finalLogitModel

DominiqueBarnes

2024-05-05

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(haven)
library(readxl)
library(MASS)
```

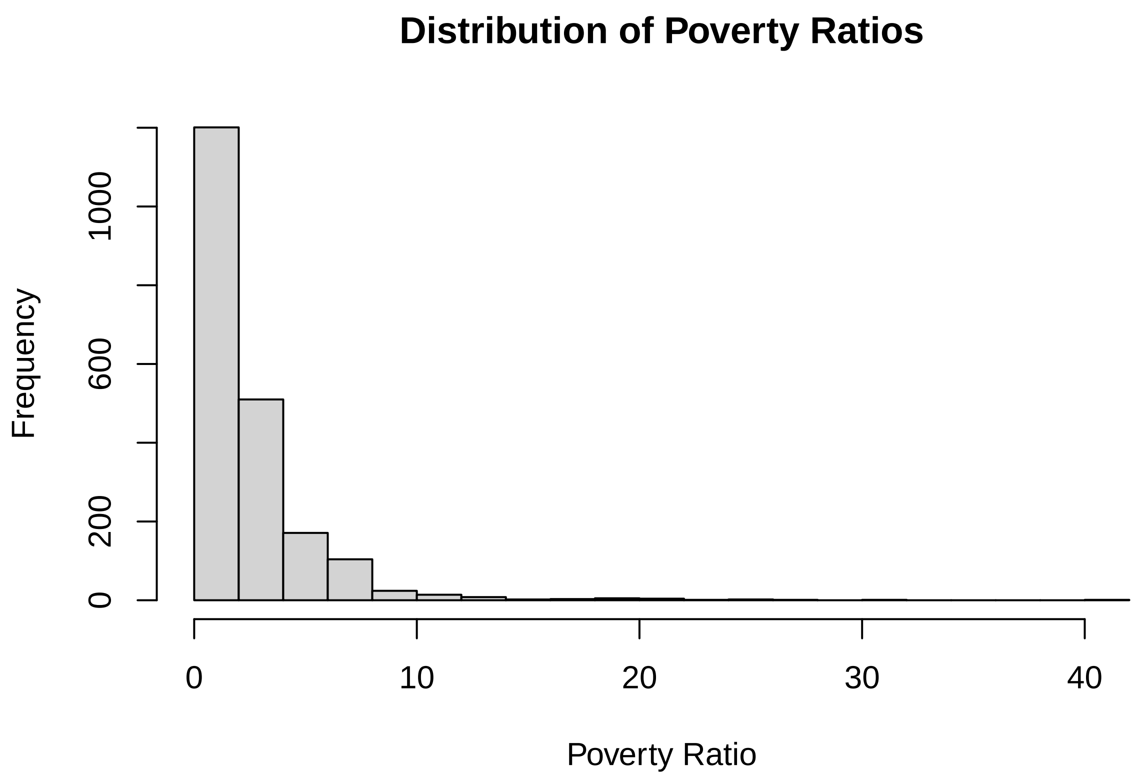
```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(ggplot2)
```

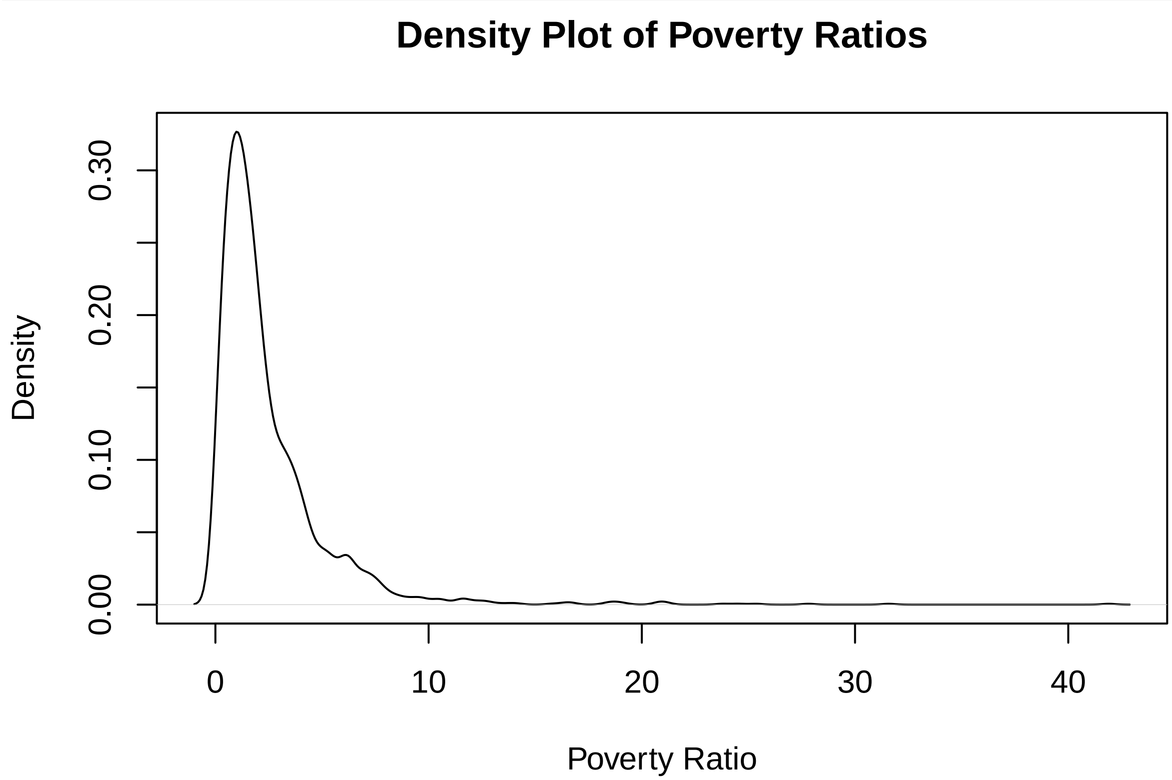
```
df <- as.data.frame(read_dta("/Users/dominiquebarnes/Desktop/SPR24_Coursework/DATA 2020/FFdata/wave6/FF_
other_var <- read.csv("/Users/dominiquebarnes/Desktop/SPR24_Coursework/DATA 2020/Final_Project/Finances/
other_df <- as.data.frame(other_var)
# Select Columns
other_var_code <- other_df$Variable
df_select<- df %>% dplyr::select(all_of(other_var_code))
```

