DATA 621: BUSINESS ANALYTICS AND DATA MINING HOMEWORK#3: LOGISTIC REGRESSION

Group 2 - Gabriel Campos, Melissa Bowman, Alexander Khaykin, & Jennifer Abinette

Last edited November 12, 2023

Contents

Overview

In this homework assignment, you will explore, analyze and model a data set containing information on crime for various neighborhoods of a major city. Each record has a response variable indicating whether or not the crime rate is above the median crime rate (1) or not (0).

Your objective is to build a *binary logistic regression model* on the training data set to predict whether the neighborhood will be at risk for high crime levels. You will provide classifications and probabilities for the evaluation data set using your binary logistic regression model. You can only use the variables given to you (or, variables that you derive from the variables provided). Below is a short description of the variables of interest in the data set:

- zn: proportion of residential land zoned for large lots (over 25000 square feet) (predictor variable)
- indus: proportion of non-retail business acres per suburb (predictor variable)
- chas: a dummy var. for whether the suburb borders the Charles River (1) or not (0) (predictor variable)
- nox: nitrogen oxides concentration (parts per 10 million) (predictor variable)
- rm: average number of rooms per dwelling (predictor variable)
- age: proportion of owner-occupied units built prior to 1940 (predictor variable)
- dis: weighted mean of distances to five Boston employment centers (predictor variable)
- rad: index of accessibility to radial highways (predictor variable)
- tax: full-value property-tax rate per $10,000 (predictor variable)
- ptratio: pupil-teacher ratio by town (predictor variable)
- lstat: lower status of the population (percent) (predictor variable)
- medv: median value of owner-occupied homes in $1000s (predictor variable)
- **target: whether the crime rate is above the median crime rate (1) or not (0) (response variable)**

Deliverables:

- A write-up submitted in PDF format. Your write-up should have four sections. Each one is described below. You may assume you are addressing me as a fellow data scientist, so do not need to shy away from technical details.
- Assigned prediction (probabilities, classifications) for the evaluation data set. Use 0.5 threshold. Include your R statistical programming code in an Appendix.

Write Up:

1. DATA EXPLORATION (25 Points) Describe the size and the variables in the crime training data set. Consider that too much detail will cause a manager to lose interest while too little detail will make the manager consider that you aren't doing your job. Some suggestions are given below. Please do NOT treat this as a check list of things to do to complete the assignment. You should have your own thoughts on what to tell the boss. These are just ideas. a. Mean / Standard Deviation / Median b. Bar Chart or Box Plot of the data c. Is the data correlated to the target variable (or to other variables?) d. Are any of the variables missing and need to be imputed/"fixed"?

2. DATA PREPARATION (25 Points) Describe how you have transformed the data by changing the original variables or creating new variables. If you did transform the data or create new variables, discuss why you did this. Here are some possible transformations. a. Fix missing values (maybe with a Mean or Median value) b. Create flags to suggest if a variable was missing c. Transform data by putting it into buckets d. Mathematical transforms such as log or square root (or, use Box-Cox) e. Combine variables (such as ratios or adding or multiplying) to create new variables

3. BUILD MODELS (25 Points) Using the training data, build at least three different binary logistic regression models, using different variables (or the same variables with different transformations). You may select the variables manually, use an approach such as Forward or Stepwise, use a different approach, or use a combination of techniques. Describe the techniques you used. If you manually selected a variable for inclusion into the model or exclusion into the model, indicate why this was done. Be sure to explain how you can make inferences from the model, as well as discuss other relevant model output. Discuss the coefficients in the models, do they make sense? Are you keeping the model even though it is counter intuitive? Why? The boss needs to know.

4. SELECT MODELS (25 Points) Decide on the criteria for selecting the best binary logistic regression model. Will you select models with slightly worse performance if it makes more sense or is more parsimonious? Discuss why you selected your model. * For the binary logistic regression model, will you use a metric such as log likelihood, AIC, ROC curve, etc.? Using the training data set, evaluate the binary logistic regression model based on (a) accuracy, (b) classification error rate, (c) precision, (d) sensitivity, (e) specificity, (f) F1 score, (g) AUC, and (h) confusion matrix. Make predictions using the evaluation data set

1 DATA EXPLORATION

1.1 Load the data

```
url_git<-
  "https://raw.githubusercontent.com/GitableGabe/Data621_Data/main/"
```

```
df_crime_eval <-
  read.csv(paste0(url_git,"crime-evaluation-data_modified.csv"))
head(df_crime_eval,n=10)
```

```
##    zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv
## 1   0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8  4.03 34.7
## 2   0  8.14    0 0.538 6.096 84.5 4.4619   4 307    21.0 10.26 18.2
## 3   0  8.14    0 0.538 6.495 94.4 4.4547   4 307    21.0 12.80 18.4
## 4   0  8.14    0 0.538 5.950 82.0 3.9900   4 307    21.0 27.71 13.2
## 5   0  5.96    0 0.499 5.850 41.5 3.9342   5 279    19.2  8.77 21.0
## 6  25  5.13    0 0.453 5.741 66.2 7.2254   8 284    19.7 13.15 18.7
## 7  25  5.13    0 0.453 5.966 93.4 6.8185   8 284    19.7 14.44 16.0
## 8   0  4.49    0 0.449 6.630 56.1 4.4377   3 247    18.5  6.53 26.6
## 9   0  4.49    0 0.449 6.121 56.8 3.7476   3 247    18.5  8.44 22.2
## 10  0  2.89    0 0.445 6.163 69.6 3.4952   2 276    18.0 11.34 21.4
```

```
df_crime_eval
```

```
##    zn indus chas   nox    rm   age    dis rad tax ptratio lstat medv
## 1   0  7.07    0 0.469 7.185  61.1 4.9671   2 242    17.8  4.03 34.7
## 2   0  8.14    0 0.538 6.096  84.5 4.4619   4 307    21.0 10.26 18.2
## 3   0  8.14    0 0.538 6.495  94.4 4.4547   4 307    21.0 12.80 18.4
## 4   0  8.14    0 0.538 5.950  82.0 3.9900   4 307    21.0 27.71 13.2
## 5   0  5.96    0 0.499 5.850  41.5 3.9342   5 279    19.2  8.77 21.0
## 6  25  5.13    0 0.453 5.741  66.2 7.2254   8 284    19.7 13.15 18.7
## 7  25  5.13    0 0.453 5.966  93.4 6.8185   8 284    19.7 14.44 16.0
## 8   0  4.49    0 0.449 6.630  56.1 4.4377   3 247    18.5  6.53 26.6
## 9   0  4.49    0 0.449 6.121  56.8 3.7476   3 247    18.5  8.44 22.2
## 10  0  2.89    0 0.445 6.163  69.6 3.4952   2 276    18.0 11.34 21.4
## 11  0 25.65    0 0.581 5.856  97.0 1.9444   2 188    19.1 25.41 17.3
## 12  0 25.65    0 0.581 5.613  95.6 1.7572   2 188    19.1 27.26 15.7
## 13  0 21.89    0 0.624 5.637  94.7 1.9799   4 437    21.2 18.34 14.3
## 14  0 19.58    0 0.605 6.101  93.0 2.2834   5 403    14.7  9.81 25.0
## 15  0 19.58    0 0.605 5.880  97.3 2.3887   5 403    14.7 12.03 19.1
## 16  0 10.59    1 0.489 5.960  92.1 3.8771   4 277    18.6 17.27 21.7
## 17  0  6.20    0 0.504 6.552  21.4 3.3751   8 307    17.4  3.76 31.5
## 18  0  6.20    0 0.507 8.247  70.4 3.6519   8 307    17.4  3.95 48.3
## 19 22  5.86    0 0.431 6.957   6.8 8.9067   7 330    19.1  3.53 29.6
## 20 90  2.97    0 0.400 7.088  20.8 7.3073   1 285    15.3  7.85 32.2
## 21 80  1.76    0 0.385 6.230  31.5 9.0892   1 241    18.2 12.93 20.1
## 22 33  2.18    0 0.472 6.616  58.1 3.3700   7 222    18.4  8.93 28.4
## 23  0  9.90    0 0.544 6.122  52.8 2.6403   4 304    18.4  5.98 22.1
## 24  0  7.38    0 0.493 6.415  40.1 4.7211   5 287    19.6  6.12 25.0
## 25  0  7.38    0 0.493 6.312  28.9 5.4159   5 287    19.6  6.15 23.0
```

```
## 26  0  5.19    0 0.515 5.895  59.6 5.6150   5 224    20.2 10.56 18.5
## 27 80  2.01    0 0.435 6.635  29.7 8.3440   4 280    17.0  5.99 24.5
## 28  0 18.10    0 0.718 3.561  87.9 1.6132  24 666    20.2  7.12 27.5
## 29  0 18.10    1 0.631 7.016  97.5 1.2024  24 666    20.2  2.96 50.0
## 30  0 18.10    0 0.584 6.348  86.1 2.0527  24 666    20.2 17.64 14.5
## 31  0 18.10    0 0.740 5.935  87.9 1.8206  24 666    20.2 34.02  8.4
## 32  0 18.10    0 0.740 5.627  93.9 1.8172  24 666    20.2 22.88 12.8
## 33  0 18.10    0 0.740 5.818  92.4 1.8662  24 666    20.2 22.11 10.5
## 34  0 18.10    0 0.740 6.219 100.0 2.0048  24 666    20.2 16.59 18.4
## 35  0 18.10    0 0.740 5.854  96.6 1.8956  24 666    20.2 23.79 10.8
## 36  0 18.10    0 0.713 6.525  86.5 2.4358  24 666    20.2 18.13 14.1
## 37  0 18.10    0 0.713 6.376  88.4 2.5671  24 666    20.2 14.65 17.7
## 38  0 18.10    0 0.655 6.209  65.4 2.9634  24 666    20.2 13.22 21.4
## 39  0  9.69    0 0.585 5.794  70.6 2.8927   6 391    19.2 14.10 18.3
## 40  0 11.93    0 0.573 6.976  91.0 2.1675   1 273    21.0  5.64 23.9
```

```r
df_crime_train <-
  read.csv(paste0(url_git,"crime-training-data_modified.csv"))
head(df_crime_train,n=10)
```

```
##    zn indus chas   nox    rm   age    dis rad tax ptratio lstat medv target
## 1   0 19.58    0 0.605 7.929  96.2 2.0459   5 403    14.7  3.70 50.0      1
## 2   0 19.58    1 0.871 5.403 100.0 1.3216   5 403    14.7 26.82 13.4      1
## 3   0 18.10    0 0.740 6.485 100.0 1.9784  24 666    20.2 18.85 15.4      1
## 4  30  4.93    0 0.428 6.393   7.8 7.0355   6 300    16.6  5.19 23.7      0
## 5   0  2.46    0 0.488 7.155  92.2 2.7006   3 193    17.8  4.82 37.9      0
## 6   0  8.56    0 0.520 6.781  71.3 2.8561   5 384    20.9  7.67 26.5      0
## 7   0 18.10    0 0.693 5.453 100.0 1.4896  24 666    20.2 30.59  5.0      1
## 8   0 18.10    0 0.693 4.519 100.0 1.6582  24 666    20.2 36.98  7.0      1
## 9   0  5.19    0 0.515 6.316  38.1 6.4584   5 224    20.2  5.68 22.2      0
## 10 80  3.64    0 0.392 5.876  19.1 9.2203   1 315    16.4  9.25 20.9      0
```

```r
df_crime_eval[is.na(df_crime_eval)]
```

```
## numeric(0)
```

```r
df_crime_train[is.na(df_crime_train)]
```

```
## numeric(0)
```

1.1.1 Data Summary

```r
summary(df_crime_train)
```

```
##        zn             indus             chas              nox        
##  Min.   :  0.00   Min.   : 0.460   Min.   :0.00000   Min.   :0.3890  
##  1st Qu.:  0.00   1st Qu.: 5.145   1st Qu.:0.00000   1st Qu.:0.4480  
##  Median :  0.00   Median : 9.690   Median :0.00000   Median :0.5380  
##  Mean   : 11.58   Mean   :11.105   Mean   :0.07082   Mean   :0.5543  
```

```
##  3rd Qu.: 16.25   3rd Qu.:18.100   3rd Qu.:0.00000   3rd Qu.:0.6240
##  Max.   :100.00   Max.   :27.740   Max.   :1.00000   Max.   :0.8710
##        rm             age              dis              rad
##  Min.   :3.863   Min.   :  2.90   Min.   : 1.130   Min.   : 1.00
##  1st Qu.:5.887   1st Qu.: 43.88   1st Qu.: 2.101   1st Qu.: 4.00
##  Median :6.210   Median : 77.15   Median : 3.191   Median : 5.00
##  Mean   :6.291   Mean   : 68.37   Mean   : 3.796   Mean   : 9.53
##  3rd Qu.:6.630   3rd Qu.: 94.10   3rd Qu.: 5.215   3rd Qu.:24.00
##  Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.00
##       tax           ptratio         lstat             medv
##  Min.   :187.0   Min.   :12.6   Min.   : 1.730   Min.   : 5.00
##  1st Qu.:281.0   1st Qu.:16.9   1st Qu.: 7.043   1st Qu.:17.02
##  Median :334.5   Median :18.9   Median :11.350   Median :21.20
##  Mean   :409.5   Mean   :18.4   Mean   :12.631   Mean   :22.59
##  3rd Qu.:666.0   3rd Qu.:20.2   3rd Qu.:16.930   3rd Qu.:25.00
##  Max.   :711.0   Max.   :22.0   Max.   :37.970   Max.   :50.00
##      target
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.4914
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

Upon a comprehensive examination of the dataset, it is noteworthy that there are no missing values, underscoring the completeness of the provided data. This absence of missing values is a positive indicator, as it eliminates the need for imputation or data filling techniques that might have otherwise been necessary.

Also, the examining the means and medians of the variables aids in understanding the distribution's symmetry, identifying outliers, assessing data consistency, and simplifying the interpretation of central tendency measures through alignment between the mean and median.

1.1.2 Correlation Matrix*

```r
# Create a correlation matrix for all variables
(matrix_cor <- cor(df_crime_train))
```

```
##                  zn       indus        chas         nox          rm         age
## zn       1.00000000 -0.53826643 -0.04016203 -0.51704518  0.31981410 -0.57258054
## indus   -0.53826643  1.00000000  0.06118317  0.75963008 -0.39271181  0.63958182
## chas    -0.04016203  0.06118317  1.00000000  0.09745577  0.09050979  0.07888366
## nox     -0.51704518  0.75963008  0.09745577  1.00000000 -0.29548972  0.73512782
## rm       0.31981410 -0.39271181  0.09050979 -0.29548972  1.00000000 -0.23281251
## age     -0.57258054  0.63958182  0.07888366  0.73512782 -0.23281251  1.00000000
## dis      0.66012434 -0.70361886 -0.09657711 -0.76888404  0.19901584 -0.75089759
## rad     -0.31548119  0.60062839 -0.01590037  0.59582984 -0.20844570  0.46031430
## tax     -0.31928408  0.73222922 -0.04676476  0.65387804 -0.29693430  0.51212452
## ptratio -0.39103573  0.39468980 -0.12866058  0.17626871 -0.36034706  0.25544785
## lstat   -0.43299252  0.60711023 -0.05142322  0.59624264 -0.63202445  0.60562001
## medv     0.37671713 -0.49617432  0.16156528 -0.43012267  0.70533679 -0.37815605
## target  -0.43168176  0.60485074  0.08004187  0.72610622 -0.15255334  0.63010625
##                 dis         rad         tax     ptratio       lstat        medv
```

```
## zn       0.66012434 -0.31548119 -0.31928408 -0.3910357 -0.43299252  0.3767171
## indus   -0.70361886  0.60062839  0.73222922  0.3946898  0.60711023 -0.4961743
## chas    -0.09657711 -0.01590037 -0.04676476 -0.1286606 -0.05142322  0.1615653
## nox     -0.76888404  0.59582984  0.65387804  0.1762687  0.59624264 -0.4301227
## rm       0.19901584 -0.20844570 -0.29693430 -0.3603471 -0.63202445  0.7053368
## age     -0.75089759  0.46031430  0.51212452  0.2554479  0.60562001 -0.3781560
## dis      1.00000000 -0.49499193 -0.53425464 -0.2333394 -0.50752800  0.2566948
## rad     -0.49499193  1.00000000  0.90646323  0.4714516  0.50310125 -0.3976683
## tax     -0.53425464  0.90646323  1.00000000  0.4744223  0.56418864 -0.4900329
## ptratio -0.23333940  0.47145160  0.47442229  1.0000000  0.37735605 -0.5159153
## lstat   -0.50752800  0.50310125  0.56418864  0.3773560  1.00000000 -0.7358008
## medv     0.25669476 -0.39766826 -0.49003287 -0.5159153 -0.73580078  1.0000000
## target  -0.61867312  0.62810492  0.61111331  0.2508489  0.46912702 -0.2705507
##              target
## zn      -0.43168176
## indus    0.60485074
## chas     0.08004187
## nox      0.72610622
## rm      -0.15255334
## age      0.63010625
## dis     -0.61867312
## rad      0.62810492
## tax      0.61111331
## ptratio  0.25084892
## lstat    0.46912702
## medv    -0.27055071
## target   1.00000000
```

Taking a glance at the correlation matrix of 'df_crime_train,' it is evident that the highly correlated variables are 'rad' and 'tax,' exhibiting a strong correlation of 91%. Considering this substantial correlation, there may be a need to explore the possibility of combining these variables.

Additionally, when assessing the correlation of variables to the target, the most correlated variables to the least correlated variables are as follows: nox (72%), age (63%), rad (62%), dis (62%), tax (61%), indus (60%), lstat (47%), zn (43%), medv (27%), ptratio (25%), rm (15%), and chas (8%).

2 DATA PREPARATION

2.1 Model 1

Examining the models without transforming or prepping the data provides us with a baseline to assess whether transforming or prepping the data will enhance the model.

