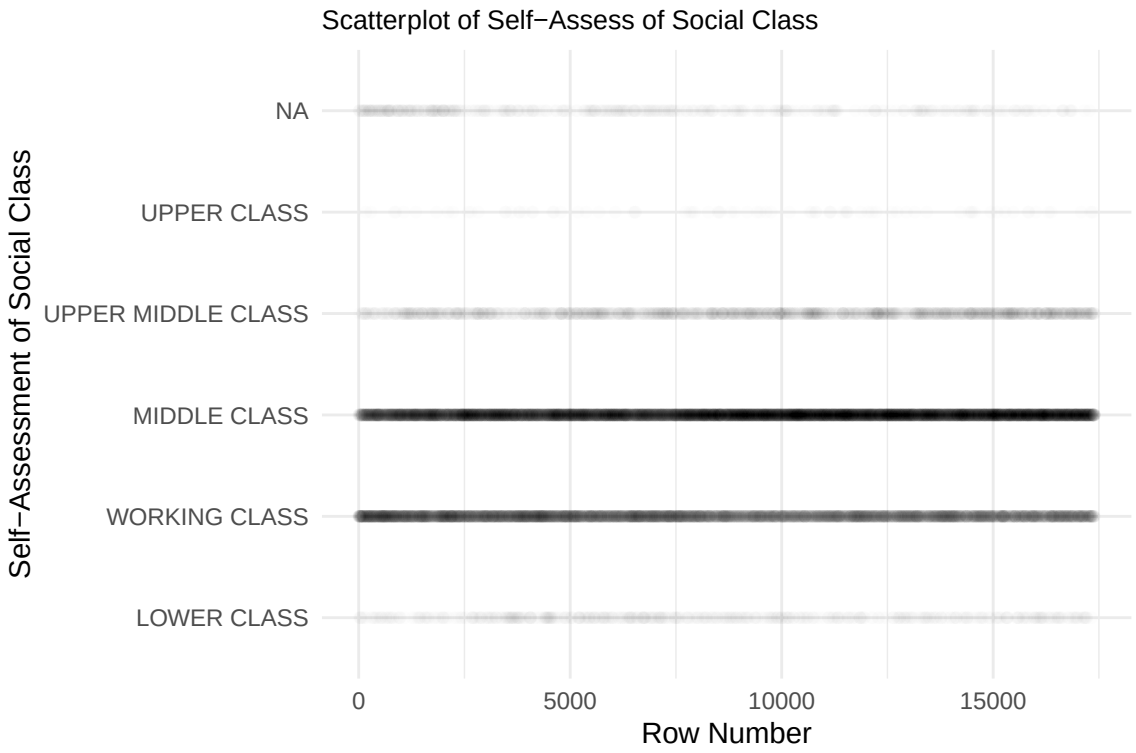


The left histogram shows the count of observations in each category without any transformation. We observe that the majority of respondents fall into the "PART GOOD, PART BAD" and "GOOD" categories, with significantly fewer respondents in the "VERY GOOD," "BAD," "VERY BAD," and "NA" categories. This indicates a skewed distribution towards the middle categories.

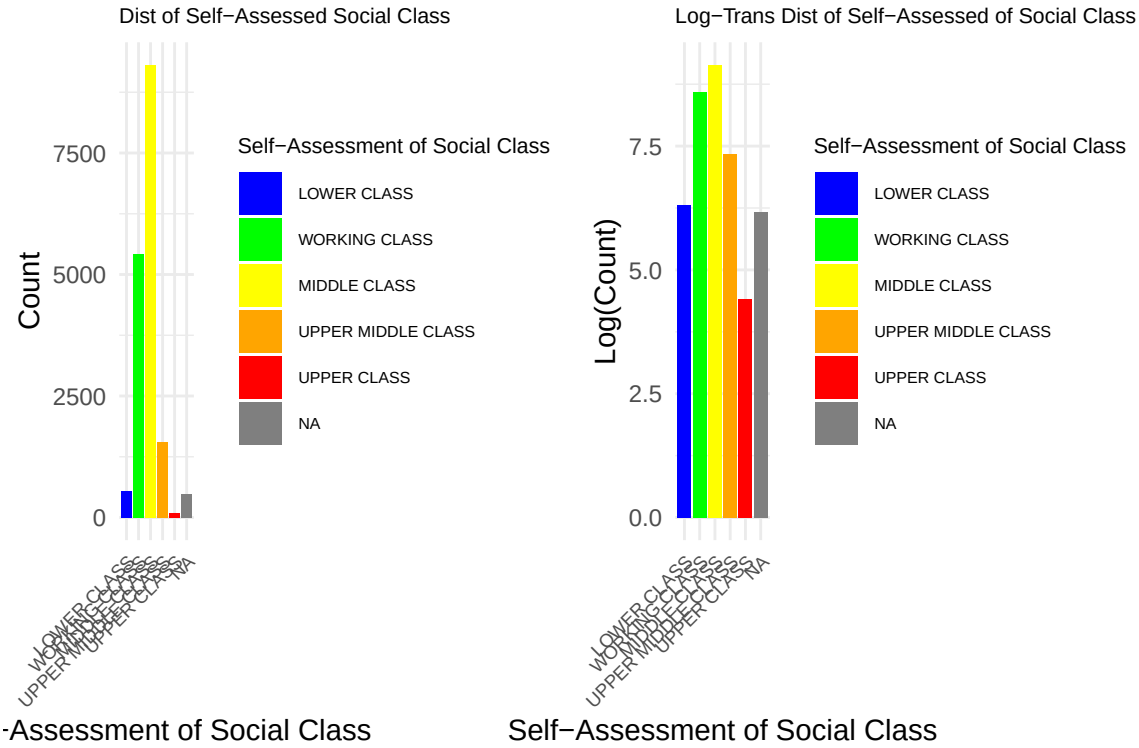
The right histogram presents a log-transformed view of the same data. This transformation helps to normalize the skewness observed in the raw counts, making the distribution appear more balanced and allowing us to better compare the relative sizes of each category. The log-transformation highlights the substantial differences in the frequencies of the categories, especially emphasizing the lower counts in the "VERY GOOD," "BAD," "VERY BAD," and "NA" categories.

C. Third potential Y: SELF-ASSESSMENT OF SOCIAL CLASS

We now scatterplot self-assessed social class versus observation index:



From the scatterplot, we observe that the majority of respondents classify themselves as "MIDDLE CLASS" or "WORKING CLASS." There are fewer observations in the "UPPER MIDDLE CLASS," "UPPER CLASS," and "LOWER CLASS" categories, and some data points are marked as "NA" (missing values). This distribution indicates that most respondents perceive themselves as belonging to the middle or working class, with fewer identifying as upper or lower class.



Regular Histogram:

The majority of respondents classify themselves as "MIDDLE CLASS" or "WORKING CLASS." A smaller number of respondents identify as "LOWER CLASS," "UPPER MIDDLE CLASS," or "UPPER CLASS." There are some missing values represented as "NA."

Log-Transformed Histogram:

The log-transformed histogram helps in visualizing the distribution more clearly, especially for categories with smaller counts. After log transformation, the "MIDDLE CLASS" and "WORKING CLASS" categories still dominate the distribution. The other categories, including "LOWER CLASS," "UPPER MIDDLE CLASS," and "UPPER CLASS," have visibly smaller log-transformed counts.

These visualizations indicate that most respondents perceive themselves as belonging to the middle or working class, with fewer identifying as upper or lower class. The log-transformed histogram effectively emphasizes the differences between categories with smaller frequencies, making it easier to compare their distributions.

We now examine the correlations between health, the current economic situation in Germany, and self-assessment of social class, which might help us understand if these variables are influenced by the same underlying common factors.

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 17398 Columns: 1209
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (1164): uniqueid, year, personid, x1, x2, x3, x4, x5, x6, x7, x8, x9, x1...
## lgl   (45): x96, x97, x98, x435, x637, x811, x812, x815, x816, x817, x818, x...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 87 rows containing missing values
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 474 rows containing missing values
```

```
## Warning: Removed 87 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 87 rows containing non-finite outside the scale range
## (`stat_density()`).
```

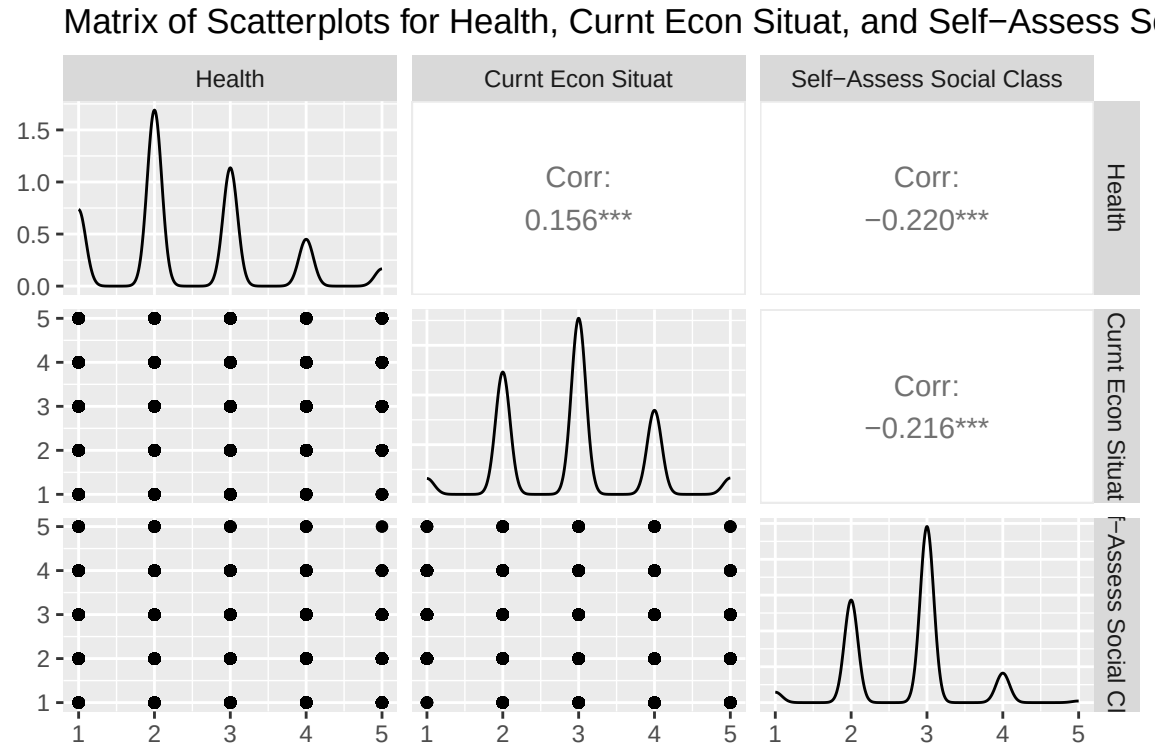
```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 548 rows containing missing values
```

```
## Warning: Removed 474 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

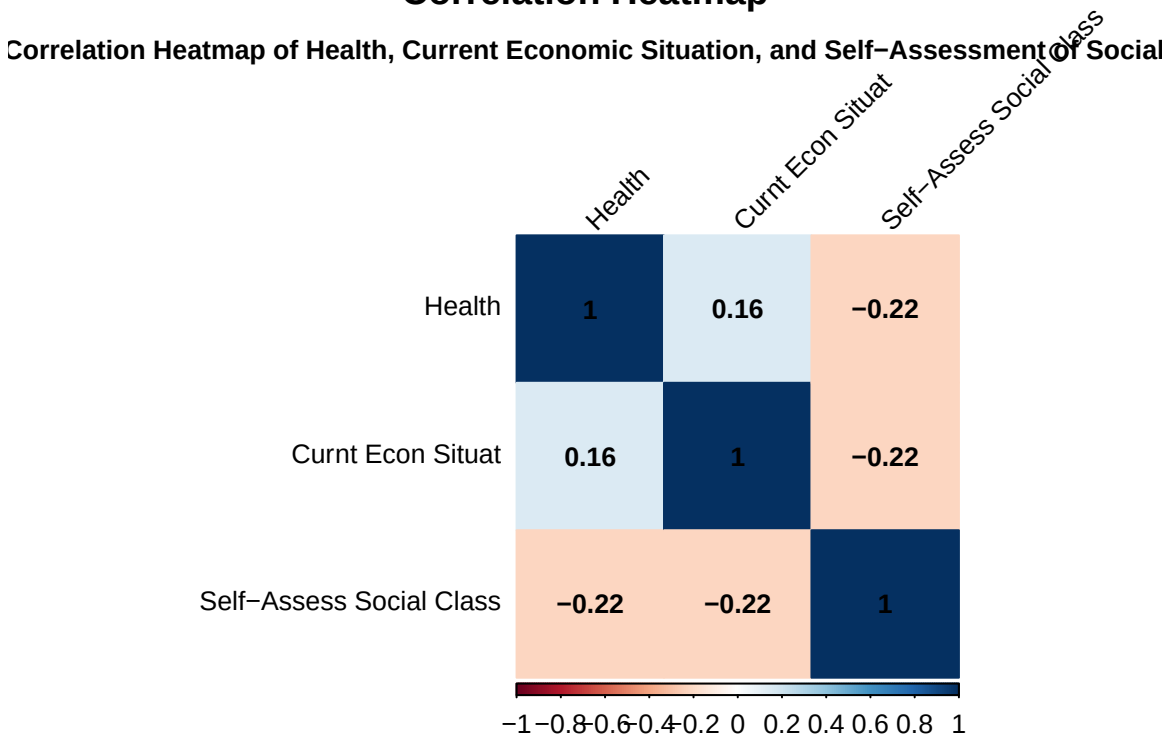
```
## Warning: Removed 548 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 474 rows containing non-finite outside the scale range
## (`stat_density()`).
```

you are looking at association between categorical variables. It makes no sense to do a scatterplot here. Also the correlations make no real sense. The pearson correlation coefficient is for two continuous variables.



Correlation Heatmap



The correlation between Health and Current Economic Situation is moderate and positive (0.16), suggesting a shared underlying factor where better health correlates with a better economic situation. The correlation between Health and Self-Assessment of Social Class is moderate and negative (-0.22), indicating that individuals with better health tend to perceive their social class as lower. The correlation between Current Economic Situation and Self-Assessment of Social Class is also moderate and negative (-0.22), showing a similar trend where a better economic situation is associated with a lower self-assessment of social class.

Overall, these analyses suggest that while there are some shared underlying factors affecting these variables, they do not move in the same direction, indicating the complexity of the relationships between health, economic perception, and social class assessment.

3.2 EDA of X variables

Let's pick some informative variables for EDA, so that we reduce the high-dimensional data into a few key features for exploration.

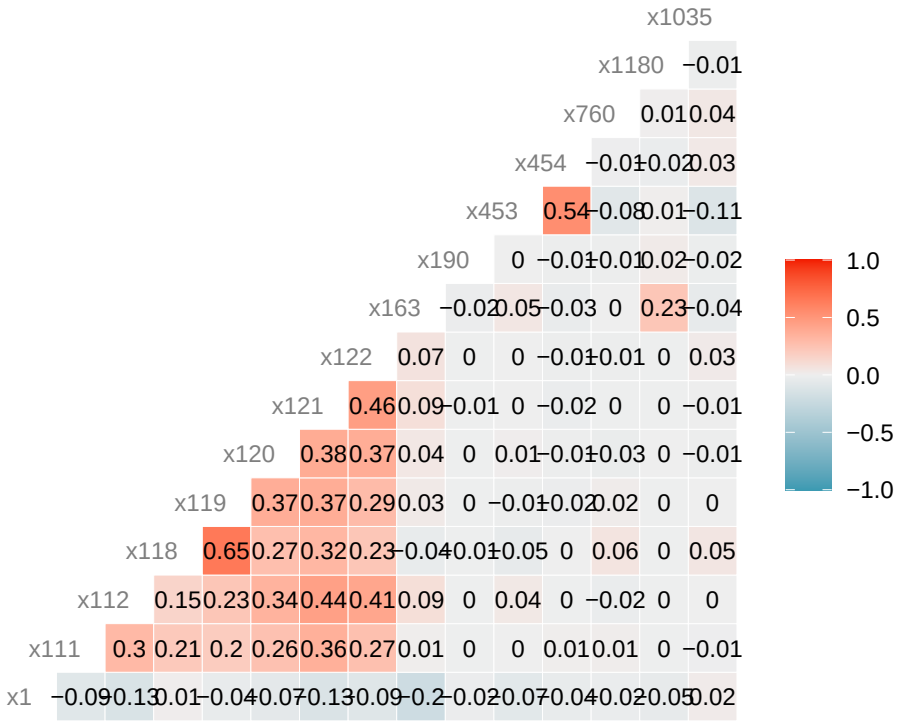
```
## x1 : Current Economic Situation in Germany
## x111 : Trust in Health Service
## x112 : Trust in Federal Constitutional Court
## x118 : Trust in Television
## x119 : Trust in Newspapers
## x120 : Trust in Universities, Higher Education
## x121 : Trust in Federal Government
## x122 : Trust in Police
## x163 : Self-Assessment of Social Class, Respondent
## x190 : Success: Depends on Own Education
## x453 : Body Height in Centimeters
## x454 : Body Weight in Kilograms
## x760 : Spouse's Age
## x1180 : Social Class of Household
## x1035 : Number of Biological Children
```

The X variables are chosen from aspects including Economical/Financial Situation, Social Class, Political Opinion, Social Acceptance, biological, family members, and education, which aims to depict a full picture of the participants.

A. Relationship between X variables

Let's check the correlations between X variables, since we need to examine multicollinearity assumption for running linear regressions later.

```
## tibble [17,398 x 15] (S3: tbl_df/tbl/data.frame)
##  $ x1   : num [1:17398] 1.79 1.39 1.61 1.39 1.61 ...
##  $ x111 : num [1:17398] 1.67 1.67 1.67 1.67 1.67 ...
##  $ x112 : num [1:17398] 1.77 1.77 1.77 1.77 1.77 ...
##  $ x118 : num [1:17398] 1.47 1.47 1.47 1.47 1.47 ...
##  $ x119 : num [1:17398] 1.56 1.56 1.56 1.56 1.56 ...
##  $ x120 : num [1:17398] 1.79 1.79 1.79 1.79 1.79 ...
##  $ x121 : num [1:17398] 1.56 1.56 1.56 1.56 1.56 ...
##  $ x122 : num [1:17398] 1.77 1.77 1.77 1.77 1.77 ...
##  $ x163 : num [1:17398] 1.1 1.1 1.1 1.39 1.1 ...
##  $ x190 : num [1:17398] 0.693 1.099 1.099 1.099 1.386 ...
##  $ x453 : num [1:17398] 5.1 5.11 5.08 5.13 5.13 ...
##  $ x454 : num [1:17398] 4.19 4.26 4.11 4.45 4.33 ...
##  $ x760 : num [1:17398] 4 4.19 4 4 4 ...
##  $ x1180: num [1:17398] 1.34 1.34 1.34 1.34 1.34 ...
##  $ x1035: num [1:17398] 0.693 1.609 1.099 1.099 1.099 ...
```



- a. We observe strong correlations among several variables**:
- x118 and x119 (0.65): There is a strong positive correlation between variables x118 (Trust in TV) and x119 (Trust in Newspapers). This suggests that individuals who trust TV also tend to trust newspapers.
- x120 and x121 (0.46): A moderately strong positive correlation exists between x120 (Trust in high Education) and x121 (Trust in Fed Govt), indicating that trust in higher education is associated with trust in the federal government.
- x121 and x122 (0.46): A similar positive correlation between x121 (Trust in Fed Govt) and x122 (Trust in Police) suggests that individuals who trust the federal government also tend to trust the police.
- x111 and x118 (0.3): There is a moderate positive correlation between x111 (Trust in Health Service) and x118 (Trust in TV).
- b. Some variables establish moderate correlations**:

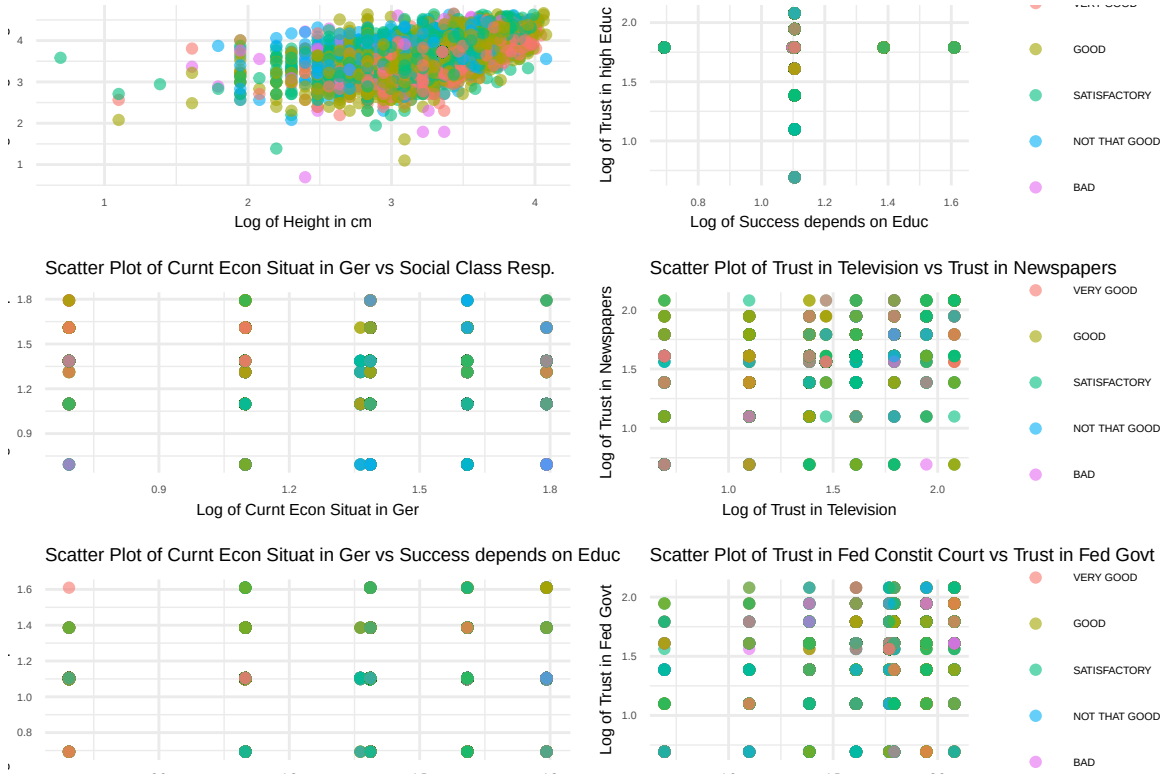
x111 and x112 (0.15): A positive correlation between x111 (Trust in Health Service) and x112 (Trust in Fed Constit Court). x118 and x119 (0.37): A positive correlation between x118 (Trust in TV) and x119 (Trust in Newspapers).

The correlation matrix reveals several pairs of variables with strong or moderate positive correlations, indicating potential multicollinearity issues. High multicollinearity can make it difficult to determine the individual effect of each predictor on the response variable, leading to unreliable statistical inferences. This indicates that methods such as variable selection, PCA, or regularization techniques are required later to help mitigate these issues.

Let's examine the relationships between various pairs of variables in the dataset, with each point colored based on health status.

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
```

```
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



- a. Height in cm vs Weight in kg (Log-transformed)**:

There is a noticeable positive correlation between height and weight, as expected. Taller individuals tend to weigh more. Points are spread across different health statuses, indicating no clear distinction based on health status in this relationship.

- b. Success Depends on Education vs Trust in High Education (Log-transformed)**:

The plot shows a clustering of points, indicating that those who believe success depends on education tend to also have trust in higher education. The majority of the points fall within the "GOOD" and "SATISFACTORY" health status categories. Current Economic Situation in Germany vs Social Class (Log-transformed):

- c. Lower numerical values on the x-axis indicate a better economic situation, while higher values indicate a worse situation.** Social class shows some variation across different economic situations, but there is no strong linear relationship. Trust in Television vs Trust in Newspapers (Log-transformed):

- d. A positive correlation is evident, with individuals who trust television also likely to trust newspapers.** Trust values are lower (indicating stronger trust) for individuals with better health statuses like "VERY GOOD" and "GOOD". Current Economic Situation in Germany vs Success Depends on Education (Log-transformed):

- e. The scatter plot suggests no strong relationship between the economic situation and the belief that success depends on education.** Health status appears to be distributed evenly across the different economic situations and beliefs about education. Trust in Federal Constitutional Court vs Trust in Federal Government (Log-transformed):

- f. A positive correlation is present, indicating that individuals who trust the Federal Constitutional Court also tend to trust the Federal Government. Trust values are lower (indicating stronger

trust) for individuals with better health statuses.

Conclusion: Positive correlations are observed between height and weight, trust in television and newspapers, and trust in the Federal Constitutional Court and the Federal Government. The color coding by health status shows that individuals with better health tend to have stronger trust (lower numerical values). The data also suggests that the current economic situation in Germany and beliefs about success depending on education are not strongly correlated with other variables.

Understanding these relationships helps in identifying patterns and potential areas for further analysis, especially considering the impact of health status on trust and perceptions of the economic situation.

Moreover, as we observed some strong correlation between certain variable pairs, we can use PCA in the analysis below to further analyze and reduce the dimensionality of this dataset. PCA combines several variables into principal components that capture the maximum variance in the data, whcih helps reduce the complexity of the dataset, mitigating multicollinearity issues, enhancing the robustness of predictive models.

B. Conditioanl distribution of X variables given health

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

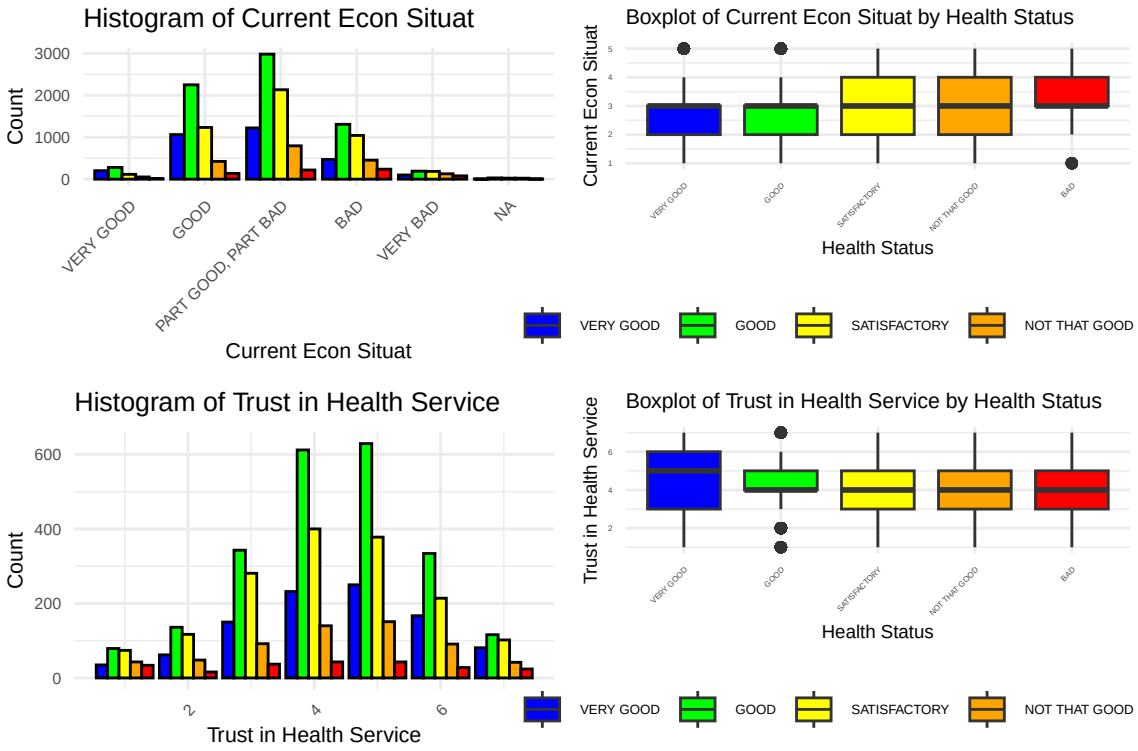
```
## Rows: 17398 Columns: 1209
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (1164): uniqueid, year, personid, x1, x2, x3, x4, x5, x6, x7, x8, x9, x1...
## lgl   (45): x96, x97, x98, x435, x637, x811, x812, x815, x816, x817, x818, x...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Understanding the conditional distribution of explanatory (X) variables given different levels of the response variable, in this case, health status, is a crucial step in many statistical analyses and machine learning tasks. This approach allows us to uncover how various factors relate to health outcomes and identify potential predictors for health-related studies. By examining the conditional distributions, we can gain insights into how different X variables (such as socioeconomic factors, trust in institutions, or personal attributes) vary across different health statuses. This can reveal significant relationships and dependencies that might not be apparent when looking at marginal distributions.

```
## Warning: Removed 87 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```
## Warning: Removed 11774 rows containing non-finite outside the scale range
## (`stat_count()`).
```

```
## Warning: Removed 11774 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Histogram and Boxplot of Current Economic Situation

- Histogram: The "GOOD" health status is the most frequently reported, followed by "SATISFACTORY". Individuals reporting "VERY GOOD" and "GOOD" health statuses are more likely to rate the current economic situation as better (lower numerical values). Conversely, those with "NOT THAT GOOD" or "BAD" health statuses tend to rate the economic situation worse (higher numerical values).
- Boxplot: Lower median values are observed for "VERY GOOD" and "GOOD" health statuses, indicating a better perception of the economic situation. There is greater variability in the economic situation ratings for individuals with "SATISFACTORY", "NOT THAT GOOD", and "BAD" health statuses, with higher median values.
- Economic Situation: Healthier individuals tend to perceive the current economic situation more positively. The ratings are more favorable (lower numerical values) for "VERY GOOD" and "GOOD" health statuses, indicating better economic perceptions.

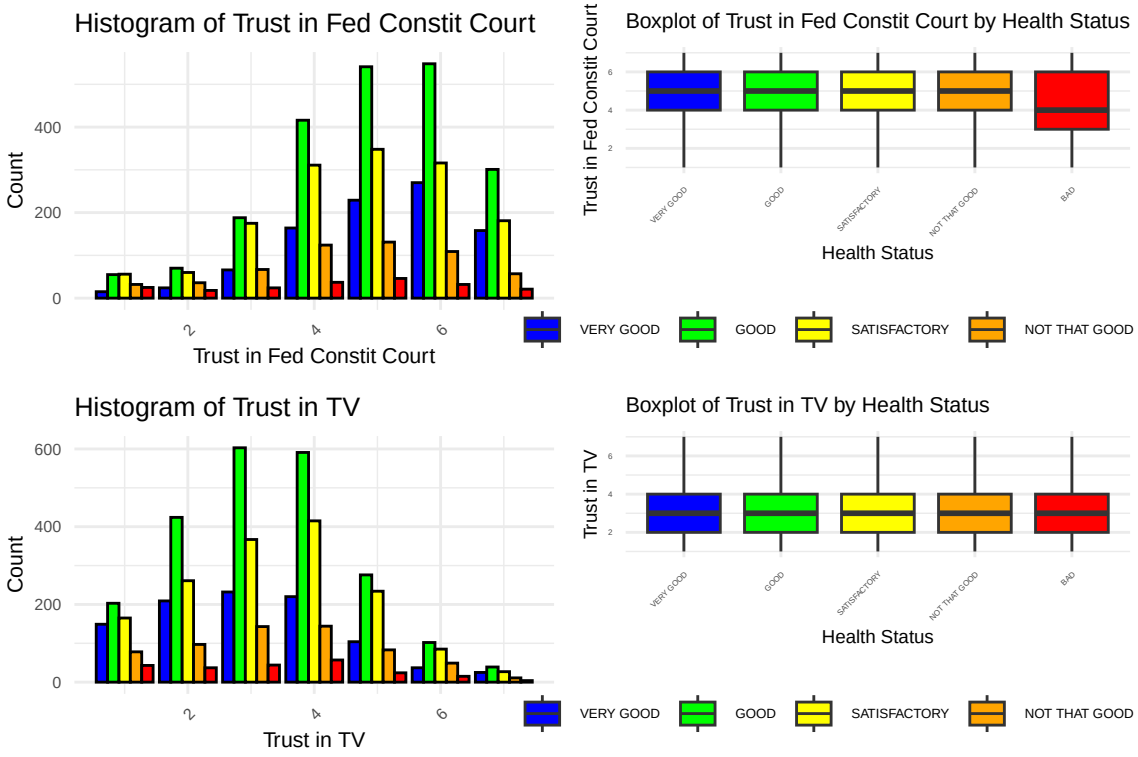
Histogram and Boxplot of Trust in Health Service

- Histogram: Higher trust levels (lower numerical values) are more common among individuals reporting "VERY GOOD" and "GOOD" health statuses. There is a noticeable decline in trust for "SATISFACTORY", "NOT THAT GOOD", and "BAD" health statuses.
- Boxplot: Individuals with "VERY GOOD" and "GOOD" health statuses generally exhibit higher trust in health services, with lower median values. The trust levels decrease for those with "SATISFACTORY", "NOT THAT GOOD", and "BAD" health statuses, showing higher median values and greater spread.
- Trust in Health Services: Trust in health services is higher among healthier individuals. Those with "VERY GOOD" and "GOOD" health statuses exhibit stronger trust (lower numerical values), whereas trust declines for individuals with poorer health statuses.