```
df_filt <- df_select %>%
  filter_all(all_vars(. !=-9 &. !=-8 &. !=-7  &.  !=-5 &. !=-4 &. !=-3 &. !=-2 &. !=-1 ))
```

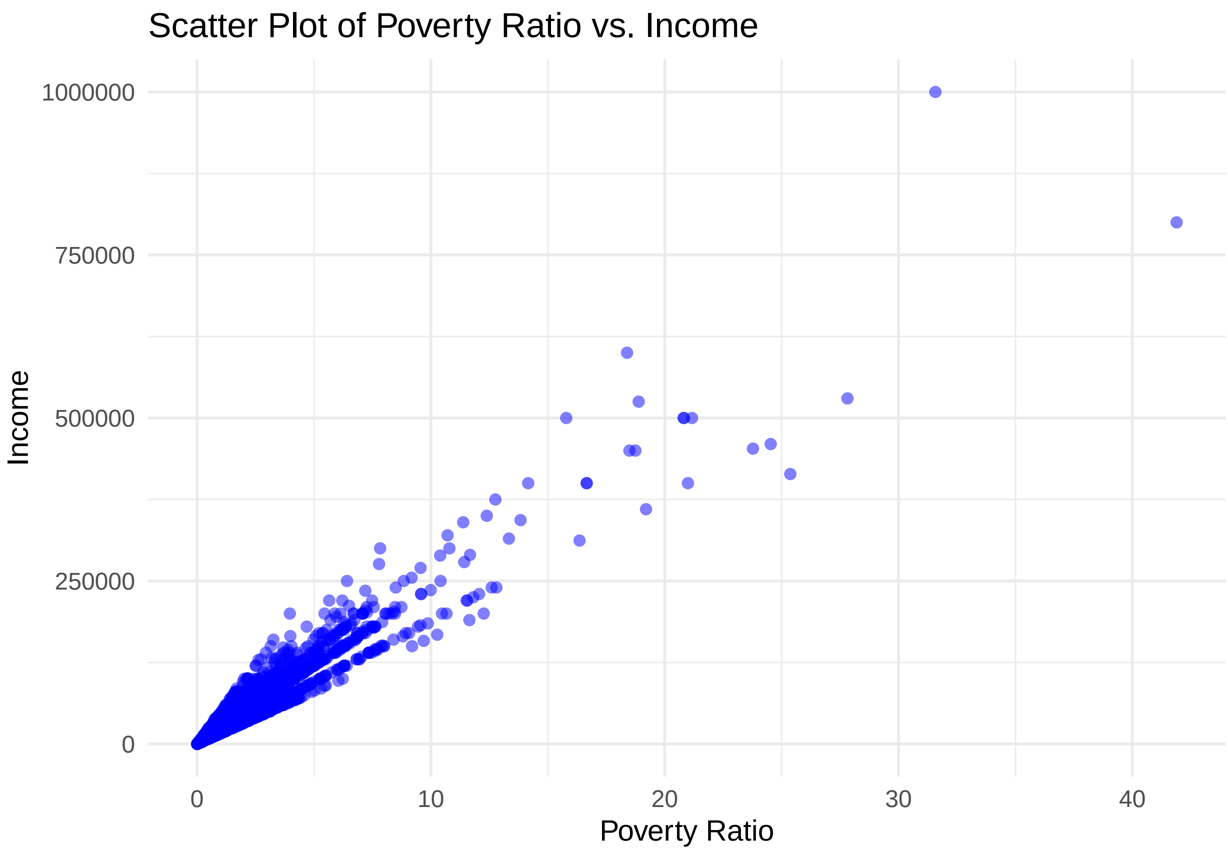
```