```r
model_1 <- glm(formula = target ~ ., family = binomial, data = df_crime_train)

summary(model_1)
```

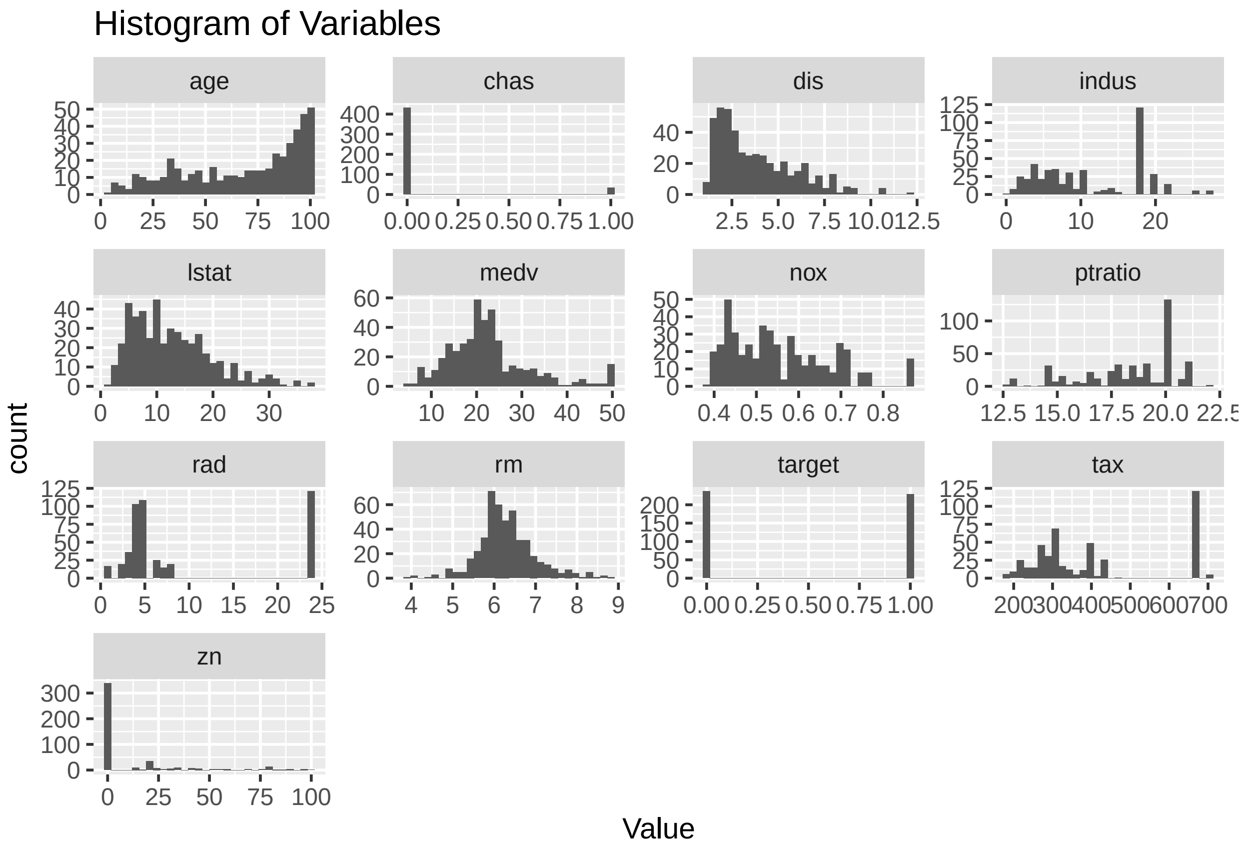
```
##
## Call:
## glm(formula = target ~ ., family = binomial, data = df_crime_train)
##
## Coefficients:
```

```
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept) -40.822934   6.632913  -6.155 7.53e-10 ***
## zn           -0.065946   0.034656  -1.903  0.05706 .
## indus        -0.064614   0.047622  -1.357  0.17485
## chas          0.910765   0.755546   1.205  0.22803
## nox          49.122297   7.931706   6.193 5.90e-10 ***
## rm           -0.587488   0.722847  -0.813  0.41637
## age           0.034189   0.013814   2.475  0.01333 *
## dis           0.738660   0.230275   3.208  0.00134 **
## rad           0.666366   0.163152   4.084 4.42e-05 ***
## tax          -0.006171   0.002955  -2.089  0.03674 *
## ptratio       0.402566   0.126627   3.179  0.00148 **
## lstat         0.045869   0.054049   0.849  0.39608
## medv          0.180824   0.068294   2.648  0.00810 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 192.05  on 453  degrees of freedom
## AIC: 218.05
##
## Number of Fisher Scoring iterations: 9
```

In the data preparation phase, we assessed the distribution of variables to determine whether they demonstrated a normalized distribution. Ensuring a normal distribution for variables in a regression model holds paramount importance for various reasons. Firstly, adherence to the assumption of normality is crucial because many statistical techniques, including those employed in regression analysis, rely on this assumption for their validity. Secondly, achieving normality in variables contributes to more accurate and efficient parameter estimates, enhancing the overall performance of the model. Furthermore, statistical inferences, such as confidence intervals and hypothesis tests, are based on normality assumptions, emphasizing the necessity of a normal distribution. Normality is also pivotal in the analysis of residuals, as normally distributed residuals signify a well-fitted model. The robustness of statistical methods is bolstered when data approximates a normal distribution, making the results more dependable and less sensitive to outliers. Lastly, normality simplifies the interpretability of coefficients, facilitating a clearer understanding of the impact of predictors on the outcome.

```
# Gather the data into a long format
df_long <- gather(df_crime_train, key = "Variable", value = "Value")

ggplot(df_long, aes(x = Value)) +
  geom_histogram() +
  facet_wrap(~Variable, scales = "free") +
  labs(title = "Histogram of Variables")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



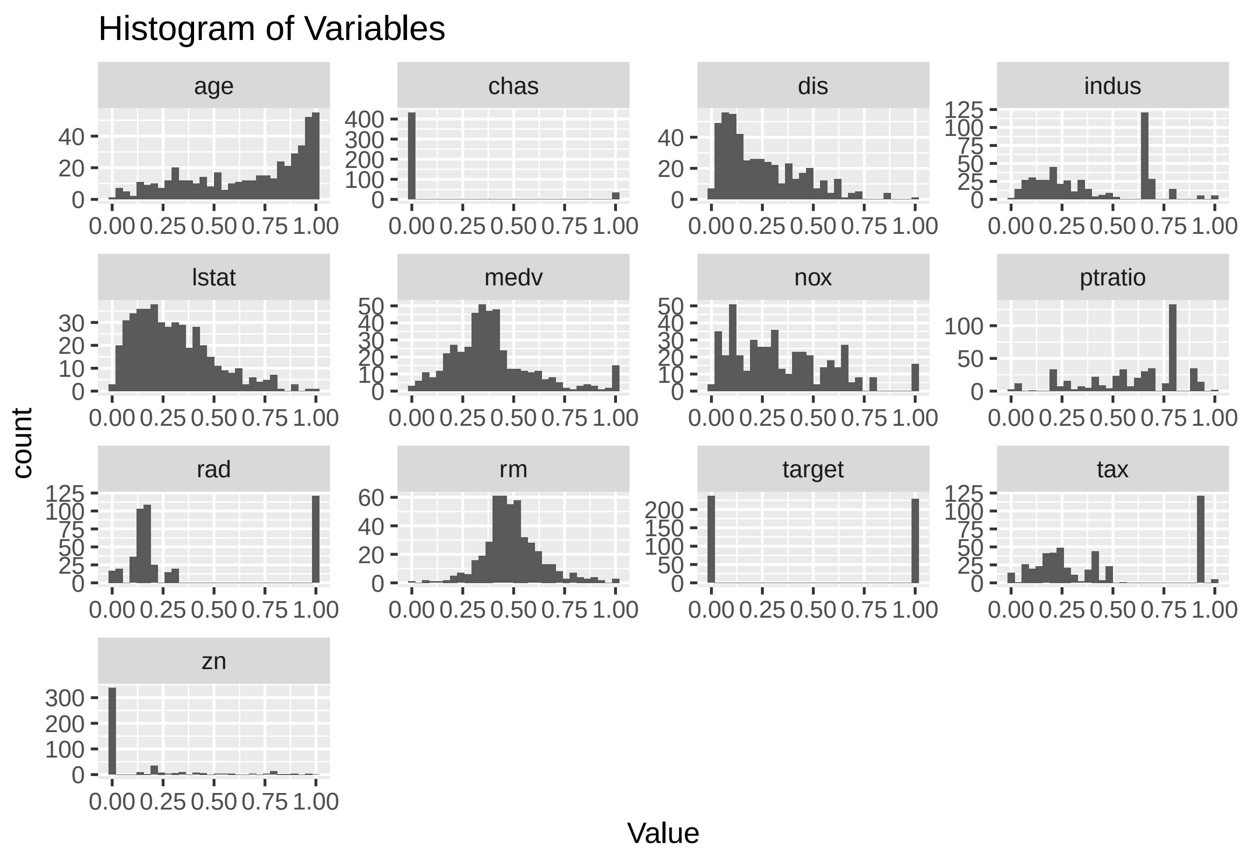
2.1.1 Scale

The variables currently lack normalization, and it is imperative to address this issue. To initiate the correction process, our first step involves applying normalization to the variables by scaling them. This entails transforming the variables to a standardized scale. Normalizing the scale of variables is particularly crucial in logistic regression. In logistic regression, the scale of the predictor variables influences the parameter estimates, and having variables on different scales might lead to uneven contributions to the model. Normalizing the scale helps ensure that each variable contributes proportionally to the logistic regression model, thereby improving the stability and interpretability of the model. This preliminary normalization step will allow us to assess whether achieving a standardized scale enhances the model's performance before further adjustments are made.

```
# Apply min-max scaling to all three variables
df_scaled <- df_crime_train
df_scaled[] <- lapply(df_crime_train, rescale)
```

```
# Gather the data into a long format
df_long_scaled <- gather(df_scaled, key = "Variable", value = "Value")

ggplot(df_long_scaled, aes(x = Value)) +
  geom_histogram() +
  facet_wrap(~Variable, scales = "free") +
  labs(title = "Histogram of Variables")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Checking correlation of scaled varibles

```r
# Create a correlation matrix for all variables
(matrix_cor <- cor(df_scaled))
```

```
##                  zn       indus        chas         nox          rm         age
## zn       1.00000000 -0.53826643 -0.04016203 -0.51704518  0.31981410 -0.57258054
## indus   -0.53826643  1.00000000  0.06118317  0.75963008 -0.39271181  0.63958182
## chas    -0.04016203  0.06118317  1.00000000  0.09745577  0.09050979  0.07888366
## nox     -0.51704518  0.75963008  0.09745577  1.00000000 -0.29548972  0.73512782
## rm       0.31981410 -0.39271181  0.09050979 -0.29548972  1.00000000 -0.23281251
## age     -0.57258054  0.63958182  0.07888366  0.73512782 -0.23281251  1.00000000
## dis      0.66012434 -0.70361886 -0.09657711 -0.76888404  0.19901584 -0.75089759
## rad     -0.31548119  0.60062839 -0.01590037  0.59582984 -0.20844570  0.46031430
## tax     -0.31928408  0.73222922 -0.04676476  0.65387804 -0.29693430  0.51212452
## ptratio -0.39103573  0.39468980 -0.12866058  0.17626871 -0.36034706  0.25544785
## lstat   -0.43299252  0.60711023 -0.05142322  0.59624264 -0.63202445  0.60562001
## medv     0.37671713 -0.49617432  0.16156528 -0.43012267  0.70533679 -0.37815605
## target  -0.43168176  0.60485074  0.08004187  0.72610622 -0.15255334  0.63010625
##                 dis         rad         tax    ptratio       lstat       medv
## zn       0.66012434 -0.31548119 -0.31928408 -0.3910357 -0.43299252  0.3767171
## indus   -0.70361886  0.60062839  0.73222922  0.3946898  0.60711023 -0.4961743
## chas    -0.09657711 -0.01590037 -0.04676476 -0.1286606 -0.05142322  0.1615653
## nox     -0.76888404  0.59582984  0.65387804  0.1762687  0.59624264 -0.4301227
## rm       0.19901584 -0.20844570 -0.29693430 -0.3603471 -0.63202445  0.7053368
## age     -0.75089759  0.46031430  0.51212452  0.2554479  0.60562001 -0.3781560
```

```
## dis      1.00000000 -0.49499193 -0.53425464 -0.2333394 -0.50752800  0.2566948
## rad     -0.49499193  1.00000000  0.90646323  0.4714516  0.50310125 -0.3976683
## tax     -0.53425464  0.90646323  1.00000000  0.4744223  0.56418864 -0.4900329
## ptratio -0.23333940  0.47145160  0.47442229  1.0000000  0.37735605 -0.5159153
## lstat   -0.50752800  0.50310125  0.56418864  0.3773560  1.00000000 -0.7358008
## medv     0.25669476 -0.39766826 -0.49003287 -0.5159153 -0.73580078  1.0000000
## target  -0.61867312  0.62810492  0.61111331  0.2508489  0.46912702 -0.2705507
##             target
## zn      -0.43168176
## indus    0.60485074
## chas     0.08004187
## nox      0.72610622
## rm      -0.15255334
## age      0.63010625
## dis     -0.61867312
## rad      0.62810492
## tax      0.61111331
## ptratio  0.25084892
## lstat    0.46912702
## medv    -0.27055071
## target   1.00000000
```

2.2 Model 2