These insights emphasize the importance of health status as a key factor influencing individuals' perceptions of their economic environment and trust in health services.

```
## Warning: Removed 12147 rows containing non-finite outside the scale range
```

```
## (`stat_count()`).

## Warning: Removed 12147 rows containing non-finite outside the scale range
## (`stat_boxplot()`).

## Warning: Removed 11801 rows containing non-finite outside the scale range
## (`stat_count()`).

## Warning: Removed 11801 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Histogram and Boxplot of Trust in the Federal Constitutional Court

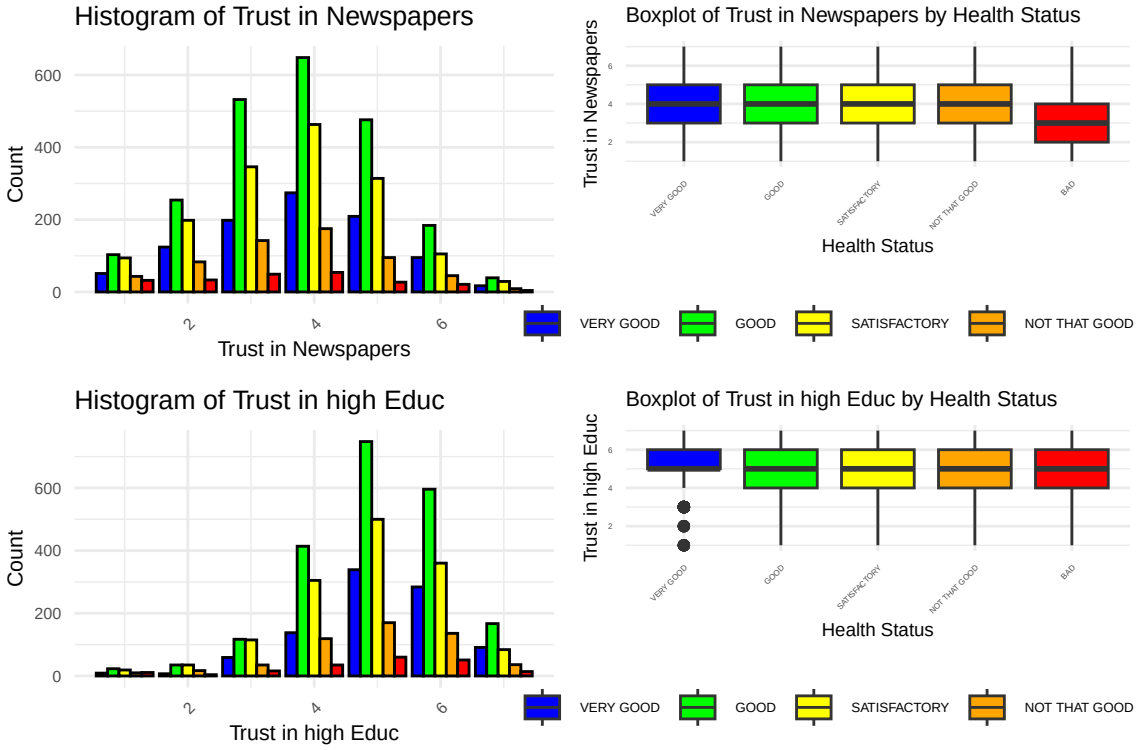
- Histogram: Higher trust levels (lower numerical values) are more frequent among individuals reporting "VERY GOOD" and "GOOD" health statuses. Trust decreases as health status worsens, with "NOT THAT GOOD" and "BAD" categories showing a broader distribution with higher numerical values.
- Boxplot: Individuals with "VERY GOOD" and "GOOD" health statuses generally exhibit higher trust in the Federal Constitutional Court, with lower median values. Trust levels are lower (higher numerical values) for those with "SATISFACTORY", "NOT THAT GOOD", and "BAD" health statuses, showing higher medians and greater spread.
- Trust in the Federal Constitutional Court: Healthier individuals tend to have higher trust in the Federal Constitutional Court. The trust levels are more favorable (lower numerical values) for "VERY GOOD" and "GOOD" health statuses, indicating stronger trust.

Histogram and Boxplot of Trust in TV

- Histogram: The histogram depicts the distribution of trust in TV across different health statuses. Similar to the previous histogram, higher trust levels (lower numerical values) are more common among individuals with "VERY GOOD" and "GOOD" health statuses. Trust in TV decreases for "SATISFACTORY", "NOT THAT GOOD", and "BAD" health statuses, with a broader distribution towards higher numerical values.

- Boxplot: The boxplot shows the variability in trust within each health status category. Individuals with "VERY GOOD" and "GOOD" health statuses generally have higher trust in TV, with lower median values. Trust levels decrease (higher numerical values) for those with poorer health statuses ("SATISFACTORY", "NOT THAT GOOD", and "BAD"), showing higher medians and greater variability.
- Trust in TV: Similarly, trust in TV is higher among healthier individuals. Those with "VERY GOOD" and "GOOD" health statuses exhibit stronger trust (lower numerical values), whereas trust decreases for individuals with poorer health statuses.

```
## Warning: Removed 11833 rows containing non-finite outside the scale range
## (`stat_count()`).

## Warning: Removed 11833 rows containing non-finite outside the scale range
## (`stat_boxplot()`).

## Warning: Removed 12239 rows containing non-finite outside the scale range
## (`stat_count()`).

## Warning: Removed 12239 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Histogram and Boxplot of Trust in Newspapers

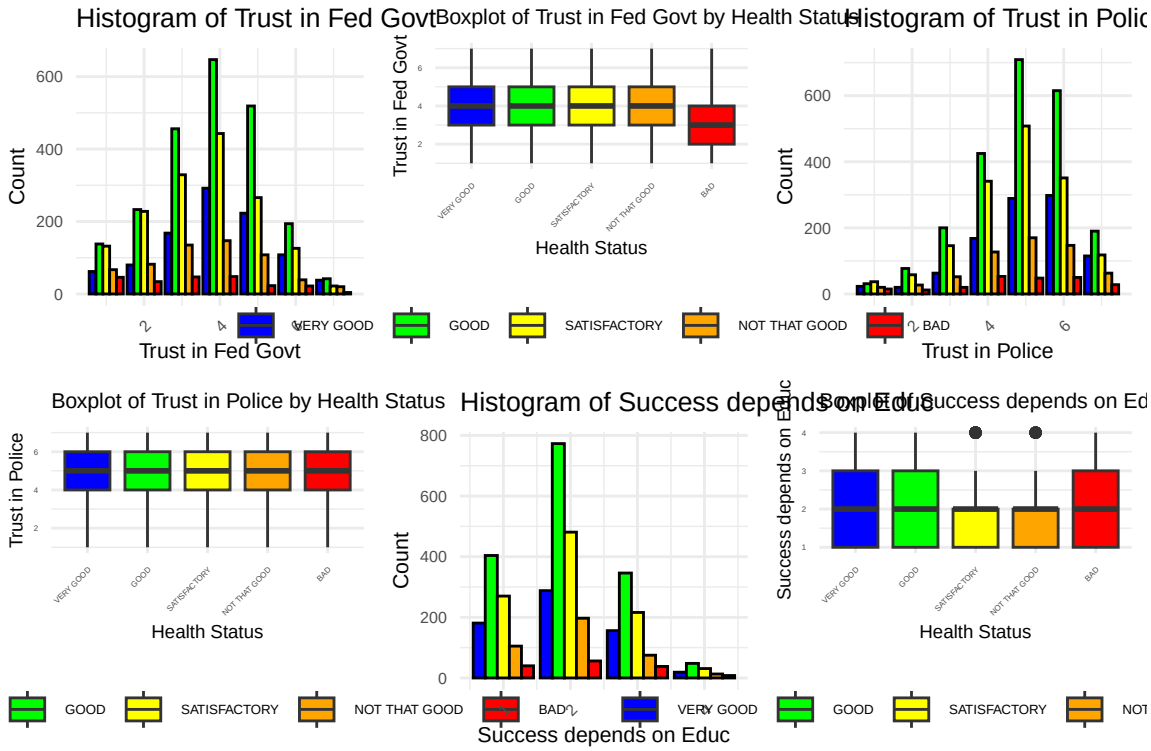
- Histogram: Higher trust levels (lower numerical values) are more frequent among individuals reporting "VERY GOOD" and "GOOD" health statuses. Trust decreases as health status worsens, with "NOT THAT GOOD" and "BAD" categories showing a broader distribution with higher numerical values.
- Boxplot: Individuals with "VERY GOOD" and "GOOD" health statuses generally exhibit higher trust in newspapers, with lower median values. Trust levels are lower (higher numerical values) for those with "SATISFACTORY", "NOT THAT GOOD", and "BAD" health statuses, showing higher medians and greater spread.
- Trust in Newspapers: Healthier individuals tend to have higher trust in newspapers. The trust levels are

more favorable (lower numerical values) for "VERY GOOD" and "GOOD" health statuses, indicating stronger trust.

Histogram and Boxplot of Trust in Higher Education

- Histogram: Higher trust levels (lower numerical values) are more common among individuals with "VERY GOOD" and "GOOD" health statuses. Trust in higher education decreases for "SATISFACTORY", "NOT THAT GOOD", and "BAD" health statuses, with a broader distribution towards higher numerical values.
- Boxplot: Individuals with "VERY GOOD" and "GOOD" health statuses generally have higher trust in higher education, with lower median values. Trust levels decrease (higher numerical values) for those with poorer health statuses ("SATISFACTORY", "NOT THAT GOOD", and "BAD"), showing higher medians and greater variability.
- Trust in Higher Education: Similarly, trust in higher education is higher among healthier individuals. Those with "VERY GOOD" and "GOOD" health statuses exhibit stronger trust (lower numerical values), whereas trust decreases for individuals with poorer health statuses.

```
## Warning: Removed 11830 rows containing non-finite outside the scale range
## (`stat_count()`).

## Warning: Removed 11830 rows containing non-finite outside the scale range
## (`stat_boxplot()`).

## Warning: Removed 11784 rows containing non-finite outside the scale range
## (`stat_count()`).

## Warning: Removed 11784 rows containing non-finite outside the scale range
## (`stat_boxplot()`).

## Warning: Removed 13653 rows containing non-finite outside the scale range
## (`stat_count()`).

## Warning: Removed 13653 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



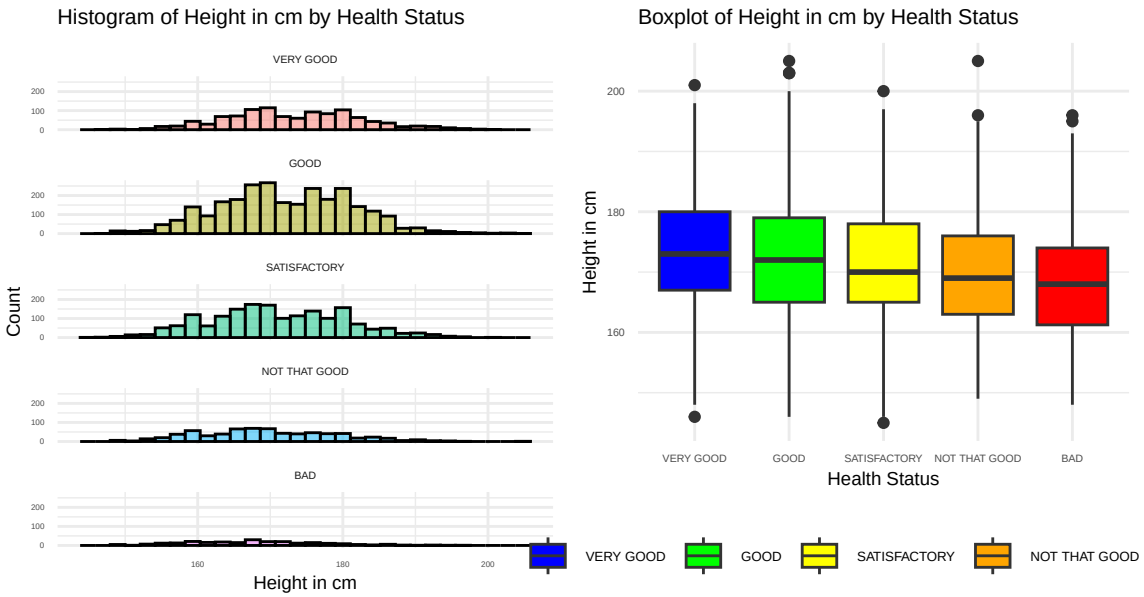
Trust in the Federal Government: Healthier individuals tend to have higher trust in the Federal Government. The trust levels are more favorable (lower numerical values) for "VERY GOOD" and "GOOD" health statuses, indicating stronger trust.

Trust in the Police: Similarly, trust in the Police is higher among healthier individuals. Those with "VERY GOOD" and "GOOD" health statuses exhibit stronger trust (lower numerical values), whereas trust decreases for individuals with poorer health statuses.

Belief that Success Depends on Education: Healthier individuals also tend to believe more strongly that success depends on education. The agreement levels are higher (lower numerical values) for "VERY GOOD" and "GOOD" health statuses.

```
## Warning: Removed 10871 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

```
## Warning: Removed 10871 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



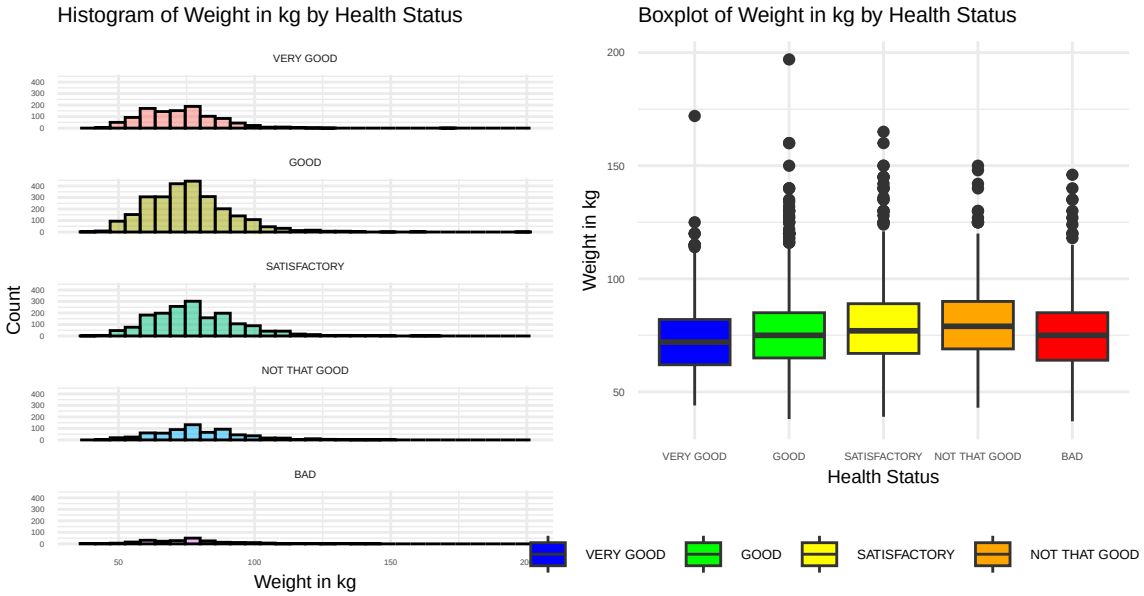
a. Height and Health: Healthier individuals (those reporting "VERY GOOD" and "GOOD" health statuses) tend to be taller on average compared to those with poorer health statuses.

b. Variability: There is less variability in height among healthier individuals, as indicated by narrower interquartile ranges and fewer outliers in the boxplots. In contrast, poorer health statuses show more variability in height, with wider IQRs and more outliers.

c. Distribution Patterns: The histograms suggest that the distribution of height becomes more flattened and spread out as health status worsens, indicating a broader range of heights among less healthy individuals.

```
## Warning: Removed 11017 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

```
## Warning: Removed 11017 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



a. Weight and Health: Healthier individuals (those reporting "VERY GOOD" and "GOOD" health sta-

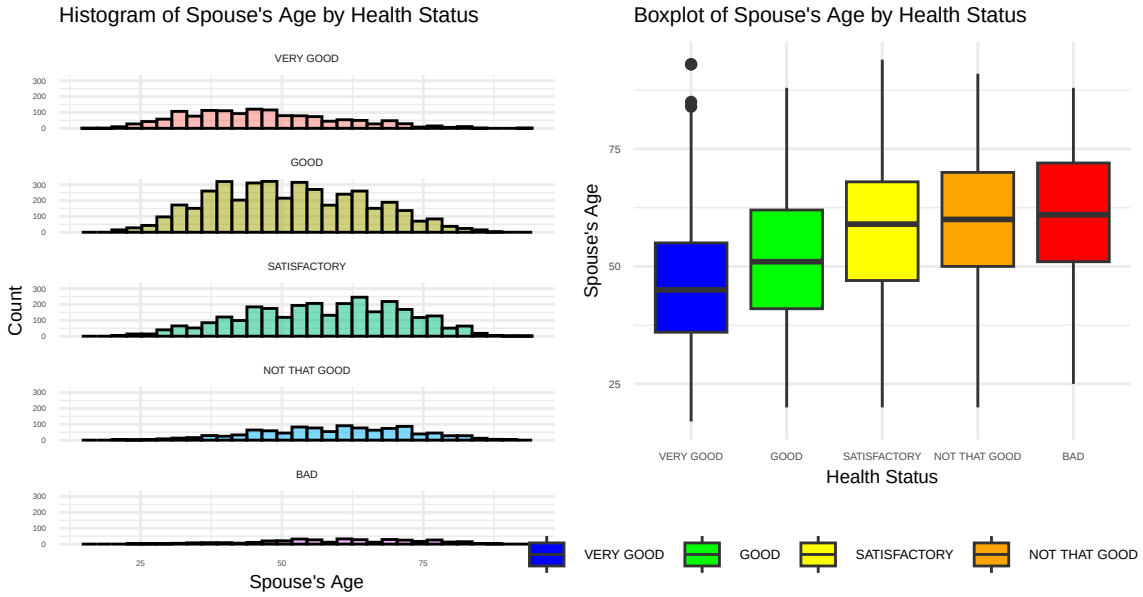
tuses) tend to have lower average weights compared to those with poorer health statuses.

b. Variability: There is less variability in weight among healthier individuals, as indicated by narrower interquartile ranges and fewer outliers in the boxplots. In contrast, poorer health statuses show more variability in weight, with wider IQRs and more outliers.

c. Distribution Patterns: The histograms suggest that the distribution of weight becomes more flattened and spread out as health status worsens, indicating a broader range of weights among less healthy individuals.

```
## Warning: Removed 7560 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

```
## Warning: Removed 7560 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



a. Spouse's Age and Health: Healthier individuals (those reporting "VERY GOOD" and "GOOD" health statuses) tend to have younger spouses on average compared to those with poorer health statuses.

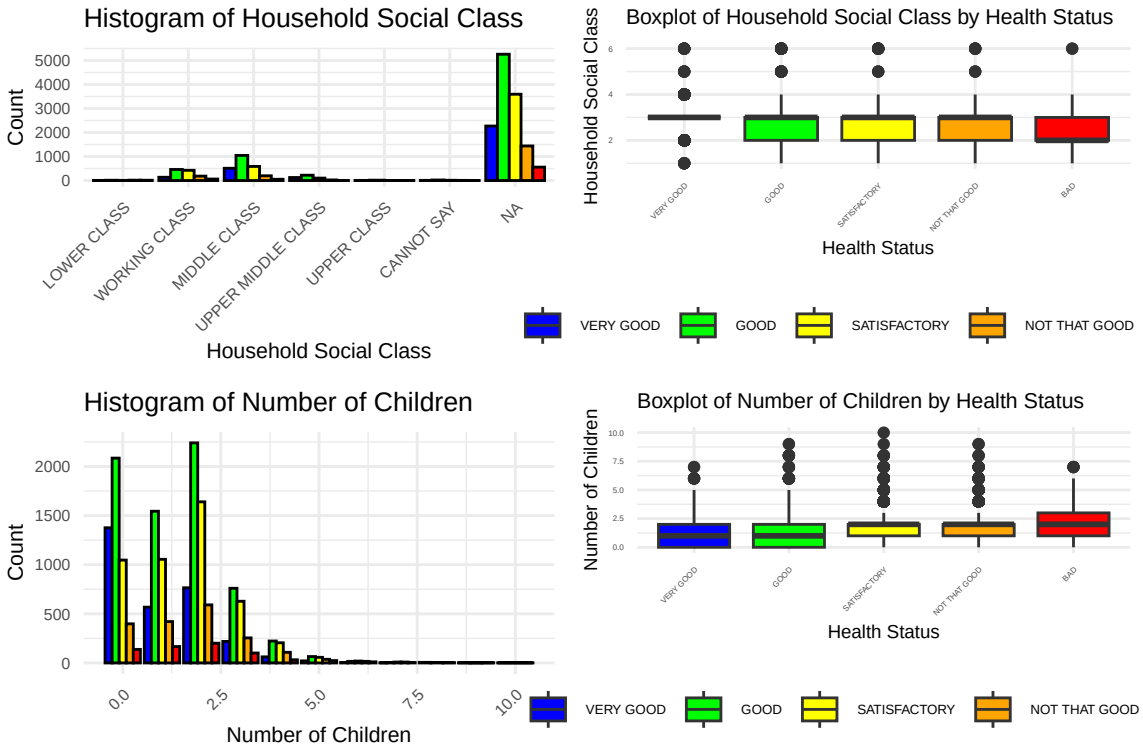
b. Variability: There is less variability in spouse's age among healthier individuals, as indicated by narrower interquartile ranges and fewer outliers in the boxplots. In contrast, poorer health statuses show more variability in spouse's age, with wider IQRs and more outliers.

c. Distribution Patterns: The histograms suggest that the distribution of spouse's age becomes more flattened and spread out as health status worsens, indicating a broader range of ages among spouses of less healthy individuals.

```
## Warning: Removed 13127 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```
## Warning: Removed 249 rows containing non-finite outside the scale range
## (`stat_count()`).
```

```
## Warning: Removed 249 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



a. Household Social Class: Healthier individuals tend to be in higher social classes (lower numerical values). The social class distribution is more concentrated for "VERY GOOD" and "GOOD" health statuses, indicating a higher proportion of middle and upper-middle-class individuals. Poorer health statuses show more variability and a tendency towards lower social classes.

b. Number of Children: Healthier individuals tend to have fewer children. The distribution of the number of children is lower for "VERY GOOD" and "GOOD" health statuses, while those with poorer health statuses show slightly higher numbers of children and more variability.

4. Data Cleaning

4.1 Selecting Threshold for missing values percentage

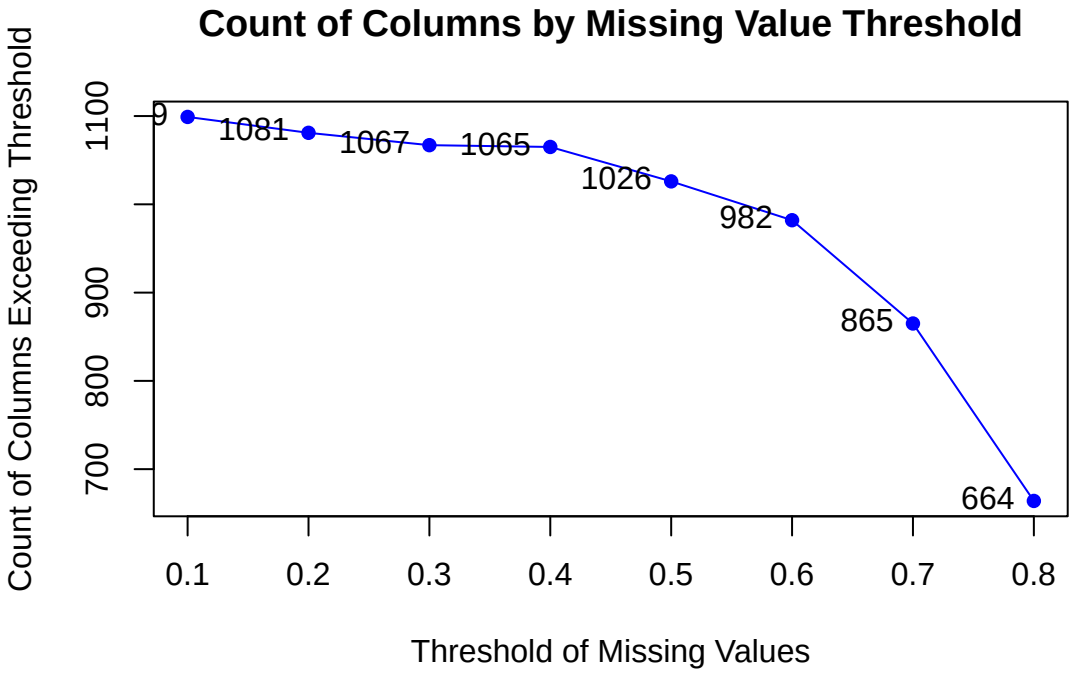
This extensive dataset contains 1,209 variables, many of which have substantial proportions of missing data. Recognizing the importance of maintaining robust and meaningful features for our analyses, we conducted a detailed assessment to determine an optimal threshold for missing values. Our evaluations included generating a scatter plot to visualize the distribution of missing values across various thresholds.

We observed that the counts of features with missing values do not significantly differ between thresholds of 30% to 40%. This stability suggested that a threshold within this range would be effective for our purposes without compromising the dataset's integrity. To enhance the accuracy and reliability of our subsequent analyses, we decided on a 30% threshold. This threshold strikes a balance, retaining a rich set of features while excluding those with excessive missing data.

Upon applying this threshold, we reduced the feature set from 1,209 to 142 variables. This substantial reduction ensures that the remaining features are more likely to contribute meaningful insights and support robust statistical analysis. The refined dataset not only aligns better with our analytical goals but also simplifies future data handling and processing tasks.

```
## Loading required package: Matrix

##
```

```
## Attaching package: 'gamlr'

## The following object is masked from 'package:MuMIn':
##
##     AICc

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:MuMIn':
##
##     importance

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine

## Loaded gbm 2.1.9

## This version of gbm is no longer under development. Consider transitioning to gbm3, https://github.c
```



4.2 Imputation

Our dataset comprises a mix of categorical and numerical features, necessitating differentiated strategies for handling missing data effectively. Imputation plays a critical role in preparing our data for robust analysis by filling in missing values using methods tailored to the nature of each feature.

For numerical features, we employ the mean imputation technique. This method involves calculating the average value of a feature and using that average to replace all missing entries. Mean imputation is particularly effective for numerical data as it maintains the overall distribution and central tendency, which is crucial for maintaining the integrity of subsequent analyses.

Categorical features, on the other hand, are treated differently. Since these features are often not ordinal and don't have a central tendency that can be represented as a mean, we use mode imputation. Here, we identify the most frequently occurring category within each feature and use this mode to fill in gaps. This method is based on the assumption that the most common category is the most likely classification for missing data points.

I don't understand this

To systematically determine the treatment of each feature as numerical or categorical, we classify any feature with fewer than 20 unique values as categorical. This threshold helps in accurately categorizing the features based on the diversity of data they contain, which in turn, ensures the appropriateness of the imputation method applied.

Following imputation, we segment our dataset into training and testing subsets to evaluate the performance of our models. Consistent with common practice in machine learning and to ensure a substantial amount of data for training, we allocate 80% of our dataset to the training set and the remaining 20% to the testing set. This split not only supports the development of more accurate models but also allows us to rigorously test these models on unseen data, thereby verifying their generalizability and robustness.

5. Which features most significantly influence individuals' self-assessed health status?

5.1 Regress self-assessed health status onto a selected subset of variables that excludes certain less relevant or redundant variables.

Variables to Exclude:

- Detailed Political Preferences: Assessments of political parties (x8 to x13), unless directly related to healthcare policies, might not be directly relevant.
- Variables with High Missing Values: Like those involving detailed political actions or less frequent political engagements which have significant missing values and may not provide substantial predictive power for health status.
- Redundant Variables: If there are multiple variables covering the same concept with slight variations, include only the most comprehensive or with the least missing data.

In this analysis, we firstly employed a generalized linear model (GLM) to investigate the factors affecting self-assessed health status using a comprehensive dataset. The results reveal significant insights into the relationships between various predictors and health outcomes. A key focus of our report is the interpretation of the p-values and the R^2 value, which provide a quantifiable measure of the statistical significance and explanatory power of the model, respectively. Below are the first 20 lines of the output (partial results).

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

##
## Call:
## glm(formula = log(health) ~ . - uniqueid - year - personid, data = training_set)
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.236e+01  3.213e+00  -3.845 0.000121 ***
## x1           2.197e-02  5.138e-03   4.277 1.91e-05 ***
## x2           6.596e-02  5.386e-03  12.246  < 2e-16 ***
## x3           1.233e-02  5.184e-03   2.379 0.017369 *
## x4           9.957e-03  5.872e-03   1.696 0.089947 .
## x7          -5.517e-03  2.238e-03  -2.465 0.013720 *
## x14          5.807e-03  3.966e-03   1.464 0.143179
## x17          7.531e-05  1.401e-04   0.537 0.590978
## x19          2.969e-02  1.167e-02   2.545 0.010938 *
## x162        -2.568e-02  5.841e-03  -4.397 1.10e-05 ***
## x163        -2.239e-02  6.881e-03  -3.255 0.001139 **
## x164        -1.386e-02  2.877e-03  -4.818 1.46e-06 ***
## x472        -3.050e-03  2.028e-03  -1.504 0.132644
## x477        -2.301e-03  3.272e-03  -0.703 0.481888
```

Based on the results from the generalized linear model (GLM), we can discern significant variables that impact self-assessed health status, enriching our understanding of the factors intimately connected with individuals' health perceptions. Here's a nuanced interpretation of the statistically significant coefficients highlighted in the analysis:

Significantly Positive Influencing Variables:

- Current Economic Situation in Germany (x1, Estimate: 0.02197, $p < 0.001$): The positive impact of x1 suggests that improvements in the broader economic environment are associated with better individual health assessments. This might reflect psychological well-being tied to economic optimism or reduced stress due to a stable economic climate.
- Own Current Financial Situation (x2, Estimate: 0.06596, $p <$ 2e-16): This variable's strong positive correlation with health status underscores the direct link between personal financial security and health. Individuals who perceive their financial situation favorably are likely to have better access to healthcare services and fewer stress-related health issues, contributing to overall better health assessments.

Significantly Negative Influencing Variables:

- Perception of Fair Share in Standard of Living (X162, Estimate: -0.02568, $p < 0.00001$): The negative coefficient for x162 suggests that feelings of economic disparity or perceived inequity in living standards are associated with poorer self-assessed health. This could reflect the psychological and physical strain of feeling economically disadvantaged.
- Self-assessment of Social Class, Respondent (x163, Estimate: -0.02239, $p < 0.001$): The significant negative impact of x163 on health highlights the influence of social class perception on health outcomes. Viewing oneself in a lower social class may be linked to chronic stress, reduced access to health-promoting resources, and greater exposure to adverse life conditions.
- Individualism Belief: Everyone Should Look Out for Themselves (x617, Estimate: -0.03100, $p < 0.001$): This variable's pronounced negative effect suggests that a strong belief in individualism might correlate with poorer health. This could be due to a lack of community support or social isolation, which are critical factors in mental and physical health.

There is the formula $R^2 = \frac{\text{Null deviance}-\text{Residual deviance}}{\text{Null deviance}}$.

```
## [1] 0.2704692
```

Using the given function and direct calculation method, we find that the R^2 in the full model will be 0.2704692.

An R^2 of approximately 0.27 suggests that while the model captures a significant portion of the variability in health outcomes, there remains a considerable amount of variance that is unaccounted for by the current

model. This is not uncommon in social science and health-related studies where human behaviors and outcomes are influenced by a complex interplay of factors, many of which are difficult to measure and quantify.

5.2 How many coefficients are used in this model and how many are significant at 10% FDR?

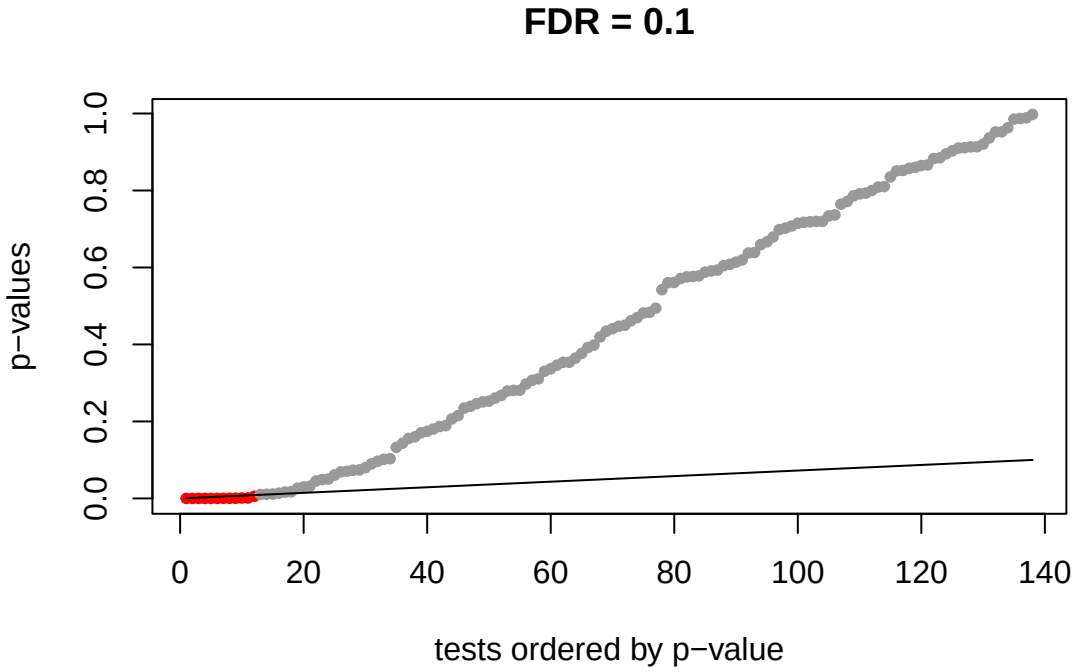
Number of coefficients in the full model:

```
## [1] 139
```

The total number of coefficients used in your generalized linear model (GLM) is 139. This total includes the intercept and 138 other predictor variables. Each coefficient corresponds to a predictor's effect on the log-transformed self-assessed health status. The full model reflects a robust set of predictors aimed at capturing the diverse influences on health perceptions. The analysis ensures broad coverage of potential factors, from economic conditions to social relationships and individual perceptions.

Number of significant coefficient at 10% FDR:

Using the false discovery rate (FDR) control method at a 10% threshold, the following variables are identified as significant. This means that these variables' coefficients are statistically significant at controlling the expected proportion of incorrectly rejected null hypotheses among the rejected hypotheses to 10%.