# cp6povco poverty ratio
poverty_ratio = df_filt$cp6povco
#Histogram
hist(poverty_ratio, breaks = 20, main = "Distribution of Poverty Ratios", xlab = "Poverty Ratio")
```



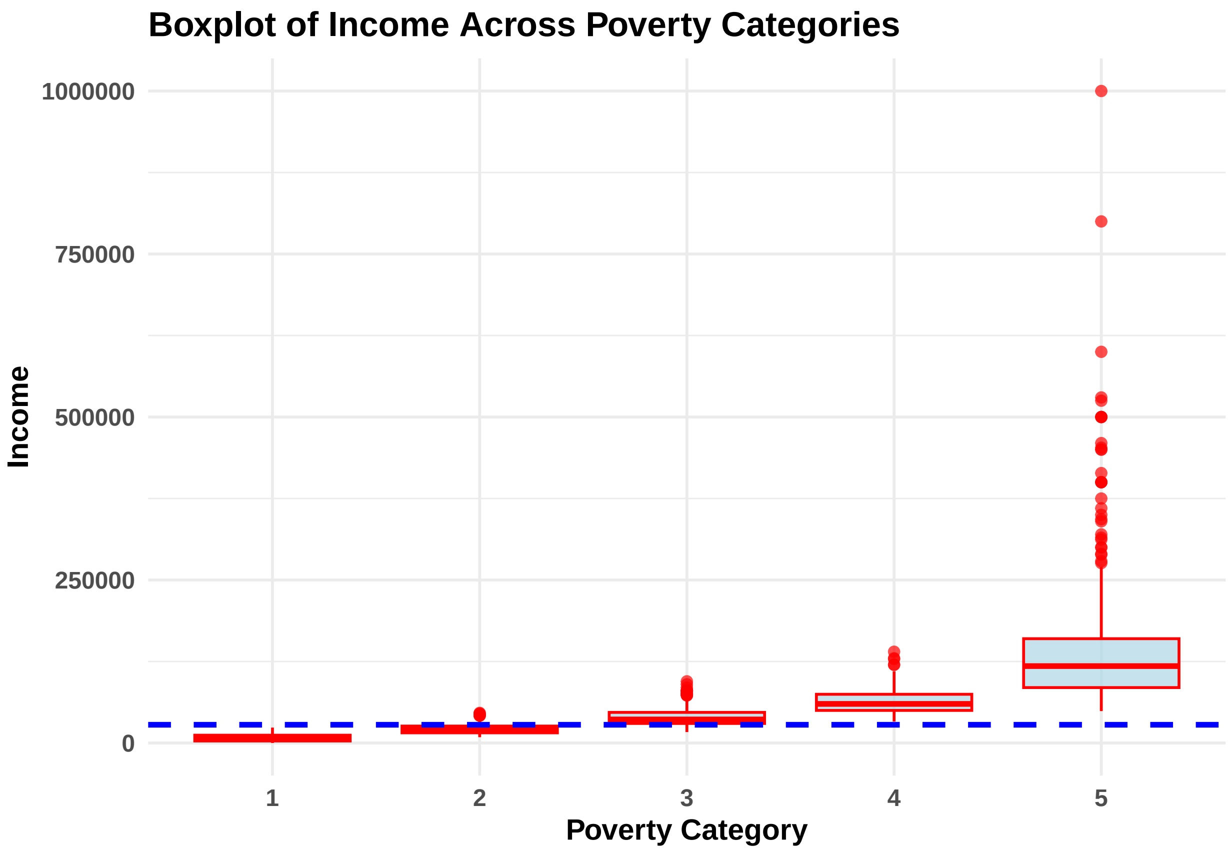