```
model_2 <- glm(formula = target ~ ., family = binomial, data = df_scaled)

(summary(model_2))
```

```
##
## Call:
## glm(formula = target ~ ., family = binomial, data = df_scaled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -17.5119     2.8741  -6.093 1.11e-09 ***
## zn           -6.5946     3.4656  -1.903  0.05706 .
## indus        -1.7627     1.2991  -1.357  0.17485
## chas          0.9108     0.7555   1.205  0.22803
## nox          23.6769     3.8231   6.193 5.90e-10 ***
## rm           -2.8887     3.5542  -0.813  0.41637
## age           3.3197     1.3413   2.475  0.01333 *
## dis           8.1230     2.5323   3.208  0.00134 **
## rad          15.3264     3.7525   4.084 4.42e-05 ***
## tax          -3.2338     1.5483  -2.089  0.03674 *
## ptratio       3.7841     1.1903   3.179  0.00148 **
## lstat         1.6623     1.9587   0.849  0.39608
## medv          8.1371     3.0732   2.648  0.00810 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 192.05  on 453  degrees of freedom
## AIC: 218.05
##
## Number of Fisher Scoring iterations: 9
```

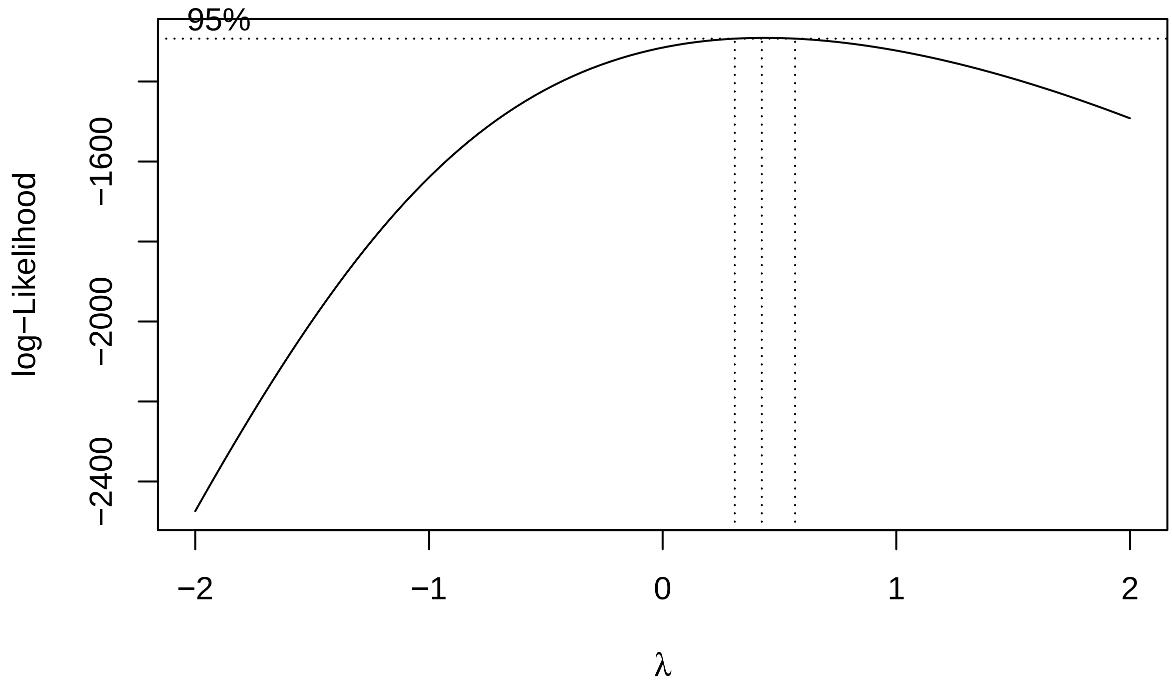
Scaling the variables did not yield improvements in the model performance. However, we will explore the potential benefits of applying the Box-Cox transformation to achieve normality in the variable distributions. The Box-Cox transformation is a statistical technique that aims to stabilize the variance and make the data more closely approximate a normal distribution. Specifically, it involves raising each data point to a power, with the power determined during the transformation process. The goal is to identify the power that maximizes the normality of the data. By doing so, Box-Cox can address issues such as skewed distributions and unequal variances, making the variables more amenable to statistical methods that assume normality. Implementing the Box-Cox transformation serves as a valuable step in preparing the variables for logistic regression, potentially enhancing the model's performance by aligning with the underlying assumptions of the chosen statistical approach.

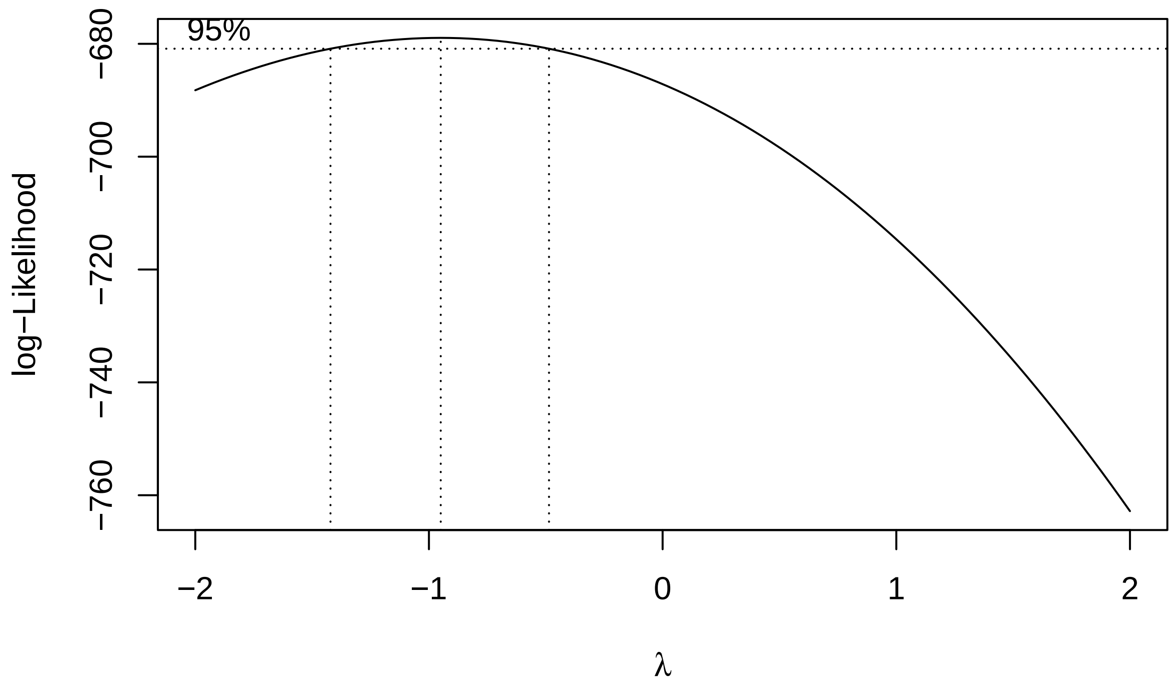
2.2.1 Box-Cox

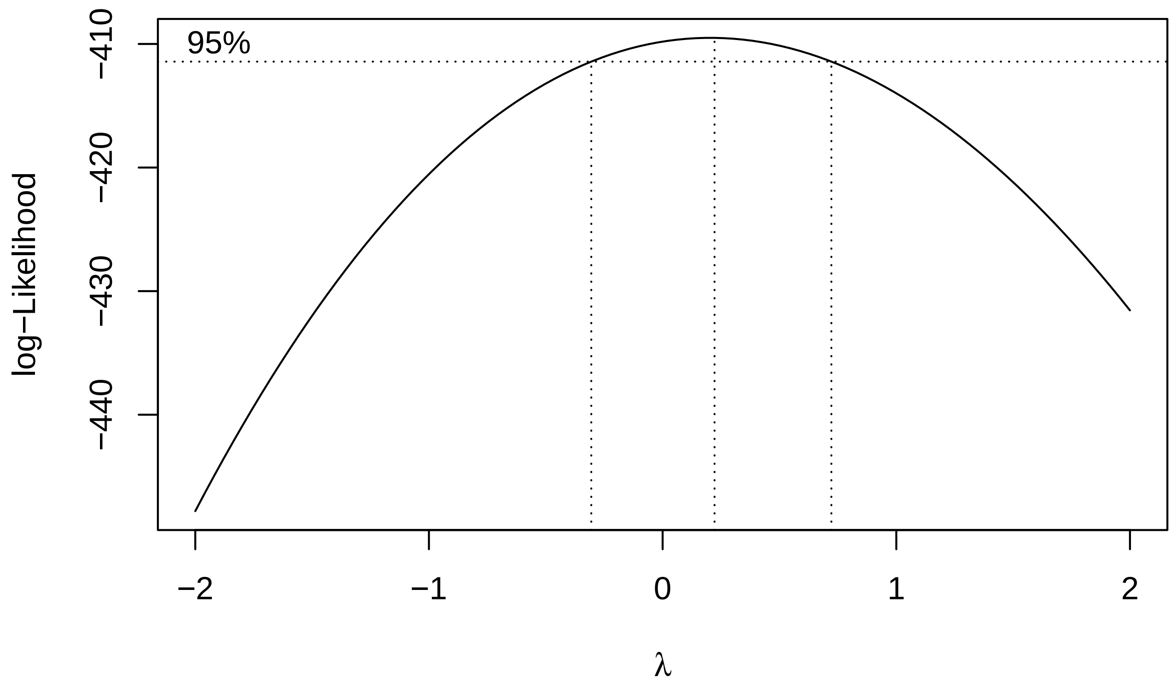
```r
df_crime_train$age <- as.numeric(df_crime_train$age)
```

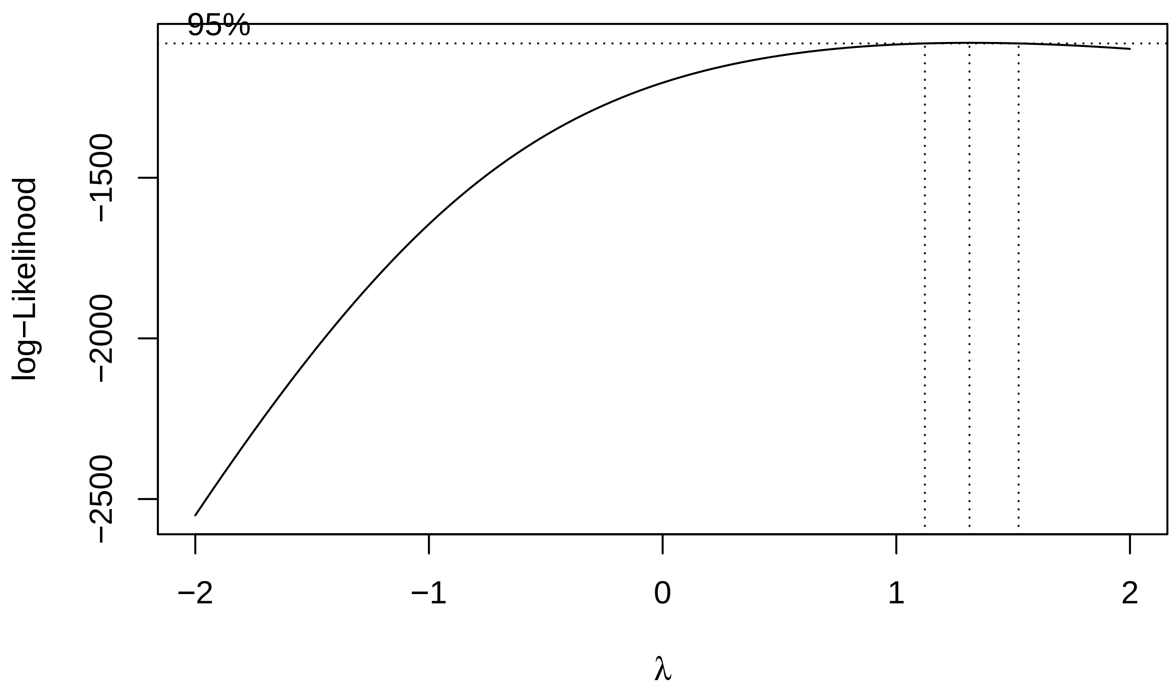
2.2.2 Transform 'df__crime__train'

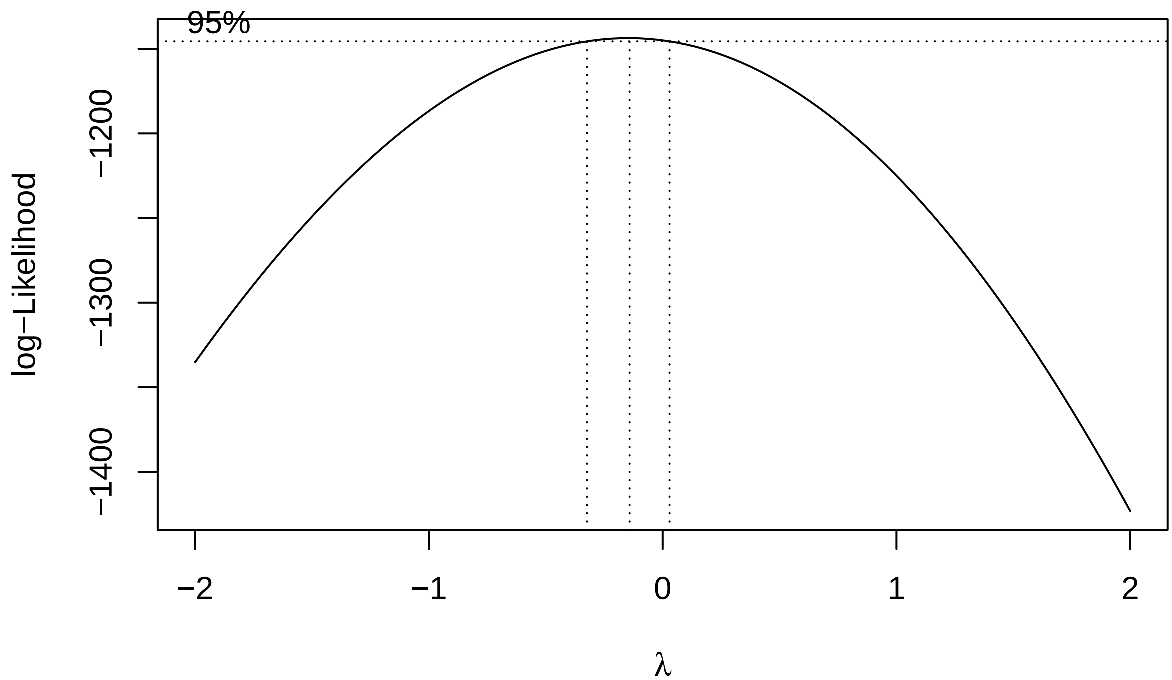
```r
# Create an empty list to store the transformed columns
col_transformed <- list()

# Define the names of columns to exclude from transformation because there variables response must be p
col_exclude <- c("target", "zn", "chas")

# Iterate through the columns in df_crime_train
for (col_name in names(df_crime_train)) {
  # Convert the column to a list and check if it's numeric and not in the exclude list
  if (is.numeric(df_crime_train[[col_name]]) && !(col_name %in% col_exclude)) {
    col_list <- as.numeric(as.list(df_crime_train[[col_name]]))

    # Find optimal lambda for Box-Cox transformation
    bc <- boxcox(col_list ~ 1, lambda = seq(-2, 2, 0.1))
    lambda_col <- bc$x[which.max(bc$y)]

    # Apply the Box-Cox transformation
    col_new <- ifelse(col_list==0, log(col_list), (col_list^lambda_col - 1) / lambda_col)