```
## FDR cutoff p-value =  0.005442358
```

	Variable Description
x1	CURRENT ECONOMIC SITUATION IN GERMANY
x2	OWN CURRENT FINANCIAL SITUATION
x162	FAIR SHARE IN STANDARD OF LIVING?
x163	SELF-ASSESSMENT OF SOCIAL CLASS
x164	TOP-BOTTOM-SCALE: SELF-CLASSIFIC.
x595	MEMBER OF A TRADE UNION
x613	GENERAL TRUST IN FELLOW MEN
x615	WITH SUCH A FUTURE, NO MORE CHILDREN

	Variable Description
x617	MOST PEOPLE DON'T CARE ABOUT OTHERS
x657	CURRENT EMPLOYMENT STATUS
x965	DEGREE OF RELATIONSHIP
x1179	START OF INTERVIEW

This selection criterion helps in mitigating the risk of type I errors typically associated with multiple comparisons. Here's a detailed look at the significant variables:

- CURRENT ECONOMIC SITUATION IN GERMANY (x1): Indicates a strong link between the national economic climate and individual health perceptions. Improved economic conditions are likely associated with better access to healthcare and lower stress levels.
- OWN CURRENT FINANCIAL SITUATION (x2): Reflects the direct impact of personal financial security on health, suggesting that individuals with stable finances tend to report better health outcomes.
- FAIR SHARE IN STANDARD OF LIVING? (x162): Suggests that perceptions of fairness and equity in economic distribution significantly affect health, possibly due to stress and dissatisfaction when perceived fairness is low.
- SELF-ASSESSMENT OF SOCIAL CLASS (x163): Highlights the influence of social stratification on health, where lower self-assessed social status is associated with poorer health.
- TOP-BOTTOM-SCALE: SELF-CLASSIFIC. (x164): Similar to x163, it underscores the impact of self-perceived social rank on health outcomes.
- MEMBER OF A TRADE UNION (x595): Indicates the potential health benefits of union membership, possibly due to better job security and working conditions.
- GENERAL TRUST IN FELLOW MEN (x613): Suggests that higher levels of general trust in society correlate with better self-assessed health, possibly due to reduced stress and increased social support.
- WITH SUCH A FUTURE, NO MORE CHILDREN (x615): Reflects pessimism about the future, significantly correlating with poorer health, potentially due to psychological distress.
- MOST PEOPLE DON'T CARE ABOUT OTHERS (x617): Shows how societal cynicism can negatively affect individual health, perhaps through mechanisms of social isolation or stress.
- CURRENT EMPLOYMENT STATUS (x657): Confirms the crucial role of employment in health status, with stable employment typically promoting better health outcomes.
- DEGREE OF RELATIONSHIP (x965): Emphasizes the importance of close personal relationships in maintaining health, likely due to emotional support and reduced feelings of loneliness.
- START OF INTERVIEW (x1179): This might relate to procedural aspects of data collection that correlate with how respondents perceive their health at the beginning of the survey process.

Conclusions and Implications

The identified significant variables offer a nuanced understanding of the factors that significantly affect self-assessed health. Economic and social variables play a pivotal role, suggesting that interventions aiming to improve public health should consider broader socio-economic improvements alongside targeted health services. Additionally, the psychological state and societal attitudes reflected in the significant predictors highlight the importance of mental health and social cohesion in overall health perceptions.

Re-run the regression with only the significant covariates, and compare the R^2 to the full model

```
## [1] 0.1743375
```

```
## R2 of regression only the significant covariates =  0.1743375
```

```
## full model R2 =  0.2704692
```

Reduction in R^2: The R^2 value decreased from 0.2704692 in the full model to 0.1743375 in the reduced model. This decrease indicates that while the significant covariates contribute to explaining the variability in self-assessed health status, they do not capture as much of the variability as the full set of covariates. This suggests several potential statistical and substantive reasons:

- Omission of Informative Variables: The reduced model includes only those variables deemed significant under the FDR threshold, potentially omitting other variables that, while not meeting the stringent criteria for significance, still contribute valuable information and variance explanation to the model.
- Loss of Data Complexity: In complex datasets, especially those involving human health and socioeconomic factors, many variables may interact in subtle ways. The full model, by including a broader array of variables, might capture complex interactions and non-linear relationships that are missed in the reduced model.

Did you try OOS experiments?

Significance vs. Contribution: Variables deemed significant are not always those that contribute most to the variance explained in the dependent variable. Some variables may have a small but statistically significant effect, or they may be significant in the presence of other variables, which helps control for confounding effects.

5.3 Fit a regression for whether the economic situation in Germany(x1) is more than 3 (Part good, part bad) (onto all variables but economic situation in Germany and self-assessed health status).

Below are the first 20 lines of the output (partial results).

```
##  [1] ""
##  [2] "Call:"
##  [3] "glm(formula = eco_situation_better ~ . - uniqueid - year - personid - "
##  [4] "    x1 - health, family = \"binomial\", data = training_set)"
##  [5] ""
##  [6] "Coefficients:"
##  [7] "              Estimate Std. Error z value Pr(>|z|)    "
##  [8] "(Intercept)  4.883e+02  2.378e+01  20.538  < 2e-16 ***"
##  [9] "x2           8.505e-01  3.771e-02  22.555  < 2e-16 ***"
## [10] "x3           5.171e-01  3.743e-02  13.815  < 2e-16 ***"
## [11] "x4          -1.540e-02  4.164e-02  -0.370 0.711491    "
## [12] "x7          -7.376e-04  1.640e-02  -0.045 0.964123    "
## [13] "x14          8.201e-02  2.935e-02   2.794 0.005201 ** "
## [14] "x17          4.285e-03  9.468e-04   4.526 6.02e-06 ***"
## [15] "x19          9.840e-02  8.097e-02   1.215 0.224259    "
## [16] "x162        -9.504e-02  4.056e-02  -2.343 0.019104 *  "
## [17] "x163         1.123e-01  5.015e-02   2.240 0.025091 *  "
## [18] "x164         3.386e-02  2.073e-02   1.634 0.102281    "
## [19] "x472        -1.038e-02  1.508e-02  -0.688 0.491190    "
## [20] "x477         6.992e-02  2.452e-02   2.852 0.004351 ** "
```

The results from the logistic regression model provided offer a comprehensive view of the relationships between various predictors and the likelihood that respondents view the economic situation in Germany as better than "part good, part bad."The model significantly improves fit compared to a null model, as evidenced by a substantial drop in the residual deviance.

Notable Coefficients: Highly Significant Predictors ($p < 0.001$)

- Intercept: Extremely large positive intercept might indicate scaling or data coding issues, or it could be influencing the estimated probabilities significantly.
- x2 (RESP. OWN CURRENT FINANCIAL SITUATION): Strong positive effect, indicating that better personal financial conditions are perceived as reflective of a better national economic situation.
- x3 (ECONOMIC SITUATION IN GERMANY IN ONE YEAR): Positive coefficient suggests that optimism about future economic prospects enhances current positive perceptions.
- x615 (WITH SUCH A FUTURE, NO MORE CHILDREN): Very strong negative coefficient implies that fears about future stability, affecting family planning decisions, are associated with negative views on current economic conditions.

Additional Significant Predictors: x14, x17, x162, x163, x477, x614, x616, x617, x630, x634, x639: These variables show varied levels of significance and effects on the perception of economic conditions, ranging from personal beliefs about societal structures to demographic characteristics.

Model Fit and Diagnostics: Residual Deviance: The reduction from the null deviance (13480.1) to 9405.3 on similar degrees of freedom suggests that the model explains a good portion of the variability in the response.

Interpretation and Implications: Economic and Psychological Predictors: The significant variables highlight both economic factors and psychological attitudes as crucial in shaping perceptions of the national economy. For instance, personal financial security, future outlooks, and broader societal trust and cynicism are pivotal. Policy Relevance: Understanding these drivers can help policymakers address public perceptions through targeted economic policies and communication strategies that address both actual economic conditions and public sentiment.

Add and describe an interaction between respondent's own financial situation and satisfaction with democracy.

Below are the first 20 lines of the output (partial results).

```
##  [1] ""
##  [2] "Call:"
##  [3] "glm(formula = eco_situation_better ~ . - uniqueid - year - personid - "
##  [4] "    x1 - health + x2 * x630, family = \"binomial\", data = training_set)"
##  [5] ""
##  [6] "Coefficients:"
##  [7] "              Estimate Std. Error z value Pr(>|z|)    "
##  [8] "(Intercept)  4.889e+02  2.379e+01  20.548  < 2e-16 ***"
##  [9] "x2           7.802e-01  1.048e-01   7.442 9.95e-14 ***"
## [10] "x3           5.172e-01  3.744e-02  13.817  < 2e-16 ***"
## [11] "x4          -1.487e-02  4.165e-02  -0.357 0.721106    "
## [12] "x7          -8.985e-04  1.640e-02  -0.055 0.956303    "
## [13] "x14          8.224e-02  2.935e-02   2.802 0.005081 ** "
## [14] "x17          4.305e-03  9.473e-04   4.545 5.50e-06 ***"
## [15] "x19          9.795e-02  8.099e-02   1.209 0.226507    "
## [16] "x162        -9.468e-02  4.057e-02  -2.334 0.019605 *  "
## [17] "x163         1.119e-01  5.015e-02   2.232 0.025618 *  "
## [18] "x164         3.404e-02  2.073e-02   1.642 0.100535    "
## [19] "x472        -1.023e-02  1.508e-02  -0.678 0.497609    "
## [20] "x477         6.960e-02  2.453e-02   2.837 0.004549 ** "
```

Primary Variables of Interest: - x2 (Respondent's Own Financial Situation): Continues to show a strong positive effect (Estimate = 0.7802, p < 1e-13), indicating that individuals' financial situations are strongly correlated with positive perceptions of the economic situation. - x630 (Respondent: Sex): By itself, not significantly influencing perceptions (Estimate = 0.2675, p = 0.202), suggesting that gender does not independently affect views on the economy.

Interaction Term - The interaction between x2 and x630 (Estimate = 0.04592, p = 0.473) is not statistically significant, indicating that the impact of one's financial situation on perceptions of economic betterment does not differ significantly between genders.

5.4 Focus only on a subset of respondents where age(x633) is greater than a specific number (e.g., 50 years).

Train the full model from previous part on this subset. Below are the first 20 lines of the output (partial results).

```
##  [1] ""
##  [2] "Call:"
##  [3] "glm(formula = log(health) ~ . - uniqueid - year - personid, data = training_set[subset, "
##  [4] "    ])"
##  [5] ""
##  [6] "Coefficients:"
##  [7] "                       Estimate Std. Error t value Pr(>|t|)    "
##  [8] "(Intercept)          -4.704e+00  4.468e+00  -1.053 0.292423    "
##  [9] "x1                    2.942e-02  9.752e-03   3.017 0.002567 ** "
## [10] "x2                    6.456e-02  7.521e-03   8.584  < 2e-16 ***"
## [11] "x3                    3.536e-03  7.128e-03   0.496 0.619878    "
## [12] "x4                    5.390e-03  9.326e-03   0.578 0.563276    "
## [13] "x7                   -5.050e-03  2.975e-03  -1.697 0.089723 .  "
## [14] "x14                   1.484e-02  5.287e-03   2.806 0.005029 ** "
## [15] "x17                  -6.016e-06  1.955e-04  -0.031 0.975454    "
## [16] "x19                   3.945e-02  1.796e-02   2.197 0.028092 *  "
## [17] "x162                 -2.680e-02  7.861e-03  -3.409 0.000656 ***"
## [18] "x163                 -1.763e-02  9.353e-03  -1.885 0.059477 .  "
## [19] "x164                 -1.357e-02  3.855e-03  -3.520 0.000435 ***"
## [20] "x472                 -3.012e-03  2.684e-03  -1.122 0.261752    "
```

The model shows a reduction in residual deviance from 910.03 to 445.85 across 5661 degrees of freedom, suggesting the model is relatively effective in capturing variability in the health variable among older adults.

Detailed Analysis of Key Predictors: Significant Positive Influences: - Current Employment Status (x657) - The coefficient value is 0.01501 with a highly significant p-value of 0.000181. This suggests that being employed is associated with better health outcomes among the elderly. Employment may provide not only financial benefits but also social interaction and a sense of purpose, all contributing to better health.

- Respondent's Own Current Financial Situation (x2) - The estimate of 0.02398 and a p-value of 0.0000614 indicate a strong positive influence on health. Financial stability is crucial in ensuring access to necessary medical care, healthier lifestyle choices, and reduced stress, all of which are vital for maintaining good health in older age.

  Significant Negative Influences:

- Member of a Trade Union (x595) - This variable shows a negative coefficient of -0.03276 with a p-value of 0.008326. While union membership often provides benefits such as job security and higher wages, the negative association might reflect the physical and mental strains associated with long-term industrial occupations typically linked to union membership.

Predict the health using this model.

```
## null deviance =  1274.693
```

```
## Residual deviance =  1234.631
```

```
## OOS R2 =  0.03142893
```

Using the testing set to predict. Below are the first 20 lines of the output (partial results).

```
##  [1] ""
##  [2] "Call:"
##  [3] "glm(formula = health_binary ~ . - uniqueid - year - personid - "
##  [4] "    health, family = \"binomial\", data = training_set)"
##  [5] ""
##  [6] "Coefficients:"
##  [7] "                        Estimate Std. Error z value Pr(>|z|)    "
##  [8] "(Intercept)           4.202e+01  2.701e+01   1.555 0.119862    "
##  [9] "x1                   -7.796e-02  6.083e-02  -1.282 0.199956    "
## [10] "x2                   -4.099e-01  4.177e-02  -9.814  < 2e-16 ***"
## [11] "x3                   -9.113e-02  4.195e-02  -2.173 0.029808 *  "
## [12] "x4                   -4.634e-02  4.789e-02  -0.968 0.333190    "
## [13] "x7                    3.145e-02  1.788e-02   1.759 0.078532 .  "
## [14] "x14                  -1.011e-01  3.133e-02  -3.227 0.001251 ** "
## [15] "x17                  -8.424e-04  1.038e-03  -0.811 0.417223    "
## [16] "x19                  -2.165e-01  8.867e-02  -2.442 0.014602 *  "
## [17] "x162                  1.733e-01  4.490e-02   3.859 0.000114 ***"
## [18] "x163                  7.601e-02  5.386e-02   1.411 0.158159    "
## [19] "x164                  9.797e-02  2.193e-02   4.468 7.90e-06 ***"
## [20] "x472                  8.352e-03  1.644e-02   0.508 0.611415    "
```

The predictive model on the testing set shows a decrease in residual deviance from 1274.693 to 933.8033, indicating the model's effectiveness in capturing significant patterns affecting health, with the predictors explaining about 26.7% of the variance in health outcomes. Notably, the model suggests that better financial situations are strongly associated with improved health outcomes, potentially due to better access to healthcare and reduced stress. While one might expect membership in a trade union to relate to better health outcomes due to support and resources, the model does not find a statistically significant association in this context.

5.4 Logistic Regression Model for Predicting Health Status

```
##           Actual
## Predicted    0    1
##         0  104   87
##         1  705 4324
```

```
## [1] "Accuracy: 0.848275862068966"
```

The results of the logistic regression model applied to the health status prediction for all variables show a commendable accuracy of approximately 84.83%. Despite this high overall accuracy, the confusion matrix reveals a significant imbalance in predictive performance. Specifically, the model exhibits a propensity for predicting positive outcomes, evidenced by a large number of false positives (705). This suggests that while the model is proficient at identifying true positive cases (4324), it struggles to accurately classify true negatives, only correctly identifying 104 cases out of a total of 809 negative cases. This imbalance may lead to an overestimation of individuals in poorer health categories and could be indicative of an inherent bias in the training data towards these outcomes. Adjusting the model to better handle class imbalance or revising the threshold for classification could help in achieving a more balanced and accurate predictive performance.

5.5 Whether age has causal effect on health status?

Investigating the causal effect of age on health status is crucial for understanding how health outcomes evolve over an individual's lifespan. Age is a fundamental demographic variable that influences a wide array of health-related factors, including susceptibility to chronic diseases, physical fitness, and overall well-being. As people age, they often experience changes in their physical and mental health, with older individuals

typically facing higher risks of conditions such as cardiovascular diseases, diabetes, and cognitive decline. By analyzing the causal effect of age on health status, we can gain valuable insights into the aging process and identify critical periods for health interventions.

Moreover, the relationship between age and health is not linear; different age groups may have distinct health challenges and needs. For instance, young adults might struggle with issues related to mental health and lifestyle choices, whereas older adults might face age-related degenerative diseases. Understanding these nuances can help tailor healthcare policies and programs to address the specific needs of various age groups effectively. By isolating the impact of age from other confounding variables, such as income and education, we can more accurately determine how age affects health and devise strategies to improve health outcomes across all stages of life.

A. Data Preparation

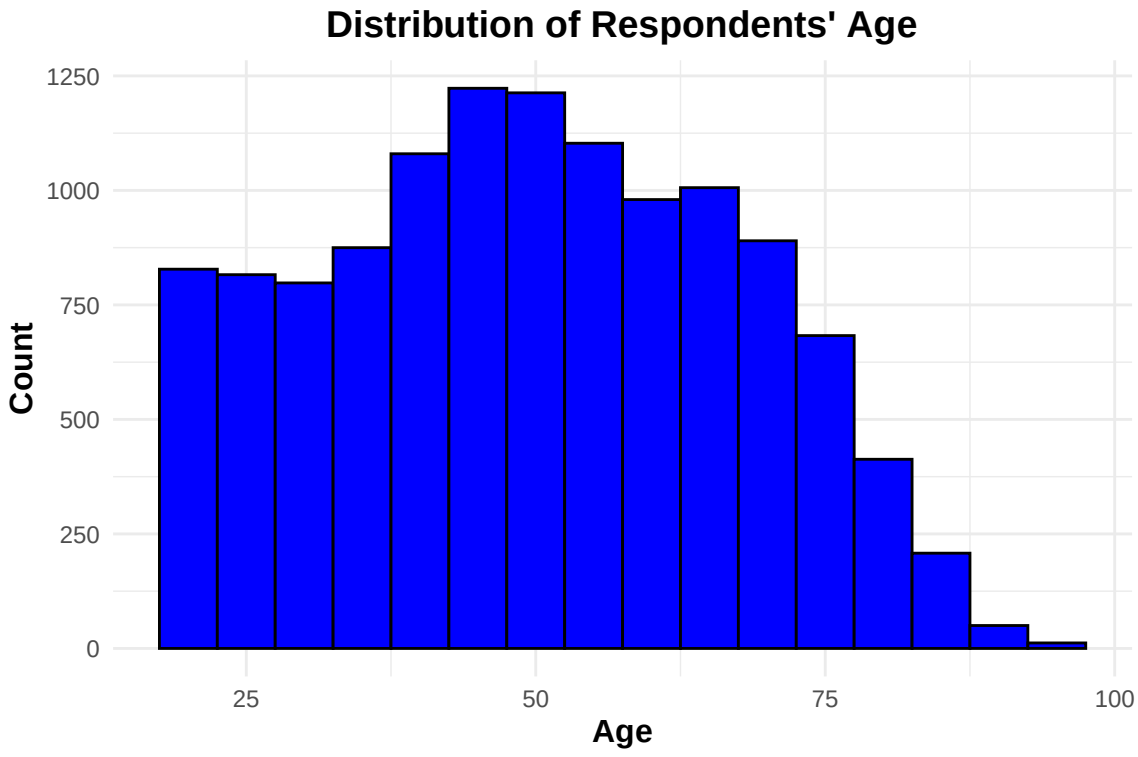
x633 (Respondent's Age): This variable represents the respondent's age, which is numerical. It captures the age of individuals in years, with values ranging from 18 to 102 units. There are 82 unique non-missing values recorded, while 48 values are missing out of a total of 24,840 observations. This variable provides a quantitative measure of the respondents' age, offering insights into the demographic distribution of the surveyed population.

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

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Rows: 12178 Columns: 142
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (142): uniqueid, year, personid, x1, x2, x3, x4, x7, x14, x17, x19, x162...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```



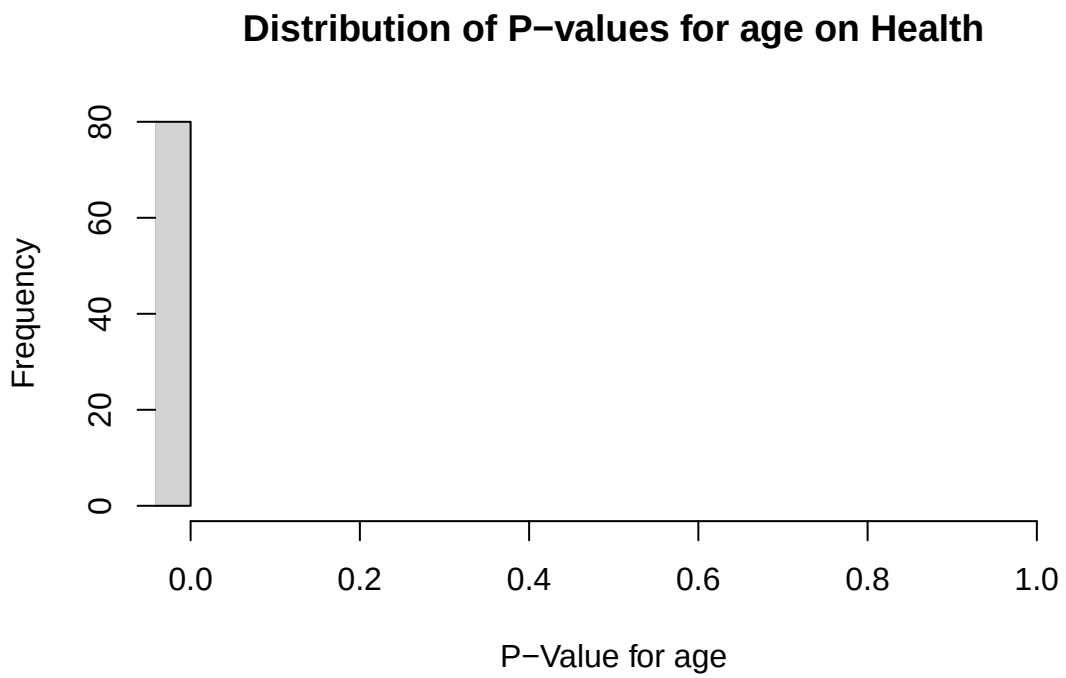
The histogram titled "Distribution of Respondents' Age" shows the age distribution of the surveyed population. The x-axis represents the age of respondents, ranging from approximately 20 to 100 years, while the y-axis displays the count of respondents within each age group. The distribution reveals a broad range of ages, with a noticeable concentration of respondents in the middle age groups, particularly between 40 and 65 years. This suggests that a significant portion of the survey participants are in their mid-life stages, which could have implications for the analysis of age-related health outcomes.

The histogram shows a relatively symmetrical distribution around the central age groups, peaking around the 50 to 55-year mark. The number of respondents begins to decline gradually after this peak, with fewer individuals represented in the older age brackets, particularly those over 75 years. Similarly, there is a moderate representation of younger adults, particularly those in their 20s and 30s, but the counts are lower compared to the middle-aged groups. This distribution highlights a diverse demographic, with a substantial representation of middle-aged and older adults, which is valuable for understanding how age influences health across different stages of life. The decline in the number of respondents in the higher age brackets might also reflect the lower population sizes and potential survey participation among older individuals.

B. Model Fitting

We will now fit a multinomial logistic regression model to explore the relationship between monthly net income and health status.

```
## [[1]]
##  [1] "ggplot2"   "dplyr"     "nnet"      "stats"     "graphics"  "grDevices"
##  [7] "utils"     "datasets"  "methods"   "base"
##
## [[2]]
##  [1] "ggplot2"   "dplyr"     "nnet"      "stats"     "graphics"  "grDevices"
##  [7] "utils"     "datasets"  "methods"   "base"
##
## [[3]]
##  [1] "ggplot2"   "dplyr"     "nnet"      "stats"     "graphics"  "grDevices"
##  [7] "utils"     "datasets"  "methods"   "base"
##
## [[4]]
##  [1] "ggplot2"   "dplyr"     "nnet"      "stats"     "graphics"  "grDevices"
##  [7] "utils"     "datasets"  "methods"   "base"
##
## [[5]]
##  [1] "ggplot2"   "dplyr"     "nnet"      "stats"     "graphics"  "grDevices"
##  [7] "utils"     "datasets"  "methods"   "base"
##
## [[6]]
##  [1] "ggplot2"   "dplyr"     "nnet"      "stats"     "graphics"  "grDevices"
##  [7] "utils"     "datasets"  "methods"   "base"
##
## [[7]]
##  [1] "ggplot2"   "dplyr"     "nnet"      "stats"     "graphics"  "grDevices"
##  [7] "utils"     "datasets"  "methods"   "base"
```



The histogram titled "Distribution of P-values for age on Health" illustrates the p-values obtained from a multinomial logistic regression analysis examining the effect of age on health outcomes. The x-axis represents the p-values, ranging from 0 to 1, while the y-axis shows the frequency of these p-values. A significant observation is the overwhelming concentration of p-values at or very near 0. This strong clustering of low p-values indicates that age is a highly significant predictor of health status for many individuals in the dataset. Essentially, it suggests that changes in age are strongly associated with variations in health outcomes, highlighting the importance of age as a demographic factor influencing health.

```
## Call:
## multinom(formula = health ~ age, data = data)
##
## Coefficients:
##   (Intercept)        age
## 2   -0.635365 0.03517557
## 3   -2.719083 0.06787202
## 4   -4.270025 0.07820457
## 5   -5.794732 0.08721977
##
## Std. Errors:
##   (Intercept)         age
## 2  0.07560192 0.001746265
## 3  0.09223799 0.001943601
## 4  0.12836445 0.002424978
## 5  0.20002449 0.003417109
##
## Residual Deviance: 31664.88
## AIC: 31680.88
```

The multinomial logistic regression model output shows the coefficients and standard errors for predicting the health status of respondents based on their age. The health status is a categorical variable with five levels: "VERY GOOD," "GOOD," "SATISFACTORY," "NOT THAT GOOD," and "BAD," with "VERY GOOD" as the reference category. The coefficients represent the log-odds of being in one of the health categories (2, 3, 4, 5) compared to the reference category (1: VERY GOOD) as a function of age.

For each health category: Health = 2 (GOOD): The coefficient for age is 0.03517557 with a standard error of 0.001746265. This positive coefficient indicates that as age increases, the log-odds of being in the "GOOD" health category compared to the "VERY GOOD" category increases. This suggests that older individuals are more likely to report their health as "GOOD" rather than "VERY GOOD." Health = 3 (SATISFACTORY): The coefficient for age is 0.06787202 with a standard error of 0.001943601. This larger positive coefficient indicates an even stronger relationship between increasing age and the likelihood of reporting health as "SATISFACTORY" compared to "VERY GOOD." Health = 4 (NOT THAT GOOD): The coefficient for age is 0.07820457 with a standard error of 0.002424978. This coefficient suggests that as age increases, individuals are much more likely to report "NOT THAT GOOD" health compared to "VERY GOOD." Health = 5 (BAD): The coefficient for age is 0.08721977 with a standard error of 0.003417109. This highest positive coefficient indicates that the probability of reporting "BAD" health increases significantly with age compared to "VERY GOOD" health. These results are consistent with common sense and existing literature on aging and health. As people age, they are more likely to experience health problems, leading to lower self-reported health status. The increasing coefficients with higher health status categories (from "GOOD" to "BAD") reflect the cumulative impact of aging on health, where older individuals are progressively more likely to report poorer health. This aligns with the general understanding that health tends to decline with age due to factors like the onset of chronic diseases, reduced physical fitness, and other age-related health issues.

C. Treatment Effect

In investigating the causal effect of age on health status, it is crucial to account for the potential confounding effects of other variables that are correlated with age. Variables such as income level (x723) and employment status are intrinsically linked to age, as older individuals often have more established careers and potentially higher incomes compared to younger individuals. If these factors are not properly accounted for, they can confound the analysis, leading to biased estimates of the true effect of age on health. By focusing on the treatment effect, we aim to isolate the impact of age itself on health status, excluding the influences of these correlated variables. This approach ensures a more accurate and reliable understanding of how changes in age specifically affect health outcomes, allowing for more targeted and effective policy interventions.

We will focus on using the significant variables for health status. Instead of using the entire dataset, we will create a subset of the dataset containing only the significant variables and use this subset for both stages of the LASSO regression. The significant variables are: x1, x2, x162, x163, x164, x595, x613, x615, x617, x657, x965, x1179.

```
## The In-Sample R-squared value is 0.4593819 .
```

The R-squared value obtained from the first stage LASSO regression is 0.459382. This R-squared value indicates that approximately 45.94% of the variance in the age variable (x633) is explained by the other predictors in the model. This suggests a moderate degree of correlation between age and the set of predictors used in the LASSO regression.

The R-squared value implies that the other predictors in the model have a moderate explanatory power regarding the age variable. This means that a substantial portion of the variation in age can be predicted based on the other variables in the dataset. Consequently, the remaining part of the age variation (about 54.06%) is not explained by these predictors, indicating a significant portion of age-related information that remains independent of the other variables.

In the context of assessing the treatment effect of age on health, this moderate R-squared value suggests that there is a considerable degree of confounding left unaccounted for by the predictors. Given that 45.94% of the variance in age is explained by the other variables, it implies that while age is somewhat predictable from these variables, a substantial residual variance in age exists. This residual variance could contribute to potential confounding, as the overlap between the treatment variable (age) and the other predictors is not complete.

However, this also means that the model has captured a substantial amount of the relevant information about age from the predictors, but there is still a considerable amount of age-related information that these

predictors do not account for. The treatment effect isolated from these predictors might therefore be more pronounced. It is essential to recognize that while the R-squared value indicates a moderate fit, it also suggests that age's unique contribution to health outcomes, independent of other variables, is still significant. Thus, the findings from the second stage regression, which assesses the effect of predicted age on health, should be interpreted with consideration of the moderate dependency of age on the other predictors. This moderate predictability allows for a more pronounced treatment effect, making it crucial to carefully analyze and account for the independent influence of age on health outcomes.

Causal LASSO

```
## The effect of predicted age (dhat) on health status for class 1 is 0 .
## The effect of predicted age (dhat) on health status for class 2 is -0.001971279 .
## The effect of predicted age (dhat) on health status for class 3 is 0 .
## The effect of predicted age (dhat) on health status for class 4 is 0 .
## The effect of predicted age (dhat) on health status for class 5 is 0.009641957 .
```

Naive LASSO