```
#Density Plot
# Assuming your poverty ratio variable is named "poverty_ratio"
plot(density(poverty_ratio), main = "Density Plot of Poverty Ratios", xlab = "Poverty Ratio", ylab = "De
```



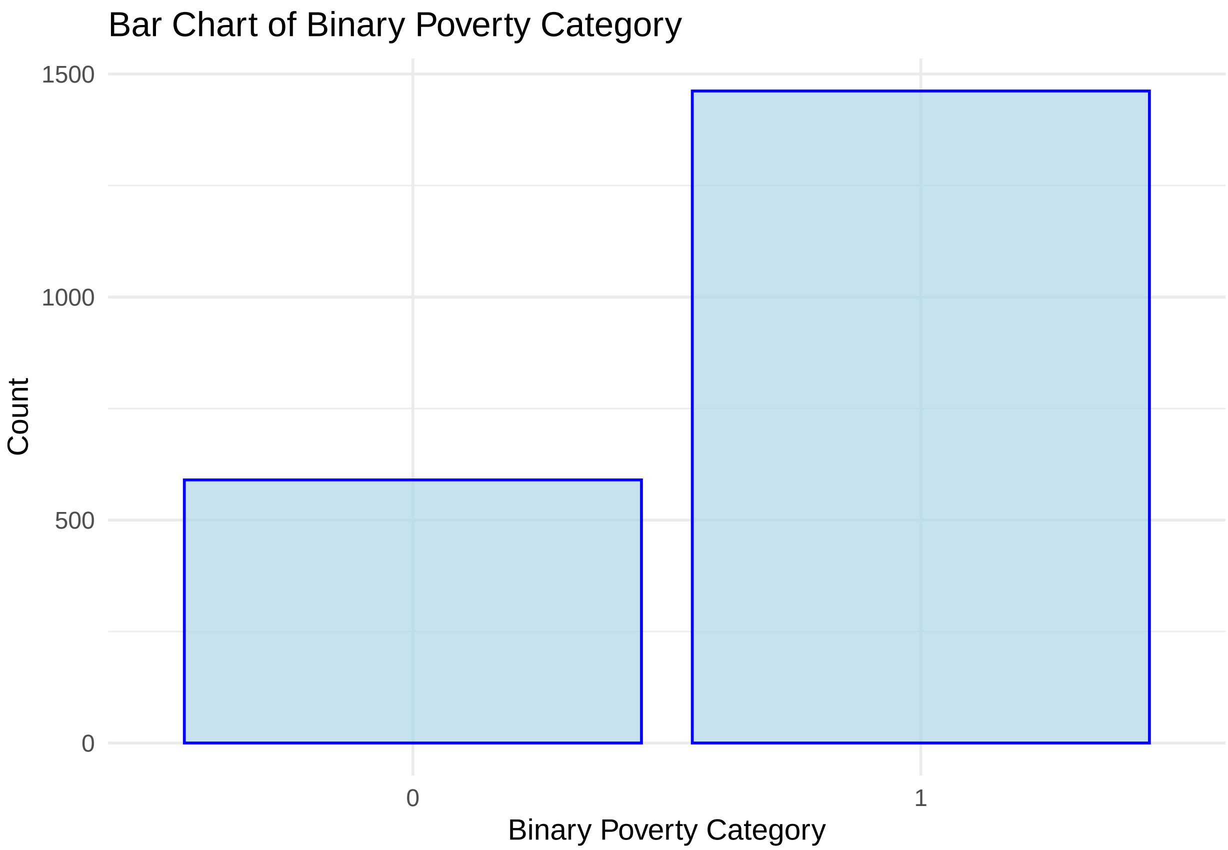
```r
# Poverty Ratio v Income
income = df_filt$cp6hhinc
ggplot(df_filt, aes(x = poverty_ratio, y = income)) +
  geom_point(color = 'blue', alpha = 0.5) +
  labs(title = 'Scatter Plot of Poverty Ratio vs. Income',
       x = 'Poverty Ratio',
       y = 'Income') +
  theme_minimal()
```