    # Store the transformed column in the list
    col_transformed[[col_name]] <- col_new
  }
}
```

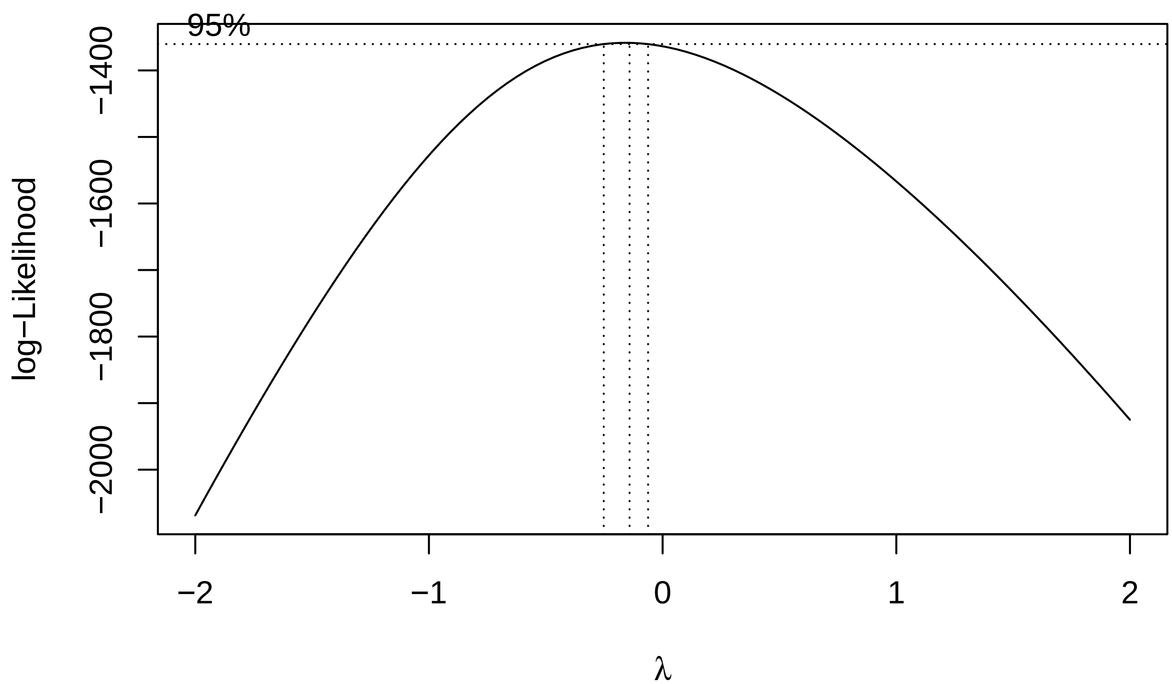

95%
log−Likelihood
−1600
−2000
−2400
−2
−1
0
1
2
λ

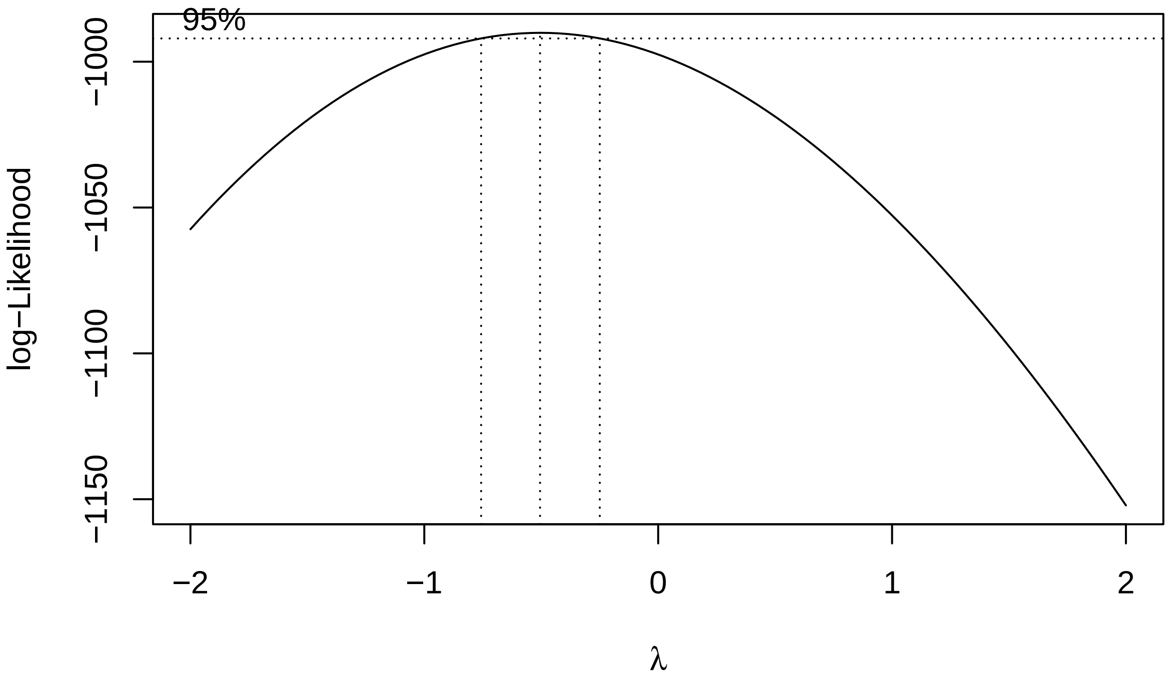

95%
−680
−700
−720
−740
−760
log−Likelihood
−2
−1
0
1
2
λ

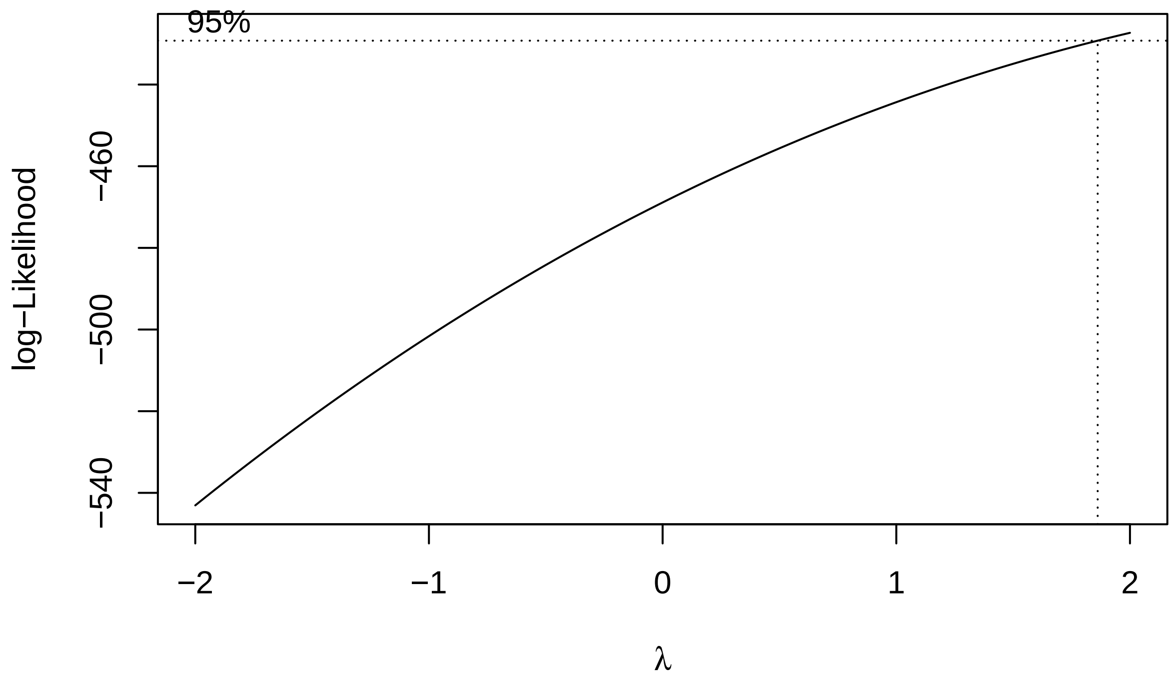

95%
log−Likelihood
−410
−420
−430
−440
−2
−1
0
1
2
λ

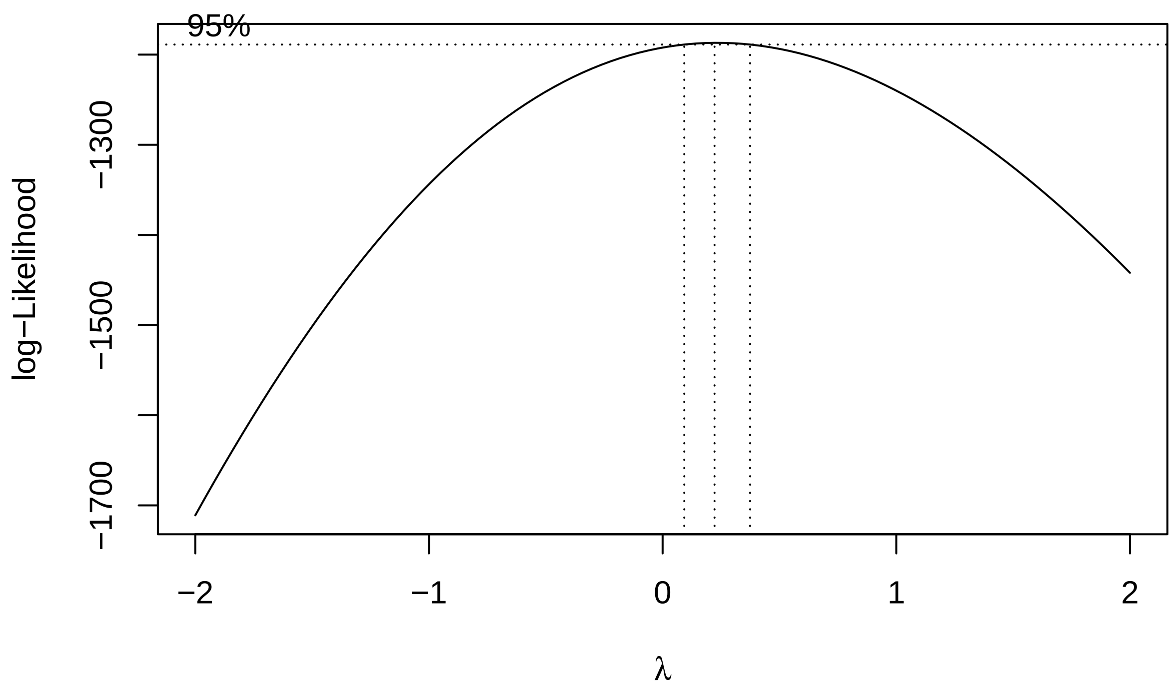

95%
log−Likelihood
−1500
−2000
−2500
−2
−1
0
1
2
λ

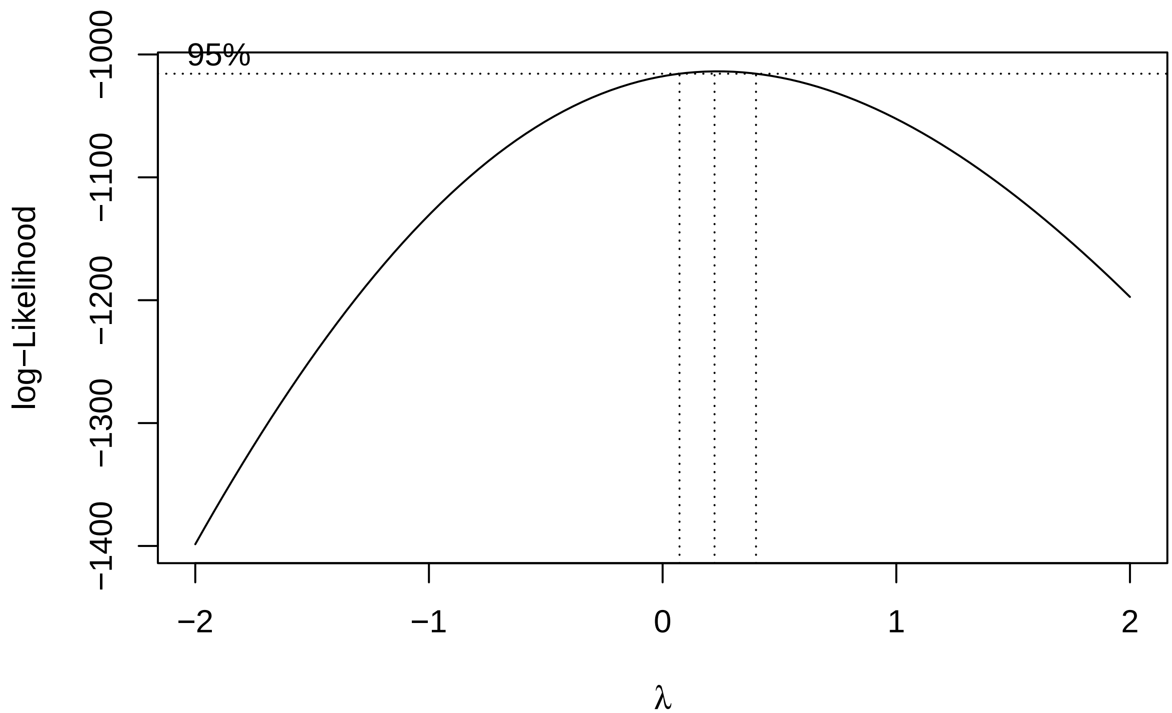

95%
log−Likelihood
−1200
−1300
−1400
−2
−1
0
1
2
λ


95%
−1400
−1600
−1800
−2000
log−Likelihood
−2
−1
0
1
2
λ


95%
log−Likelihood
−1000
−1050
−1100
−1150
−2
−1
0
1
2
λ


95%
log−Likelihood
−460
−500
−540
−2
−1
0
1
2
λ


95%
log−Likelihood
−1300
−1500
−1700
−2
−1
0
1
2
λ



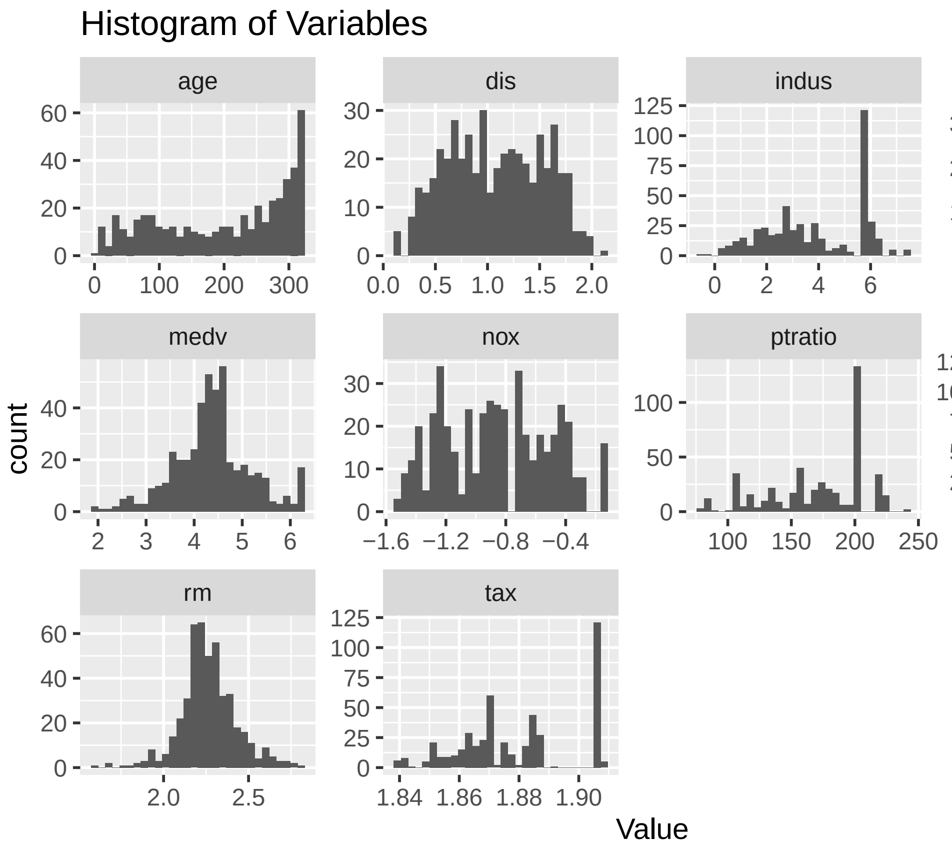