```
## The effect of d on health status from a naive lasso for class 1 is -0.0682922 .
## The effect of d on health status from a naive lasso for class 2 is -0.03219982 .
## The effect of d on health status from a naive lasso for class 3 is 0 .
## The effect of d on health status from a naive lasso for class 4 is 0.006873968 .
## The effect of d on health status from a naive lasso for class 5 is 0.01387759 .
```

Class 1: - Causal LASSO: The coefficient for dhat is 0. - Naive LASSO: The coefficient for d is -0.0682922. In the causal LASSO, a coefficient of 0 for dhat means that predicted age does not affect the probability of being in health status Class 1. In contrast, the naive LASSO shows a coefficient of -0.0682922, which implies that for every one-unit increase in age, the log-odds of being in Class 1 decreases by -0.0682922. This suggests that the naive LASSO's estimate may be confounded by other variables that the causal LASSO controls for, resulting in no effect in the causal model.

Class 2: - Causal LASSO: The coefficient for dhat is -0.0019713. - Naive LASSO: The coefficient for d is -0.0321998. In the causal LASSO, a coefficient of -0.0019713 for dhat means that for every one-unit increase in predicted age, the log-odds of being in Class 2 decreases slightly by -0.0019713. The naive LASSO shows a stronger negative effect with a coefficient of r coef_value2[2], implying that for every one-unit increase in age, the log-odds of being in Class 2 decreases by -0.0321998. The smaller coefficient in the causal LASSO suggests that some of the observed effect in the naive LASSO is due to confounding variables.

Class 3: - Causal LASSO: The coefficient for dhat is 0. - Naive LASSO: The coefficient for d is 0. Both models indicate that there is no effect of age on the log-odds of being in Class 3. This consistency suggests that for this class, the relationship between age and health status is not confounded by other variables.

Class 4: - Causal LASSO: The coefficient for dhat is 0. - Naive LASSO: The coefficient for d is 0.006874. The causal LASSO suggests no effect of predicted age on the log-odds of being in Class 4, with a coefficient of 0 for dhat. The naive LASSO shows a coefficient of 0.006874, meaning that for every one-unit increase in age, the log-odds of being in Class 4 increases slightly by 0.006874. This discrepancy suggests that the naive LASSO's coefficient is influenced by confounding variables, which the causal LASSO controls for.

Class 5: - Causal LASSO: The coefficient for dhat is 0.009642. - Naive LASSO: The coefficient for d is 0.0138776. In the causal LASSO, a coefficient of 0.009642 for dhat means that for every one-unit increase in predicted age, the log-odds of being in Class 5 increases by 0.009642. The naive LASSO shows a slightly larger coefficient of 0.0138776, indicating that for every one-unit increase in age, the log-odds of being in Class 5 increases by 0.0138776. The small difference between the two models suggests that while there is some confounding in the naive model, it does not substantially alter the observed effect of age on health status for this class.

The comparison shows that the naive LASSO often produces stronger effects than the causal LASSO, particularly for Classes 1 and 2. This pattern suggests that the naive LASSO's estimates are inflated due to

confounding variables that the causal LASSO adjusts for. The causal LASSO's coefficients, being closer to zero, indicate a more accurate estimation of the true effect of age, free from the bias introduced by correlated predictors.

For Class 1, the naive LASSO indicates a negative effect of -0.0682922, while the causal LASSO shows no effect (0), highlighting potential confounding in the naive model. Similarly, for Class 2, the naive LASSO's coefficient is -0.0321998 compared to -0.0019713 in the causal LASSO, again suggesting confounding.

For Class 3, both models agree that age has no effect on health status, suggesting no confounding. For Class 4, the naive LASSO indicates a slight positive effect (0.006874), which disappears in the causal model, indicating that the naive estimate was likely confounded. Finally, for Class 5, both models suggest a positive relationship, with the naive LASSO at 0.0138776 and the causal LASSO at 0.009642. The small difference indicates that while confounding is present, it does not significantly impact the observed effect for this class.

The coefficients from the causal LASSO are generally smaller in magnitude than those from the naive LASSO, which is consistent with the notion that naive estimates are inflated due to the presence of confounders. This observation underscores the importance of using methods like causal LASSO to obtain unbiased estimates of treatment effects.

The positive coefficients for age in Class 5 across both models suggest that, for this health status category, increasing age is associated with an improvement in health status or a higher likelihood of being in this class. However, the generally small coefficients (close to zero) across all classes indicate that the effect of age on health status is relatively weak.

D. Answer to the question

Based on the analysis, the causal LASSO results suggest that age has little to no causal effect on health status for most categories, with the exception of a small positive effect for the "BAD" health status category. This finding implies that age alone does not significantly determine health status, and other factors may play a more crucial role. The discrepancies between the naive and causal LASSO results highlight the importance of accounting for confounding variables to obtain accurate estimates of causal effects.

Specifically, the naive LASSO shows a negative coefficient for the "VERY GOOD" health status category, indicating that an increase in age is associated with a lower likelihood of being in this category. However, this effect disappears in the causal model, suggesting that the naive estimate was confounded by other variables. Similarly, the naive LASSO's stronger negative coefficient for the "GOOD" health status category is also reduced in the causal model, indicating that the observed relationship in the naive model was partly due to confounding.

The consistent null effect for the "SATISFACTORY" health status category in both the naive and causal models indicates that age does not have a significant impact on the likelihood of being in this health status category, irrespective of confounders. For the "NOT THAT GOOD" health status category, the naive LASSO suggests a slight positive effect of age, but this effect is not present in the causal model, further underscoring the role of confounding variables in the naive estimate.

The small positive effect observed in the causal LASSO for the "BAD" health status category, with a coefficient of 0.009641957, suggests that as age increases, the likelihood of being in the "BAD" health status category slightly increases. This effect is also seen in the naive LASSO, although it is slightly larger, indicating that while confounding is present, it does not significantly alter the observed effect for this category.

In summary, the analysis underscores that age alone does not have a substantial causal impact on most health status categories, except for a slight positive influence on the "BAD" health status. This highlights the nuanced role of age in determining health outcomes and emphasizes the importance of controlling for confounding variables to accurately assess causal relationships. The findings suggest that other factors beyond age may play a more significant role in influencing health status, and future research should focus on identifying and accounting for these factors to better understand the determinants of health outcomes.

6. Classification: Economic Situation in Germany

6.1 Objective

We aim to explore the various factors that influence the economic situation of individuals in Germany. Our study will primarily utilize features derived from both political and personal aspects, including education, religion, and housing conditions, to assess their impact on economic status.

In our recent lectures, we learned that decision tree models are not only interpretable but can also be highly effective in prediction tasks. To build on this foundation, our current research intends to compare the predictive performance of different tree-based models—specifically, simple classification trees, Random Forests, and boosting trees.

6.2 Data Preparation

Given that our target variable, x1:current economic situation, is highly skewed with fewer observations in the "VERY GOOD" and "VERY BAD" categories, we have opted to simplify the response levels to improve the model's predictive accuracy and streamline the classification process. Specifically, we will merge "GOOD" and "VERY GOOD" into a single "GOOD" category, and "BAD" and "VERY BAD" into "BAD". The remaining responses will be labeled as "NEUTRAL". This reclassification reduces the original five-level response to a more manageable three-level system, potentially reducing variability and enhancing the model's performance.By consolidating the categories in this manner, we aim to address the distribution's skewness and improve the stability and reliability of our subsequent analyses.

In the initial phase of our analysis, feature selection will be conducted manually to ensure relevance and potential for insight. Our selection process focuses on three primary aspects:

1. Political Engagement: We begin by exploring variables related to political engagement, which reflect individual participation and interest in political events. For example:

- x14:POLITICAL INTEREST and x17:VOTING INTENTION measure the level of interest and intention to vote, respectively.
- x19:DID YOU VOTE IN LAST FEDERAL ELECTION directly indicates past voting behavior, providing a concrete measure of political participation.

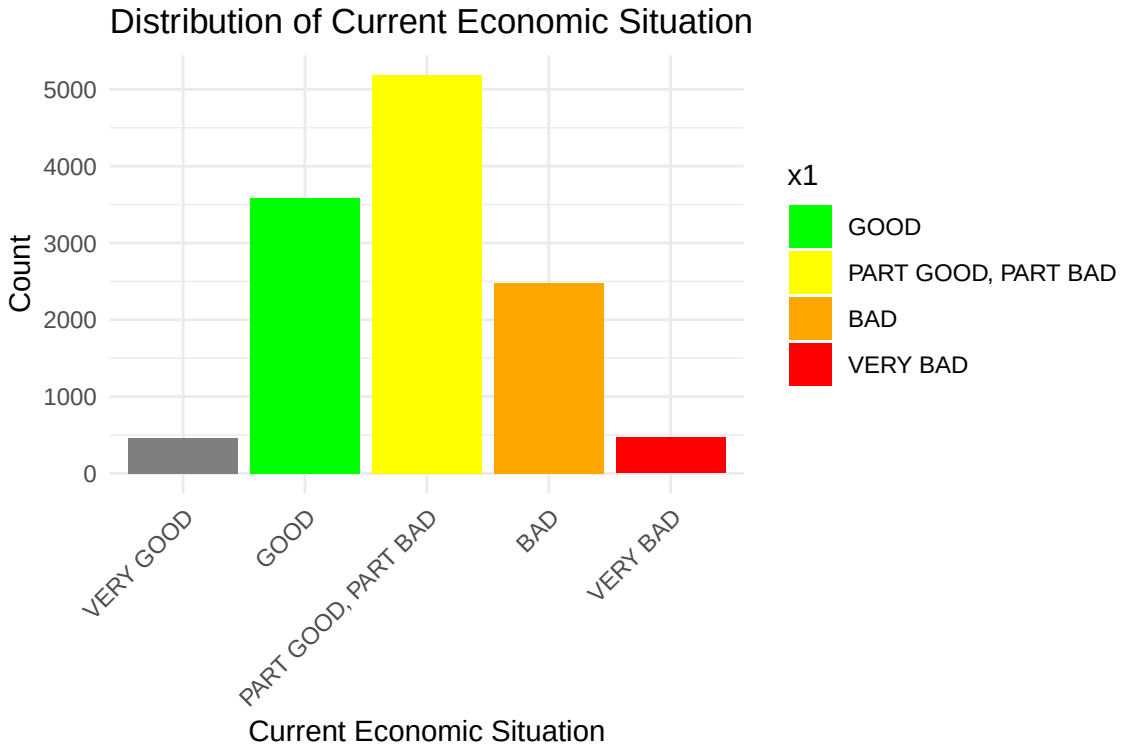
2. Socioeconomic Self-Assessment: This category encompasses features that allow individuals to self-report their socioeconomic status, which could influence or reflect their economic situation:

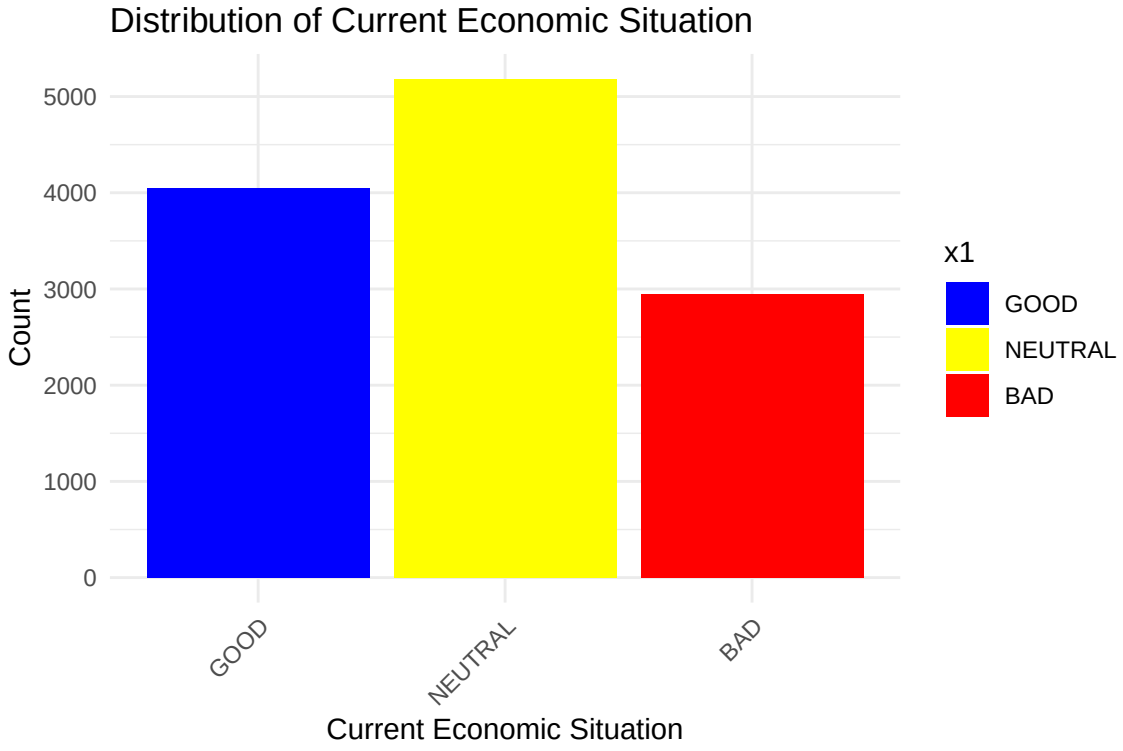
- x163:SELF-ASSESSMENT OF SOCIAL CLASS offers insights into how individuals perceive their social standing.
- x472:RELIGIOUS DENOMINATION and x1184:EDUCATION provide background on religious and educational contexts, respectively.
- x1184:HOUSEHOLD CLASSIFICATION might give additional data on living arrangements that correlate with economic conditions.

3. Perceptions of Economic and Social Trends: The final set of features captures individuals' perspectives on broader societal trends, which can impact and reflect their personal economic outlook:

- x614:LIFE IS GETTING WORSE FOR COMMON PEOPLE and x615:WITH SUCH A FUTURE, NO MORE CHILDREN are indicative of pessimism and concern about future prospects, potentially influencing economic behaviors and attitudes.

After the data preparation, there are 21 features from those aspects for further classification models.





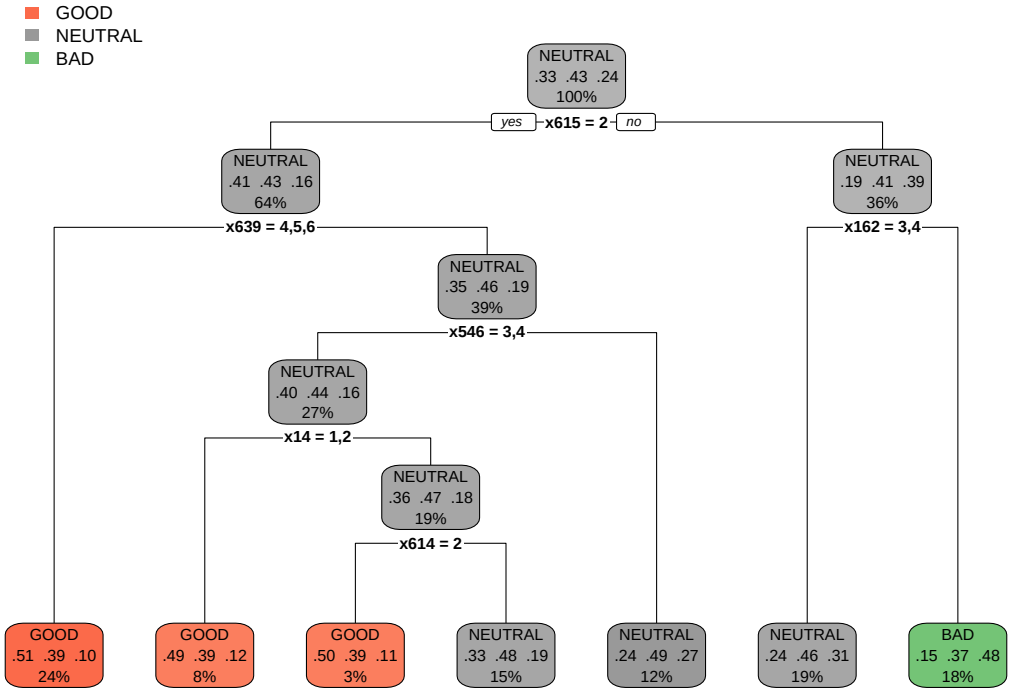
6.3 Classification Desicion Tree

For our decision tree analysis, we employ the rpart package, which excels in visualizing and managing categorical feature splits, akin to the basic tree function. Central to rpart is the complexity parameter (cp), a threshold that determines the minimum required improvement for a split, helping to prevent overfitting by avoiding unnecessary complexity in the model. Adjusting cp allows for fine-tuning the tree's depth, ensuring

the model remains interpretable and generalizes well to new data, thus optimizing both its accuracy and robustness.

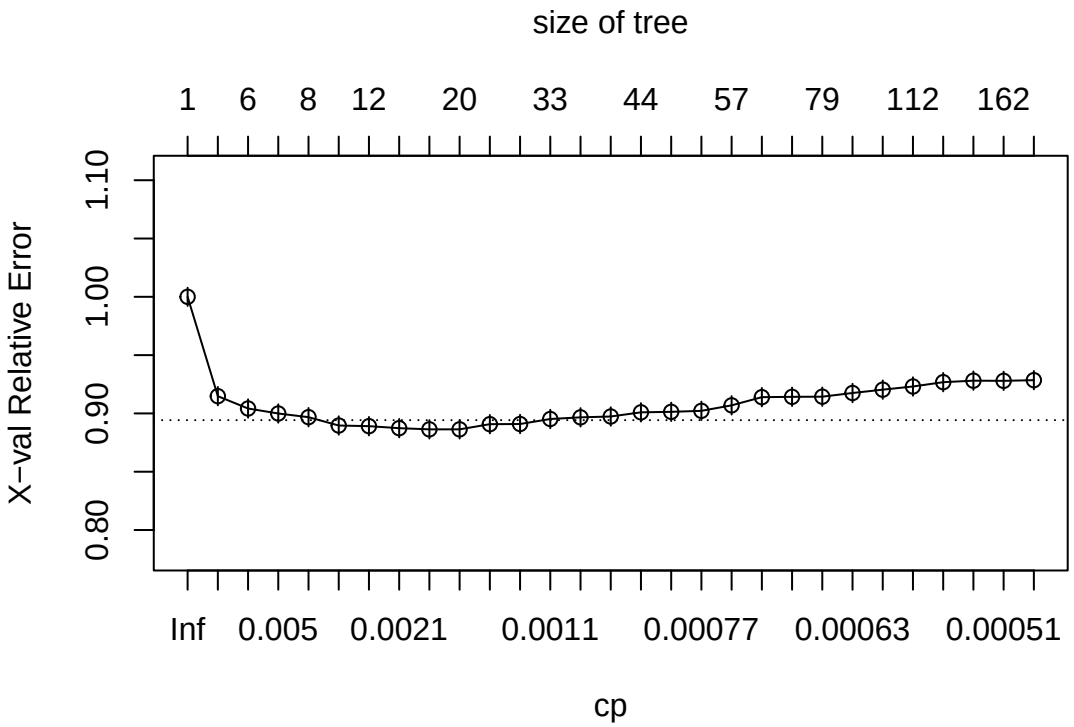
In our analysis, we utilized a shallow decision tree with 7 leaf nodes to ensure interpretability. Notably, one of the leaf nodes categorizes economic situations as "BAD", influenced primarily by responses to x615=2 ("WITH SUCH A FUTURE, NO MORE CHILDREN: Agree") and x162=3 ("FAIR SHARE IN STANDARD OF LIVING: Less"). This suggests that individuals concerned about future prospects and perceiving their share in the standard of living as inadequate are more likely to view their economic situation negatively. This aligns with the intuition that financial insecurities can diminish the desire for children due to perceived economic strains.

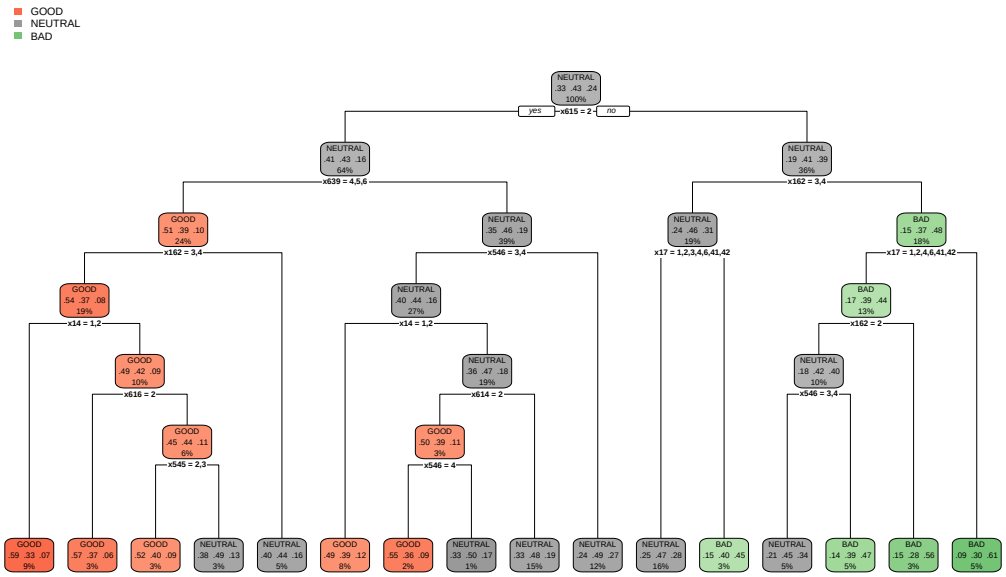
Conversely, several nodes categorized economic situations as "GOOD", particularly where responses indicated a lack of financial concern and satisfaction with their standard of living, which typically reflects a stable or prosperous economic condition. The accuracy of these "GOOD" predictions reaffirms the tree's effectiveness in capturing significant economic indicators that align with general expectations of wealth and economic stability.



To optimize the out-of-sample (OOS) performance of our decision tree model, we employed cross-validation using the rpart package in R. The decision tree was initially fit with a 10-fold cross-validation, setting the complexity parameter (cp) to a fine-tuned threshold of 0.0005 to balance tree depth with prediction accuracy.

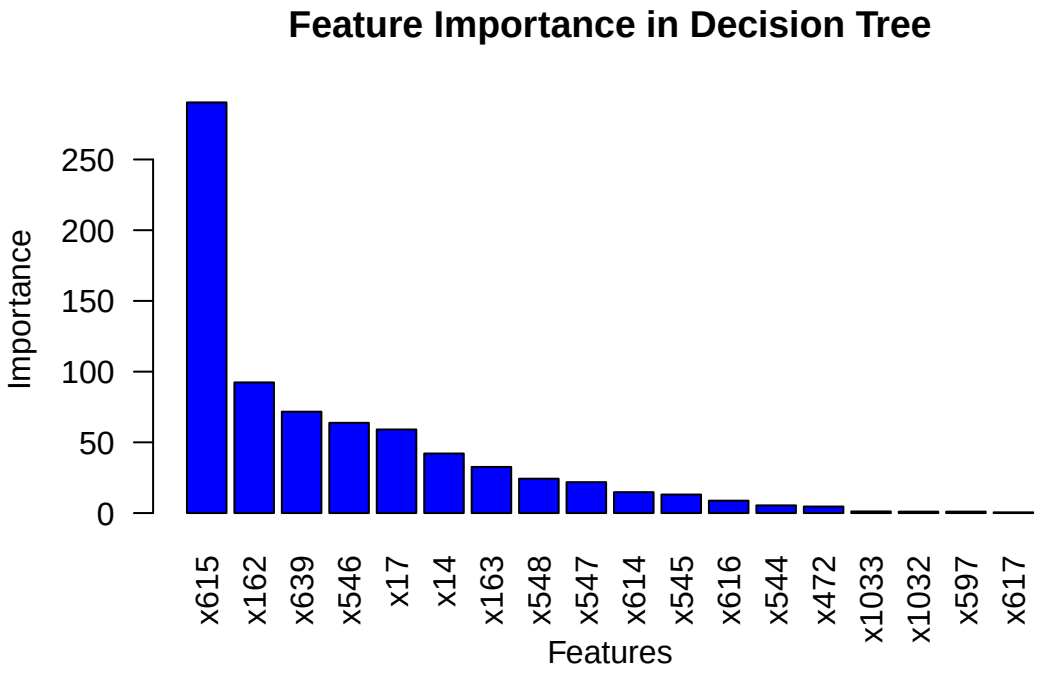
After fitting the tree, we scrutinized the cross-validation error rates against various tree sizes. The analysis revealed that a tree size of 22 nodes minimized the cross-validation error, indicating the most efficient model complexity for our dataset. Notably, this model configuration resulted in a cross-validation error substantially lower than a baseline approach of predicting a constant "NEUTRAL" for each case—exhibiting about a 10% improvement in accuracy. This demonstrates the added value of a precisely pruned model over simpler, less nuanced methods.






```
##           Actual
## Predicted GOOD NEUTRAL  BAD
##    GOOD    1784    1195  292
##    NEUTRAL 2026    3349 1674
##    BAD      237     639  982
```

In our decision tree analysis, the feature importance graph highlights that the variable x615:WITH SUCH A FUTURE, NO MORE CHILDREN is the most influential, indicating that attitudes towards future childrearing strongly predict economic outlooks, being 150% more important than the next significant factor, x162:FAIR SHARE IN STANDARD OF LIVING. Other notable contributors include x639:GENERAL SCHOOL LEAVING CERTIFICATE and x546:POLITICAL GOALS: FIGHT RISING PRICES, which show how educational achievements and economic concerns interplay in shaping perceptions of economic stability. This analysis emphasizes the critical impact of personal and societal outlooks on economic conditions.



We evaluated the performance of our decision tree model using the best tree derived from cross-validation to make predictions on the test set. The model demonstrated an accuracy of approximately 48%, closely aligning with the cross-validation findings, which suggests that the model effectively prevents overfitting. The confusion matrix reveals a notable tendency within the model to misclassify both "GOOD" and "BAD" categories as "NEUTRAL." This bias towards a "NEUTRAL" prediction may indicate a need for further model tuning or considering more complex ensemble methods such as bagging and boosting, which could enhance discrimination between classes and potentially improve overall accuracy.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction GOOD NEUTRAL  BAD
##    GOOD     750     561  124
##    NEUTRAL  873    1398  712
##    BAD       99     295  408
##
## Overall Statistics
##
##                Accuracy : 0.4897
##                  95% CI : (0.476, 0.5033)
##     No Information Rate : 0.4318
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.1847
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: GOOD Class: NEUTRAL Class: BAD
## Sensitivity               0.4355         0.6202    0.32797
## Specificity               0.8042         0.4656    0.90091
## Pos Pred Value            0.5226         0.4687    0.50873
```

```
## Neg Pred Value            0.7432         0.6173    0.81077
## Prevalence                0.3299         0.4318    0.23831
## Detection Rate            0.1437         0.2678    0.07816
## Detection Prevalence      0.2749         0.5715    0.15364
## Balanced Accuracy         0.6199         0.5429    0.61444
```

6.4 Ensemble Methods

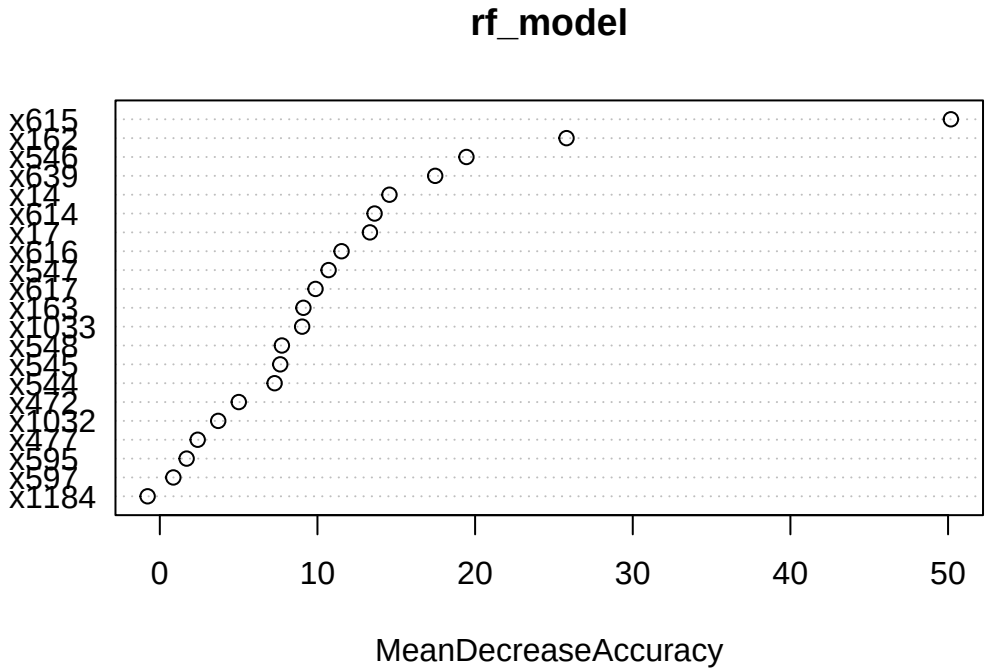
Ensemble learning is a machine learning paradigm where multiple models (often called "weak learners") are trained to solve the same problem and combined to get better results. The main hypothesis is that when weak models are correctly combined we can obtain more accurate and/or robust models. There are two major types of ensmble methods: 1. bagging, that often considers homogeneous weak learners, learns them independently from each other in parallel and combines them following some kind of deterministic averaging process;2. boosting, that often considers homogeneous weak learners, learns them sequentially in a very adaptative way (a base model depends on the previous ones) and combines them following a deterministic strategy.

The random forest approach is a bagging method where deep trees, fitted on bootstrap samples, are combined to produce an output with lower variance. However, random forests also use another trick to make the multiple fitted trees a bit less correlated with each others: when growing each tree, instead of only sampling over the observations in the dataset to generate a bootstrap sample, we also sample over features and keep only a random subset of them to build the tree.

The Random Forest classifier will be run 500 times to effectively reduce variance and enhance model stability. Despite these efforts, when tested on the dataset, the model exhibits an accuracy of approximately 48.9%, which is suboptimal. Interestingly, it still displays a propensity to categorize outcomes as "NEUTRAL" more frequently than might be expected. Notably, the four most significant features identified by the Random Forest closely match those determined by the simpler decision tree model, suggesting consistency in the features deemed most predictive across different modeling approaches.

```
##           Actual
## Predicted GOOD NEUTRAL  BAD
##   GOOD     813     644  149
##   NEUTRAL  785    1292  645
##   BAD      124     318  450
```

```
## [1] "Accuracy: 0.489463601532567"
```



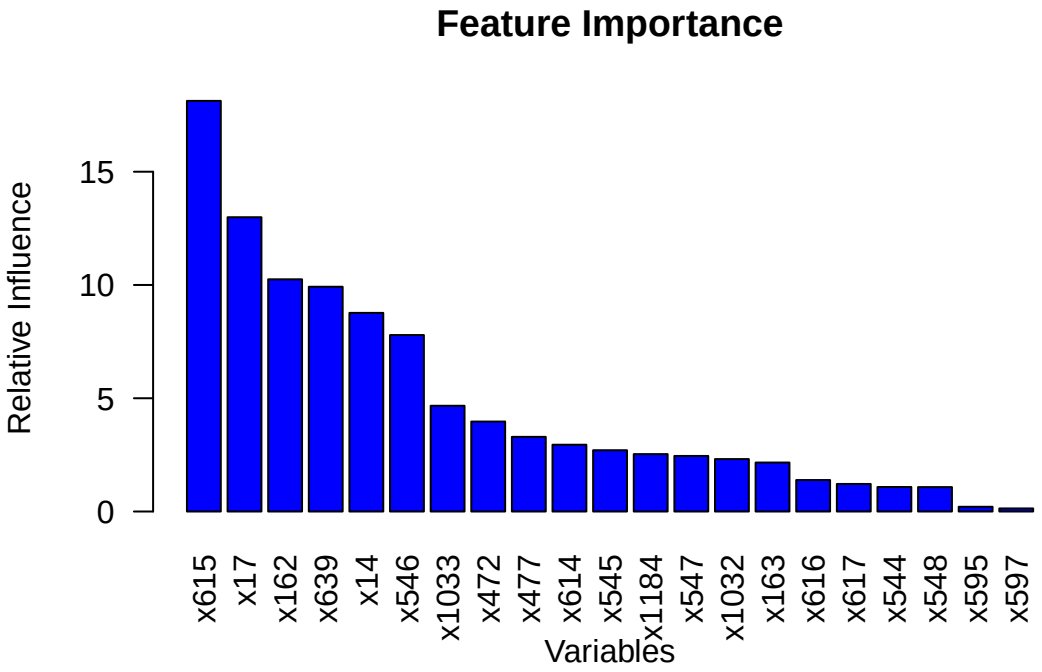
Boosting methods operate similarly to bagging techniques by aggregating a family of models to enhance performance, though with a key distinction. Unlike bagging, which primarily focuses on reducing variance by averaging predictions from various models, boosting strategically fits a sequence of weak learners in a highly adaptive manner. Each model in the sequence prioritizes misclassified observations by previous models, incrementally focusing on the most challenging data to transform a series of weak learners into a potent collective model with lower bias.

We proceeded to fit the gradient boosting tree classifier, setting the learning rate at 0.1 and iterating through 100 runs. The resulting out-of-sample (OOS) accuracy was 50.5%, mirroring the performance of a simpler tree model, suggesting no significant gain from the more complex boosting approach in this instance. Notably, the second most important feature identified by this model was x17: VOTING INTENTION: FEDERAL ELECTION, which underscores the potential influence of political engagement on economic perceptions. This feature's prominence indicates a nuanced change from previous models, highlighting the interconnectedness of political participation and economic outcomes.

```
## Warning: Setting `distribution = "multinomial"` is ill-advised as it is
## currently broken. It exists only for backwards compatibility. Use at your own
## risk.
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.0986             nan     0.1000    0.0306
##      2        1.0759             nan     0.1000    0.0234
##      3        1.0597             nan     0.1000    0.0170
##      4        1.0472             nan     0.1000    0.0148
##      5        1.0366             nan     0.1000    0.0112
##      6        1.0277             nan     0.1000    0.0095
##      7        1.0204             nan     0.1000    0.0079
##      8        1.0141             nan     0.1000    0.0065
##      9        1.0089             nan     0.1000    0.0058
##     10        1.0040             nan     0.1000    0.0039
##     20        0.9789             nan     0.1000    0.0010
##     40        0.9612             nan     0.1000   -0.0002
##     60        0.9530             nan     0.1000   -0.0006
##     80        0.9465             nan     0.1000   -0.0005
```

```
##     100         0.9411             nan     0.1000   -0.0006
```



6.5 Summary

In this analysis, we explore factors influencing the economic situation of individuals in Germany, focusing on three key aspects: Political Engagement, Socioeconomic Self-Assessment, and Perceptions of Economic and Social Trends. Using decision trees, Random Forests, and gradient boosting, we identified several critical drivers of economic outcomes. Notably, the desire to have children emerged as a significant predictor, alongside education level and living standards. The influence of political participation, as indicated by voting behavior, also proved pivotal.

To improve predictions of Germany's economic situation, we employed various classification tree models, achieving an out-of-sample accuracy of approximately 48.9%. While ensemble methods like Random Forest and gradient boosting slightly increased accuracy to around 50%, they required significantly more computational resources, which may not justify the marginal gain.

One potential issue identified is data imbalance, which predisposes the models to predict the most common outcome. To address this and potentially enhance model performance, oversampling techniques could be considered in future analyses.

7. What aspects have the most influence on German Self-Assessed Social Class? Are they similar to the aspects influencing Current Economic Situation?

Self-assessed social class and current economic situation are sometimes perceived as similar, but there are subtle differences between them. While self-assessed social class often reflects a broader, more stable perception of one's position in the social hierarchy, influenced by long-term factors such as education, occupation, and family background, the current economic situation tends to be more dynamic, reflecting immediate financial status and economic conditions, such as income, job security, and recent financial changes.

An interesting aspect to analyze is whether these two perceptions are driven by the same set of variables. By examining the underlying factors, we can determine if the same socio-economic dimensions influence both

self-assessed social class and the current economic situation. This analysis aims to provide insights into the similarities and differences in the determinants of these two important Socio-economic perceptions.

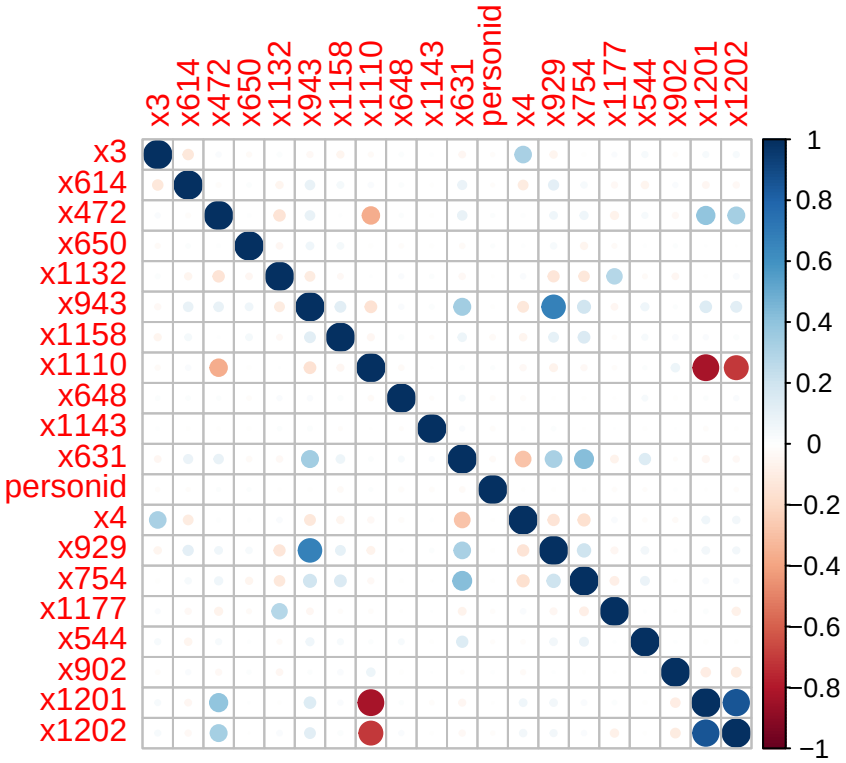
We now check the correlation matrix of the explanatory (X) variables in the cleaned training dataset. Since the cleaned dataset still has a very high dimension, we will randomly pick 20 from them to check for the correlation.