```r
# Poverty Cat v Income

ggplot(df_filt, aes(x = factor(cp6povca), y = income)) +
  geom_boxplot(color = 'red', fill = 'lightblue', alpha = 0.7) +
  geom_hline(yintercept = 28000, color = "blue", linetype = "dashed", linewidth = 1) +
  labs(title = 'Boxplot of Income Across Poverty Categories',
       x = 'Poverty Category',
       y = 'Income') +
  theme_minimal()+
  theme(axis.text.x = element_text(face = "bold"),
        axis.text.y = element_text(face = "bold"),
        axis.title = element_text(face = "bold"),
        plot.title = element_text(face = "bold"))
```




```r
df_filt$binary_poverty_category <- NA
# Assign 0 to rows where poverty category is below or equal to 2, and 1 otherwise
df_filt$binary_poverty_category[df_filt$cp6povca <= 2] <- 0
df_filt$binary_poverty_category[df_filt$cp6povca > 2] <- 1

ggplot(df_filt, aes(x = factor(binary_poverty_category))) +
  geom_bar(fill = 'lightblue', color = 'blue', alpha = 0.7) +
  labs(title = 'Bar Chart of Binary Poverty Category',
       x = 'Binary Poverty Category',
       y = 'Count') +
  theme_minimal()
```




```
dependent_vars <- c('cp6hhsize','p6a3','p6a4','p6i19','p6i20_8','p6i21','p6j7','p6j8','p6j9','p6j11','p6

independent_var <- df_filt$binary_poverty_category

final_df <- df_filt %>% dplyr::select(all_of(dependent_vars))
```

```
data <-final_df


binary_vars <- c('p6a4','p6i19','p6i20_8','p6j14','p6j15','p6j24','p6j30','p6j33','p6j34','p6j35','p6j3

#cat_vars <- c('p6k19_code_pub','p6k36_code_pub')

cont_vars <- c('cp6hhsize','p6a3','p6i21','p6j7','p6j8','p6j9','p6j11','p6j25','p6j31','p6k34','p6k65',


#One hot encoding Categorical Variables
# Perform one-hot encoding for each categorical variable
#encoded_data <- model.matrix(~ . - 1, data = data[, cat_vars])
# Combine the encoded data with the original data frame
#data <- cbind(data, encoded_data)

#Binary
# Replace 2 with 0 for each binary variable
for (var in binary_vars) {
  data[[var]] <- ifelse(data[[var]] == 2, 0, data[[var]])
}
```

```r
data[, cont_vars] <- scale(data[, cont_vars])
```

```r
# Set the seed for reproducibility
set.seed(123)