```r
# Convert the list of transformed columns into a DataFrame
df_transformed <- as.data.frame(col_transformed)
```

2.2.3 Gather

Examining the variables after applying the Box-Cox transformation

```r
# Gather the data into a long format
data_transformed_long <- gather(df_transformed, key = "Variable", value = "Value")

ggplot(data_transformed_long, aes(x = Value)) +
  geom_histogram() +
  facet_wrap(~Variable, scales = "free") +
  labs(title = "Histogram of Variables")
```

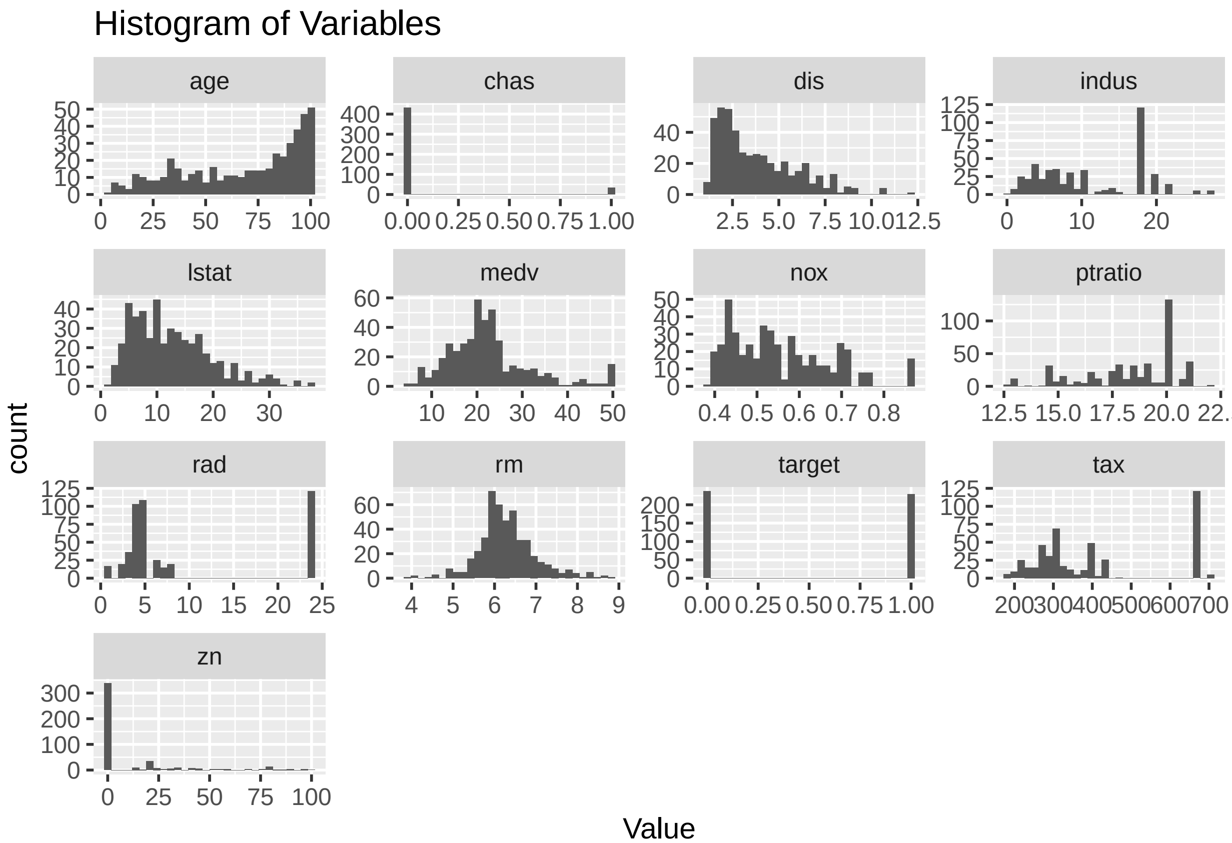
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



After examining the histograms, it's evident that the variables 'dis,' 'lstat,' 'medv,' and 'nox' have undergone a transformation resulting in a more normal distribution. Subsequently, these transformed variables will replace their original counterparts in the original dataset. The objective is to assess whether having more normalized variables contributes to the creation of a better model. This replacement aligns with the intention of leveraging the Box-Cox transformation to enhance the normality of the variables, potentially leading to improvements in the model's performance.

```
# Gather the data into a long format
df_long <- gather(df_crime_train, key = "Variable", value = "Value")

ggplot(df_long, aes(x = Value)) +
  geom_histogram() +
  facet_wrap(~Variable, scales = "free") +
  labs(title = "Histogram of Variables")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



2.2.4 Consolidate 'df_crime_train' data with 'transformed'

```
# Move
df_crime_train_with_transformed <- df_transformed %>%
  dplyr::select(dis, lstat, medv, nox) %>%
  mutate(dis_t = dis, lstat_t = lstat, medv_t = medv, nox_t = nox)%>%
               dplyr::select(-c(dis, lstat, medv, nox))
```

2.2.5 Combining Results

```
# Combine data frames by adding columns
result <- cbind(df_crime_train_with_transformed, df_crime_train %>%
               dplyr::select(-c(dis, lstat, medv, nox)))
```

2.3 Correlation Matrix with 'df_crime_train'

```
# Create a correlation matrix for all variables
(matrix_cor <- cor(result))
```

```
##               dis_t     lstat_t     medv_t       nox_t          zn       indus
## dis_t    1.00000000 -0.56179715  0.4015341 -0.87709320  0.57641370 -0.75792603
## lstat_t -0.56179715  1.00000000 -0.8263703  0.62045618 -0.49640280  0.61605309
## medv_t   0.40153414 -0.82637027  1.0000000 -0.50211171  0.38117040 -0.54583768
## nox_t   -0.87709320  0.62045618 -0.5021117  1.00000000 -0.61422595  0.78007417
## zn       0.57641370 -0.49640280  0.3811704 -0.61422595  1.00000000 -0.53826643
## indus   -0.75792603  0.61605309 -0.5458377  0.78007417 -0.53826643  1.00000000
## chas    -0.07750927 -0.06338501  0.1527892  0.08085077 -0.04016203  0.06118317
## rm       0.25918152 -0.67343224  0.6629534 -0.29807776  0.31981410 -0.39271181
## age     -0.78183574  0.61820150 -0.4425546  0.79350670 -0.57258054  0.63958182
## rad     -0.56530309  0.48965607 -0.4770309  0.61533605 -0.31548119  0.60062839
## tax     -0.62675351  0.55590617 -0.5646188  0.66553959 -0.31928408  0.73222922
## ptratio -0.23748298  0.41969279 -0.5141646  0.25253161 -0.39103573  0.39468980
## target  -0.65585498  0.45542422 -0.3435728  0.75332427 -0.43168176  0.60485074
##                chas          rm         age         rad         tax    ptratio
## dis_t   -0.07750927  0.25918152 -0.78183574 -0.56530309 -0.62675351 -0.2374830
## lstat_t -0.06338501 -0.67343224  0.61820150  0.48965607  0.55590617  0.4196928
## medv_t   0.15278916  0.66295338 -0.44255459 -0.47703086 -0.56461880 -0.5141646
## nox_t    0.08085077 -0.29807776  0.79350670  0.61533605  0.66553959  0.2525316
## zn      -0.04016203  0.31981410 -0.57258054 -0.31548119 -0.31928408 -0.3910357
## indus    0.06118317 -0.39271181  0.63958182  0.60062839  0.73222922  0.3946898
## chas     1.00000000  0.09050979  0.07888366 -0.01590037 -0.04676476 -0.1286606
## rm       0.09050979  1.00000000 -0.23281251 -0.20844570 -0.29693430 -0.3603471
## age      0.07888366 -0.23281251  1.00000000  0.46031430  0.51212452  0.2554479
## rad     -0.01590037 -0.20844570  0.46031430  1.00000000  0.90646323  0.4714516
## tax     -0.04676476 -0.29693430  0.51212452  0.90646323  1.00000000  0.4744223
## ptratio -0.12866058 -0.36034706  0.25544785  0.47145160  0.47442229  1.0000000
## target   0.08004187 -0.15255334  0.63010625  0.62810492  0.61111331  0.2508489
##              target
## dis_t   -0.65585498
## lstat_t  0.45542422
## medv_t  -0.34357282
## nox_t    0.75332427
## zn      -0.43168176
## indus    0.60485074
## chas     0.08004187
## rm      -0.15255334
## age      0.63010625
## rad      0.62810492
## tax      0.61111331
## ptratio  0.25084892
## target   1.00000000
```

2.4 Apply Scaling

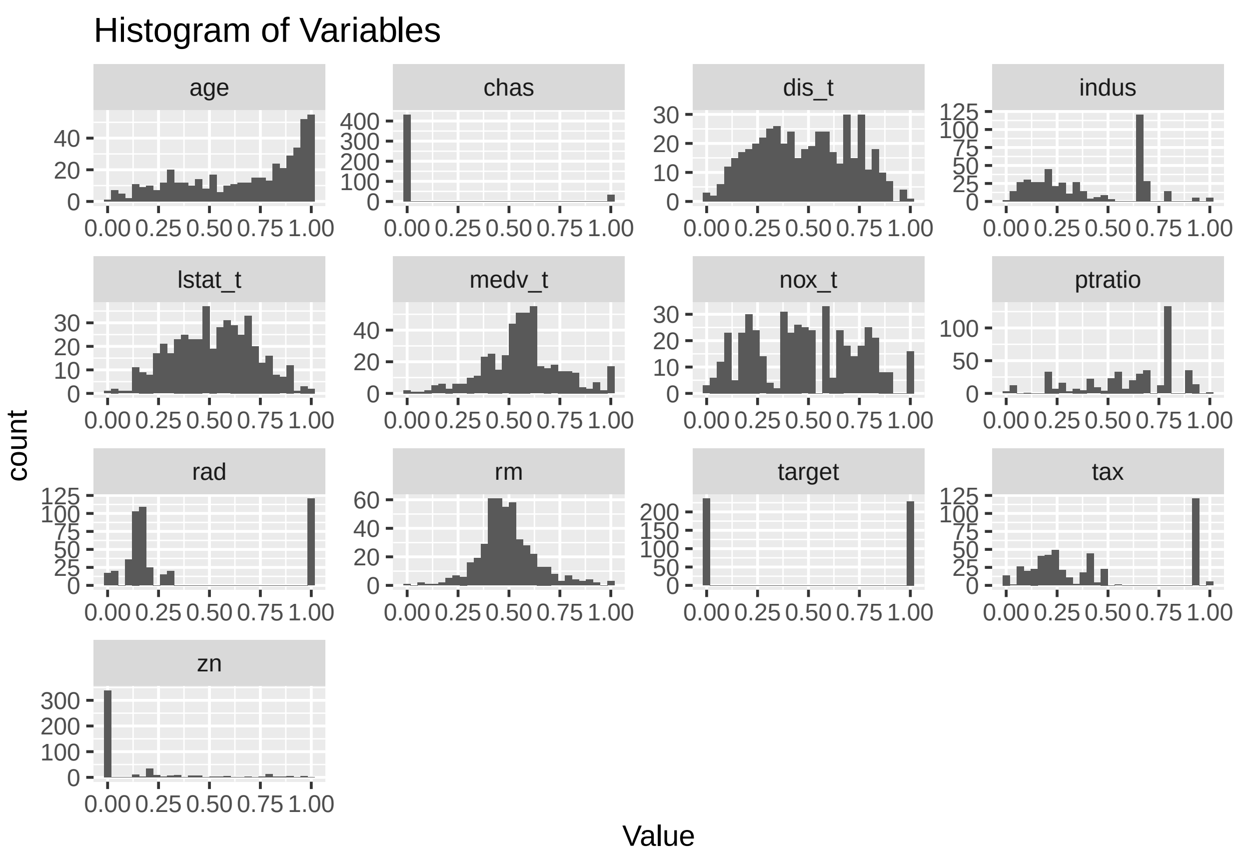
Additionally, we will normalize the variables using the same scaling technique once again. This step ensures consistency in the treatment of variables and allows us to maintain a standardized scale across the dataset.

```r
# Apply min-max scaling to all three variables
df_scaled <- result
df_scaled[] <- lapply(result, rescale)
```

2.5 Gather Scaled Data

```r
# Gather the data into a long format
df_crime_train_with_transformed <- gather(df_scaled, key = "Variable", value = "Value")

ggplot(df_crime_train_with_transformed, aes(x = Value)) +
  geom_histogram() +
  facet_wrap(~Variable, scales = "free") +
  labs(title = "Histogram of Variables")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Certainly, now that we have completed the necessary data preprocessing steps, including variable transformation and normalization, it's time to proceed with building the models.

3 BUILD MODELS

Let us explore three different models including:

- A - Backward Elimination with AIC Criterion
- B - Forward Selection with AIC Criterion
- C - Forward Selection + Interactions + Non-transformed Variables

3.1 A - Backward Elimination with AIC Criterion

3.1.1 A1 - ALL Variables -AIC 222.37

```r
# Including all variables
A1_back_elim <- glm(formula = target ~ ., family = binomial (link="logit"), data = df_scaled)

summary(A1_back_elim)
```

```
##
## Call:
## glm(formula = target ~ ., family = binomial(link = "logit"),
##     data = df_scaled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -24.8993     4.7369  -5.256 1.47e-07 ***
## dis_t         7.6061     2.1514   3.535 0.000407 ***
## lstat_t       0.6406     2.0150   0.318 0.750550
## medv_t        8.5589     3.7925   2.257 0.024021 *
## nox_t        19.7762     3.1080   6.363 1.98e-10 ***
## zn           -2.0891     2.7615  -0.757 0.449345
## indus        -0.3741     1.2381  -0.302 0.762554
## chas          0.8386     0.7557   1.110 0.267133
## rm           -1.3986     3.2690  -0.428 0.668771
## age           3.5322     1.3393   2.637 0.008353 **
## rad          14.2778     3.7287   3.829 0.000129 ***
## tax          -2.3200     1.5460  -1.501 0.133457
## ptratio       3.7771     1.2130   3.114 0.001847 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 196.37  on 453  degrees of freedom
## AIC: 222.37
##
## Number of Fisher Scoring iterations: 9
```

```r
drop1(A1_back_elim,test="Chi")
```

```
## Single term deletions
##
## Model:
## target ~ dis_t + lstat_t + medv_t + nox_t + zn + indus + chas +
##     rm + age + rad + tax + ptratio
##         Df Deviance    AIC    LRT  Pr(>Chi)
## <none>       196.37 222.37
## dis_t    1   210.51 234.51 14.139 0.0001698 ***
## lstat_t  1   196.47 220.47  0.101 0.7505401
## medv_t   1   202.03 226.03  5.653 0.0174232 *
```