7.1 Multicollinearity analysis on the cleaned training data

Do you need all of this inside a report?

```
##                    x3        x614         x472         x650         x1132
## x3          1.000000000 -0.12053841  0.023741862 -0.022503052  0.0193155409
## x614       -0.120538410  1.00000000 -0.019006885  0.016153136 -0.0520640759
## x472        0.023741862 -0.01900688  1.000000000  0.006135557 -0.1459534087
## x650       -0.022503052  0.01615314  0.006135557  1.000000000 -0.0411656956
## x1132       0.019315541 -0.05206408 -0.145953409 -0.041165696  1.0000000000
## x943       -0.035339941  0.09762527  0.098595648  0.060047740 -0.1033611850
## x1158      -0.050269206  0.04590205  0.002369571  0.043556750 -0.0306965627
## x1110      -0.023071218  0.02629843 -0.360784889 -0.009174941  0.0009959477
## x648        0.015844163  0.01582366  0.013298502 -0.016495092  0.0217018756
## x1143       0.005360999 -0.00668389 -0.007361311 -0.008054305  0.0140253507
## x631       -0.043065438  0.08748141  0.096475609 -0.026836525 -0.0381429686
## personid   -0.013983802 -0.01166199  0.010954103  0.010790946 -0.0026012195
## x4          0.325897625 -0.10253714  0.009947759  0.006587101  0.0263186100
## x929       -0.054900507  0.11343616  0.060648937  0.038783814 -0.1392070054
## x754       -0.009999384  0.03820118  0.069892052 -0.051956510 -0.1213457421
## x1177       0.020348252 -0.03676133 -0.068368428 -0.020585941  0.2802471164
## x544        0.034901134 -0.05541302  0.033750332  0.004856587 -0.0203632708
## x902       -0.016998943  0.02376465 -0.035072256  0.012450604 -0.0413562352
## x1201       0.032749355 -0.04491715  0.395972131  0.001123453 -0.0045281309
## x1202       0.034515998 -0.03709926  0.338641019 -0.000120860  0.0298374977
##                   x943        x1158         x1110         x648         x1143
## x3         -0.03533994 -0.050269206 -0.0230712176  0.015844163  0.0053609987
## x614        0.09762527  0.045902045  0.0262984253  0.015823660 -0.0066838903
## x472        0.09859565  0.002369571 -0.3607848892  0.013298502 -0.0073613113
## x650        0.06004774  0.043556750 -0.0091749414 -0.016495092 -0.0080543046
## x1132      -0.10336118 -0.030696563  0.0009959477  0.021701876  0.0140253507
## x943        1.00000000  0.128720000 -0.1541768533  0.024892430  0.0192684446
## x1158       0.12872000  1.000000000 -0.0486101827  0.019887937 -0.0107417968
## x1110      -0.15417685 -0.048610183  1.0000000000 -0.026746057 -0.0006877045
## x648        0.02489243  0.019887937 -0.0267460569  1.000000000 -0.0107527878
## x1143       0.01926844 -0.010741797 -0.0006877045 -0.010752788  1.0000000000
## x631        0.34147919  0.076201796  0.0158826706  0.031591189  0.0210546456
## personid   -0.00709510 -0.028444974  0.0100974698 -0.019329578 -0.0177206792
## x4         -0.12911700 -0.055307427 -0.0373346260 -0.019682445  0.0064920140
## x929        0.67831098  0.104296774 -0.0628709041 -0.004947582  0.0150334473
## x754        0.19370712  0.155037344 -0.0344669914 -0.006865434 -0.0009712930
## x1177      -0.04299676 -0.014510602 -0.0155014423 -0.001740452  0.0001562966
## x544        0.06661416  0.034367398  0.0124164489  0.030965257 -0.0072078104
## x902        0.01171136  0.027112357  0.0748061680 -0.009846260  0.0001088021
## x1201       0.14892377  0.023589516 -0.8230485168  0.013492144 -0.0167559908
## x1202       0.12860985  0.019843472 -0.7032442494  0.024035904 -0.0063388654
##                   x631     personid           x4         x929         x754
## x3         -0.04306544 -0.013983802  0.325897625 -0.054900507 -0.009999384
## x614        0.08748141 -0.011661987 -0.102537142  0.113436155  0.038201178
## x472        0.09647561  0.010954103  0.009947759  0.060648937  0.069892052
```

```
## x650     -0.02683652  0.010790946  0.006587101  0.038783814 -0.051956510
## x1132    -0.03814297 -0.002601220  0.026318610 -0.139207005 -0.121345742
## x943      0.34147919 -0.007095100 -0.129117001  0.678310983  0.193707117
## x1158     0.07620180 -0.028444974 -0.055307427  0.104296774  0.155037344
## x1110     0.01588267  0.010097470 -0.037334626 -0.062870904 -0.034466991
## x648      0.03159119 -0.019329578 -0.019682445 -0.004947582 -0.006865434
## x1143     0.02105465 -0.017720679  0.006492014  0.015033447 -0.000971293
## x631      1.00000000 -0.027964384 -0.288243340  0.328409439  0.429571979
## personid -0.02796438  1.000000000 -0.004604511 -0.007885148 -0.005620255
## x4       -0.28824334 -0.004604511  1.000000000 -0.142796552 -0.166728572
## x929      0.32840944 -0.007885148 -0.142796552  1.000000000  0.205271089
## x754      0.42957198 -0.005620255 -0.166728572  0.205271089  1.000000000
## x1177    -0.06125328 -0.014321495  0.027095039 -0.047180136 -0.080527195
## x544      0.14208201 -0.010919912 -0.015200898  0.058867763  0.083647676
## x902     -0.01022154  0.004991090 -0.022474052  0.003428218 -0.004537474
## x1201    -0.04910247 -0.001871022  0.061312641  0.058234391  0.028795872
## x1202    -0.03716388 -0.014268685  0.055693550  0.050264339  0.025803649
##                 x1177         x544          x902        x1201        x1202
## x3       0.0203482521  0.034901134 -0.0169989429  0.032749355  0.034515998
## x614    -0.0367613306 -0.055413017  0.0237646497 -0.044917155 -0.037099255
## x472    -0.0683684282  0.033750332 -0.0350722560  0.395972131  0.338641019
## x650    -0.0205859408  0.004856587  0.0124506040  0.001123453 -0.000120860
## x1132    0.2802471164 -0.020363271 -0.0413562352 -0.004528131  0.029837498
## x943    -0.0429967567  0.066614155  0.0117113588  0.148923769  0.128609847
## x1158   -0.0145106025  0.034367398  0.0271123570  0.023589516  0.019843472
## x1110   -0.0155014423  0.012416449  0.0748061680 -0.823048517 -0.703244249
## x648    -0.0017404524  0.030965257 -0.0098462595  0.013492144  0.024035904
## x1143    0.0001562966 -0.007207810  0.0001088021 -0.016755991 -0.006338865
## x631    -0.0612532843  0.142082010 -0.0102215442 -0.049102469 -0.037163876
## personid -0.0143214955 -0.010919912  0.0049910898 -0.001871022 -0.014268685
## x4       0.0270950389 -0.015200898 -0.0224740525  0.061312641  0.055693550
## x929    -0.0471801361  0.058867763  0.0034282185  0.058234391  0.050264339
## x754    -0.0805271950  0.083647676 -0.0045374738  0.028795872  0.025803649
## x1177    1.0000000000  0.002941180 -0.0022913998  0.016450435 -0.071915874
## x544     0.0029411796  1.000000000  0.0029274432 -0.009892195 -0.006079661
## x902    -0.0022913998  0.002927443  1.0000000000 -0.096699493 -0.102778058
## x1201    0.0164504352 -0.009892195 -0.0966994928  1.000000000  0.852214002
## x1202   -0.0719158738 -0.006079661 -0.1027780582  0.852214002  1.000000000
```



Based on the observed correlation matrix, many variables have moderate correlations, some have strong correlations, indicating some degree of related movement. This is consistent with our findings of multicollinearity during the EDA. For instance, correlations between financial status and education level could suggest that higher education levels are associated with better financial situations. Moreover, there are some negative correlations, though they are relatively few and not very strong. These suggest that under certain conditions, some aspects (e.g., number of children in household and political opinions) may move in opposite directions, possibly due to different priorities or influences in household decision-making and political views.

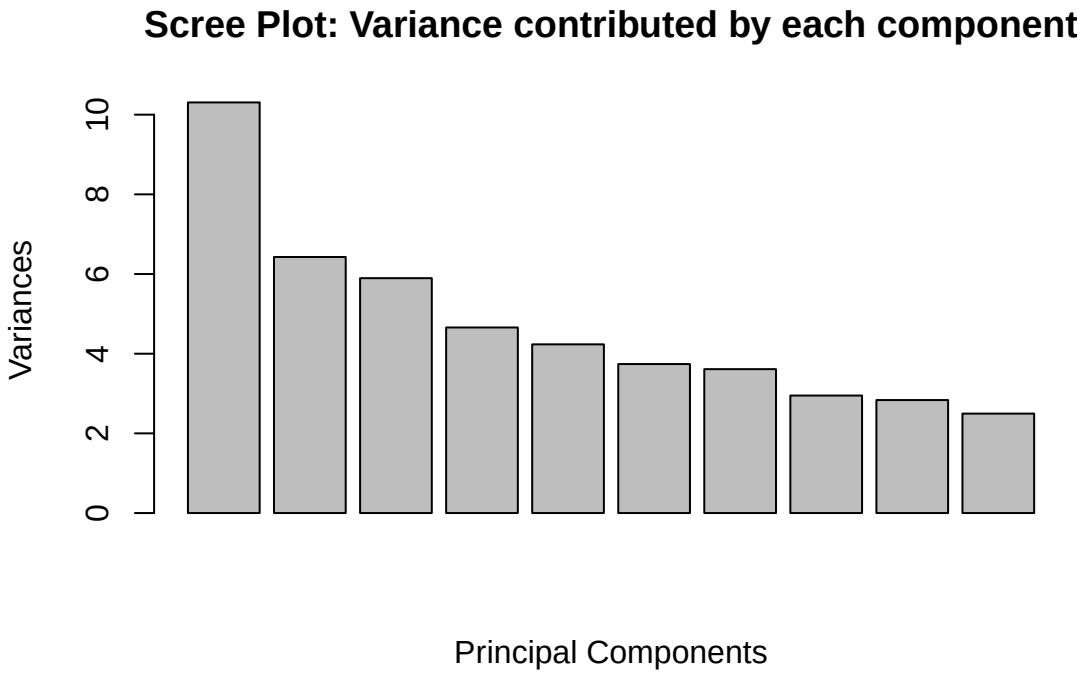
High correlations across several variables suggest that one or more common factors (like socio-economic status, educational background, or household structure) might be driving these correlations. Running regression on these factors directly might cause high multicollinearity, which violates one of the regression assumptions. Principal Component Analysis (PCA) can be useful in extracting these underlying common factors as latent variables to reduce dimensions and provide better predictions.

Therefore, by using PCA, we can effectively reduce multicollinearity, simplify our analysis, and potentially improve the predictive power of our models. This analysis will help us determine if self-assessed social class and current economic situation are influenced by the same set of variables, providing valuable insights into the determinants of these socio-economic perceptions.

7.2 Principal Component Analysis

A. Variance contribution - Scree Plot

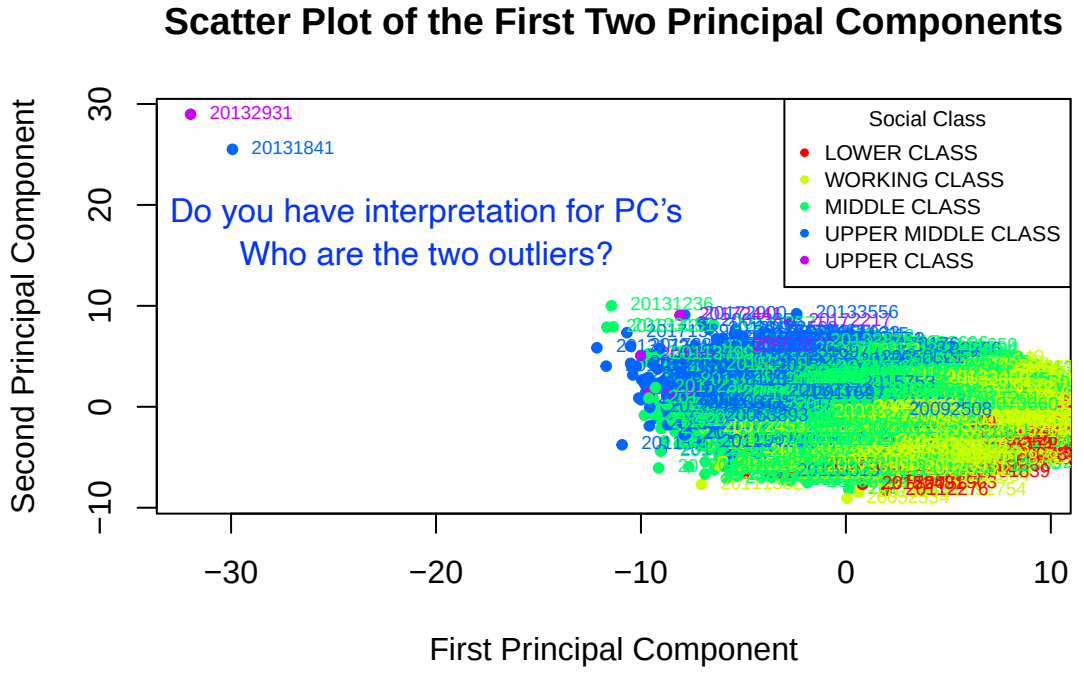
We now fit the PCA, and the scree plot below displays the variance contributed by each principal component (PC) in the PCA analysis, which helps determine the number of significant principal components to retain for further analysis.



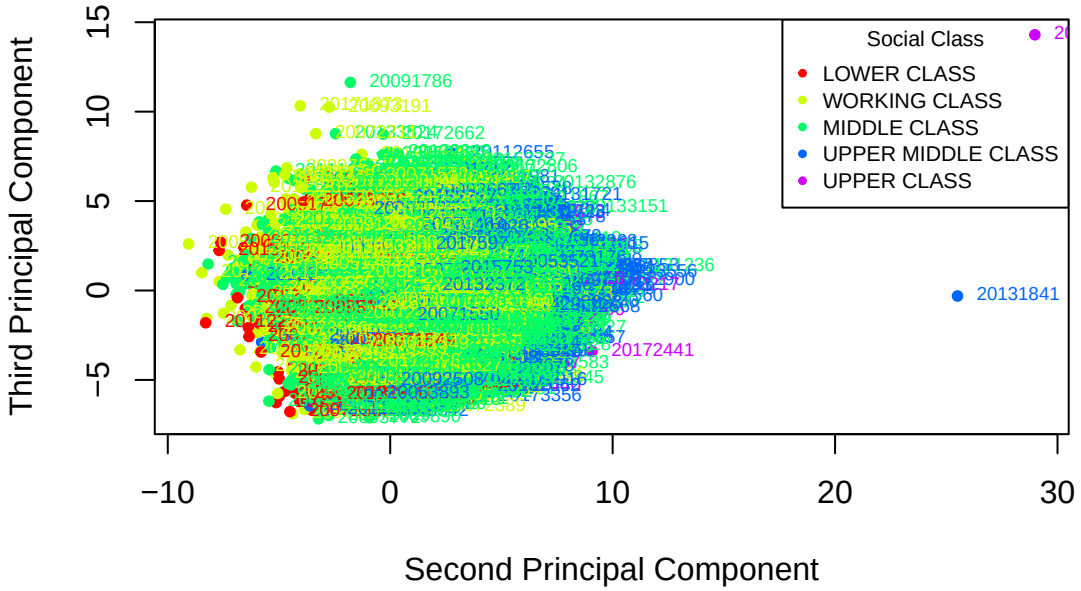
The first principal component (PC1) contributes the most significant variance, with a value of approximately 10. The second principal component (PC2) also contributes a substantial amount of variance, with a value around 7. Together, PC1 and PC2 capture the majority of the variance in the dataset, indicating they contain the most critical information about the underlying structure of the data.

To understand the contribution more clearly, we next scatterplot the PC1 versus PC2, PC2 versus PC3 with colored social class groups, which helps to determine their separation ability and thereby the significance better.

B. Principal component's effectiveness, relationships - Scatterplots



Scatter Plot of the Second and Third Principal Components

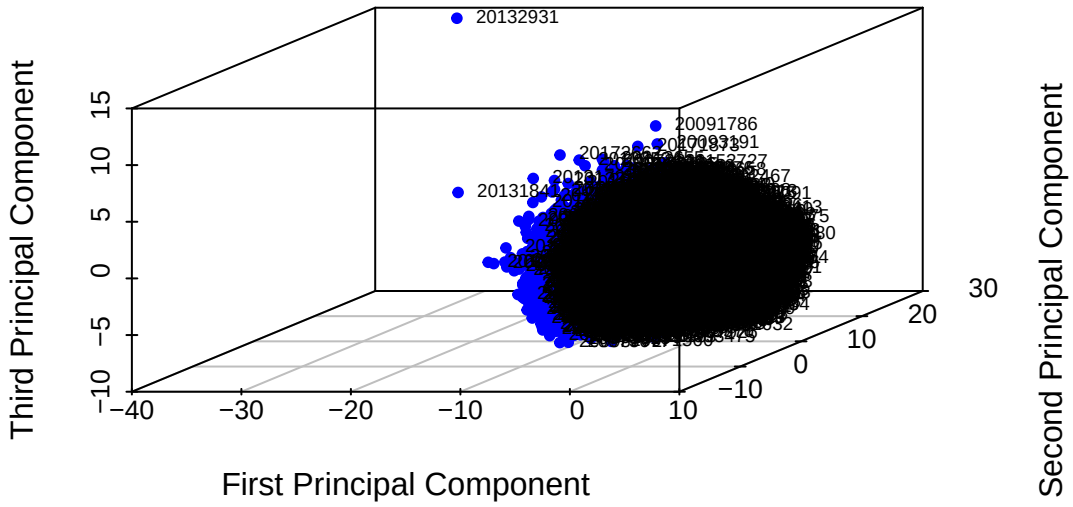


The scatter plot of the first two principal components (PC1 and PC2) shows a clear separation across different social class groups. Specifically, the larger the PC1 values, the lower the self-assessed social class. PC2 also shows a similar separation except that the smaller the PC2 values, the lower the social class, which suggests a latent factor opposite to PC1's latent factor's direction. This indicates that the first two principal components capture significant variance that differentiates between the social classes, suggesting these components are influenced by variables strongly related to social class. This makes sense as we see above that the first two PCs contribute the majority of variance.

The majority of the data points are clustered around the middle and lower class regions, represented by different colors. Notably, the upper-class individuals (in purple) are separated from the rest, indicating distinct characteristics captured by the first two principal components.

The second scatter plot shows that PC3 is ambiguous to read, so PC3 might be meaningless.

Scatter Plot of the First Three Principal Components



A few outliers are visible, particularly the point 20132931 far from the main cluster, which is due to missing data values and the imputed values might not fit well.

C. Principal components' interpretaions - Rotations

In order to insight the weightings of each feature in PC1 and PC2, we need to check their rotations/loadings:

```
##        Variable    Loading Abs_Loading  PC
## x6331      x633  0.2939757   0.2939757 PC2
## x6311      x631 -0.2757537   0.2757537 PC2
## x9691      x969  0.2322269   0.2322269 PC2
## x9671      x967 -0.2141797   0.2141797 PC2
## x639       x639 -0.2132930   0.2132930 PC1
## x10391    x1039 -0.2006375   0.2006375 PC2
## x896       x896 -0.1948971   0.1948971 PC1
## x7541      x754 -0.1941396   0.1941396 PC2
## x923       x923 -0.1899607   0.1899607 PC1
## x943       x943 -0.1861965   0.1861965 PC1
```

The largest loadings in terms of absolute values in PC1 and PC2 are:

- x633 (RESPONDENT: AGE) and x631 (RESPONDENT: YEAR OF BIRTH): Age and year of birth are closely related variables and likely contribute significantly to a principal component that captures the age-related demographic characteristics of the respondents.
- x969 (2.PERSON IN HOUSEHOLD: AGE): The age of the second person in the household could indicate household structure, potentially capturing the presence of a partner or child. This variable might be associated with family demographics or household composition.
- x639 (RESP: GENERAL SCHOOL LEAVING CERTIFICATE) and x896 (FATHER: GENERAL SCHOOL LEAVING CERTIFIC.): These variables reflect the educational background of the respondent and their father. These variables likely load heavily on a component related to educational attainment within families.
- x1039 (CHILDREN NOT LIVING IN THE HOUSEHOLD?): This variable indicates whether the respondent has children who do not live with them, which could be significant in understanding family dynamics and household composition.
- x754 (RESPONDENT: MARITAL STATUS): Marital status is a key demographic variable that often influences various aspects of a respondent's life, including economic status, social support, and living arrangements.
- x943 (MOTHER: ISCED 1997 - 5 LEVELS): The ISCED (International Standard Classification of Education) level of the mother reflects her educational attainment. This variable can be crucial in understanding the educational background and socio-economic status of the respondent's family.

The principal components with the greatest loadings seem to capture significant demographic and socio-economic characteristics of the respondents:

- Demographic Characteristics: Age of respondents and household members (x633, x631, x969).
- Educational Background: General school leaving certificates of respondents and their parents (x639, x896, x943).
- Family Structure: Marital status and the presence of children not living in the household (x1039, x754).

Based on the information above, we can interpret the first 2 principal components (the most significant ones) as follows:

- First Principal Component (PC1):

High Loadings: RESPONDENT: AGE (x633), RESPONDENT: YEAR OF BIRTH (x631), 2.PERSON IN HOUSEHOLD: AGE (x969) Interpretation: This component likely captures the age-related variance in the

data. It represents a combination of variables related to the ages of the respondent and other household members. This component may differentiate between younger and older respondents, reflecting generational differences.

- Second Principal Component (PC2):

High Loadings: RESP: GENERAL SCHOOL LEAVING CERTIFICATE (x639), FATHER: GENERAL SCHOOL LEAVING CERTIFIC. (x896), MOTHER: ISCED 1997 (x943) Interpretation: This component appears to capture educational attainment within families. It combines the educational levels of the respondent and their parents, highlighting differences in educational backgrounds. This component may be indicative of socio-economic status and educational mobility across generations.

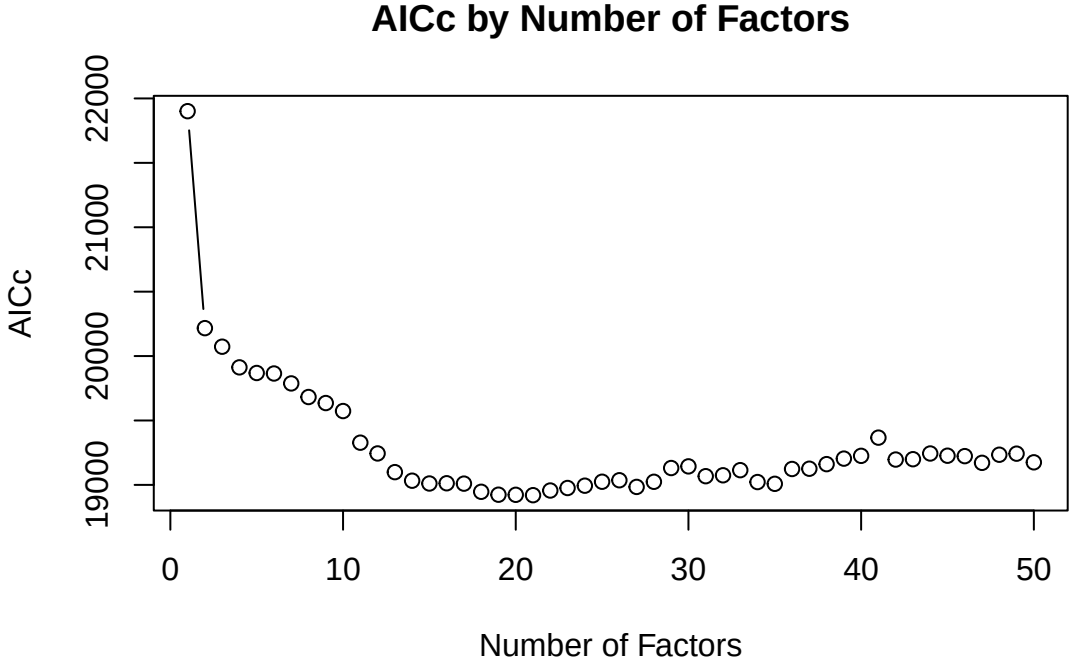
7.3 PCR (principal component regression) using 'logistic regression on first K' technique

A. Model fitting

We now conduct a multinomial PCR (principal component regression) to determine which PCs are the most significant ones affecting self-assessed social class. We regress x163 (self-assessed social class) over the principal components, using the 'logistic regression on first K' technique.

```
## Best number of PCs (based on AICc): 21
```

```
## Best number of PCs (based on BIC): 14
```




```
## Call:
## multinom(formula = response_variable ~ ., data = zdf[, 1:K, drop = FALSE])
##
## Coefficients:
##   (Intercept)        PC1        PC2       PC3       PC4         PC5         PC6
## 2   4.2225232 -0.3691169 -0.4822853 0.2375873 0.1000593 -0.25346245 -0.07232383
## 3   5.2175098 -0.7793623 -0.7828957 0.2783476 0.2221916 -0.26364448 -0.11812959
## 4   2.2425665 -1.1739541 -1.0709534 0.3593881 0.3276563 -0.23752458 -0.14157650
## 5  -0.7689582 -0.9577600 -1.1825254 0.2453662 0.3964626 -0.07822814 -0.01831439
##           PC7       PC8        PC9       PC10      PC11       PC12      PC13
## 2 -0.02512360 0.2223804 -0.2539794 -0.2439826 0.1846035 0.03354894 0.2313167
```

```
## 3  0.05669684 0.3393915 -0.2745608 -0.1378456 0.3647080 0.17652691 0.3974027
## 4  0.17491575 0.4436561 -0.2632593 -0.1153585 0.5330706 0.24336885 0.4924532
## 5  0.11711188 0.4075234 -0.5223447 -0.2606658 0.4446328 0.21175000 0.4367094
##        PC14        PC15         PC16       PC17        PC18       PC19
## 2 0.1247352  0.03608592  0.019646392 0.03118427 -0.06717532 -0.2037766
## 3 0.2500701  0.12113637 -0.001028981 0.08483739 -0.19190159 -0.1178231
## 4 0.2772440  0.10878180  0.039098286 0.05692380 -0.29636164 -0.1374943
## 5 0.2923292 -0.18220050 -0.209556931 0.07144711 -0.17602482 -0.1027219
##          PC20         PC21
## 2 -0.01761564  0.033335808
## 3 -0.03040913 -0.026819053
## 4 -0.05877150 -0.007473357
## 5 -0.32719996  0.085243686
##
## Std. Errors:
##   (Intercept)        PC1        PC2        PC3        PC4        PC5        PC6
## 2   0.1483510 0.03292079 0.03172066 0.02314459 0.02424775 0.03416386 0.03255733
## 3   0.1481590 0.03378111 0.03281484 0.02401886 0.02537536 0.03480794 0.03341487
## 4   0.1606657 0.03698687 0.03667793 0.02822637 0.03293056 0.03873572 0.03781259
## 5   0.2863488 0.05646671 0.06739497 0.06322338 0.08119197 0.07317185 0.07439420
##          PC7        PC8        PC9       PC10       PC11       PC12       PC13
## 2 0.03246595 0.03984811 0.03852346 0.04449996 0.03871315 0.03876557 0.04822611
## 3 0.03345329 0.04093722 0.03960677 0.04572616 0.04006861 0.04010309 0.04929690
## 4 0.03854276 0.04586870 0.04507525 0.05188616 0.04717861 0.04693427 0.05415287
## 5 0.08145777 0.08599636 0.08958284 0.09224177 0.09978420 0.09854890 0.08900124
##         PC14       PC15       PC16       PC17       PC18       PC19       PC20
## 2 0.04771180 0.04623317 0.04384038 0.04450511 0.04298443 0.05091569 0.04925215
## 3 0.04866814 0.04713857 0.04516346 0.04579884 0.04455134 0.05158227 0.05050365
## 4 0.05399219 0.05287064 0.05255802 0.05309284 0.05329662 0.05711907 0.05689951
## 5 0.09261482 0.10067103 0.11169199 0.10564867 0.11743976 0.10188011 0.10876888
##         PC21
## 2 0.04759489
## 3 0.04897929
## 4 0.05651411
## 5 0.10885710
##
## Residual Deviance: 18742.78
## AIC: 18918.78
```

According to the first-K technique of PCR, PC1 and PC2 have relatively large coefficients across categories and lower standard errors compared to other PCs, indicating their significant contribution to the model. For instance, the coefficients for PC1 and PC2 are notably large in magnitude, particularly for categories 3 and 4. The coefficients for PC1 and PC2 are consistently larger in absolute value compared to other principal components. This indicates that these components have a stronger effect on the response variable.

The standard errors for PC1 and PC2 are relatively low, resulting in higher z-values (coefficient divided by standard error), which suggests that these components are statistically significant predictors.

While PCs like PC3, PC4, and PC5 also have non-negligible coefficients, their contributions are smaller compared to PC1 and PC2. Some higher-order PCs have even smaller coefficients and higher standard errors, indicating less significance.