# Assuming 'data' is your preprocessed dataset
n <- nrow(data)
train_indices <- sample(1:n, 0.8 * n)  # 80% for training
data_train <- data[train_indices, ]
data_test <- data[-train_indices, ]
```

```r
#Correlation Matrics
correlation_matrix <- cor(data_train[, dependent_vars])
highly_correlated_pairs <- which(correlation_matrix > 0.7 & correlation_matrix != 1, arr.ind = TRUE)

#Remove one variable from each highly correlated pair
vars_to_remove <- rownames(correlation_matrix)[highly_correlated_pairs[, "col"]]
data_train_filtered <- data_train[, !colnames(data_train) %in% vars_to_remove]
data_test_filtered <- data_test[, !colnames(data_test) %in% vars_to_remove]
```

```r
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```r
ctrl <- trainControl(method = "cv", number = 5)
data_train_filtered$binary_poverty_category <- factor(data_train_filtered$binary_poverty_category, level

# Build Logistic Regression Model
model_logit <- train(binary_poverty_category ~. , data = data_train_filtered, method = "glm"  , trContr
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(model_logit)
```

```
##
## Call:
## NULL
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -0.4446363  1.5251369  -0.292 0.770639    
## cp6hhsize   -0.6793373  0.0860925  -7.891 3.00e-15 ***
## p6a3         0.0509705  0.0833278   0.612 0.540745    
## p6a4        -0.4202767  0.2476502  -1.697 0.089686 .  
## p6i21        0.2132001  0.2134175   0.999 0.317804    
## p6j7         2.4780057  0.4878655   5.079 3.79e-07 ***
## p6j11        0.4691230  0.1053989   4.451 8.55e-06 ***
## p6j24       -1.7862652  1.9733917  -0.905 0.365373    
## p6j25       -0.0830534  0.0905701  -0.917 0.359139    
## p6j30       -0.9075611  2.6456288  -0.343 0.731567    
## p6j31        0.0826560  0.0926418   0.892 0.372280    
## p6j33        0.4427439  0.1912393   2.315 0.020606 *  
## p6j36       -0.4281449  0.1968054  -2.175 0.029595 *  
## p6j37        0.0161149  2.4447185   0.007 0.994741    
## p6j38       -0.7026217  2.6064043  -0.270 0.787487    
## p6j39        1.6397589  2.0829802   0.787 0.431154    
## p6j40       -0.2615772  2.3730612  -0.110 0.912229    
## p6j41        0.0779611  2.0421258   0.038 0.969547    
## p6j42        0.1352813  1.8990512   0.071 0.943210    
## p6j43        0.2705381  1.8498161   0.146 0.883723    
## p6j44        2.5629234  1.9490214   1.315 0.188517    
## p6j45        2.3981263  2.9311505   0.818 0.413270    
## p6j46       -3.6337170  2.8683360  -1.267 0.205213    
## p6j47       -3.4381576  2.0271261  -1.696 0.089872 .  
## p6k13        0.0845278  0.0410143   2.061 0.039309 *  
## p6k69        0.1388736  0.1712816   0.811 0.417486    
## p6k70        0.4656053  0.2028899   2.295 0.021741 *  
## p6k71        0.1437858  0.1145834   1.255 0.209531    
## p6k72        1.0780888  0.3416151   3.156 0.001600 ** 
## p6b31       -0.5656642  0.2218858  -2.549 0.010792 *  
## p6e24        0.0198070  0.0311842   0.635 0.525324    
## p6e29        0.3033316  0.0849487   3.571 0.000356 ***
## p6f9        -0.0812659  0.0274839  -2.957 0.003108 ** 
## p6f10        0.2380860  0.1023862   2.325 0.020052 *  
## p6j18       -0.1716464  1.5145290  -0.113 0.909766    
## p6j19        0.0859274  0.0856381   1.003 0.315678    
## p6j20        0.0052994  0.2397307   0.022 0.982364    
## p6j21        2.3890647  1.3504761   1.769 0.076885 .  
## p6j22        0.0928322  0.0967960   0.959 0.337534    
## p6j23        0.3948067  0.2072518   1.905 0.056785 .  
## p6j26       -0.1036160  0.3095420  -0.335 0.737822    
## p6j29        0.0398093  0.0288447   1.380 0.167549    
## p6j32       -0.0291015  0.4199321  -0.069 0.944750    
## p6j48        0.0183444  0.3996101   0.046 0.963385    
## p6j49       -0.0601015  0.4248156  -0.141 0.887493    
## p6j50        0.2754051  0.3312998   0.831 0.405812    
## p6j51       -0.0506869  0.3747205  -0.135 0.892402    
## p6j52       -0.0130880  0.3244407  -0.040 0.967822    
## p6j53        0.0455324  0.3000749   0.152 0.879395    
## p6j54        0.0601113  0.2954242   0.203 0.838764    
## p6j55        0.4440053  0.3095066   1.435 0.151413    
## p6j56        0.5191372  0.4690572   1.107 0.268394    
## p6j57       -0.5703141  0.4684040  -1.218 0.223388    
## p6j58       -0.5096514  0.3227979  -1.579 0.114369    
```