```
## nox_t    1   267.11 291.11 70.734 < 2.2e-16 ***
## zn       1   197.00 221.00  0.625 0.4292117
## indus    1   196.47 220.47  0.092 0.7618252
## chas     1   197.63 221.63  1.258 0.2621052
## rm       1   196.56 220.56  0.184 0.6683571
## age      1   203.89 227.89  7.518 0.0061092 **
## rad      1   236.11 260.11 39.737 2.905e-10 ***
## tax      1   198.66 222.66  2.287 0.1304285
## ptratio  1   207.17 231.17 10.798 0.0010161 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.1.1.1 Observations

- AIC of 222.37
- Residual Deviance of 196.37 on 453 df
- 6 of 12 variable coefficients and the intercept coefficient are significant ($p < .05$)
- Indus variable has largest p-value of .76

3.1.2 A2 - Removed Variable (indus) with Largest P-Value -AIC 220.46

```r
# Removed variables: indus
A2_back_elim <- glm(formula = target ~ dis_t
                   + medv_t + nox_t + zn
                   + lstat_t + chas + rm
                   + age + rad + tax
                   + ptratio
                   , family = binomial, data = df_scaled)

summary(A2_back_elim)
```

```
##
## Call:
## glm(formula = target ~ dis_t + medv_t + nox_t + zn + lstat_t +
##     chas + rm + age + rad + tax + ptratio, family = binomial,
##     data = df_scaled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -24.9602     4.7432  -5.262 1.42e-07 ***
## dis_t         7.7014     2.1333   3.610 0.000306 ***
## medv_t        8.5315     3.8057   2.242 0.024976 *
## nox_t        19.5489     3.0038   6.508 7.61e-11 ***
## zn           -2.0940     2.7472  -0.762 0.445919
## lstat_t       0.6210     2.0117   0.309 0.757535
## chas          0.7981     0.7449   1.071 0.283960
## rm           -1.3351     3.2649  -0.409 0.682583
## age           3.5174     1.3381   2.629 0.008570 **
## rad          14.6196     3.5681   4.097 4.18e-05 ***
## tax          -2.4723     1.4514  -1.703 0.088499 .
## ptratio       3.7537     1.2128   3.095 0.001967 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 196.46  on 454  degrees of freedom
## AIC: 220.46
##
## Number of Fisher Scoring iterations: 9
```

3.1.2.1 Observations

- AIC decreased to 220.46 (previously 222.37)
- Residual Deviance increased to 196.46 on 454 df (previously 196.37 on 453 df)
- 6 of 11 variable coefficients and the intercept coefficient are significant ($p < .05$)
- lstat_t variable has largest p-value of .757

3.1.3 A3 - Removed Next Variable with Largest P-Value (lstat_t) -AIC 218.56

```r
# Removed variables: indus, lstat_t
A3_back_elim <- glm(formula = target ~ dis_t
                   + medv_t + nox_t + zn
                   + chas + rm
                   + age + rad + tax
                   + ptratio
                   , family = binomial, data = df_scaled)

summary(A3_back_elim)
```

```
##
## Call:
## glm(formula = target ~ dis_t + medv_t + nox_t + zn + chas + rm +
##     age + rad + tax + ptratio, family = binomial, data = df_scaled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -24.4294     4.3986  -5.554 2.79e-08 ***
## dis_t         7.6576     2.1242   3.605 0.000312 ***
## medv_t        8.2748     3.7043   2.234 0.025493 *
## nox_t        19.5767     3.0023   6.521 7.00e-11 ***
## zn           -1.9451     2.6770  -0.727 0.467470
## chas          0.8327     0.7417   1.123 0.261568
## rm           -1.6828     3.0604  -0.550 0.582421
## age           3.6495     1.2680   2.878 0.004001 **
## rad          14.6939     3.5543   4.134 3.56e-05 ***
## tax          -2.4863     1.4504  -1.714 0.086475 .
## ptratio       3.7661     1.2120   3.107 0.001888 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 196.56  on 455  degrees of freedom
## AIC: 218.56
##
## Number of Fisher Scoring iterations: 9
```

3.1.3.1 Observations

- AIC decreased to 218.56 (previously 220.46)
- Residual Deviance increased to 196.56 on 455 df (previously 196.46 on 454 df)
- 6 of 10 variable coefficients and the intercept coefficient are significant ($p < .05$)
- rm variable has largest p-value of .58

3.1.4 A4 - Removed Next Variable with Largest P-Value (rm) -AIC 216.86

```r
# Removed variables: indus, lstat_t, rm
A4_back_elim <- glm(formula = target ~ dis_t
                   + medv_t + nox_t + zn
                   + chas
                   + age + rad + tax
                   + ptratio
                   , family = binomial, data = df_scaled)

summary(A4_back_elim)
```

```
##
## Call:
## glm(formula = target ~ dis_t + medv_t + nox_t + zn + chas + age +
##     rad + tax + ptratio, family = binomial, data = df_scaled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -23.5368     4.0142  -5.863 4.54e-09 ***
## dis_t         7.3709     2.0399   3.613 0.000302 ***
## medv_t        6.6201     2.0910   3.166 0.001546 **
## nox_t        19.3079     2.9428   6.561 5.35e-11 ***
## zn           -2.2262     2.6417  -0.843 0.399379
## chas          0.8862     0.7418   1.195 0.232181
## age           3.3159     1.0999   3.015 0.002571 **
## rad          14.4449     3.4990   4.128 3.65e-05 ***
## tax          -2.5264     1.4380  -1.757 0.078953 .
## ptratio       3.5046     1.0997   3.187 0.001438 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 196.86  on 456  degrees of freedom
```

```
## AIC: 216.86
##
## Number of Fisher Scoring iterations: 9
```

3.1.4.1 Observations

- AIC decreased to 216.86 (previously 218.56)
- Residual Deviance increased to 196.86 on 456 df (previously 196.56 on 455 df)
- Same 6 variable coefficients of 9 and the intercept coefficient are significant (p < .05)
- zn variable has largest p-value of .399

3.1.5 A5 - Removed Next Variable with Largest P-Value (zn) -AIC 215.64

```r
# Removed variables: indus, lstat_t, rm, zn
A5_back_elim <- glm(formula = target ~ dis_t
                    + medv_t + nox_t
                    + chas
                    + age + rad + tax
                    + ptratio
                    , family = binomial, data = df_scaled)

summary(A5_back_elim)
```

```
##
## Call:
## glm(formula = target ~ dis_t + medv_t + nox_t + chas + age +
##     rad + tax + ptratio, family = binomial, data = df_scaled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -23.9105     4.0029  -5.973 2.33e-09 ***
## dis_t         7.2060     2.0184   3.570 0.000357 ***
## medv_t        6.5630     2.0839   3.149 0.001636 **
## nox_t        19.7319     2.9288   6.737 1.62e-11 ***
## chas          0.9890     0.7201   1.373 0.169619
## age           3.3632     1.0959   3.069 0.002150 **
## rad          14.3641     3.4113   4.211 2.55e-05 ***
## tax          -2.5650     1.4078  -1.822 0.068458 .
## ptratio       3.7976     1.0527   3.608 0.000309 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 197.64  on 457  degrees of freedom
## AIC: 215.64
##
## Number of Fisher Scoring iterations: 9
```

3.1.5.1 Observations

- AIC decreased to 215.64 (previously 216.86)
- Residual Deviance increased to 197.64 on 457 df (previously 196.86 on 456 df)
- Same 6 of 8 variable coefficients and the intercept coefficient are significant ($p < .05$)
- chas variable has largest p-value of .170

3.1.6 A6 - Removed Next Variable with Largest P-Value (chas) -AIC 215.57

```r
# Removed variables: indus, lstat_t, rm, zn, chas
A6_back_elim <- glm(formula = target ~ dis_t
                   + medv_t + nox_t
                   + age + rad + tax
                   + ptratio
                   , family = binomial, data = df_scaled)

summary(A6_back_elim)
```

```
##
## Call:
## glm(formula = target ~ dis_t + medv_t + nox_t + age + rad + tax +
##     ptratio, family = binomial, data = df_scaled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -23.321      3.938  -5.922 3.18e-09 ***
## dis_t          6.858      1.972   3.477 0.000506 ***
## medv_t         6.401      2.068   3.096 0.001963 **
## nox_t         19.218      2.869   6.699 2.09e-11 ***
## age            3.496      1.093   3.198 0.001386 **
## rad           14.976      3.381   4.429 9.45e-06 ***
## tax           -2.800      1.402  -1.997 0.045810 *
## ptratio        3.577      1.034   3.459 0.000542 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 199.57  on 458  degrees of freedom
## AIC: 215.57
##
## Number of Fisher Scoring iterations: 9
```

3.1.6.1 Observations

- AIC decreased to 215.57 (previously 215.64)
- Residual Deviance increased to 199.57 on 458 df (previously 197.64 on 457 df)
- 7 of 7 variable coefficients and the intercept coefficient are significant ($p < .05$)

```r
anova(A1_back_elim, A6_back_elim, test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ dis_t + lstat_t + medv_t + nox_t + zn + indus + chas +
##     rm + age + rad + tax + ptratio
## Model 2: target ~ dis_t + medv_t + nox_t + age + rad + tax + ptratio
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       453     196.37
## 2       458     199.57 -5  -3.2003   0.6691
```

3.1.7 BEST MODEL: A6_back_elim

- Predictors: dis_t + medv_t + nox_t + age + rad + tax + ptratio
- Best AIC of 215.57
- Variable Coefficients - As shown below, our model indicates that the crime rate is more likely to be over the median with greater nitrogen oxide concentration (nox), accessibility to radial highways (rad), weighted mean of distances to five Boston employment centers (dis), proportion of owner-occupied units built prior to 1940 (age), median value of owner-occupied homes in 1000s (medv), pupil-teacher ratio by town (ptratio), and less likely to be over the median with greater full-value property-tax rate per 10,000 (tax)

Variable Coefficients:

```r
(A6_beta <- coef(A6_back_elim))
```

```
## (Intercept)       dis_t      medv_t       nox_t         age         rad
##  -23.320694    6.858256    6.400997   19.218101    3.495959   14.976203
##         tax     ptratio
##   -2.800110    3.577214
```

Model Odds Ratios:

```r
format(exp(A6_beta), scientific = F)
```

```
##                   (Intercept)                          dis_t
## "        0.00000000007446485" "      951.70619909853917306"
##                        medv_t                          nox_t
## "      602.44553028409688977" "221980693.93047717213630676"
##                           age                            rad
## "       32.98191723503830985" "  3192144.02063742792233825"
##                           tax                        ptratio
## "        0.06080338020394439" "       35.77373841187006320"
```

3.2 B - Forward Selection with AIC Criterion

3.2.1 B1 - Start with Variable nox_t with Lowest P-Value -AIC 295.88

```r
B1_forward <- glm(formula = target ~ nox_t
                  , family = binomial, data = df_scaled)

summary(B1_forward)
```

```
##
## Call:
## glm(formula = target ~ nox_t, family = binomial, data = df_scaled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.8921     0.5575  -10.57   <2e-16 ***
## nox_t        12.0417     1.0967   10.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 291.88  on 464  degrees of freedom
## AIC: 295.88
##
## Number of Fisher Scoring iterations: 6
```

3.2.1.1 Observations

- AIC of 295.88
- Residual Deviance of 291.88 on 464 df
- 1 of 1 variable coefficients and the intercept coefficient are significant ($p < .05$)
- Rad variable has next smallest p-value

3.2.2 B2 - Add Variable with Next Lowest P-Value (rad) -AIC 243.42

```r
B2_forward <- glm(formula = target ~ nox_t
                  + rad
                  , family = binomial, data = df_scaled)

summary(B2_forward)
```

```
##
## Call:
## glm(formula = target ~ nox_t + rad, family = binomial, data = df_scaled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -7.535      0.873  -8.631  < 2e-16 ***
## nox_t         10.832      1.303   8.312  < 2e-16 ***
## rad           11.780      2.531   4.654 3.26e-06 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 237.42  on 463  degrees of freedom
## AIC: 243.42
##
## Number of Fisher Scoring iterations: 8
```

3.2.2.1 Observations

- AIC decreased to 243.42 (previously 295.88)
- Residual Deviance decreased to 237.42 on 463 df (previously 291.88 on 464 df)
- 2 of 2 variable coefficients and the intercept coefficient are significant (p < .05)
- dist_t variable has next smallest p-value

3.2.3 B3 - Add Variable with Next Lowest P-Value (dist_t) -AIC 237.33

```
B3_forward <- glm(formula = target ~ nox_t
                  + rad
                  + dis_t
                  , family = binomial, data = df_scaled)

summary(B3_forward)
```

```
##
## Call:
## glm(formula = target ~ nox_t + rad + dis_t, family = binomial,
##     data = df_scaled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -11.929      1.918  -6.218 5.03e-10 ***
## nox_t         15.300      2.211   6.921 4.50e-12 ***
## rad           12.530      2.654   4.721 2.34e-06 ***
## dis_t          4.295      1.547   2.776  0.00551 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 229.33  on 462  degrees of freedom
## AIC: 237.33
##
## Number of Fisher Scoring iterations: 8
```

3.2.3.1 Observations

- AIC decreased to 237.33 (previously 243.42)

- Residual Deviance decreased to 229.33 on 462 df (previously 237.42 on 463 df)
- 3 of 3 variable coefficients and the intercept coefficient are significant ($p < .05$)
- ptratio variable has next smallest p-value

3.2.4 B4 - Add Variable with Next Lowest P-Value (ptratio) -AIC 237.74

```
B4_forward <- glm(formula = target ~ nox_t
                  + rad
                  + dis_t
                  + ptratio
                  , family = binomial, data = df_scaled)

summary(B4_forward)
```

```
##
## Call:
## glm(formula = target ~ nox_t + rad + dis_t + ptratio, family = binomial,
##     data = df_scaled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -12.9027     2.1365  -6.039 1.55e-09 ***
## nox_t        15.6838     2.2925   6.841 7.85e-12 ***
## rad          13.6412     2.8807   4.735 2.19e-06 ***
## dis_t         4.3507     1.5524   2.803  0.00507 **
## ptratio       0.9104     0.7210   1.263  0.20668
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 227.74  on 461  degrees of freedom
## AIC: 237.74
##
## Number of Fisher Scoring iterations: 8
```

3.2.4.1 Observations

- AIC increased to 237.74 (previously 237.33)
- Residual Deviance decreased to 227.74 on 461 df (previously 229.33 on 462 df)
- 3 of 4 variable coefficients and the intercept coefficient are significant ($p < .05$)
- age variable has next smallest p-value

3.2.5 B5 - Add Variable with Next Lowest P-Value (age), Exclude ptratio -AIC 235.17

Have not included ptratio as the previous model B4 demonstrated that including this variable decreased the model fit as the AIC increased rather than decreased.

```r
B5_forward <- glm(formula = target ~ nox_t
                  + rad
                  + dis_t
                  + age
                  , family = binomial, data = df_scaled)

summary(B5_forward)
```

```
##
## Call:
## glm(formula = target ~ nox_t + rad + dis_t + age, family = binomial,
##     data = df_scaled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -13.0917     2.0626  -6.347 2.19e-10 ***
## nox_t        14.1593     2.2661   6.248 4.15e-10 ***
## rad          12.6002     2.6862   4.691 2.72e-06 ***
## dis_t         5.0251     1.6247   3.093  0.00198 **
## age           1.8354     0.9132   2.010  0.04445 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 225.17  on 461  degrees of freedom
## AIC: 235.17
##
## Number of Fisher Scoring iterations: 8
```

3.2.5.1 Observations

- AIC has decreased to 235.17 (previously 237.74 and 237.33)
- Residual Deviance decreased to 225.17 df (previously 227.74 on 461 df)
- 4 of 4 variable coefficients and the intercept coefficient are significant ($p < .05$)

3.2.6 BEST MODEL: B5_forward

- Predictors: nox_t + rad + dis_t + age
- Best AIC of 235.17
- Intuitive Variable Coefficients - As shown below, the crime rate is more likely to be over the median with greater nitrogen oxide concentration (nox), accessibility to radial highways (rad), weighted mean of distances to five Boston employment centers (dis) and proportion of owner-occupied units built prior to 1940 (age).