B. Prediction based on test dataset, model evaluation

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction    1    2    3    4    5
##          1   18   10    7    0    0
##          2  110  943  475    4    4
##          3   27  682 2409  362   13
##          4    0    1   63   83    9
##          5    0    0    0    0    0
##
## Overall Statistics
##
##                Accuracy : 0.6615
##                  95% CI : (0.6485, 0.6743)
##     No Information Rate : 0.5659
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3569
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity          0.116129   0.5764   0.8155  0.18486 0.000000
## Specificity          0.996644   0.8345   0.5216  0.98470 1.000000
## Pos Pred Value       0.514286   0.6139   0.6897  0.53205      NaN
## Neg Pred Value       0.973578   0.8119   0.6844  0.92773 0.995019
## Prevalence           0.029693   0.3134   0.5659  0.08602 0.004981
## Detection Rate       0.003448   0.1807   0.4615  0.01590 0.000000
## Detection Prevalence 0.006705   0.2943   0.6692  0.02989 0.000000
## Balanced Accuracy    0.556386   0.7055   0.6686  0.58478 0.500000
```

The evalaution above shows that accuracy is 0.6651. This indicates that the model correctly predicted 66.51% of the instances. While this is above the No Information Rate (NIR) of 0.5659, it shows room for improvement.

The model performs well for Class 3 with a high recall (0.8155), indicating that it correctly identifies most instances of this class. The sensitivity for Class 1 and Class 5 is very low, indicating that the model struggles to identify these classes correctly. The balanced accuracy for the model (average of sensitivity and specificity) is moderate for some classes but poor for others, indicating an uneven performance across different classes.

Overall, there appears to be a significant class imbalance, particularly with Class 5 having no correct predictions. This imbalance likely affects the overall performance metrics and suggests a need for handling class imbalance, possibly through techniques like oversampling, undersampling, or using class weights.

7.4 Answering the motivating question

The analysis identified the most significant factors influencing German Self-Assessed Social Class through principal component analysis (PCA), which reveals that age-related factors (captured by PC1) and educational attainment within families (captured by PC2) are the most significant factors. These findings suggest that both generational dynamics and educational backgrounds play crucial roles in how individuals perceive their social status in Germany.

PC1:High Loadings: RESPONDENT: AGE (x633), RESPONDENT: YEAR OF BIRTH (x631), 2.PERSON IN HOUSEHOLD: AGE (x969) The strong association with age-related variables suggests that generational factors and household age composition significantly influence one's self-assessed social class in Germany.

PC2:High Loadings: RESP: GENERAL SCHOOL LEAVING CERTIFICATE (x639), FATHER: GEN-

ERAL SCHOOL LEAVING CERTIFIC. (x896), MOTHER: ISCED 1997 (x943) The influence of educational attainment variables underscores the importance of educational background and familial educational achievements in shaping one's perception of their social class.

Our analysis for the second motivating question shows that: WITH SUCH A FUTURE, NO MORE CHILDREN (x615), RESP: FAIR SHARE IN STANDARD OF LIVING? (x162), RESP: GENERAL SCHOOL LEAVING CERTIFICATE (x639), POLITICAL GOALS: FIGHT RISING PRICES (x546) are the variables impacting current economical situation in Germany the most.

Common Influence: Both self-assessed social class and current economic situation are influenced by the respondent's educational attainment (x639), highlighting the importance of education in socio-economic perceptions.

Distinct Influences: Self-Assessed Social Class: Strongly influenced by age-related factors and the educational background of the family. Current Economic Situation: Influenced by future outlook regarding children, perception of fairness in the standard of living, political priorities, and immediate financial concerns like living costs and children raising.

Conclusively, while there is some overlap in the factors influencing self-assessed social class and the current economic situation, key differences highlight the nuanced nature of these perceptions. Self-assessed social class reflects broader, long-term socio-economic factors, particularly generational and educational backgrounds. In contrast, the current economic situation is influenced by immediate concerns about future prospects, fairness in the standard of living, political priorities, and the costs associated with living and raising children. This distinction underscores the complexity of socio-economic perceptions and the varied dimensions that contribute to how individuals perceive their social and economic standing.

8. Conclusion

This report investigates various socio-economic aspects influencing individual perceptions and statuses within German society. The analysis is structured around four primary questions, each addressing different dimensions of social class, health status, economic situation, and the intricate relationships between these variables. Through a comprehensive approach involving data exploration, cleaning, imputation, statistical modeling, and machine learning techniques, we aim to uncover significant factors and their interdependencies.

Initially, we undertook extensive data cleaning and imputation to ensure the quality and completeness of our dataset. This process involved handling missing values, correcting inconsistencies, and normalizing data where necessary. Specifically, missing values were imputed using suitable techniques such as mean imputation for numerical variables and mode imputation for categorical variables. This rigorous data preprocessing ensured that our analyses were based on robust and reliable data.

Following data cleaning, we conducted an Exploratory Data Analysis (EDA) to understand the distribution of variables, detect any anomalies, and identify potential relationships. EDA included generating summary statistics, visualizing distributions through histograms and box plots, and examining correlations using heatmaps. This initial exploration provided crucial insights and guided our subsequent modeling efforts by highlighting key variables and potential interaction effects.

Firstly, we explored the determinants of German Self-Assessed Social Class. Principal Component Analysis (PCA) revealed that age-related factors and educational attainment within families are the most significant influences. Specifically, the first principal component (PC1) is strongly associated with age-related variables, suggesting that generational dynamics and household age composition significantly influence one's self-assessed social class. The second principal component (PC2) underscores the importance of educational background and familial educational achievements, indicating that both personal and parental education levels play crucial roles in shaping individuals' perceptions of their social status.

Next, we examined the factors impacting the current economic situation in Germany. Our analysis identified that future outlook regarding children, perception of fairness in the standard of living, educational attainment, and political priorities are the most significant influences. The variable indicating concerns about

future prospects and the economic feasibility of raising children emerged as a particularly strong predictor, reflecting the substantial impact of financial insecurities on individuals' economic perceptions. Additionally, the perceived fairness in the standard of living and immediate financial concerns, such as living costs and political goals, notably influence how individuals assess their current economic situation.

To delve deeper into the relationship between age and health status, we applied causal LASSO techniques, which suggested that age alone does not have a substantial causal impact on most health status categories. The analysis highlighted that other factors beyond age might play more significant roles in influencing health outcomes, emphasizing the complexity of health determinants. This finding was further corroborated by comparing naive and causal models, revealing the importance of accounting for confounding variables to obtain accurate estimates of causal effects.

Our investigation into the classification of the economic situation employed decision trees, Random Forests, and gradient boosting techniques. We found that while decision trees provided interpretable insights, ensemble methods like Random Forests and gradient boosting slightly increased prediction accuracy. However, the marginal gain in accuracy from these more complex models did not justify the significantly higher computational resources required. A notable finding from this part of the analysis was the issue of data imbalance, which predisposed the models to predict the most common outcome. This suggests that future analyses could benefit from addressing data imbalance through techniques such as oversampling.

Finally, by comparing the influences on self-assessed social class and current economic situation, we observed some overlap, particularly in the impact of educational attainment. However, key differences were also evident. While self-assessed social class reflects broader, long-term socio-economic factors, particularly generational and educational backgrounds, the current economic situation is influenced by more immediate concerns. These include future prospects, perceptions of fairness, political priorities, and the costs associated with living and raising children. This distinction highlights the nuanced nature of socio-economic perceptions and underscores the varied dimensions contributing to how individuals perceive their social and economic standing.

In conclusion, our comprehensive analysis elucidates the complex interplay of factors influencing socio-economic perceptions in Germany. The findings underscore the importance of both long-term educational and generational factors, as well as immediate economic concerns, in shaping individuals' views on their social class and economic situation. These insights have important implications for policymakers aiming to address socio-economic disparities and enhance the well-being of the population. Future research should continue to explore these relationships, particularly considering the dynamic nature of economic conditions and their impact on social perceptions.

I thought this project was a bit confused. I was disappointed by the EDA where your plots did not make much of a sense. I appreciated the tree analysis.

I was missing out of sample comparisons. Your report was messy. I was hoping for a more coherent story with minimal R output copy pasted.

Appendix

```r
knitr::opts_chunk$set(echo = FALSE,fig.align='center',cache=FALSE)
# Function to check if a package is installed and install it if not
install_if_missing <- function(packages) {
  for (pkg in packages) {
    if (!requireNamespace(pkg, quietly = TRUE)) {
      install.packages(pkg)
    }
    library(pkg, character.only = TRUE)
  }
}

# List of necessary packages
packages <- c("dplyr", "ggplot2", "readr", "reshape2", "gridExtra", "GGally", "corrplot", "scatterplot3

# Install and load packages
install_if_missing(packages)

raw_data <- read_csv("train.csv")
raw_data$health <- factor(raw_data$health, levels = c(1, 2, 3, 4, 5),
                          labels = c("VERY GOOD", "GOOD", "SATISFACTORY", "NOT THAT GOOD", "BAD"))
dimensions <- dim(raw_data)

# Select the first 20 columns
raw_data_first_20 <- raw_data %>% dplyr::select(1:40)

# Calculate missing values and percentages for the first 20 columns
missing_values_first_20 <- colSums(is.na(raw_data_first_20))
total_entries_first_20 <- nrow(raw_data_first_20)
missing_values_percentage_first_20 <- (missing_values_first_20 / total_entries_first_20) * 100
missing_values_percentage_first_20 <- round(missing_values_percentage_first_20, 2)

# Calculate total missing percentage for the entire dataset
total_missing_values <- sum(is.na(raw_data))
total_entries_all <- prod(dim(raw_data))
total_missing_percentage <- (total_missing_values / total_entries_all) * 100
total_missing_percentage <- round(total_missing_percentage, 2)

# Convert to percentage format with '%' sign
missing_values_percentage_first_20 <- paste0(missing_values_percentage_first_20, "%")
total_missing_percentage <- paste0(total_missing_percentage, "%")

# Print the properties
cat("Dimensions:\n", dimensions, "\n\n")
cat("Missing Values by Column (Percentage) for the First 40 Columns:\n")
print(missing_values_percentage_first_20)
cat("\nTotal Missing Percentage for the Entire Dataset:", total_missing_percentage, "\n\n")

# Add a row number column to the dataset
raw_data$row_number <- 1:nrow(raw_data)

# Scatterplot of row number vs health status
```

```
scatter_plot <- ggplot(raw_data, aes(x = row_number, y = health)) +
  geom_point(alpha = 0.009) +
  labs(title = "Scatterplot of Health Status", x = "Row Number", y = "Health Status") +
  theme_minimal() +
  theme(plot.title = element_text(size = 10))  # Reduce the plot title size

# Print the scatterplot with increased width
print(scatter_plot)
# Histogram of the response variable 'health' with different colors
histogram_health <- ggplot(raw_data, aes(x = health, fill = health)) +
  geom_bar() +
  labs(title = "Dist of Health Status", x = "Health Status", y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(size = 10)) +  # Reduce the plot title size
  scale_fill_manual(values = c("VERY GOOD" = "blue", "GOOD" = "green", "SATISFACTORY" = "yellow", "NOT

# Log-transformed histogram with different colors
# First, compute counts and then log-transform
health_counts <- raw_data %>%
  group_by(health) %>%
  summarise(count = n())

log_transformed_health <- ggplot(health_counts, aes(x = health, y = log(count), fill = health)) +
  geom_bar(stat = "identity") +
  labs(title = "Log-Trans Dist of Health Status", x = "Health Status", y = "Log(Count)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(size = 10)) +  # Reduce the plot title size
  scale_fill_manual(values = c("VERY GOOD" = "blue", "GOOD" = "green", "SATISFACTORY" = "yellow", "NOT

# Arrange the plots in a single row with increased width
options(repr.plot.width = 16, repr.plot.height = 6)
grid.arrange(histogram_health, log_transformed_health, ncol = 2)

# Convert x1 to a factor with appropriate labels
raw_data$x1 <- factor(raw_data$x1, levels = c(1, 2, 3, 4, 5),
                      labels = c("VERY GOOD", "GOOD", "PART GOOD, PART BAD", "BAD", "VERY BAD"))

# Add a row number column to the dataset
raw_data$row_number <- 1:nrow(raw_data)

# Scatterplot of row number vs current economic situation
scatter_plot_x1 <- ggplot(raw_data, aes(x = row_number, y = x1)) +
  geom_point(alpha = 0.009) +
  labs(title = "Scatterplot of Current Econ Situat", x = "Row Number", y = "Current Economic Situation"
  theme_minimal() +
  theme(plot.title = element_text(size = 10))  # Reduce the plot title size

# Print the scatterplot with increased width
print(scatter_plot_x1)
# Histogram of the variable x1 with different colors
histogram_x1 <- ggplot(raw_data, aes(x = x1, fill = x1)) +
```

```
  geom_bar() +
  labs(title = "Dist of Current Econ Situat", x = "Current Economic Situation", y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(size = 10)) +  # Reduce the plot title size
  scale_fill_manual(values = c("VERY GOOD" = "blue", "GOOD" = "green", "PART GOOD, PART BAD" = "yellow"

# Log-transformed histogram with different colors
# First, compute counts and then log-transform
x1_counts <- raw_data %>%
  group_by(x1) %>%
  summarise(count = n())

log_transformed_x1 <- ggplot(x1_counts, aes(x = x1, y = log(count), fill = x1)) +
  geom_bar(stat = "identity") +
  labs(title = "Log-Trans Dist of Current Econ Situat", x = "Current Economic Situation", y = "Log(Coun
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(size = 10)) +  # Reduce the plot title size
  scale_fill_manual(values = c("VERY GOOD" = "blue", "GOOD" = "green", "PART GOOD, PART BAD" = "yellow"

# Arrange the plots in a single row with increased width
options(repr.plot.width = 16, repr.plot.height = 6)
grid.arrange(histogram_x1, log_transformed_x1, ncol = 2)
# Convert x163 to a factor with appropriate labels
raw_data$x163 <- factor(raw_data$x163, levels = c(1, 2, 3, 4, 5),
                        labels = c("LOWER CLASS", "WORKING CLASS", "MIDDLE CLASS", "UPPER MIDDLE CLASS"
raw_data$row_number <- 1:nrow(raw_data)

# Scatterplot of row number vs self-assessment of social class
scatter_plot_x163 <- ggplot(raw_data, aes(x = row_number, y = x163)) +
  geom_point(alpha = 0.009) +
  labs(title = "Scatterplot of Self-Assess of Social Class", x = "Row Number", y = "Self-Assessment of
  theme_minimal() +
  theme(plot.title = element_text(size = 10))  # Reduce the plot title size

# Print the scatterplot with increased width
print(scatter_plot_x163)
# Histogram of the variable x163 with different colors
histogram_x163 <- ggplot(raw_data, aes(x = x163, fill = x163)) +
  geom_bar() +
  labs(title = "Dist of Self-Assessed Social Class", x = "Self-Assessment of Social Class", y = "Count"
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 7),
        plot.title = element_text(size = 8),  # Reduce the plot title size
        legend.text = element_text(size = 6),  # Adjust legend text size
        legend.title = element_text(size = 8)) +  # Adjust legend title size
  scale_fill_manual(values = c("LOWER CLASS" = "blue", "WORKING CLASS" = "green", "MIDDLE CLASS" = "yel
  guides(fill = guide_legend(title = "Self-Assessment of Social Class"))

# Log-transformed histogram with different colors
# First, compute counts and then log-transform
x163_counts <- raw_data %>%
```

```
  group_by(x163) %>%
  summarise(count = n())

log_transformed_x163 <- ggplot(x163_counts, aes(x = x163, y = log(count), fill = x163)) +
  geom_bar(stat = "identity") +
  labs(title = "Log-Trans Dist of Self-Assessed of Social Class", x = "Self-Assessment of Social Class"
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 7),
        plot.title = element_text(size = 8),  # Reduce the plot title size
        legend.text = element_text(size = 6),  # Adjust legend text size
        legend.title = element_text(size = 8)) +  # Adjust legend title size
  scale_fill_manual(values = c("LOWER CLASS" = "blue", "WORKING CLASS" = "green", "MIDDLE CLASS" = "yel
  guides(fill = guide_legend(title = "Self-Assessment of Social Class"))

# Arrange the plots in a single row with increased width
options(repr.plot.width = 24, repr.plot.height = 6)
grid.arrange(histogram_x163, log_transformed_x163, ncol = 2)

raw_data <- read_csv("train.csv")
raw_data$health <- factor(raw_data$health, levels = c(1, 2, 3, 4, 5),
                          labels = c("VERY GOOD", "GOOD", "SATISFACTORY", "NOT THAT GOOD", "BAD"))
# Subset the relevant columns and convert factors to numeric
subset_data <- raw_data[, c("health", "x1", "x163")]
subset_data$health <- as.numeric(subset_data$health)
subset_data$x1 <- as.numeric(subset_data$x1)
subset_data$x163 <- as.numeric(subset_data$x163)

# Create a ggpairs plot
ggpairs(subset_data,
        title = "Matrix of Scatterplots for Health, Curnt Econ Situat, and Self-Assess Social Class",
        labeller = as_labeller(c(health = "Health", x1 = "Curnt Econ Situat", x163 = "Self-Assess Social

# Rename the variables
colnames(subset_data) <- c("Health", "Curnt Econ Situat", "Self-Assess Social Class")

# Convert health variable to numeric for correlation calculation
subset_data$Health <- as.numeric(as.factor(subset_data$Health))

# Calculate the correlation matrix
cor_matrix <- cor(subset_data, use = "complete.obs")

# Plot the heatmap of the correlation matrix
corrplot(cor_matrix, method = "color", addCoef.col = "black",
         title = "Correlation Heatmap",
         tl.cex = 0.8, number.cex = 0.8, mar = c(0,0,1,0),
         tl.col = "black", tl.srt = 45, cl.pos = "b")
title(main = "Correlation Heatmap of Health, Current Economic Situation, and Self-Assessment of Social

# Important variables and their respective titles
important_vars <- list(
  x1 = "Current Economic Situation in Germany",
  x111 = "Trust in Health Service",
  x112 = "Trust in Federal Constitutional Court",
```

```
  x118 = "Trust in Television",
  x119 = "Trust in Newspapers",
  x120 = "Trust in Universities, Higher Education",
  x121 = "Trust in Federal Government",
  x122 = "Trust in Police",
  x163 = "Self-Assessment of Social Class, Respondent",
  x190 = "Success: Depends on Own Education",
  x453 = "Body Height in Centimeters",
  x454 = "Body Weight in Kilograms",
  x760 = "Spouse's Age",
  x1180 = "Social Class of Household",
  x1035 = "Number of Biological Children"
)

# Print out the variable names and their interpretations
for (var in names(important_vars)) {
  cat(paste(var, ":", important_vars[[var]], "\n"))
}

important_vars <- c("x1", "x111", "x112", "x118", "x119", "x120", "x121", "x122", "x163", "x190", "x453

# Select the important variables from the dataset using dplyr::select
data_selected <- raw_data %>% dplyr::select(all_of(important_vars))

# Convert variables to numeric if they are not already
data_selected <- data_selected %>% mutate(across(everything(), ~ as.numeric(as.character(.))))

# Fill missing values with the mean of each column
data_selected <- data_selected %>% mutate(across(everything(), ~ ifelse(is.na(.), mean(., na.rm = TRUE)

# Log transform the data, adding a small constant to avoid log(0) issues
data_log_transformed <- data_selected %>% mutate(across(everything(), ~ log(. + 1)))

# Check the structure of the log-transformed data
str(data_log_transformed)

# Create the correlation matrix plot
ggcorr(data_log_transformed, label = TRUE, label_size = 3, label_round = 2, hjust = 1, size = 3, color =

# Simplified version of axis titles
vars <- list(
  x1 = list(title = "Curnt Econ Situat in Ger", levels = c(1, 2, 3, 4, 5), labels = c("VERY GOOD", "GOO
  x111 = list(title = "Trust in Health Service"),
  x112 = list(title = "Trust in Fed Constit Court"),
  x118 = list(title = "Trust in Television"),
  x119 = list(title = "Trust in Newspapers"),
  x120 = list(title = "Trust in high Educ"),
  x121 = list(title = "Trust in Fed Govt"),
  x122 = list(title = "Trust in Police"),
  x190 = list(title = "Success depends on Educ"),
  x453 = list(title = "Height in cm"),
  x454 = list(title = "Weight in kg"),
  x760 = list(title = "Spouse's Age"),
```

```
  x1180 = list(title = "Household Social Class", levels = c(1, 2, 3, 4, 5, 6), labels = c("LOWER CLASS"
  x1035 = list(title = "Number of Children")
)

# Function to create scatter plots for log-transformed pairs of variables
create_scatter_plot <- function(data, vars_pair, x_label, y_label, show_legend = TRUE) {
  # Select the important variables and health from the dataset
  data_selected <- data %>% dplyr::select(all_of(c(vars_pair, "health")))

  # Convert variables to numeric if they are not already, preserving factor levels
  data_selected <- data_selected %>% mutate(across(all_of(vars_pair), ~ as.numeric(factor(.))))

  # Fill missing values with the mean of each column
  data_selected <- data_selected %>% mutate(across(all_of(vars_pair), ~ ifelse(is.na(.), mean(., na.rm =

  # Log transform the data, adding a small constant to avoid log(0) issues
  data_log_transformed <- data_selected %>% mutate(across(all_of(vars_pair), ~ log(. + 1)))

  # Create scatter plot for log-transformed variables
  scatter_plot <- ggplot(data_log_transformed, aes_string(x = vars_pair[1], y = vars_pair[2], color = "
    geom_point(alpha = 0.6) +
    labs(title = paste("Scatter Plot of", x_label, "vs", y_label),
         x = paste("Log of", x_label),
         y = paste("Log of", y_label),
         color = "Health Status") +
    theme_minimal() +
    theme(plot.title = element_text(size = 7),
          axis.title.x = element_text(size = 6),
          axis.title.y = element_text(size = 6),
          axis.text = element_text(size = 4),
          legend.title = element_blank(),
          legend.text = element_text(size = 4))

  if (!show_legend) {
    scatter_plot <- scatter_plot + theme(legend.position = "none")
  }

  return(scatter_plot)
}

# Create scatter plots for the specified pairs of variables
plot1 <- create_scatter_plot(raw_data, c("x453", "x454"),
                             vars$x453$title, vars$x454$title, show_legend = FALSE)

plot2 <- create_scatter_plot(raw_data, c("x190", "x120"),
                             vars$x190$title, vars$x120$title, show_legend = TRUE)

plot3 <- create_scatter_plot(raw_data, c("x1", "x163"),
                             vars$x1$title, "Social Class Resp.", show_legend = FALSE)

plot4 <- create_scatter_plot(raw_data, c("x118", "x119"),
                             vars$x118$title, vars$x119$title, show_legend = TRUE)
```

```r
plot5 <- create_scatter_plot(raw_data, c("x1", "x190"),
                             vars$x1$title, vars$x190$title, show_legend = FALSE)

plot6 <- create_scatter_plot(raw_data, c("x112", "x121"),
                             vars$x112$title, vars$x121$title, show_legend = TRUE)

# Display the plots 2 in a row with added spacing
grid.arrange(grobs = list(plot1, plot2, plot3, plot4, plot5, plot6),
             layout_matrix = rbind(c(1, 2),
                                   c(3, 4),
                                   c(5, 6)),
             widths = unit(c(8, 8), "cm"),
             heights = unit(rep(3.8, 3.8), "cm"),
             padding = unit(1, "lines"))

raw_data <- read_csv("train.csv")
raw_data$health <- factor(raw_data$health, levels = c(1, 2, 3, 4, 5),
                          labels = c("VERY GOOD", "GOOD", "SATISFACTORY", "NOT THAT GOOD", "BAD"))
# Function to create histogram and boxplot for a variable
create_plots <- function(data, var, var_title, levels=NULL, labels=NULL) {
  if (!is.null(levels) && !is.null(labels)) {
    data[[var]] <- factor(data[[var]], levels = levels, labels = labels)
  }

  # Histogram
  p_hist <- ggplot(data, aes_string(x = var, fill = "health")) +
    geom_bar(position = "dodge", color = "black") +
    labs(title = paste("Histogram of", var_title), x = var_title, y = "Count") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 6),
          axis.text.y = element_text(size = 6),
          axis.title = element_text(size = 8),
          plot.title = element_text(size = 10),
          legend.position = "none") +
    scale_fill_manual(values = c("VERY GOOD" = "blue", "GOOD" = "green", "SATISFACTORY" = "yellow", "NO
    guides(fill = guide_legend(title = NULL))

  # Boxplot
  p_box <- ggplot(data, aes_string(x = "health", y = as.numeric(data[[var]]), fill = "health")) +
    geom_boxplot() +
    labs(title = paste("Boxplot of", var_title, "by Health Status"), x = "Health Status", y = var_title
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 3),
          axis.text.y = element_text(size = 3),
          axis.title = element_text(size = 7),
          plot.title = element_text(size = 8),
          legend.position = "bottom",
          legend.text = element_text(size = 4.5),
          legend.title = element_blank()) +
    scale_fill_manual(values = c("VERY GOOD" = "blue", "GOOD" = "green", "SATISFACTORY" = "yellow", "NO
    guides(fill = guide_legend(title = NULL))

  return(list(p_hist, p_box))
```

```
}

# Function to create subgroup histograms and boxplots for specific variables
create_subgroup_plots <- function(data, var, var_title) {
  # Histogram
  p_hist <- ggplot(data, aes_string(x = var, fill = "health")) +
    geom_histogram(position = "identity", alpha = 0.5, color = "black", bins = 30) +
    facet_wrap(~health, ncol = 1) +
    labs(title = paste("Histogram of", var_title, "by Health Status"), x = var_title, y = "Count") +
    theme_minimal() +
    theme(axis.text.x = element_text(size = 3),
          axis.text.y = element_text(size = 3),
          axis.title = element_text(size = 7),
          plot.title = element_text(size = 8),
          strip.text = element_text(size = 4),  # Reduce the strip text size
          legend.position = "none")

  # Boxplot
  p_box <- ggplot(data, aes_string(x = "health", y = var, fill = "health")) +
    geom_boxplot() +
    labs(title = paste("Boxplot of", var_title, "by Health Status"), x = "Health Status", y = var_title
    theme_minimal() +
    theme(axis.text.x = element_text(size = 4),
          axis.text.y = element_text(size = 4),
          axis.title = element_text(size = 7),
          plot.title = element_text(size = 8),
          strip.text = element_text(size = 8),  # Reduce the strip text size
          legend.position = "bottom",
          legend.text = element_text(size = 4.5),
          legend.title = element_blank()) +
    scale_fill_manual(values = c("VERY GOOD" = "blue", "GOOD" = "green", "SATISFACTORY" = "yellow", "NO
    guides(fill = guide_legend(title = NULL))

  return(list(p_hist, p_box))
}

# Create plots for each variable
plot_list <- list()

# Variables with their respective titles
vars <- list(
  x1 = list(title = "Current Econ Situat", levels = c(1, 2, 3, 4, 5), labels = c("VERY GOOD", "GOOD", "
  x111 = list(title = "Trust in Health Service"),
  x112 = list(title = "Trust in Fed Constit Court"),
  x118 = list(title = "Trust in TV"),
  x119 = list(title = "Trust in Newspapers"),
  x120 = list(title = "Trust in high Educ"),
  x121 = list(title = "Trust in Fed Govt"),
  x122 = list(title = "Trust in Police"),
  x190 = list(title = "Success depends on Educ"),
  x453 = list(title = "Height in cm"),
  x454 = list(title = "Weight in kg"),
  x760 = list(title = "Spouse's Age"),
```

```
  x1180 = list(title = "Household Social Class", levels = c(1, 2, 3, 4, 5, 6), labels = c("LOWER CLASS"
  x1035 = list(title = "Number of Children")
)

for (var in names(vars)) {
  var_title <- vars[[var]]$title
  levels <- vars[[var]]$levels
  labels <- vars[[var]]$labels
  if (var %in% c("x453", "x454", "x760")) {
    plots <- create_subgroup_plots(raw_data, var, var_title)
    plot_list <- c(plot_list, plots)
  } else {
    plots <- create_plots(raw_data, var, var_title, levels, labels)
    plot_list <- c(plot_list, plots)
  }
}

grid.arrange(
  grobs = list(
    ggplotGrob(plot_list[[1]]), ggplotGrob(plot_list[[2]]),
    ggplotGrob(plot_list[[3]]), ggplotGrob(plot_list[[4]])
  ),
  ncol = 2,
  nrow = 2,
  widths = c(0.6, 0.6),
  heights = c(1, 1),
  padding = unit(1, "lines")
)

grid.arrange(
  grobs = list(
    ggplotGrob(plot_list[[5]]), ggplotGrob(plot_list[[6]]),
    ggplotGrob(plot_list[[7]]), ggplotGrob(plot_list[[8]])
  ),
  ncol = 2,
  nrow = 2,
  widths = c(0.6, 0.6),
  heights = c(1, 1),
  padding = unit(1, "lines")
)

grid.arrange(
  grobs = list(
    ggplotGrob(plot_list[[9]]), ggplotGrob(plot_list[[10]]),
    ggplotGrob(plot_list[[11]]), ggplotGrob(plot_list[[12]])
  ),
  ncol = 2,
  nrow = 2,
  widths = c(0.6, 0.6),
  heights = c(1, 1),
  padding = unit(1, "lines")
)
```

```
grid.arrange(
  grobs = list(
    ggplotGrob(plot_list[[13]]), ggplotGrob(plot_list[[14]]),
    ggplotGrob(plot_list[[15]]), ggplotGrob(plot_list[[16]]),
    ggplotGrob(plot_list[[17]]), ggplotGrob(plot_list[[18]])
  ),
  ncol = 3,
  nrow = 2,
  widths = c(0.6, 0.6, 0.6),
  heights = c(1, 1),
  padding = unit(1, "lines")
)
# Display plots for indices (19, 20)
grid.arrange(grobs = list(ggplotGrob(plot_list[[19]]), ggplotGrob(plot_list[[20]])), ncol = 2, widths =
# Display plots for indices (21, 22)
grid.arrange(grobs = list(ggplotGrob(plot_list[[21]]), ggplotGrob(plot_list[[22]])), ncol = 2, widths =
# Display plots for indices (23, 24)
grid.arrange(grobs = list(ggplotGrob(plot_list[[23]]), ggplotGrob(plot_list[[24]])), ncol = 2, widths =
grid.arrange(
  grobs = list(
    ggplotGrob(plot_list[[25]]), ggplotGrob(plot_list[[26]]),
    ggplotGrob(plot_list[[27]]), ggplotGrob(plot_list[[28]])
  ),
  ncol = 2,
  nrow = 2,
  widths = c(1, 1),
  heights = c(2, 2),  # Adjusted heights to give equal space to both rows
  padding = unit(1, "lines")
)

library(gamlr)
library(tree)
library(randomForest)
library(rpart)
library(rpart.plot)
library(caret)
library(ggplot2)
library(gridExtra)

library(gbm)
df = read.csv("train.csv")
# Calculate the percentage of missing values for each column
missing_percentages <- sapply(df,function(x) mean(is.na(x)))

# Define the thresholds
thresholds <- c(0.1,0.2, 0.3,0.4, 0.5,0.6, 0.7,0.8)

# Calculate the number of columns exceeding each threshold
counts_above_thresholds <- sapply(thresholds, function(t) sum(missing_percentages > t))

# Create a plot of thresholds vs. counts
plot(thresholds, counts_above_thresholds, type = "o", col = "blue", pch = 16,
     xlab = "Threshold of Missing Values", ylab = "Count of Columns Exceeding Threshold",
```

```
     main = "Count of Columns by Missing Value Threshold")

# Add labels to each point for clarity
text(thresholds, counts_above_thresholds, labels = counts_above_thresholds, pos = 2)
df = df[,missing_percentages < 0.3]
# Calculate the number of unique values for each column
num_unique_values <- sapply(df, function(x) length(unique(x)))


# Function to calculate the mode
mode_function <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

# Function to perform imputation based on the number of unique values
impute_column <- function(x) {
  num_unique <- length(unique(na.omit(x)))
  if (num_unique < 20) {
    # Impute with mode if fewer than 20 unique non-NA values
    fill_value <- mode_function(na.omit(x))
  } else {
    # Impute with mean otherwise
    fill_value <- mean(x, na.rm = TRUE)
  }
  # Replace NA values with the determined fill value
  x[is.na(x)] <- fill_value
  return(x)
}

# Apply the imputation function to each column in the DataFrame
df_imputed <- data.frame(sapply(df, impute_column))

# Set seed for reproducibility
set.seed(0)

# Determine the size of the dataset
n <- nrow(df_imputed)

# Specify the proportion for the training set (e.g., 70% training, 30% test)
train_size <- floor(0.7 * n)

# Randomly sample indices for the training data
train_indices <- sample(seq_len(n), size = train_size)

# Create the training and testing sets
train_set <- df_imputed[train_indices, ]
test_set <- df_imputed[-train_indices, ]



# Save the training set to a CSV file
write.csv(train_set, "training_set.csv", row.names = FALSE)
```

```r
# Save the testing set to a CSV file
write.csv(test_set, "testing_set.csv", row.names = FALSE)
library(dplyr)
library(caret)
library(pROC)
library(knitr)

training_set <- read.csv("training_set.csv")
training_set$health <- as.numeric(as.character(training_set$health))

health <- glm(log(health) ~ . -uniqueid -year -personid, data=training_set)
model_summary1 <- summary(health)
summary_lines <- capture.output(model_summary1)
first_ten_lines <- summary_lines[1:20]
cat(first_ten_lines, sep = "\n")
source('deviance.R')
# compute R^2 for the full model
R2_all <- R2(y=log(training_set$health), pred=predict(health), family="gaussian")
R2_all
coefficients <- coef(health)
length(coefficients)
var_descriptions <- c(
  x1 = "CURRENT ECONOMIC SITUATION IN GERMANY",
  x2 = "OWN CURRENT FINANCIAL SITUATION",
  x162 = "FAIR SHARE IN STANDARD OF LIVING?",
  x163 = "SELF-ASSESSMENT OF SOCIAL CLASS",
  x164 = "TOP-BOTTOM-SCALE: SELF-CLASSIFIC.",
  x595 = "MEMBER OF A TRADE UNION",
  x613 = "GENERAL TRUST IN FELLOW MEN",
  x615 = "WITH SUCH A FUTURE, NO MORE CHILDREN",
  x617 = "MOST PEOPLE DON'T CARE ABOUT OTHERS",
  x657 = "CURRENT EMPLOYMENT STATUS",
  x965 = "DEGREE OF RELATIONSHIP",
  x1179 = "START OF INTERVIEW"
)

# extract pvalues
pvals <- summary(health)$coef[-1,4]

# sort pvalues ascending
pvals_ordered<-pvals[order(pvals,decreasing=F)]

# FDR cut at 10%
q = 0.1
source("fdr.R")

cutoff <- fdr_cut(pvals_ordered, q, plotit=TRUE)
cat("FDR cutoff p-value = ", cutoff, "\n")

significant_vars = names(pvals)[pvals <= cutoff]

# Use var_descriptions to get the full names of significant variables
significant_descriptions <- var_descriptions[significant_vars]
```

```r
# Display significant variables with descriptions using kable
kable(as.data.frame(significant_descriptions), col.names = c("Variable Description"))

# exclude not-significant covariates as well as mortgage
pricey_sig <- glm(log(health) ~ x1 + x2 + x162 + x163 + x164 + x595 + x613 + x615 + x617 + x657 + x965 +

# find the new R^2
R2_new = R2(y=log(training_set$health),pred=predict(pricey_sig),family="gaussian")
R2_new

# R2 of only the significant covariate
cat("R2 of regression only the significant covariates = ", R2_new, "\n")
# full model R2
cat("full model R2 = ", R2_all, "\n")

 # Create a binary response variable
  training_set <- training_set %>%
    mutate(eco_situation_better = ifelse(x1 > 3, 1, 0))

eco_situationy <- glm(eco_situation_better ~ . -uniqueid -year -personid - x1 - health, data=training_s
model_summary2 <- summary(eco_situationy)
summary_output <- capture.output(summary(eco_situationy))
print(summary_output[1:20])
# add interaction between respondent's own financial situation(x2) and satisfaction with democracy(x6).
eco_situation_better_int <- glm(eco_situation_better ~ .-uniqueid -year -personid - x1 - health + x2*x6
model_summary3 <- summary(eco_situation_better_int)
summary_output3 <- capture.output(summary(eco_situation_better_int))
print(summary_output3[1:20])

# Load the dplyr package for data manipulation
if (!require("dplyr")) install.packages("dplyr")
library(dplyr)

# Assuming 'training_set' is your main dataset and it's already loaded
# Filter the dataset to include only respondents older than 50 years based on x633
subset <- which(training_set$x633>50)

# train the full model on this subset
health_sub <- glm(log(health) ~ .-uniqueid -year -personid, data=training_set[subset,])
model_summary4 <- summary(health_sub)
summary_output4 <- capture.output(summary(health_sub))
print(summary_output4[1:20])


# predict the left-out homes using this model
pred_left = predict(health_sub, newdata=training_set[-subset,])

# compute OOS deviance
source("deviance.R")

# null deviance for OOS
ybar <- mean(log(training_set$health[-subset]))
D0 <- deviance(y=log(training_set$health[-subset]), pred=ybar,family="gaussian")
```

```r
cat("null deviance = ", D0, "\n")

# residual deviance for OOS
D <- deviance(y=log(training_set$health[-subset]), pred=pred_left,family="gaussian")
cat("Residual deviance = ", D , "\n")

# R2 = 1 - Residual Deviance / Null Deviance
R2=1-D/D0
cat("OOS R2 = ", R2 , "\n")
training_set <- training_set %>%
mutate(health_binary = ifelse(health %in% c(1, 2, 3), 1, 0))
health_model <- glm(health_binary ~ . - uniqueid - year - personid - health, data=training_set, family=
model_summary5<- summary(health_model)
summary_output5 <- capture.output(summary(health_model))
print(summary_output5[1:20])

training_set <- read.csv("training_set.csv")
training_set <- training_set %>%
  mutate(health_binary = ifelse(health %in% c(1, 2, 3), 1, 0))

model <- glm(health_binary ~ . - uniqueid - year - personid - health, data=training_set, family="binomia

testing_set <- read.csv("testing_set.csv")


testing_set <- testing_set %>%
  mutate(health_binary = ifelse(health %in% c(1, 2, 3), 1, 0))

# Similarly, transform 'health' into a binary variable in the testing dataset
testing_set <- testing_set %>%
  mutate(health_binary = ifelse(health %in% c(1, 2, 3), 1, 0))


# Use the trained model to predict the health status on the testing dataset
# Predict probabilities of the outcomes based on the logistic model
predicted_probs <- predict(model, newdata = testing_set, type = "response")