```
## p6k3_8        0.0155093  0.0298358   0.520 0.603188
## p6k5         -0.9052249  0.2802980  -3.230 0.001240 **
## p6k54         0.3698972  0.1490267   2.482 0.013062 *
## p6k59        -0.3389772  0.1687320  -2.009 0.044540 *
## p6k60         3.0186372  1.6839210   1.793 0.073033 .
## p6k61         0.0001958  0.0001293   1.514 0.130037
## p6k62         0.4035617  0.2710308   1.489 0.136490
## k6e35k       -0.0309731  0.0262160  -1.181 0.237421
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1965.6  on 1640  degrees of freedom
## Residual deviance: 1189.6  on 1579  degrees of freedom
## AIC: 1313.6
##
## Number of Fisher Scoring iterations: 7
```

```r
#Predictions and Model Eval
predictions_logit <- predict(model_logit, newdata = data_test_filtered, type = "prob")
```

```r
# Extract predicted probabilities for the positive class
predicted_probs <- predictions_logit[, 2]

# Calculate predictions (convert probabilities to binary predictions)
predicted_class <- ifelse(predicted_probs > 0.5, 1, 0)

# Calculate accuracy
accuracy <- mean(predicted_class == data_test_filtered$binary_poverty_category)
cat("Accuracy:", accuracy, "\n")
```

```
## Accuracy: 0.8199513
```

```r
# Calculate confusion matrix
confusion <- table(predicted_class, data_test_filtered$binary_poverty_category)

# Calculate precision, recall, specificity, and F1 score
TP <- confusion[2, 2]
FP <- confusion[1, 2]
TN <- confusion[1, 1]
FN <- confusion[2, 1]

precision <- TP / (TP + FP)
recall <- TP / (TP + FN)
specificity <- TN / (TN + FP)
f1_score <- 2 * (precision * recall) / (precision + recall)