Variable Coefficients:

```r
(B5_beta <- coef(B5_forward))
```

```
## (Intercept)       nox_t         rad       dis_t         age
##  -13.091717   14.159316   12.600212    5.025085    1.835371
```

Model Odds Ratios:

```r
format(exp(B5_beta), scientific = F)
```

```
##           (Intercept)                  nox_t                    rad
## "      0.000002062243" "1410304.295350710163" " 296621.483488471014"
##                 dis_t                    age
## "    152.183224878728" "      6.267460190568"
```

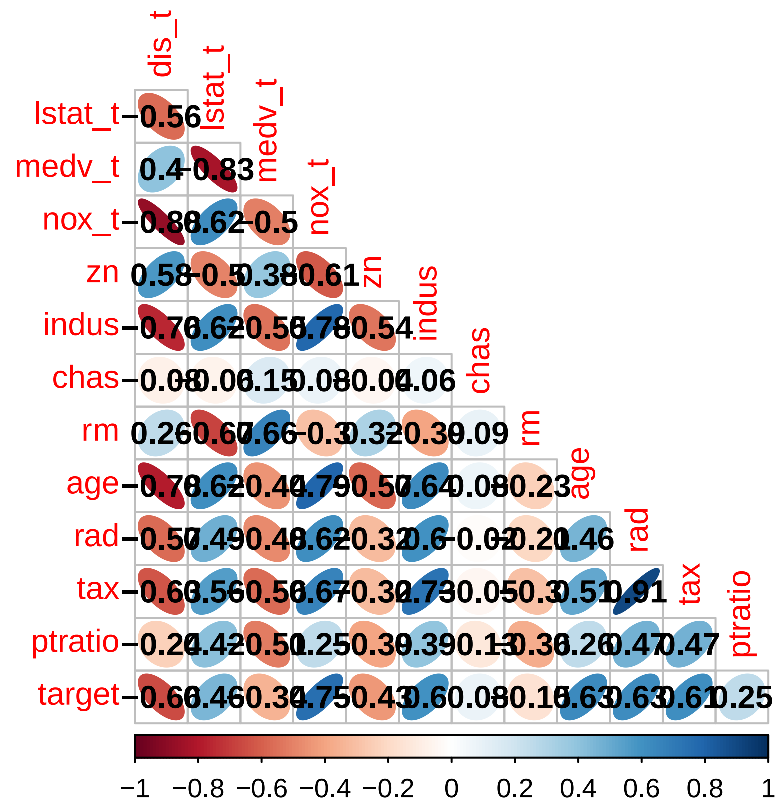
3.3 C - Forward Selection + Interactions + Non-transformed Variables

3.3.1 Correlations between Variables

```r
cor(df_scaled, y=df_scaled$target)
```

```
##                 [,1]
## dis_t   -0.65585498
## lstat_t  0.45542422
## medv_t  -0.34357282
## nox_t    0.75332427
## zn      -0.43168176
## indus    0.60485074
## chas     0.08004187
## rm      -0.15255334
## age      0.63010625
## rad      0.62810492
## tax      0.61111331
## ptratio  0.25084892
## target   1.00000000
```

```r
df_scaled %>%
  cor(.,) %>%
  corrplot(., method = "ellipse", type = "lower",addCoef.col = 'black', diag = FALSE)
```




```r
cor(df_scaled, y=df_scaled$nox_t)
```

```
##                [,1]
## dis_t   -0.87709320
## lstat_t  0.62045618
## medv_t  -0.50211171
## nox_t    1.00000000
## zn      -0.61422595
## indus    0.78007417
## chas     0.08085077
## rm      -0.29807776
## age      0.79350670
## rad      0.61533605
## tax      0.66553959
## ptratio  0.25253161
## target   0.75332427
```

3.3.2 C1 - Add Interaction between nox_t & dist_t because of strong negative relationship with each other -AIC 237.02

Starting out with our Forward Selection Model B5 and adding the interaction term for nox_t & dis_t as they have a correlation of -.877.

```
C1_model <- glm(formula = target ~
                    nox_t
                   + rad
                + dis_t
                   + age
                  + nox_t*dis_t
                   , family = binomial, data = df_scaled)

summary(C1_model)
```

```
##
## Call:
## glm(formula = target ~ nox_t + rad + dis_t + age + nox_t * dis_t,
##     family = binomial, data = df_scaled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -12.4184     2.6582  -4.672 2.99e-06 ***
## nox_t        12.9311     3.8119   3.392 0.000693 ***
## rad          12.6157     2.6990   4.674 2.95e-06 ***
## dis_t         3.8283     3.4577   1.107 0.268212
## age           1.8020     0.9203   1.958 0.050216 .
## nox_t:dis_t   2.4060     6.1546   0.391 0.695847
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 225.02  on 460  degrees of freedom
## AIC: 237.02
##
## Number of Fisher Scoring iterations: 8
```

3.3.2.1 Observations Compared to the original B5 Model: * AIC has increased to 237.02 (previously 235.17) * Residual Deviance decreased to 225.02 on 460 df (previously 225.17 on 461 df) * Only nox_t and rad variable coefficients and the intercept coefficient are significant ($p < .05$) * Adding the interaction term has affected the goodness of fit negatively as not only is it not significant, but the variable coefficients for dis_t and age are no longer significant.

3.3.3 C2 - Add Interaction terms between all predictors (excluding nox_t x dis_t) -AIC 236.07

Starting out with our Forward Selection Model B5 and adding the interaction terms for all the variables besides the one previously tested above to determine if any interaction terms may be beneficial to the model.

```
C2_model <- glm(formula = target ~
                    nox_t + rad + dis_t + age
                  + nox_t*rad + nox_t*age
                  + rad*dis_t + rad*age
                  + dis_t*age
                   , family = binomial, data = df_scaled)
```

```r
summary(C2_model)
```

```
##
## Call:
## glm(formula = target ~ nox_t + rad + dis_t + age + nox_t * rad +
##     nox_t * age + rad * dis_t + rad * age + dis_t * age, family = binomial,
##     data = df_scaled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -15.119      6.950  -2.175 0.029600 *
## nox_t         13.495      6.055   2.229 0.025829 *
## rad           67.200     22.411   2.999 0.002713 **
## dis_t          8.577      7.498   1.144 0.252660
## age           -8.971      8.292  -1.082 0.279319
## nox_t:rad    -51.356     14.845  -3.460 0.000541 ***
## nox_t:age     15.043      8.337   1.804 0.071174 .
## rad:dis_t    -54.349     23.925  -2.272 0.023108 *
## rad:age       -3.242     13.943  -0.233 0.816122
## dis_t:age      8.165      7.287   1.120 0.262548
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 216.07  on 456  degrees of freedom
## AIC: 236.07
##
## Number of Fisher Scoring iterations: 9
```

3.3.3.1 Observations

- AIC has decreased to 236.07 (previously 237.02)
- Residual Deviance decreased to 216.07 on 456 df (previously 225.02 on 460 df)
- Interaction term nox_t*rad has a significant variable coefficient with p-value of 0.000541, indicating it could be beneficial to add to our model

3.3.4 C3 - Add Interaction term with smallest p-value (nox_t x rad) -AIC 236.69

Starting out with our Forward Selection Model B5 again, and adding the interaction term between nox_t and rad as our previous model C2 indicated the variable coefficient for this interaction was very small.

```r
C3_model <- glm(formula = target ~
                   nox_t + rad + dis_t + age
                 + nox_t*rad
                  , family = binomial, data = df_scaled)

summary(C3_model)
```

```
##
```

```
## Call:
## glm(formula = target ~ nox_t + rad + dis_t + age + nox_t * rad,
##     family = binomial, data = df_scaled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -14.2549     2.6919  -5.296 1.19e-07 ***
## nox_t        16.7365     4.2771   3.913 9.11e-05 ***
## rad          18.8953     9.0986   2.077  0.03783 *
## dis_t         5.0258     1.6285   3.086  0.00203 **
## age           1.7829     0.9113   1.956  0.05041 .
## nox_t:rad   -14.0984    18.8918  -0.746  0.45550
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 224.69  on 460  degrees of freedom
## AIC: 236.69
##
## Number of Fisher Scoring iterations: 10
```

3.3.4.1 Observations

- AIC has increased to 236.69 (previously 236.07)
- Residual Deviance increased to 224.69 on 460 df (previously 216.07 on 456 df)
- Interaction term nox_t*rad is not significant in this model where the other interaction terms were removed. Additionally, the variable coefficient for age is no longer significant and since the AIC increased, we determine that including the interaction term negatively impacts our model fit.

3.3.5 C4 - Original Model without Transformations -AIC 244.17

We have been using the transformed version of the data in df_scaled, but would the results be similar if we used the original dataset?

```r
C4_no_transform <- glm(formula = target ~ nox
                  + rad
                  + dis
                  + age
                  , family = binomial, data = df_crime_train)

summary(C4_no_transform)
```

```
##
## Call:
## glm(formula = target ~ nox + rad + dis + age, family = binomial,
##     data = df_crime_train)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -20.224999   3.199415  -6.321 2.59e-10 ***
```

```
## nox          28.296396   5.071309   5.580 2.41e-08 ***
## rad           0.521295   0.109765   4.749 2.04e-06 ***
## dis           0.239127   0.147455   1.622   0.1049
## age           0.017349   0.009008   1.926   0.0541 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 234.17  on 461  degrees of freedom
## AIC: 244.17
##
## Number of Fisher Scoring iterations: 8
```

3.3.5.1 Observations Compared to the B5 Model using Transformed Variables: * AIC has increased to 244.17 (previously 235.17) * Residual Deviance increased to 234.17 on 461 df (previously 225.17 on 461 df) * Dis and Age variable coefficients are not significant

3.3.6 C5 - Remove age variable -AIC 245.96

If we were doing forward selection on the original data then we likely would not have added the age variable as a predictor as the dis variable may not have been significant when added before it. So let's see what our model looks like using nox, rad and dis and excluding age.

```r
# Removed age
C5_no_transform <- glm(formula = target ~ nox
                   + rad
                   + dis
                   , family = binomial, data = df_crime_train)

summary(C5_no_transform)
```

```
##
## Call:
## glm(formula = target ~ nox + rad + dis, family = binomial, data = df_crime_train)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -20.4385     3.1815  -6.424 1.33e-10 ***
## nox          31.4710     4.8497   6.489 8.63e-11 ***
## rad           0.5226     0.1089   4.799 1.60e-06 ***
## dis           0.1803     0.1438   1.254     0.21
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 237.96  on 462  degrees of freedom
## AIC: 245.96
##
## Number of Fisher Scoring iterations: 8
```

3.3.6.1 Observations

- AIC has increased to 245.96 (previously 244.17)
- Residual Deviance increased to 237.96 on 462 df (previously 234.17 on 461 df)
- Dis variable coefficient is still not significant

3.3.7 C6 - Remove dis variable -AIC 245.51

If we were doing forward selection on the original data then we may also have to exclude the dis variable.

```r
# Removed age and dis
C6_no_transform <- glm(formula = target ~ nox
                    + rad
                    , family = binomial, data = df_crime_train)

summary(C6_no_transform)
```

```
##
## Call:
## glm(formula = target ~ nox + rad, family = binomial, data = df_crime_train)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -17.4532     1.9488  -8.956  < 2e-16 ***
## nox          27.1964     3.2317   8.415  < 2e-16 ***
## rad           0.5139     0.1082   4.750 2.04e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 239.51  on 463  degrees of freedom
## AIC: 245.51
##
## Number of Fisher Scoring iterations: 8
```

3.3.7.1 Observations

- AIC has decreased to 245.51 (previously 245.96)
- Residual Deviance increased to 239.51 on 463 df (previously 237.96 on 462 df)
- All variable coefficients and intercept are significant

3.3.8 BEST MODEL: C6_no_transform

- Predictors: nox_t + rad
- Forward Selection model using original dataset rather than the transformed dataset
- AIC of 245.51

4 MODEL SELECTION

4.1 Selection Criteria to Consider

*Simplicity of Model, AIC, and Variable Coefficients

4.1.1 Backward Elimination Model - A6_back_elim

- Predictors: dis_t + medv_t + nox_t + age + rad + tax + ptratio
- Best AIC of 215.57
- Variable Coefficients - As shown below, our model indicates that the crime rate is more likely to be over the median with greater nitrogen oxide concentration (nox), accessibility to radial highways (rad), weighted mean of distances to five Boston employment centers (dis), proportion of owner-occupied units built prior to 1940 (age), median value of owner-occupied homes in 1000s (medv), pupil-teacher ratio by town (ptratio), and less likely to be over the median with greater full-value property-tax rate per 10,000 (tax)

4.1.2 Forward Selection Model - B5_forward

- Predictors: nox_t + rad + dis_t + age
- Best AIC of 235.17
- Intuitive Variable Coefficients - the crime rate is more likely to be over the median with greater nitrogen oxide concentration (nox), accessibility to radial highways (rad), weighted mean of distances to five Boston employment centers (dis) and proportion of owner-occupied units built prior to 1940 (age).

4.1.3 Forward Selection Model on Untransformed Data: C6_no_transform

- Predictors: nox_t + rad
- Forward Selection model using original dataset rather than the transformed dataset
- AIC of 245.51

4.2 Selected Model

We chose the Backward Elimination Model using the transformed dataset (A6_back_elim) as it has the lowest AIC and the variable coefficients make sense. Although the other two models are simpler given they have less predictors, they do have higher AICs in comparison.