# Convert probabilities to binary outcomes using a threshold of 0.5
predicted_classes <- ifelse(predicted_probs > 0.5, 1, 0)

# You can also create a confusion matrix to evaluate predictions
confusion_matrix <- table(Predicted = predicted_classes, Actual = testing_set$health_binary)

# Print the confusion matrix
print(confusion_matrix)

# Optionally, calculate accuracy or other performance metrics
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
print(paste("Accuracy:", accuracy))


library(readr)
library(readtext)
```

```r
library(SnowballC)
library(tidytext)
library(gamlr)
library(nnet)
library(dplyr)
library(glmnet)
library(ggplot2)
library(parallel)
library(doParallel)
data <- read_csv("training_set.csv")

# Identify and convert categorical variables (unique values < 20) to factors
for (var in names(data)) {
  if (length(unique(data[[var]])) < 20) {
    data[[var]] <- as.factor(data[[var]])
  }
}

# Ensure the health column is a factor
data$health <- as.factor(data$health)

# Convert the age column to numeric
data$x633 <- as.numeric(data$x633)

# Define the response variable and the treatment
health <- data$health
age <- data$x633


# Plot the distribution of Age using a histogram
ggplot(data, aes(x = x633)) +
  geom_histogram(binwidth = 5, fill = "blue", color = "black") +
  xlab("Age") +
  ylab("Count") +
  ggtitle("Distribution of Respondents' Age") +
  theme_minimal() +  # Apply a minimal theme for a cleaner look
  theme(
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold")
  )
# Function to perform multinomial logistic regression and return p-values
margreg <- function(i) {
  model <- multinom(health ~ age, data = data)
  summary_model <- summary(model)
  coefficients <- summary_model$coefficients
  std_errors <- summary_model$standard.errors
  z_values <- coefficients / std_errors
  p_values <- 2 * (1 - pnorm(abs(z_values)))
  return(list(p_values = as.vector(p_values), summary_model = summary_model))
}

# Setup parallel processing
```

```r
cl <- makeCluster(detectCores() - 1)  # Use one less than the number of available cores
clusterExport(cl, c("data", "health", "age", "margreg"))  # Export variables and functions to the clust
clusterEvalQ(cl, {
  library(nnet)
  library(dplyr)
  library(ggplot2)
})  # Ensure necessary libraries are loaded on each cluster

# Run the regression in parallel
P <- 1:10  # Dummy list to run the function multiple times
results <- parLapply(cl, P, function(x) margreg(x))

# Extract p-values and the summary model from one of the runs
mrgpvals_age <- unlist(lapply(results, function(res) res$p_values))
summary_model <- results[[1]]$summary_model

# Stop the cluster
stopCluster(cl)

# Plot the distribution of p-values
hist(mrgpvals_age, main="Distribution of P-values for age on Health", xlab="P-Value for age", ylab="Fre

# Print the summary model from one of the runs
print(summary_model)
# Select only the significant variables
significant_vars <- data[,c("x1", "x2", "x162", "x163", "x164", "x595", "x613", "x615", "x617", "x657",


# Prepare the data for LASSO regression using model.matrix to create dummy variables
predictors <- model.matrix(~ . - 1, data = significant_vars)

# Stage 1 LASSO: fit a model for age on other predictors using Gaussian family
model_age <- cv.glmnet(predictors, data$x633, alpha = 1, family = "gaussian")

# Predict age using the fitted LASSO model
dhat <- predict(model_age, s = "lambda.min", newx = predictors)

# Calculate the R-squared value
R2 <- cor(drop(dhat), data$x633)^2
cat("The In-Sample R-squared value is", R2, ".\n")

dhat <- as.numeric(dhat)

# Combine predicted age (dhat) with other predictors
predictors_combined <- cbind(dhat, significant_vars, age)

# Detect number of cores and register parallel backend
num_cores <- detectCores()
cl <- makeCluster(num_cores)
registerDoParallel(cl)

# Stage 2: Fit a multinomial logistic regression model using glmnet with parallel processing
model_health <- cv.glmnet(as.matrix(predictors_combined), as.matrix(data$health), family = "multinomial
```

```r
# Stop the cluster after model fitting
stopCluster(cl)

# Extract coefficients for the predicted age (dhat) for each class
coefficients_list <- coef(model_health, s = "lambda.min")
coef_value <- numeric(length(coefficients_list))

# Assuming we find the correct row name for dhat, extract and print the coefficient
for (i in 1:length(coefficients_list)) {
  coef_matrix <- as.matrix(coefficients_list[[i]])
  coef_value[i] <- coef_matrix["dhat", ]
  cat("The effect of predicted age (dhat) on health status for class", i, "is", coef_value[i], ".\n")
}

# Combine predicted age with other predictors
predictors_combined2 <- cbind(significant_vars, age)

# Detect number of cores and register parallel backend
num_cores <- detectCores()
cl <- makeCluster(num_cores)
registerDoParallel(cl)

# Stage 2: Fit a multinomial logistic regression model using glmnet with parallel processing
model_health2 <- cv.glmnet(as.matrix(predictors_combined2), as.matrix(data$health), family = "multinomia

# Stop the cluster after model fitting
stopCluster(cl)

# Extract coefficients for the predicted age for each class
coefficients_list2 <- coef(model_health2, s = "lambda.min")
coef_value2 <- numeric(length(coefficients_list))

# Assuming we find the correct row name for dhat, extract and print the coefficient
for (i in 1:length(coefficients_list2)) {
  coef_matrix2 <- as.matrix(coefficients_list2[[i]])
  coef_value2[i] <- coef_matrix2["age", ]
  cat("The effect of d on health status from a naive lasso for class", i, "is", coef_value2[i], ".\n")
}

# transfer my response
train_set$x1 <- factor(train_set$x1, levels = c(1, 2, 3, 4, 5),
                       labels = c(" VERY GOOD", "GOOD", "PART GOOD, PART BAD", "BAD", "VERY BAD"))
test_set$x1 <- factor(test_set$x1, levels = c(1, 2, 3, 4, 5),
                       labels =  c(" VERY GOOD", "GOOD", "PART GOOD, PART BAD", "BAD", "VERY BAD"))

# Histogram of the variable x1 with different colors
histogram_x1 <- ggplot(train_set, aes(x = x1, fill = x1)) +
                geom_bar() +
                labs(title = "Distribution of Current Economic Situation", x = "Current Economic Situati
                theme_minimal() +
                theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
                scale_fill_manual(values = c("VERY GOOD" = "blue", "GOOD" = "green", "PART GOOD, PART BA
```

```r
# Display the first histogram
print(histogram_x1)

# transfer my response
train_set$x1 <- factor(train_set$x1, levels = c(" VERY GOOD", "GOOD", "PART GOOD, PART BAD", "BAD", "VE
                       labels = c("GOOD", "GOOD", "NEUTRAL", "BAD", "BAD"))
test_set$x1 <- factor(test_set$x1, levels = c(" VERY GOOD", "GOOD", "PART GOOD, PART BAD", "BAD", "VERY
                       labels = c("GOOD", "GOOD", "NEUTRAL", "BAD", "BAD"))

# second histogram with 3 level
histogram_x1_2 <- ggplot(train_set, aes(x = x1, fill = x1)) +
  geom_bar() +
  labs(title = "Distribution of Current Economic Situation", x = "Current Economic Situation", y = "Cou
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = c("GOOD" = "blue", "NEUTRAL" = "yellow", "BAD" = "red"))

print(histogram_x1_2)

train_set = read.csv("training_set.csv")
test_set = read.csv("testing_set.csv")

# Loop through each column in the dataframe
for (col_name in names(train_set)) {
  # Check if the number of unique non-NA values is less than 20
  if (length(unique(na.omit(train_set[[col_name]]))) < 20) {

    # Convert the column to a factor (categorical variable)
    train_set[[col_name]] <- as.factor(train_set[[col_name]])
    test_set[[col_name]] <- as.factor(test_set[[col_name]])
  }
}




# keep the following variables
var_lists = c("x1", "x14", "x17", "x597", "x544", "x545","x546","x547","x548", "x162", "x163", "x595",

train_set = train_set[,var_lists]
test_set = test_set[,var_lists]
# transfer my response
train_set$x1 <- factor(train_set$x1, levels = c(1, 2, 3, 4, 5),
                       labels = c("GOOD", "GOOD", "NEUTRAL", "BAD", "BAD"))
test_set$x1 <- factor(test_set$x1, levels = c(1, 2, 3, 4, 5),
                       labels = c("GOOD", "GOOD", "NEUTRAL", "BAD", "BAD"))
tree <- rpart(x1 ~., data = train_set, method = "class",cp=0.005)
rpart.plot(tree)
# Fit the initial tree with xval for cross-validation (10-fold by default)
tree_model <- rpart(x1 ~ ., data = train_set, method = "class",
                    control = rpart.control(xval = 10, cp=0.0005))

# Find the best cp value (minimal xerror)
```

```r
best_cp <- tree_model$cptable[which.min(tree_model$cptable[,"xerror"]), "CP"]

plotcp(tree_model)
# Prune the tree with the best cp
pruned_tree <- prune(tree_model, cp = best_cp)

# Plot the pruned tree
rpart.plot(pruned_tree, extra = 104)  # shows splits with detailed labels

# Predict and evaluate on a test set or via additional cross-validation
predicted_labels <- predict(pruned_tree, newdata = train_set, type = "class")
actual_labels = train_set$x1
#confusionMatrix(data = predicted_labels, reference = actual_labels)

conf_matrix <- table(Predicted = predicted_labels, Actual = actual_labels)
print(conf_matrix)
# Get the variable importance
feature_importance <- pruned_tree$variable.importance


# Plotting feature importance
barplot(feature_importance, main = "Feature Importance in Decision Tree",
        xlab = "Features", ylab = "Importance", las = 2, col = "blue")

# Predict and evaluate on a test set or via additional cross-validation
predicted_labels <- predict(pruned_tree, newdata = test_set, type = "class")
actual_labels = test_set$x1
confusionMatrix(data = predicted_labels, reference = actual_labels)

rf_model <- randomForest(x1 ~ ., data = train_set, ntree = 500, importance = TRUE)

predictions <- predict(rf_model, test_set)
confusion_matrix <- table(Predicted = predictions, Actual = test_set$x1)
print(confusion_matrix)

accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
print(paste("Accuracy:", accuracy))
varImpPlot(rf_model, type = 1)
set.seed(0)
gbm_model <- gbm(x1 ~ .,
                 data = train_set,
                 distribution = "multinomial",
                 n.trees = 100,
                 interaction.depth = 3,
                 shrinkage = 0.1,
                 cv.folds = 5,
                 n.minobsinnode = 10,
                 verbose = TRUE)


# Predict on full data
preds <- predict(gbm_model, test_set, n.trees = 100, type = "response")
predicted_classes <- apply(preds, 1, which.max) - 1
```

```r
# Confusion Matrix
confusion_matrix=table(Predicted = predicted_classes, Actual = test_set$x1)

accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)

print(confusion_matrix)
print(paste("Accuracy:", accuracy))
# Get relative influence
relative_influence <- summary(gbm_model, plotit = FALSE)

# Sort the data by relative influence
sorted_influence <- relative_influence[order(-relative_influence$rel.inf),]

# Create a bar plot
barplot(sorted_influence$rel.inf, names.arg = sorted_influence$var,
        main = "Feature Importance",
        xlab = "Variables",
        ylab = "Relative Influence",
        las = 2,  # makes the variable names perpendicular to the axis
        col = "blue")
training_set <- read.csv("training_set.csv")
# Randomly select 20 columns from the dataset
set.seed(41201)  # Set seed for reproducibility
selected_columns <- sample(names(training_set), 20)

# Subset the dataset to include only the selected columns
subset_training_set <- training_set[, selected_columns]

# Compute the correlation matrix
correlation_matrix <- cor(subset_training_set, use = "complete.obs")

# Print the correlation matrix
print(correlation_matrix)

# Optional: Visualize the correlation matrix using corrplot
corrplot(correlation_matrix, method = "circle")

# Remove the column x163 from the dataset
training_set <- training_set[, !names(training_set) %in% "x163"]

# Perform PCA on the full dataset, scaling the data
mypca <- prcomp(training_set, scale = TRUE)

# Plot the scree plot
plot(mypca, main = "Scree Plot: Variance contributed by each component")
title(xlab = "Principal Components")
# Read the CSV file into a data frame
training_set <- read.csv("training_set.csv")

# Extract the x163 column for coloring
x163 <- training_set$x163

# Ensure the uniqueid column is included in the dataset and extracted separately
```

```r
uniqueid <- training_set$uniqueid

# Remove the x163 and uniqueid columns from the dataset used for PCA
training_set <- training_set[, !names(training_set) %in% c("x163", "uniqueid")]

# Perform PCA on the full dataset, scaling the data
mypca <- prcomp(training_set, scale = TRUE)

# Predict the principal components
pcs <- predict(mypca)

# Define colors based on x163 values
colors <- as.factor(x163)
palette <- rainbow(length(levels(colors)))

# Define labels for x163
x163_labels <- c("LOWER CLASS", "WORKING CLASS", "MIDDLE CLASS", "UPPER MIDDLE CLASS", "UPPER CLASS")

# Plot the scatter plot of the first two principal components
plot(pcs[,1], pcs[,2], xlab = "First Principal Component",
     ylab = "Second Principal Component",
     main = "Scatter Plot of the First Two Principal Components", pch = 20,
     col = palette[colors])

# Add text labels to the points using the uniqueid column
text(pcs[,1], pcs[,2], labels = uniqueid, pos = 4, cex = 0.6, col = palette[colors])

# Add smaller legend to the plot
legend("topright", legend = x163_labels, col = palette, pch = 20, title = "Social Class", cex = 0.7)

# Define colors based on x163 values
colors <- as.factor(x163)
palette <- rainbow(length(levels(colors)))

# Define labels for x163
x163_labels <- c("LOWER CLASS", "WORKING CLASS", "MIDDLE CLASS", "UPPER MIDDLE CLASS", "UPPER CLASS")

# Plot the scatter plot of the second and third principal components
plot(pcs[,2], pcs[,3], xlab = "Second Principal Component",
     ylab = "Third Principal Component",
     main = "Scatter Plot of the Second and Third Principal Components", pch = 20,
     col = palette[colors])

# Add text labels to the points using the uniqueid column
text(pcs[,2], pcs[,3], labels = uniqueid, pos = 4, cex = 0.6, col = palette[colors])

# Add smaller legend to the plot
legend("topright", legend = x163_labels, col = palette, pch = 20, title = "Social Class", cex = 0.7)

# Plot the scatter plot of the first three principal components
s3d <- scatterplot3d(pcs[,1], pcs[,2], pcs[,3], xlab = "First Principal Component",
                     ylab = "Second Principal Component", zlab = "Third Principal Component",
                     main = "Scatter Plot of the First Three Principal Components", pch = 20, color = '
```

```r
# Add text labels to the points using the uniqueid column
s3d_coords <- s3d$xyz.convert(pcs[,1], pcs[,2], pcs[,3])
text(s3d_coords$x, s3d_coords$y, labels = uniqueid, pos = 4, cex = 0.6)

loadings <- mypca$rotation[,1:2]
abs_loadings <- abs(loadings)

# Create a data frame to store the sorted loadings with variable names
sorted_loadings <- data.frame()

for (i in 1:ncol(abs_loadings)) {
  sorted_loadings <- rbind(sorted_loadings,
                           data.frame(Variable = rownames(abs_loadings),
                                      Loading = loadings[,i],
                                      Abs_Loading = abs_loadings[,i],
                                      PC = colnames(abs_loadings)[i]))
}

# Sort the data frame by absolute loading values in descending order
sorted_loadings <- sorted_loadings[order(-sorted_loadings$Abs_Loading), ]
print(head(sorted_loadings, 10))
training_set <- read.csv("training_set.csv")
response_variable <- training_set$x163
training_set <- training_set[, !names(training_set) %in% "x163"]

# Perform PCA on the full dataset, scaling the data
mypca <- prcomp(training_set, scale = TRUE)

# Get the principal component scores
pcs <- predict(mypca)
zdf <- as.data.frame(pcs)

# Function to calculate AICc for a multinomial model
calculateAICc <- function(model) {
  AICc(model)
}

# Perform multinomial logistic regression fits on 1:20 factors and calculate AICc for each
kfits <- lapply(1:50, function(K) {
  multinom(response_variable ~ ., data = zdf[, 1:K, drop = FALSE])
})
# Calculate AICc for each model
aicc <- sapply(kfits, calculateAICc)
best_K_AICc <- which.min(aicc)
cat("Best number of PCs (based on AICc):", best_K_AICc, "\n")

# Calculate BIC for each model
bic <- sapply(kfits, BIC)
best_K_BIC <- which.min(bic)
cat("Best number of PCs (based on BIC):", best_K_BIC, "\n")

# Plot AICc by number of factors
plot(aicc, type = 'b', xlab = "Number of Factors",
```

```
     ylab = "AICc",
     main = "AICc by Number of Factors")
# Select the best model based on AICc
best_model <- kfits[[best_K_AICc]]
summary(best_model)
# Load the testing dataset
testing_set <- read.csv("testing_set.csv")

# Store the true labels for evaluation
true_labels <- testing_set$x163

# Remove the response variable column (x163) from the testing set
testing_set <- testing_set[, !names(testing_set) %in% "x163"]

# Perform PCA on the testing dataset, scaling the data
testing_pca <- predict(mypca, newdata = testing_set)
testing_pcs <- as.data.frame(testing_pca)

# Predict using the best model
predictions <- predict(best_model, newdata = testing_pcs[, 1:best_K_AICc, drop = FALSE], type = "class")

# Load necessary library for evaluation
library(caret)

# Create a confusion matrix
confusion_matrix <- confusionMatrix(predictions, as.factor(true_labels))

# Print the confusion matrix and other evaluation metrics
print(confusion_matrix)


model_summary1

model_summary2

model_summary3

model_summary4

model_summary5
```

```
##
## Call:
## glm(formula = log(health) ~ . - uniqueid - year - personid, data = training_set)
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.236e+01  3.213e+00  -3.845 0.000121 ***
## x1           2.197e-02  5.138e-03   4.277 1.91e-05 ***
## x2           6.596e-02  5.386e-03  12.246  < 2e-16 ***
## x3           1.233e-02  5.184e-03   2.379 0.017369 *
## x4           9.957e-03  5.872e-03   1.696 0.089947 .
## x7          -5.517e-03  2.238e-03  -2.465 0.013720 *
## x14          5.807e-03  3.966e-03   1.464 0.143179
```

```
## x17          7.531e-05  1.401e-04   0.537 0.590978
## x19          2.969e-02  1.167e-02   2.545 0.010938 *
## x162        -2.568e-02  5.841e-03  -4.397 1.10e-05 ***
## x163        -2.239e-02  6.881e-03  -3.255 0.001139 **
## x164        -1.386e-02  2.877e-03  -4.818 1.46e-06 ***
## x472        -3.050e-03  2.028e-03  -1.504 0.132644
## x477        -2.301e-03  3.272e-03  -0.703 0.481888
## x544        -1.074e-02  9.919e-03  -1.082 0.279089
## x545         3.851e-03  9.313e-03   0.414 0.679195
## x546        -5.655e-03  9.990e-03  -0.566 0.571318
## x547         6.373e-03  9.321e-03   0.684 0.494162
## x548        -1.891e-02  1.160e-02  -1.630 0.103116
## x595        -4.168e-02  1.152e-02  -3.616 0.000300 ***
## x596        -1.100e-02  9.581e-03  -1.149 0.250784
## x597         3.998e-02  1.997e-02   2.002 0.045280 *
## x613         1.917e-02  5.603e-03   3.421 0.000625 ***
## x614        -1.923e-02  1.076e-02  -1.788 0.073875 .
## x615        -2.700e-02  8.305e-03  -3.252 0.001151 **
## x616         3.503e-03  1.041e-02   0.336 0.736523
## x617        -3.100e-02  8.730e-03  -3.551 0.000385 ***
## x630         6.736e-04  1.121e-02   0.060 0.952087
## x631        -9.938e-03  1.161e-02  -0.856 0.391937
## x632        -1.539e-03  1.660e-03  -0.928 0.353614
## x633        -1.413e-03  1.160e-02  -0.122 0.903058
## x634        -4.287e-03  1.173e-02  -0.365 0.714837
## x635        -5.586e-05  5.182e-05  -1.078 0.281134
## x638         9.629e-03  3.545e-02   0.272 0.785938
## x639        -9.016e-04  5.114e-03  -0.176 0.860066
## x640         8.741e-04  1.035e-03   0.845 0.398398
## x641        -1.587e-02  1.967e-02  -0.807 0.419592
## x642         2.674e-02  2.377e-02   1.125 0.260727
## x643         3.954e-02  3.870e-02   1.022 0.306965
## x644        -7.013e-03  1.256e-02  -0.558 0.576624
## x645         2.741e-03  1.319e-02   0.208 0.835351
## x646         2.080e-02  3.840e-02   0.542 0.588095
## x647         1.260e-02  2.458e-02   0.513 0.608229
## x648         6.260e-03  1.745e-02   0.359 0.719842
## x649         1.157e-02  2.466e-02   0.469 0.638925
## x650         2.117e-02  2.283e-02   0.927 0.353836
## x651         1.661e-02  2.197e-02   0.756 0.449677
## x652        -3.668e-02  3.618e-02  -1.014 0.310717
## x655        -2.306e-02  1.171e-02  -1.968 0.049043 *
## x657         1.337e-02  3.301e-03   4.050 5.15e-05 ***
## x723        -3.144e-06  8.114e-06  -0.387 0.698424
## x725        -2.718e-06  7.997e-06  -0.340 0.733905
## x728        -3.504e-06  9.166e-06  -0.382 0.702282
## x729        -3.383e-05  2.917e-05  -1.160 0.246221
## x730         3.922e-05  3.428e-05   1.144 0.252604
## x754         4.418e-03  4.871e-03   0.907 0.364467
## x893         4.084e-03  5.546e-03   0.736 0.461548
## x896        -6.602e-03  6.119e-03  -1.079 0.280630
## x897         1.143e-03  3.047e-03   0.375 0.707667
## x898         1.570e-02  2.239e-02   0.701 0.483283
## x901        -1.464e-02  1.404e-02  -1.043 0.297066
```

```
## x902       -4.393e-02  1.832e-02  -2.398 0.016487 *
## x906       -3.403e-02  3.496e-02  -0.973 0.330374
## x907       -5.062e-02  3.693e-02  -1.371 0.170487
## x908       -6.887e-02  3.673e-02  -1.875 0.060786 .
## x909       -2.491e-02  1.335e-01  -0.187 0.851930
## x910        8.364e-03  1.373e-02   0.609 0.542362
## x911       -8.865e-03  7.018e-03  -1.263 0.206558
## x912        1.108e-03  6.768e-04   1.637 0.101645
## x920       -3.271e-06  2.950e-06  -1.109 0.267399
## x921       -8.427e-04  7.161e-04  -1.177 0.239258
## x923        1.643e-04  6.206e-04   0.265 0.791239
## x929       -8.760e-04  6.646e-03  -0.132 0.895129
## x930        5.930e-04  3.521e-03   0.168 0.866279
## x931       -1.774e-03  1.573e-02  -0.113 0.910242
## x934       -4.341e-03  1.794e-02  -0.242 0.808843
## x935       -7.319e-03  2.021e-02  -0.362 0.717284
## x939       -1.709e-02  3.876e-02  -0.441 0.659335
## x940        1.885e-02  4.374e-02   0.431 0.666478
## x941       -1.230e-04  4.189e-02  -0.003 0.997656
## x942       -4.999e-02  1.668e-01  -0.300 0.764344
## x943        7.428e-04  1.246e-02   0.060 0.952485
## x961        9.946e-04  2.143e-02   0.046 0.962989
## x963       -5.165e-03  9.668e-03  -0.534 0.593194
## x964        4.179e-03  1.164e-02   0.359 0.719657
## x965       -7.572e-03  2.724e-03  -2.780 0.005442 **
## x966        1.722e-02  1.214e-02   1.419 0.155976
## x967       -2.318e-04  2.136e-03  -0.109 0.913572
## x968        9.330e-05  1.171e-03   0.080 0.936494
## x969       -3.982e-05  2.169e-03  -0.018 0.985351
## x970       -7.024e-04  4.770e-03  -0.147 0.882933
## x1032       4.711e-04  3.587e-04   1.313 0.189155
## x1033      -2.509e-03  9.575e-03  -0.262 0.793318
## x1035       4.787e-04  3.444e-02   0.014 0.988911
## x1036      -1.006e-02  4.188e-02  -0.240 0.810165
## x1037       5.536e-04  3.438e-02   0.016 0.987152
## x1038       7.145e-03  4.200e-02   0.170 0.864940
## x1039       2.623e-03  7.282e-03   0.360 0.718732
## x1106      -2.739e-02  1.399e-02  -1.958 0.050268 .
## x1108       1.921e-04  2.173e-04   0.884 0.376789
## x1110       9.115e-03  5.205e-03   1.751 0.079956 .
## x1132      -3.383e-03  5.812e-03  -0.582 0.560523
## x1133      -4.997e-03  3.009e-03  -1.661 0.096811 .
## x1141      -2.392e-03  2.491e-03  -0.960 0.336872
## x1143      -3.888e-04  4.127e-04  -0.942 0.346185
## x1144       2.601e-04  2.392e-03   0.109 0.913389
## x1145       1.661e-06  1.182e-06   1.405 0.160063
## x1146      -4.054e-03  2.802e-02  -0.145 0.884965
## x1147       3.620e-05  3.626e-04   0.100 0.920461
## x1148      -2.444e-02  3.379e-02  -0.723 0.469493
## x1149       1.829e-02  3.146e-02   0.581 0.561093
## x1150       5.191e-04  3.932e-04   1.320 0.186784
## x1151       1.070e-02  3.684e-02   0.291 0.771423
## x1152       2.007e-04  3.878e-04   0.518 0.604798
## x1153      -9.240e-03  5.079e-03  -1.819 0.068902 .
```

```
## x1157        5.936e-03  4.789e-03   1.239 0.215189
## x1158        2.718e-02  3.479e-02   0.781 0.434611
## x1159        6.030e-02  3.366e-02   1.792 0.073220 .
## x1160        1.710e-02  3.074e-02   0.556 0.578161
## x1161        3.011e-02  3.962e-02   0.760 0.447204
## x1162        9.415e-03  5.007e-02   0.188 0.850851
## x1164        1.005e-02  8.458e-03   1.188 0.234735
## x1165       -2.941e-02  1.325e-02  -2.220 0.026415 *
## x1166       -7.468e-03  3.440e-03  -2.171 0.029957 *
## x1167        4.812e-03  9.694e-03   0.496 0.619612
## x1175       -6.957e-03  3.236e-03  -2.150 0.031595 *
## x1176        3.577e-03  7.593e-03   0.471 0.637632
## x1177       -9.527e-04  8.537e-03  -0.112 0.911140
## x1178        1.425e-02  5.625e-03   2.533 0.011331 *
## x1179       -2.055e-02  2.383e-03  -8.621  < 2e-16 ***
## x1181        7.840e-08  4.335e-08   1.809 0.070533 .
## x1182        5.915e-03  7.675e-03   0.771 0.440868
## x1183        1.151e-03  4.481e-04   2.568 0.010230 *
## x1184       -4.845e-04  2.698e-03  -0.180 0.857515
## x1185        5.873e-04  4.322e-04   1.359 0.174223
## x1201        2.641e-02  1.970e-02   1.340 0.180242
## x1202       -8.635e-05  1.711e-04  -0.505 0.613824
## x1203       -9.639e-04  3.802e-03  -0.254 0.799873
## x1204        1.011e-03  1.806e-03   0.560 0.575482
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1489544)
##
##     Null deviance: 2458.1  on 12177  degrees of freedom
## Residual deviance: 1793.3  on 12039  degrees of freedom
## AIC: 11512
##
## Number of Fisher Scoring iterations: 2

##
## Call:
## glm(formula = eco_situation_better ~ . - uniqueid - year - personid -
##     x1 - health, family = "binomial", data = training_set)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.883e+02  2.378e+01  20.538  < 2e-16 ***
## x2           8.505e-01  3.771e-02  22.555  < 2e-16 ***
## x3           5.171e-01  3.743e-02  13.815  < 2e-16 ***
## x4          -1.540e-02  4.164e-02  -0.370 0.711491
## x7          -7.376e-04  1.640e-02  -0.045 0.964123
## x14          8.201e-02  2.935e-02   2.794 0.005201 **
## x17          4.285e-03  9.468e-04   4.526 6.02e-06 ***
## x19          9.840e-02  8.097e-02   1.215 0.224259
## x162        -9.504e-02  4.056e-02  -2.343 0.019104 *
## x163         1.123e-01  5.015e-02   2.240 0.025091 *
## x164         3.386e-02  2.073e-02   1.634 0.102281
## x472        -1.038e-02  1.508e-02  -0.688 0.491190
```

```
## x477         6.992e-02  2.452e-02   2.852 0.004351 **
## x544         5.126e-02  6.980e-02   0.734 0.462705
## x545        -9.730e-02  6.633e-02  -1.467 0.142390
## x546        -4.158e-02  7.150e-02  -0.581 0.560912
## x547         2.551e-02  6.502e-02   0.392 0.694785
## x548         6.617e-02  8.489e-02   0.780 0.435676
## x595        -4.706e-03  8.685e-02  -0.054 0.956788
## x596         1.821e-02  7.122e-02   0.256 0.798130
## x597         1.711e-01  1.579e-01   1.084 0.278488
## x613        -7.127e-02  5.082e-02  -1.403 0.160765
## x614        -2.662e-01  9.589e-02  -2.776 0.005505 **
## x615        -4.852e-01  5.672e-02  -8.554  < 2e-16 ***
## x616        -3.301e-01  8.969e-02  -3.680 0.000233 ***
## x617        -1.827e-01  6.866e-02  -2.661 0.007793 **
## x630         4.061e-01  8.226e-02   4.937 7.94e-07 ***
## x631        -5.231e-02  8.324e-02  -0.628 0.529743
## x632        -1.834e-03  1.198e-02  -0.153 0.878341
## x633        -5.582e-02  8.321e-02  -0.671 0.502308
## x634         2.550e-01  8.733e-02   2.920 0.003502 **
## x635        -1.746e-03  4.631e-04  -3.770 0.000163 ***
## x638        -2.093e-01  2.756e-01  -0.759 0.447654
## x639        -1.095e-01  3.934e-02  -2.783 0.005381 **
## x640        -4.050e-03  8.287e-03  -0.489 0.625058
## x641         2.018e-01  1.493e-01   1.352 0.176515
## x642         7.199e-02  1.710e-01   0.421 0.673717
## x643        -2.414e-01  2.664e-01  -0.906 0.364826
## x644         1.224e-01  9.829e-02   1.245 0.212949
## x645         5.809e-02  1.019e-01   0.570 0.568613
## x646        -3.952e-01  2.990e-01  -1.322 0.186288
## x647         1.625e-02  1.980e-01   0.082 0.934582
## x648        -4.264e-02  1.356e-01  -0.314 0.753212
## x649        -3.311e-01  2.009e-01  -1.648 0.099265 .
## x650        -1.913e-01  1.895e-01  -1.009 0.312926
## x651        -2.300e-01  1.849e-01  -1.244 0.213547
## x652         1.243e-01  2.966e-01   0.419 0.675091
## x655         4.728e-02  9.217e-02   0.513 0.607962
## x657        -5.841e-02  2.395e-02  -2.439 0.014737 *
## x723        -6.932e-05  7.125e-05  -0.973 0.330608
## x725         1.110e-04  6.963e-05   1.594 0.110908
## x728         1.089e-04  6.800e-05   1.601 0.109368
## x729         2.931e-04  2.164e-04   1.355 0.175570
## x730        -4.210e-04  2.555e-04  -1.647 0.099496 .
## x754         5.202e-02  3.586e-02   1.451 0.146860
## x893         1.651e-02  3.956e-02   0.417 0.676516
## x896        -1.945e-02  4.765e-02  -0.408 0.683164
## x897        -5.679e-02  2.242e-02  -2.532 0.011326 *
## x898        -1.725e-01  1.597e-01  -1.080 0.280307
## x901        -9.265e-03  9.676e-02  -0.096 0.923714
## x902         1.019e-02  1.342e-01   0.076 0.939482
## x906        -1.302e-02  2.423e-01  -0.054 0.957150
## x907         2.276e-01  2.638e-01   0.863 0.388285
## x908         2.860e-01  2.664e-01   1.074 0.283031
## x909         2.320e+00  9.769e-01   2.375 0.017539 *
## x910         1.079e-01  9.684e-02   1.115 0.265044
```