cat("Precision:", precision, "\n")
```

```
## Precision: 0.8865979
```

```r
cat("Recall:", recall, "\n")
```

```
## Recall: 0.8628763
```

```r
cat("Specificity:", specificity, "\n")
```

```
## Specificity: 0.7053571
```

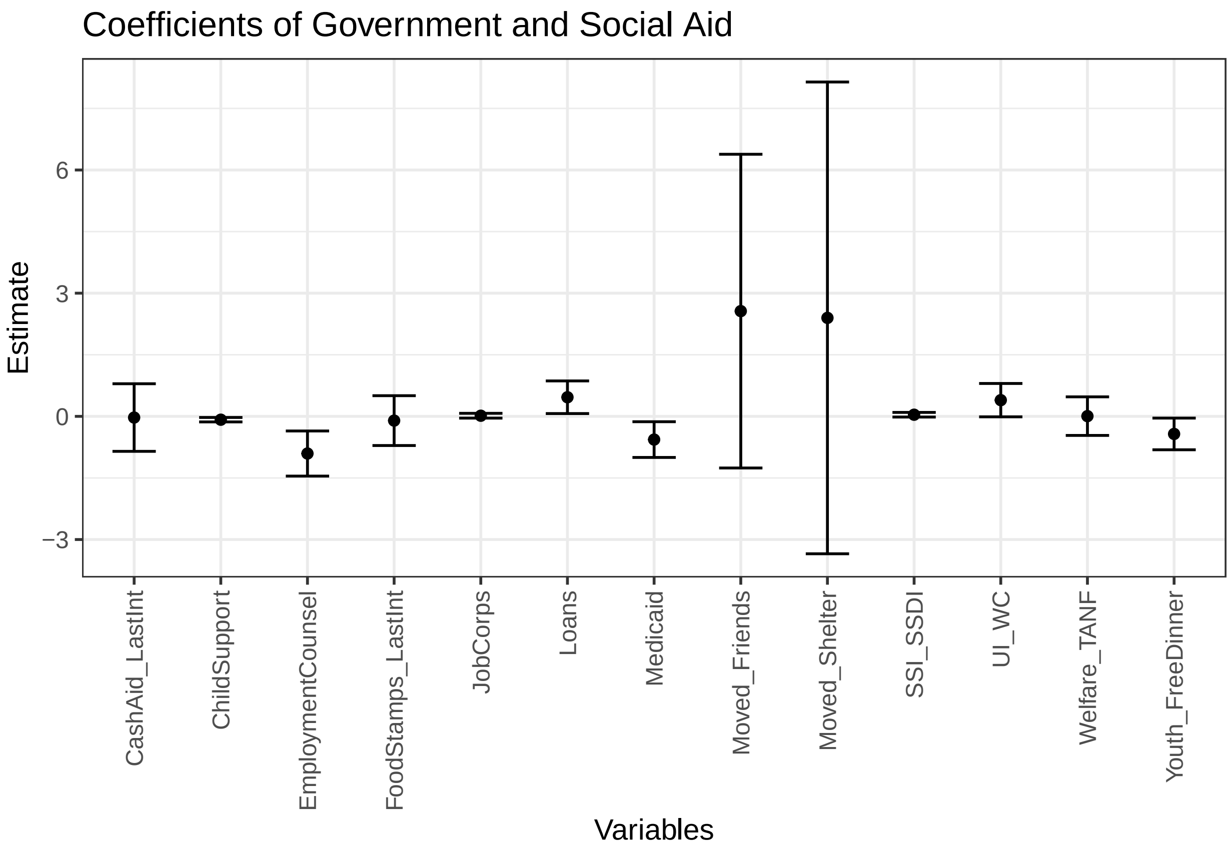
```r
cat("F1 Score:", f1_score, "\n")
```

```
## F1 Score: 0.8745763
```

```r
print(confusion)
```

```
##
## predicted_class   0   1
##               0  79  33
##               1  41 258
```

```r
gov_social_aid_vars <- c('p6j36','p6j44','p6j45','p6k70','p6b31','p6f9','p6j20','p6j23','p6j26','p6j29'

gov_social_aid_names <- c('Youth_FreeDinner','Moved_Friends','Moved_Shelter','Loans','Medicaid','ChildS


# Extract coefficients and standard errors for the variables of interest
coefficients <- summary(model_logit)$coef[, 1]
se <- summary(model_logit)$coef[, 2]

# Create a data frame
df <- data.frame(
  Variables = gov_social_aid_names,
  Estimate = coefficients[gov_social_aid_vars],
  se = se[gov_social_aid_vars]
)

# Calculate confidence intervals
df$lower <- df$Estimate - 1.96 * df$se
df$upper <- df$Estimate + 1.96 * df$se

# Plot
ggplot(df, aes(x = Variables, y = Estimate)) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.5) +
  geom_point() +
  theme_bw() +
  labs(title = "Coefficients of Government and Social Aid", y = "Estimate")+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



Coefficients of Government and Social Aid
Estimate
6
3
0
−3
CashAid_LastInt
ChildSupport
EmploymentCounsel
FoodStamps_LastInt
JobCorps
Loans
Medicaid
Moved_Friends
Moved_Shelter
SSI_SSDI
UI_WC
Welfare_TANF
Youth_FreeDinner
Variables