```
## x911       -2.575e-02  5.187e-02  -0.497 0.619510
## x912        2.885e-03  5.079e-03   0.568 0.570000
## x920        1.819e-05  2.272e-05   0.801 0.423398
## x921        2.166e-03  5.392e-03   0.402 0.687960
## x923        1.565e-03  4.692e-03   0.334 0.738722
## x929       -1.021e-02  5.223e-02  -0.195 0.845024
## x930       -1.670e-02  2.031e-02  -0.822 0.411001
## x931        1.333e-01  1.113e-01   1.197 0.231196
## x934        2.456e-02  1.158e-01   0.212 0.832042
## x935        1.434e-01  1.285e-01   1.117 0.264206
## x939       -1.186e-01  2.650e-01  -0.448 0.654413
## x940        1.925e-01  3.052e-01   0.631 0.528251
## x941       -4.223e-02  2.844e-01  -0.149 0.881932
## x942        9.319e-01  9.459e-01   0.985 0.324534
## x943       -7.397e-03  8.574e-02  -0.086 0.931246
## x961       -1.106e-01  1.569e-01  -0.705 0.480597
## x963       -7.522e-02  7.230e-02  -1.040 0.298169
## x964        8.994e-02  8.638e-02   1.041 0.297789
## x965       -2.436e-02  2.018e-02  -1.207 0.227304
## x966        1.270e-01  8.915e-02   1.425 0.154138
## x967       -2.314e-02  1.708e-02  -1.354 0.175623
## x968       -1.278e-02  8.769e-03  -1.457 0.145127
## x969       -2.669e-02  1.735e-02  -1.538 0.123937
## x970       -4.273e-02  3.507e-02  -1.218 0.223085
## x1032      -4.873e-03  2.657e-03  -1.834 0.066605 .
## x1033       1.526e-01  7.133e-02   2.139 0.032426 *
## x1035      -5.095e-02  2.455e-01  -0.208 0.835566
## x1036       2.052e-01  3.043e-01   0.674 0.500102
## x1037       9.861e-02  2.450e-01   0.402 0.687337
## x1038      -1.407e-01  3.051e-01  -0.461 0.644717
## x1039      -1.756e-02  5.304e-02  -0.331 0.740614
## x1106      -1.822e-01  1.023e-01  -1.780 0.075002 .
## x1108      -3.274e-03  1.642e-03  -1.994 0.046147 *
## x1110       7.967e-02  3.647e-02   2.184 0.028927 *
## x1132      -1.586e-03  4.332e-02  -0.037 0.970794
## x1133       1.924e-02  2.206e-02   0.872 0.383055
## x1141       4.478e-02  1.929e-02   2.321 0.020268 *
## x1143       1.296e-03  3.080e-03   0.421 0.673836
## x1144      -1.045e-02  1.793e-02  -0.583 0.560062
## x1145      -1.694e-05  8.487e-06  -1.996 0.045923 *
## x1146      -1.491e-01  1.903e-01  -0.784 0.433245
## x1147      -4.472e-04  2.544e-03  -0.176 0.860432
## x1148       2.027e-01  2.270e-01   0.893 0.371870
## x1149       2.219e-01  2.154e-01   1.030 0.303097
## x1150       1.288e-04  2.772e-03   0.046 0.962932
## x1151      -2.837e-01  2.479e-01  -1.144 0.252470
## x1152      -1.705e-03  2.719e-03  -0.627 0.530550
## x1153      -4.536e-02  3.679e-02  -1.233 0.217601
## x1157       7.663e-02  3.460e-02   2.215 0.026775 *
## x1158      -2.642e-01  2.421e-01  -1.091 0.275092
## x1159      -1.483e-01  2.349e-01  -0.631 0.527816
## x1160      -5.188e-01  2.159e-01  -2.402 0.016291 *
## x1161      -4.400e-04  2.710e-01  -0.002 0.998704
## x1162      -2.488e-01  3.386e-01  -0.735 0.462493
```

```
## x1164       -2.855e-02  5.744e-02  -0.497 0.619232
## x1165       -1.184e-01  8.909e-02  -1.328 0.184039
## x1166       -7.346e-02  2.672e-02  -2.749 0.005976 **
## x1167       -1.300e-01  6.859e-02  -1.896 0.058014 .
## x1175        2.898e-02  2.401e-02   1.207 0.227364
## x1176        9.539e-02  5.312e-02   1.796 0.072516 .
## x1177        5.631e-02  6.358e-02   0.886 0.375835
## x1178       -2.347e-02  4.102e-02  -0.572 0.567190
## x1179        2.223e-03  1.880e-02   0.118 0.905860
## x1181       -6.436e-07  3.621e-07  -1.777 0.075529 .
## x1182       -7.411e-02  5.839e-02  -1.269 0.204385
## x1183       -3.973e-03  3.386e-03  -1.174 0.240577
## x1184        4.290e-02  2.021e-02   2.123 0.033787 *
## x1185       -6.892e-04  3.409e-03  -0.202 0.839774
## x1201        5.469e-02  1.426e-01   0.384 0.701296
## x1202       -1.417e-03  1.257e-03  -1.127 0.259612
## x1203       -1.608e-02  2.811e-02  -0.572 0.567328
## x1204       -2.533e-02  1.340e-02  -1.891 0.058681 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13480.1  on 12177  degrees of freedom
## Residual deviance:  9405.3  on 12040  degrees of freedom
## AIC: 9681.3
##
## Number of Fisher Scoring iterations: 5
```

```
##
## Call:
## glm(formula = eco_situation_better ~ . - uniqueid - year - personid -
##     x1 - health + x2 * x630, family = "binomial", data = training_set)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.889e+02  2.379e+01  20.548  < 2e-16 ***
## x2           7.802e-01  1.048e-01   7.442 9.95e-14 ***
## x3           5.172e-01  3.744e-02  13.817  < 2e-16 ***
## x4          -1.487e-02  4.165e-02  -0.357 0.721106
## x7          -8.985e-04  1.640e-02  -0.055 0.956303
## x14          8.224e-02  2.935e-02   2.802 0.005081 **
## x17          4.305e-03  9.473e-04   4.545 5.50e-06 ***
## x19          9.795e-02  8.099e-02   1.209 0.226507
## x162        -9.468e-02  4.057e-02  -2.334 0.019605 *
## x163         1.119e-01  5.015e-02   2.232 0.025618 *
## x164         3.404e-02  2.073e-02   1.642 0.100535
## x472        -1.023e-02  1.508e-02  -0.678 0.497609
## x477         6.960e-02  2.453e-02   2.837 0.004549 **
## x544         5.116e-02  6.982e-02   0.733 0.463716
## x545        -9.704e-02  6.635e-02  -1.463 0.143580
## x546        -4.127e-02  7.152e-02  -0.577 0.563944
## x547         2.556e-02  6.504e-02   0.393 0.694323
## x548         6.556e-02  8.491e-02   0.772 0.440032
```

```
## x595        -3.917e-03  8.681e-02  -0.045 0.964013
## x596         1.803e-02  7.122e-02   0.253 0.800107
## x597         1.723e-01  1.578e-01   1.092 0.274812
## x613        -7.173e-02  5.083e-02  -1.411 0.158210
## x614        -2.666e-01  9.591e-02  -2.780 0.005439 **
## x615        -4.851e-01  5.673e-02  -8.551  < 2e-16 ***
## x616        -3.304e-01  8.970e-02  -3.683 0.000230 ***
## x617        -1.822e-01  6.866e-02  -2.654 0.007947 **
## x630         2.675e-01  2.099e-01   1.275 0.202466
## x631        -5.212e-02  8.325e-02  -0.626 0.531249
## x632        -1.875e-03  1.198e-02  -0.157 0.875629
## x633        -5.567e-02  8.321e-02  -0.669 0.503481
## x634         2.547e-01  8.732e-02   2.917 0.003537 **
## x635        -1.744e-03  4.628e-04  -3.768 0.000165 ***
## x638        -2.088e-01  2.756e-01  -0.758 0.448679
## x639        -1.090e-01  3.935e-02  -2.769 0.005621 **
## x640        -3.873e-03  8.291e-03  -0.467 0.640415
## x641         2.019e-01  1.493e-01   1.352 0.176281
## x642         7.536e-02  1.710e-01   0.441 0.659424
## x643        -2.389e-01  2.664e-01  -0.897 0.369792
## x644         1.226e-01  9.824e-02   1.248 0.212213
## x645         5.870e-02  1.019e-01   0.576 0.564603
## x646        -3.934e-01  2.990e-01  -1.315 0.188359
## x647         1.731e-02  1.980e-01   0.087 0.930329
## x648        -4.295e-02  1.356e-01  -0.317 0.751503
## x649        -3.324e-01  2.008e-01  -1.656 0.097763 .
## x650        -1.941e-01  1.896e-01  -1.024 0.305960
## x651        -2.305e-01  1.849e-01  -1.247 0.212498
## x652         1.175e-01  2.967e-01   0.396 0.692047
## x655         4.658e-02  9.217e-02   0.505 0.613272
## x657        -5.815e-02  2.396e-02  -2.427 0.015211 *
## x723        -6.923e-05  7.125e-05  -0.972 0.331205
## x725         1.081e-04  6.971e-05   1.550 0.121074
## x728         1.084e-04  6.802e-05   1.593 0.111092
## x729         2.889e-04  2.167e-04   1.333 0.182611
## x730        -4.146e-04  2.557e-04  -1.621 0.104954
## x754         5.158e-02  3.587e-02   1.438 0.150414
## x893         1.689e-02  3.957e-02   0.427 0.669529
## x896        -1.956e-02  4.766e-02  -0.410 0.681483
## x897        -5.658e-02  2.235e-02  -2.532 0.011355 *
## x898        -1.721e-01  1.596e-01  -1.078 0.280818
## x901        -8.893e-03  9.676e-02  -0.092 0.926769
## x902         9.997e-03  1.342e-01   0.075 0.940610
## x906        -1.092e-02  2.422e-01  -0.045 0.964051
## x907         2.302e-01  2.637e-01   0.873 0.382705
## x908         2.865e-01  2.663e-01   1.076 0.281945
## x909         2.311e+00  9.735e-01   2.374 0.017574 *
## x910         1.066e-01  9.685e-02   1.101 0.270978
## x911        -2.571e-02  5.189e-02  -0.496 0.620230
## x912         2.853e-03  5.080e-03   0.562 0.574405
## x920         1.817e-05  2.272e-05   0.800 0.423847
## x921         2.167e-03  5.392e-03   0.402 0.687778
## x923         1.596e-03  4.694e-03   0.340 0.733825
## x929        -1.036e-02  5.223e-02  -0.198 0.842828
```

```
## x930        -1.696e-02  2.032e-02  -0.835 0.403962
## x931         1.352e-01  1.114e-01   1.214 0.224891
## x934         2.601e-02  1.158e-01   0.225 0.822264
## x935         1.450e-01  1.285e-01   1.129 0.258988
## x939        -1.181e-01  2.650e-01  -0.446 0.655802
## x940         1.931e-01  3.050e-01   0.633 0.526625
## x941        -4.191e-02  2.843e-01  -0.147 0.882801
## x942         9.435e-01  9.465e-01   0.997 0.318807
## x943        -6.409e-03  8.575e-02  -0.075 0.940417
## x961        -1.088e-01  1.569e-01  -0.693 0.488269
## x963        -7.604e-02  7.233e-02  -1.051 0.293083
## x964         9.112e-02  8.640e-02   1.055 0.291568
## x965        -2.469e-02  2.018e-02  -1.223 0.221195
## x966         1.208e-01  8.957e-02   1.349 0.177405
## x967        -2.317e-02  1.708e-02  -1.356 0.175036
## x968        -1.275e-02  8.769e-03  -1.455 0.145782
## x969        -2.665e-02  1.735e-02  -1.536 0.124519
## x970        -4.223e-02  3.508e-02  -1.204 0.228667
## x1032       -4.843e-03  2.658e-03  -1.822 0.068471 .
## x1033        1.522e-01  7.134e-02   2.133 0.032930 *
## x1035       -5.453e-02  2.452e-01  -0.222 0.824001
## x1036        2.090e-01  3.039e-01   0.688 0.491701
## x1037        1.019e-01  2.447e-01   0.416 0.677122
## x1038       -1.434e-01  3.047e-01  -0.471 0.637988
## x1039       -1.675e-02  5.305e-02  -0.316 0.752248
## x1106       -1.824e-01  1.023e-01  -1.782 0.074682 .
## x1108       -3.260e-03  1.642e-03  -1.985 0.047121 *
## x1110        7.974e-02  3.647e-02   2.187 0.028777 *
## x1132       -1.813e-03  4.333e-02  -0.042 0.966624
## x1133        1.929e-02  2.206e-02   0.874 0.381860
## x1141        4.494e-02  1.930e-02   2.329 0.019859 *
## x1143        1.291e-03  3.080e-03   0.419 0.675037
## x1144       -1.046e-02  1.793e-02  -0.583 0.559776
## x1145       -1.697e-05  8.487e-06  -2.000 0.045528 *
## x1146       -1.491e-01  1.903e-01  -0.784 0.433249
## x1147       -4.370e-04  2.543e-03  -0.172 0.863543
## x1148        2.041e-01  2.273e-01   0.898 0.369084
## x1149        2.230e-01  2.155e-01   1.035 0.300810
## x1150        1.363e-04  2.772e-03   0.049 0.960771
## x1151       -2.862e-01  2.483e-01  -1.153 0.248970
## x1152       -1.679e-03  2.720e-03  -0.617 0.537042
## x1153       -4.519e-02  3.679e-02  -1.228 0.219323
## x1157        7.681e-02  3.460e-02   2.220 0.026423 *
## x1158       -2.619e-01  2.421e-01  -1.082 0.279303
## x1159       -1.457e-01  2.349e-01  -0.620 0.535203
## x1160       -5.184e-01  2.158e-01  -2.402 0.016307 *
## x1161        3.497e-03  2.710e-01   0.013 0.989704
## x1162       -2.407e-01  3.388e-01  -0.710 0.477452
## x1164       -2.878e-02  5.747e-02  -0.501 0.616603
## x1165       -1.186e-01  8.912e-02  -1.331 0.183085
## x1166       -7.338e-02  2.672e-02  -2.746 0.006033 **
## x1167       -1.304e-01  6.860e-02  -1.900 0.057390 .
## x1175        2.897e-02  2.401e-02   1.206 0.227657
## x1176        9.571e-02  5.312e-02   1.802 0.071579 .
```

```
## x1177        5.613e-02  6.358e-02   0.883 0.377340
## x1178       -2.323e-02  4.102e-02  -0.566 0.571289
## x1179        2.427e-03  1.880e-02   0.129 0.897294
## x1181       -6.474e-07  3.622e-07  -1.787 0.073886 .
## x1182       -7.430e-02  5.840e-02  -1.272 0.203289
## x1183       -3.956e-03  3.386e-03  -1.168 0.242688
## x1184        4.263e-02  2.022e-02   2.109 0.034979 *
## x1185       -7.192e-04  3.410e-03  -0.211 0.832948
## x1201        5.598e-02  1.426e-01   0.393 0.694638
## x1202       -1.427e-03  1.257e-03  -1.135 0.256342
## x1203       -1.621e-02  2.812e-02  -0.576 0.564295
## x1204       -2.524e-02  1.340e-02  -1.884 0.059530 .
## x2:x630      4.592e-02  6.402e-02   0.717 0.473231
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13480.1  on 12177  degrees of freedom
## Residual deviance:  9404.8  on 12039  degrees of freedom
## AIC: 9682.8
##
## Number of Fisher Scoring iterations: 5
```

```
##
## Call:
## glm(formula = log(health) ~ . - uniqueid - year - personid, data = training_set[subset,
##     ])
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -4.704e+00  4.468e+00  -1.053 0.292423
## x1                   2.942e-02  9.752e-03   3.017 0.002567 **
## x2                   6.456e-02  7.521e-03   8.584  < 2e-16 ***
## x3                   3.536e-03  7.128e-03   0.496 0.619878
## x4                   5.390e-03  9.326e-03   0.578 0.563276
## x7                  -5.050e-03  2.975e-03  -1.697 0.089723 .
## x14                  1.484e-02  5.287e-03   2.806 0.005029 **
## x17                 -6.016e-06  1.955e-04  -0.031 0.975454
## x19                  3.945e-02  1.796e-02   2.197 0.028092 *
## x162                -2.680e-02  7.861e-03  -3.409 0.000656 ***
## x163                -1.763e-02  9.353e-03  -1.885 0.059477 .
## x164                -1.357e-02  3.855e-03  -3.520 0.000435 ***
## x472                -3.012e-03  2.684e-03  -1.122 0.261752
## x477                 1.050e-03  4.162e-03   0.252 0.800781
## x544                -2.904e-02  1.384e-02  -2.098 0.035907 *
## x545                -1.109e-02  1.206e-02  -0.920 0.357825
## x546                -2.183e-02  1.362e-02  -1.603 0.109099
## x547                -1.716e-02  1.199e-02  -1.432 0.152244
## x548                -8.078e-03  1.578e-02  -0.512 0.608803
## x595                -3.354e-02  1.572e-02  -2.134 0.032902 *
## x596                 8.131e-03  1.221e-02   0.666 0.505406
## x597                 2.516e-02  2.340e-02   1.075 0.282441
## x613                 1.777e-02  7.660e-03   2.319 0.020404 *
```

```
## x614                -1.548e-02  1.539e-02  -1.006 0.314562
## x615                -2.931e-02  1.096e-02  -2.674 0.007507 **
## x616                 1.363e-02  1.497e-02   0.911 0.362513
## x617                -3.120e-02  1.178e-02  -2.649 0.008090 **
## x630                -1.849e-02  1.825e-02  -1.013 0.311235
## x631                 4.828e-03  1.758e-02   0.275 0.783665
## x632                -4.526e-04  2.393e-03  -0.189 0.850010
## x633                 1.101e-02  1.756e-02   0.627 0.530424
## x634                 1.337e-02  2.005e-02   0.667 0.504954
## x635                -1.519e-04  1.109e-04  -1.370 0.170677
## x638                 3.986e-02  5.957e-02   0.669 0.503421
## x639                 3.888e-03  7.625e-03   0.510 0.610143
## x640                 1.695e-04  1.495e-03   0.113 0.909723
## x641                 1.211e-02  3.087e-02   0.392 0.694904
## x642                 4.132e-02  3.226e-02   1.281 0.200298
## x643                 7.118e-02  4.679e-02   1.521 0.128228
## x644                -1.926e-02  1.662e-02  -1.158 0.246729
## x645                -1.850e-02  1.801e-02  -1.027 0.304450
## x646                -9.969e-04  5.835e-02  -0.017 0.986368
## x647                -1.142e-02  3.216e-02  -0.355 0.722580
## x648                 2.367e-03  2.426e-02   0.098 0.922271
## x649                 4.119e-03  3.334e-02   0.124 0.901691
## x650                 5.494e-03  3.051e-02   0.180 0.857122
## x651                -1.537e-02  3.047e-02  -0.504 0.614006
## x652                -3.398e-03  5.367e-02  -0.063 0.949513
## x655                -1.069e-02  1.672e-02  -0.639 0.522556
## x657                 3.086e-02  5.064e-03   6.093 1.18e-09 ***
## x723                -1.792e-05  1.092e-05  -1.641 0.100874
## x725                 3.969e-06  1.017e-05   0.390 0.696328
## x728                -2.522e-06  1.678e-05  -0.150 0.880503
## x729                -5.635e-05  5.847e-05  -0.964 0.335206
## x730                 5.592e-05  6.550e-05   0.854 0.393257
## x754                 1.819e-02  8.585e-03   2.118 0.034193 *
## x893                 7.853e-04  6.962e-03   0.113 0.910192
## x896                -6.972e-03  9.145e-03  -0.762 0.445834
## x897                -1.271e-03  4.458e-03  -0.285 0.775500
## x898                -5.774e-03  2.889e-02  -0.200 0.841626
## x901                 1.510e-02  1.883e-02   0.802 0.422542
## x902                -3.012e-02  2.433e-02  -1.238 0.215763
## x906                -2.261e-03  5.024e-02  -0.045 0.964111
## x907                 1.387e-02  5.342e-02   0.260 0.795076
## x908                -4.889e-02  5.384e-02  -0.908 0.363862
## x909                 7.971e-02  1.972e-01   0.404 0.686118
## x910                -8.618e-03  1.913e-02  -0.451 0.652351
## x911                -6.648e-03  8.725e-03  -0.762 0.446112
## x912                 6.824e-04  8.746e-04   0.780 0.435243
## x920                -8.914e-07  4.196e-06  -0.212 0.831761
## x921                -4.941e-05  1.006e-03  -0.049 0.960821
## x923                -2.219e-04  8.390e-04  -0.264 0.791415
## x929                -2.374e-03  1.065e-02  -0.223 0.823691
## x930                -3.527e-03  6.590e-03  -0.535 0.592513
## x931                -3.693e-03  1.943e-02  -0.190 0.849261
## x934                 2.760e-02  3.132e-02   0.881 0.378325
## x935                 4.380e-02  3.725e-02   1.176 0.239682
```

```
## x939                1.083e-01  7.157e-02   1.513 0.130313
## x940                1.610e-01  8.418e-02   1.912 0.055932 .
## x941                1.512e-01  8.468e-02   1.785 0.074273 .
## x942                2.231e-01  3.177e-01   0.702 0.482579
## x943               -2.047e-02  1.882e-02  -1.088 0.276760
## x961               -6.411e-02  3.694e-02  -1.736 0.082695 .
## x963               -5.029e-02  2.706e-02  -1.859 0.063112 .
## x964                4.439e-02  3.016e-02   1.472 0.141154
## x965               -5.548e-05  5.405e-03  -0.010 0.991810
## x966                1.523e-02  2.026e-02   0.752 0.452269
## x967               -3.264e-03  2.764e-03  -1.181 0.237714
## x968               -7.616e-04  1.622e-03  -0.470 0.638623
## x969               -4.577e-03  2.845e-03  -1.609 0.107700
## x970               -2.465e-02  9.319e-03  -2.646 0.008174 **
## x1032               6.303e-04  6.259e-04   1.007 0.313939
## x1033              -1.018e-02  1.946e-02  -0.523 0.601048
## x1035              -6.015e-02  5.483e-02  -1.097 0.272750
## x1036               1.452e-03  9.345e-02   0.016 0.987607
## x1037               5.688e-02  5.476e-02   1.039 0.298926
## x1038               3.096e-02  9.307e-02   0.333 0.739394
## x1039               3.864e-03  8.863e-03   0.436 0.662880
## x1106              -2.110e-02  1.671e-02  -1.263 0.206679
## x1108               4.580e-04  3.041e-04   1.506 0.132098
## x1110               1.107e-02  6.869e-03   1.611 0.107201
## x1132              -3.660e-03  7.978e-03  -0.459 0.646411
## x1133              -7.992e-03  4.279e-03  -1.868 0.061867 .
## x1141              -3.143e-03  3.541e-03  -0.888 0.374829
## x1143              -9.685e-04  5.500e-04  -1.761 0.078331 .
## x1144               1.816e-03  3.284e-03   0.553 0.580222
## x1145               1.205e-07  1.793e-06   0.067 0.946408
## x1146              -1.573e-02  3.720e-02  -0.423 0.672465
## x1147               1.476e-04  4.705e-04   0.314 0.753829
## x1148              -2.401e-03  4.374e-02  -0.055 0.956225
## x1149               3.573e-02  4.253e-02   0.840 0.400831
## x1150               5.906e-04  5.170e-04   1.142 0.253357
## x1151              -1.667e-02  4.862e-02  -0.343 0.731682
## x1152               2.730e-04  4.635e-04   0.589 0.555854
## x1153              -6.969e-03  6.888e-03  -1.012 0.311723
## x1157               3.681e-03  6.687e-03   0.551 0.581984
## x1158               8.762e-02  6.294e-02   1.392 0.163950
## x1159               1.215e-01  6.226e-02   1.952 0.050971 .
## x1160               1.382e-01  5.785e-02   2.389 0.016949 *
## x1161               8.043e-02  6.839e-02   1.176 0.239643
## x1162              -2.203e-02  8.274e-02  -0.266 0.790035
## x1164               7.998e-03  1.042e-02   0.768 0.442773
## x1165              -2.480e-02  1.617e-02  -1.534 0.125033
## x1166              -6.278e-03  4.991e-03  -1.258 0.208492
## x1167               7.150e-04  1.179e-02   0.061 0.951629
## x1175              -5.572e-03  4.330e-03  -1.287 0.198216
## x1176               5.497e-03  1.060e-02   0.518 0.604150
## x1177              -3.556e-03  1.175e-02  -0.303 0.762270
## x1178               1.352e-02  7.628e-03   1.772 0.076458 .
## x1179              -1.851e-02  3.092e-03  -5.987 2.27e-09 ***
## x1181               5.576e-08  5.931e-08   0.940 0.347229
```

```
## x1182                 7.078e-03  1.039e-02   0.681 0.495634
## x1183                 1.205e-03  6.027e-04   2.000 0.045583 *
## x1184                 6.164e-04  3.634e-03   0.170 0.865315
## x1185                 4.790e-04  5.738e-04   0.835 0.403810
## x1201                 5.878e-02  2.728e-02   2.155 0.031194 *
## x1202                -1.034e-04  2.300e-04  -0.450 0.653066
## x1203                -1.804e-03  5.129e-03  -0.352 0.725083
## x1204                -8.432e-04  2.390e-03  -0.353 0.724256
## eco_situation_better  1.604e-03  1.866e-02   0.086 0.931487
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.126312)
##
##     Null deviance: 910.03  on 5801  degrees of freedom
## Residual deviance: 715.18  on 5662  degrees of freedom
## AIC: 4601.3
##
## Number of Fisher Scoring iterations: 2
```

```
##
## Call:
## glm(formula = health_binary ~ . - uniqueid - year - personid -
##     health, family = "binomial", data = training_set)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           4.202e+01  2.701e+01   1.555 0.119862
## x1                   -7.796e-02  6.083e-02  -1.282 0.199956
## x2                   -4.099e-01  4.177e-02  -9.814  < 2e-16 ***
## x3                   -9.113e-02  4.195e-02  -2.173 0.029808 *
## x4                   -4.634e-02  4.789e-02  -0.968 0.333190
## x7                    3.145e-02  1.788e-02   1.759 0.078532 .
## x14                  -1.011e-01  3.133e-02  -3.227 0.001251 **
## x17                  -8.424e-04  1.038e-03  -0.811 0.417223
## x19                  -2.165e-01  8.867e-02  -2.442 0.014602 *
## x162                  1.733e-01  4.490e-02   3.859 0.000114 ***
## x163                  7.601e-02  5.386e-02   1.411 0.158159
## x164                  9.797e-02  2.193e-02   4.468 7.90e-06 ***
## x472                  8.352e-03  1.644e-02   0.508 0.611415
## x477                 -5.404e-02  2.615e-02  -2.067 0.038756 *
## x544                  3.397e-02  7.243e-02   0.469 0.639095
## x545                  3.824e-02  6.620e-02   0.578 0.563480
## x546                  5.100e-02  7.334e-02   0.695 0.486813
## x547                 -2.077e-03  6.537e-02  -0.032 0.974659
## x548                  5.975e-02  9.100e-02   0.657 0.511455
## x595                  1.104e-01  9.783e-02   1.129 0.259067
## x596                  2.957e-02  7.446e-02   0.397 0.691226
## x597                 -8.228e-02  1.673e-01  -0.492 0.622809
## x613                 -4.134e-02  4.978e-02  -0.830 0.406278
## x614                 -1.681e-02  1.002e-01  -0.168 0.866748
## x615                  9.405e-02  6.468e-02   1.454 0.145952
## x616                 -1.393e-01  9.628e-02  -1.447 0.147977
## x617                  2.300e-01  7.681e-02   2.994 0.002755 **
```

```
## x630                3.303e-02  9.458e-02   0.349 0.726899
## x631               -1.468e-02  9.160e-02  -0.160 0.872641
## x632                1.074e-04  1.320e-02   0.008 0.993507
## x633               -5.025e-02  9.153e-02  -0.549 0.583005
## x634               -2.702e-02  9.693e-02  -0.279 0.780435
## x635                1.611e-04  4.718e-04   0.341 0.732741
## x638               -3.973e-01  2.797e-01  -1.420 0.155525
## x639                3.218e-02  4.366e-02   0.737 0.461051
## x640               -8.620e-03  1.030e-02  -0.837 0.402647
## x641               -1.378e-03  1.747e-01  -0.008 0.993707
## x642               -1.507e-01  1.839e-01  -0.820 0.412499
## x643               -1.658e-01  2.504e-01  -0.662 0.508042
## x644               -1.276e-03  1.064e-01  -0.012 0.990434
## x645                1.831e-02  1.139e-01   0.161 0.872245
## x646               -1.509e-01  3.287e-01  -0.459 0.646206
## x647               -2.965e-01  2.054e-01  -1.443 0.148983
## x648               -7.023e-02  1.470e-01  -0.478 0.632845
## x649               -1.134e-01  2.102e-01  -0.539 0.589599
## x650               -2.372e-01  1.988e-01  -1.193 0.232878
## x651               -2.482e-01  1.943e-01  -1.278 0.201414
## x652                3.593e-01  3.844e-01   0.935 0.350016
## x655                9.949e-02  9.977e-02   0.997 0.318705
## x657               -2.082e-01  2.846e-02  -7.317 2.53e-13 ***
## x723                9.265e-05  9.830e-05   0.942 0.345937
## x725                1.282e-04  9.324e-05   1.375 0.169092
## x728               -1.084e-04  8.958e-05  -1.210 0.226118
## x729               -3.424e-04  2.937e-04  -1.166 0.243625
## x730                2.886e-04  3.211e-04   0.899 0.368807
## x754               -1.202e-02  4.186e-02  -0.287 0.774043
## x893                1.116e-02  4.074e-02   0.274 0.784173
## x896                4.479e-02  5.461e-02   0.820 0.412075
## x897                1.940e-02  2.243e-02   0.865 0.386959
## x898                1.080e-01  1.643e-01   0.657 0.511169
## x901                1.136e-01  1.070e-01   1.062 0.288190
## x902                2.941e-01  1.504e-01   1.956 0.050449 .
## x906                2.167e-01  2.798e-01   0.774 0.438650
## x907                4.411e-01  3.141e-01   1.404 0.160185
## x908                3.296e-01  3.107e-01   1.061 0.288784
## x909               -5.984e-01  9.772e-01  -0.612 0.540280
## x910               -8.315e-02  1.082e-01  -0.769 0.442178
## x911                2.399e-02  5.528e-02   0.434 0.664277
## x912               -4.618e-03  5.498e-03  -0.840 0.401026
## x920                2.647e-05  2.543e-05   1.041 0.297870
## x921               -5.850e-03  6.064e-03  -0.965 0.334685
## x923                3.068e-03  5.211e-03   0.589 0.555997
## x929                3.971e-02  6.208e-02   0.640 0.522382
## x930               -4.028e-02  5.522e-02  -0.730 0.465691
## x931               -6.784e-02  1.183e-01  -0.573 0.566434
## x934                4.808e-02  2.151e-01   0.224 0.823137
## x935                9.928e-02  2.669e-01   0.372 0.709915
## x939                5.066e-01  4.615e-01   1.098 0.272331
## x940                3.190e-01  5.425e-01   0.588 0.556523
## x941                4.643e-01  5.652e-01   0.821 0.411405
## x942                1.808e+00  2.666e+00   0.678 0.497654
```

```
## x943                -1.414e-02  1.036e-01  -0.137 0.891418
## x961                -2.710e-02  1.812e-01  -0.150 0.881120
## x963                 4.998e-02  9.381e-02   0.533 0.594221
## x964                -7.776e-02  1.101e-01  -0.706 0.480040
## x965                 6.530e-03  2.413e-02   0.271 0.786686
## x966                -4.953e-02  1.052e-01  -0.471 0.637771
## x967                -1.633e-03  1.652e-02  -0.099 0.921245
## x968                -1.055e-02  9.763e-03  -1.081 0.279733
## x969                 3.359e-03  1.680e-02   0.200 0.841484
## x970                 3.158e-02  4.205e-02   0.751 0.452676
## x1032               -8.506e-04  3.063e-03  -0.278 0.781239
## x1033               -2.136e-02  8.489e-02  -0.252 0.801375
## x1035                1.218e-01  2.826e-01   0.431 0.666570
## x1036               -5.965e-02  3.537e-01  -0.169 0.866088
## x1037               -1.552e-01  2.823e-01  -0.550 0.582595
## x1038                1.104e-01  3.552e-01   0.311 0.755999
## x1039               -6.987e-02  5.410e-02  -1.292 0.196506
## x1106                3.002e-01  1.080e-01   2.780 0.005433 **
## x1108               -3.970e-03  1.697e-03  -2.340 0.019298 *
## x1110                1.864e-02  4.053e-02   0.460 0.645593
## x1132                6.302e-02  4.779e-02   1.319 0.187231
## x1133                9.478e-03  2.379e-02   0.398 0.690359
## x1141                6.163e-04  2.110e-02   0.029 0.976693
## x1143                1.911e-03  3.336e-03   0.573 0.566608
## x1144               -1.411e-02  1.944e-02  -0.726 0.468076
## x1145               -2.609e-07  9.326e-06  -0.028 0.977680
## x1146               -7.997e-02  2.442e-01  -0.327 0.743334
## x1147               -2.327e-03  2.896e-03  -0.803 0.421776
## x1148                2.501e-01  3.195e-01   0.783 0.433714
## x1149                3.098e-01  2.794e-01   1.109 0.267446
## x1150                2.432e-03  3.144e-03   0.774 0.439188
## x1151               -4.776e-01  3.576e-01  -1.335 0.181742
## x1152               -3.355e-03  3.123e-03  -1.074 0.282669
## x1153                2.892e-02  4.100e-02   0.705 0.480517
## x1157               -4.406e-02  4.005e-02  -1.100 0.271300
## x1158               -2.127e-01  2.556e-01  -0.832 0.405261
## x1159               -4.909e-01  2.486e-01  -1.974 0.048333 *
## x1160               -1.958e-01  2.237e-01  -0.875 0.381488
## x1161               -1.568e-01  2.988e-01  -0.525 0.599793
## x1162               -3.141e-01  3.435e-01  -0.914 0.360512
## x1164               -3.592e-02  5.728e-02  -0.627 0.530582
## x1165                6.642e-02  8.372e-02   0.793 0.427566
## x1166                5.023e-02  3.073e-02   1.635 0.102143
## x1167                5.189e-02  7.237e-02   0.717 0.473374
## x1175                8.270e-03  2.572e-02   0.322 0.747758
## x1176               -7.223e-03  5.756e-02  -0.125 0.900133
## x1177                4.179e-02  7.099e-02   0.589 0.556096
## x1178               -3.909e-02  4.542e-02  -0.861 0.389414
## x1179                1.197e-01  1.838e-02   6.515 7.28e-11 ***
## x1181               -2.602e-07  3.653e-07  -0.712 0.476363
## x1182               -2.698e-02  6.285e-02  -0.429 0.667763
## x1183               -8.409e-03  3.682e-03  -2.284 0.022370 *
## x1184                2.091e-02  2.206e-02   0.948 0.343251
## x1185               -1.551e-03  3.490e-03  -0.444 0.656832
```

```
## x1201                 1.705e-01  1.589e-01   1.074 0.283040
## x1202                 5.944e-04  1.382e-03   0.430 0.667078
## x1203                -1.117e-02  3.125e-02  -0.358 0.720713
## x1204                 1.221e-02  1.476e-02   0.827 0.408344
## eco_situation_better -7.844e-03  1.104e-01  -0.071 0.943337
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10043.4  on 12177  degrees of freedom
## Residual deviance:  8207.1  on 12038  degrees of freedom
## AIC: 8487.1
##
## Number of Fisher Scoring iterations: 6
```