4.2.1 Regression Summary for Selected Model

```r
summary(A6_back_elim)
```

```
##
## Call:
## glm(formula = target ~ dis_t + medv_t + nox_t + age + rad + tax +
##     ptratio, family = binomial, data = df_scaled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)  -23.321      3.938  -5.922 3.18e-09 ***
## dis_t          6.858      1.972   3.477 0.000506 ***
## medv_t         6.401      2.068   3.096 0.001963 **
## nox_t         19.218      2.869   6.699 2.09e-11 ***
## age            3.496      1.093   3.198 0.001386 **
## rad           14.976      3.381   4.429 9.45e-06 ***
## tax           -2.800      1.402  -1.997 0.045810 *
## ptratio        3.577      1.034   3.459 0.000542 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 199.57  on 458  degrees of freedom
## AIC: 215.57
##
## Number of Fisher Scoring iterations: 9
```

4.3 Evaluate Selected Binary Logistic Regression Model

```
# Add the predicted class based on selected model
df_scaled_classification <- df_scaled
df_scaled_classification$PRED = predict(A6_back_elim, new = df_scaled_classification, type="response")
df_scaled_classification$PRED_CLASS <- ifelse(df_scaled_classification$PRED > 0.5, 1, 0)
```

```
table(df_scaled_classification$PRED_CLASS, df_scaled_classification$target)
```

```
##
##       0   1
##   0 220  18
##   1  17 211
```

4.3.1 Confusion Matrix & Statistics

```
ls_class <- relevel(factor(df_scaled_classification$target), ref = "1") ## changes it from the default r
ls_scr_class <- relevel(factor(df_scaled_classification$PRED_CLASS), ref = "1")

confusionMatrix(data=ls_scr_class, reference = ls_class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   0
##          1 211  17
##          0  18 220
##
##                Accuracy : 0.9249
##                  95% CI : (0.8971, 0.9471)
```

```
##     No Information Rate : 0.5086
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8497
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9214
##             Specificity : 0.9283
##          Pos Pred Value : 0.9254
##          Neg Pred Value : 0.9244
##              Prevalence : 0.4914
##          Detection Rate : 0.4528
##    Detection Prevalence : 0.4893
##       Balanced Accuracy : 0.9248
##
##        'Positive' Class : 1
##
```

4.3.2 ROC Curve & AUC

```r
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.3.1
```

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

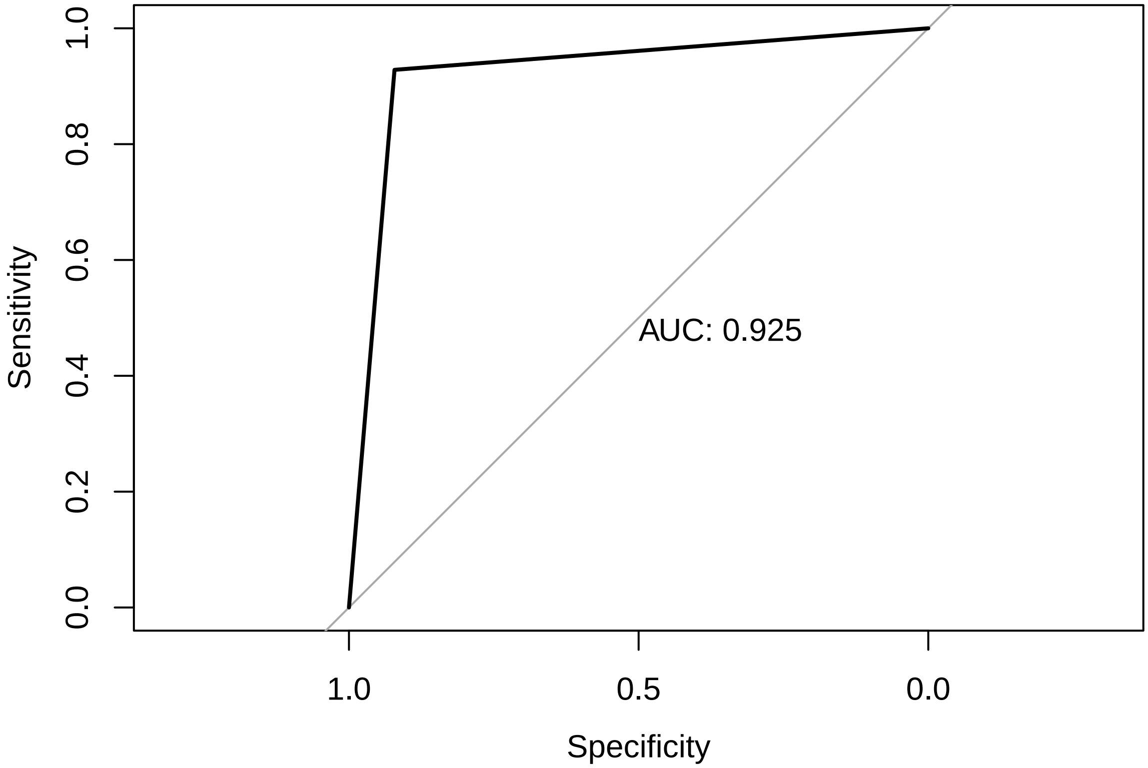
```
##
## Attaching package: 'pROC'
```

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

```r
roc(as.numeric(ls_class), as.numeric(ls_scr_class), plot = TRUE, print.auc = TRUE)
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```




```
##
## Call:
## roc.default(response = as.numeric(ls_class), predictor = as.numeric(ls_scr_class),     plot = TRUE, p
##
## Data: as.numeric(ls_scr_class) in 229 controls (as.numeric(ls_class) 1) < 237 cases (as.numeric(ls_cl
## Area under the curve: 0.9248
```

4.4 Predictions for Evaluation Dataset

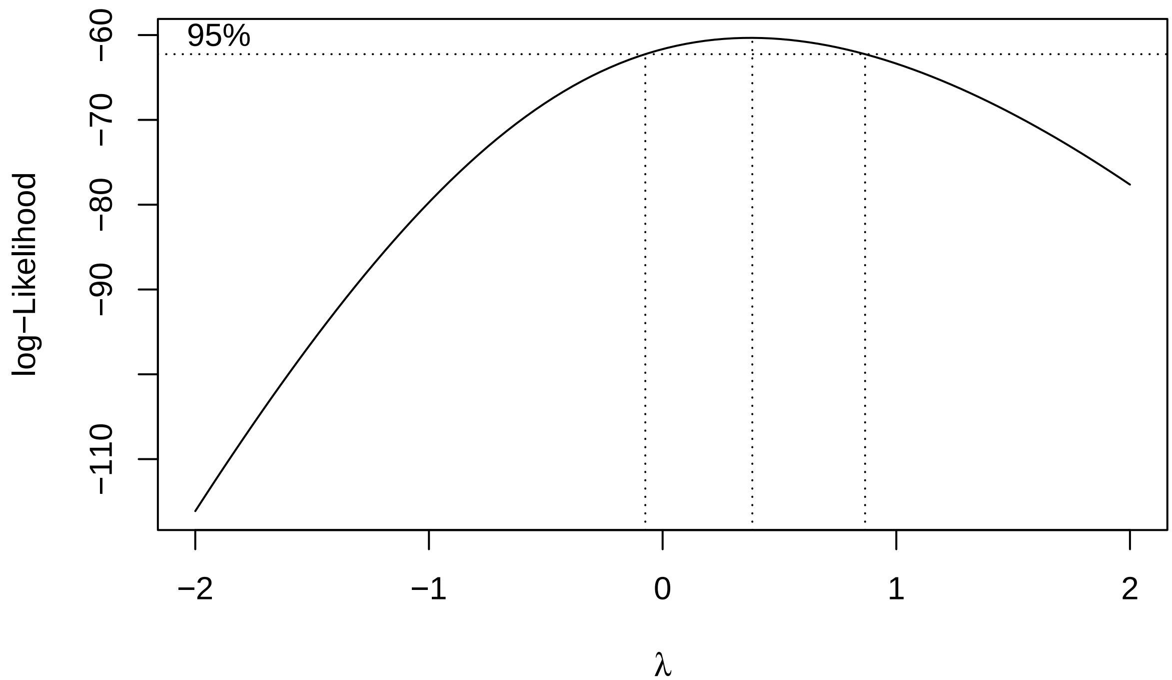
4.4.1 Transform 'df_crime_eval' as did for training dataset

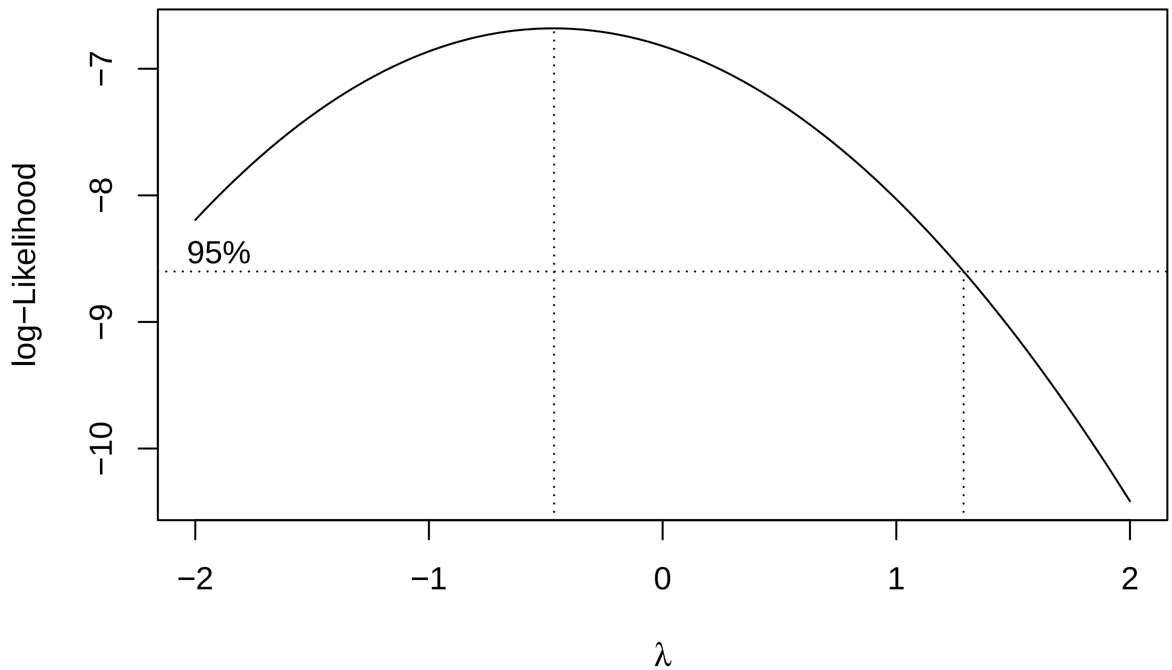
```r
summary(df_crime_eval)
```

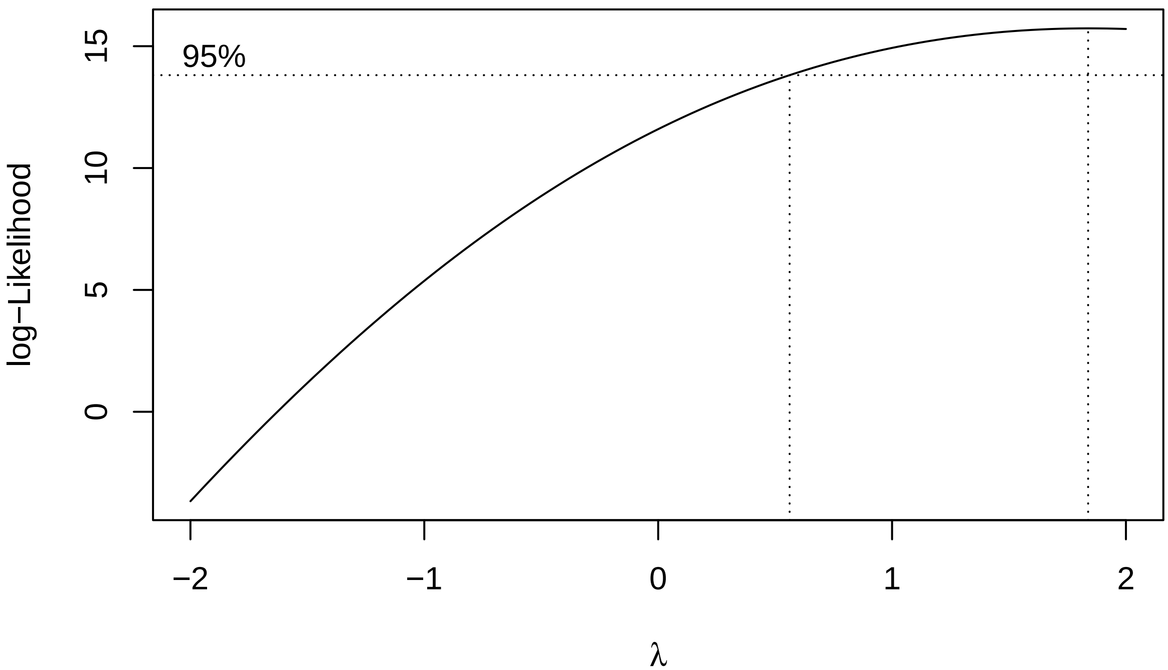
```
##         zn             indus             chas            nox
##  Min.   : 0.000   Min.   : 1.760   Min.   :0.00   Min.   :0.3850
##  1st Qu.: 0.000   1st Qu.: 5.692   1st Qu.:0.00   1st Qu.:0.4713
##  Median : 0.000   Median : 8.915   Median :0.00   Median :0.5380
##  Mean   : 8.875   Mean   :11.507   Mean   :0.05   Mean   :0.5592
##  3rd Qu.: 0.000   3rd Qu.:18.100   3rd Qu.:0.00   3rd Qu.:0.6258
##  Max.   :90.000   Max.   :25.650   Max.   :1.00   Max.   :0.7400
##         rm             age              dis             rad
##  Min.   :3.561   Min.   :  6.80   Min.   :1.202   Min.   : 1.000
##  1st Qu.:5.874   1st Qu.: 56.62   1st Qu.:2.041   1st Qu.: 4.000
```

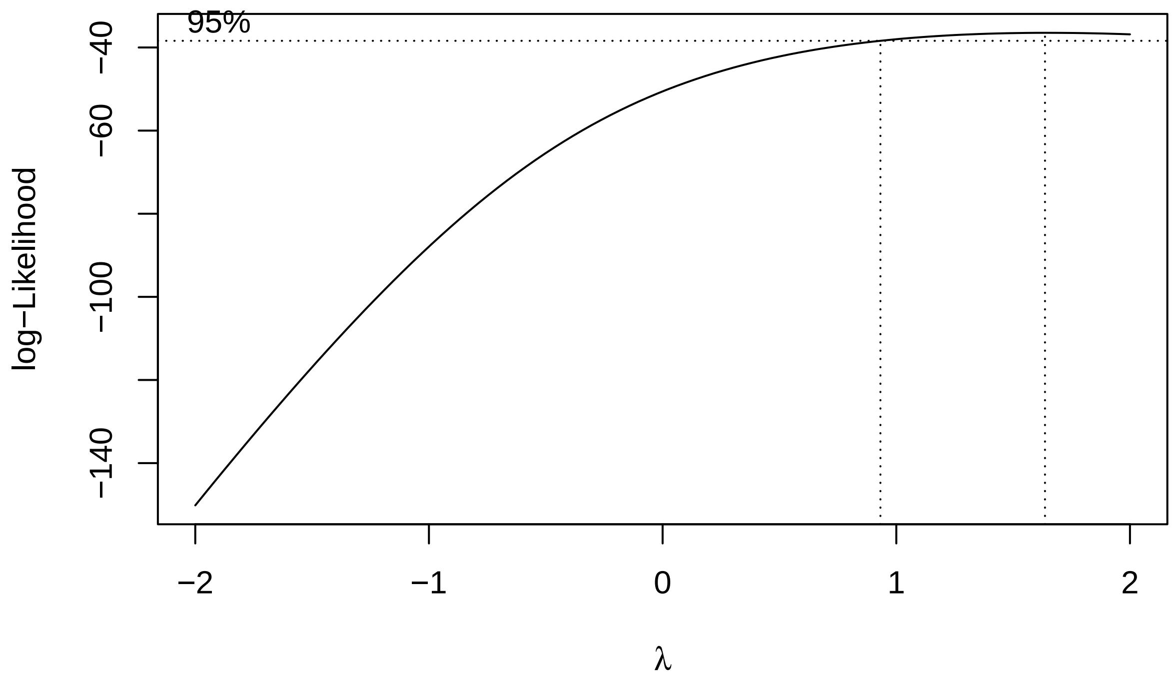
```
##  Median :6.143   Median : 83.25   Median :3.373   Median : 5.000
##  Mean   :6.214   Mean   : 70.99   Mean   :3.787   Mean   : 9.775
##  3rd Qu.:6.532   3rd Qu.: 93.10   3rd Qu.:4.527   3rd Qu.:24.000
##  Max.   :8.247   Max.   :100.00   Max.   :9.089   Max.   :24.000
##       tax           ptratio          lstat             medv
##  Min.   :188.0   Min.   :14.70   Min.   : 2.960   Min.   : 8.40
##  1st Qu.:276.8   1st Qu.:18.40   1st Qu.: 6.435   1st Qu.:16.98
##  Median :307.0   Median :19.60   Median :11.685   Median :20.55
##  Mean   :393.5   Mean   :19.12   Mean   :12.905   Mean   :21.88
##  3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:17.363   3rd Qu.:25.00
##  Max.   :666.0   Max.   :21.20   Max.   :34.020   Max.   :50.00
```

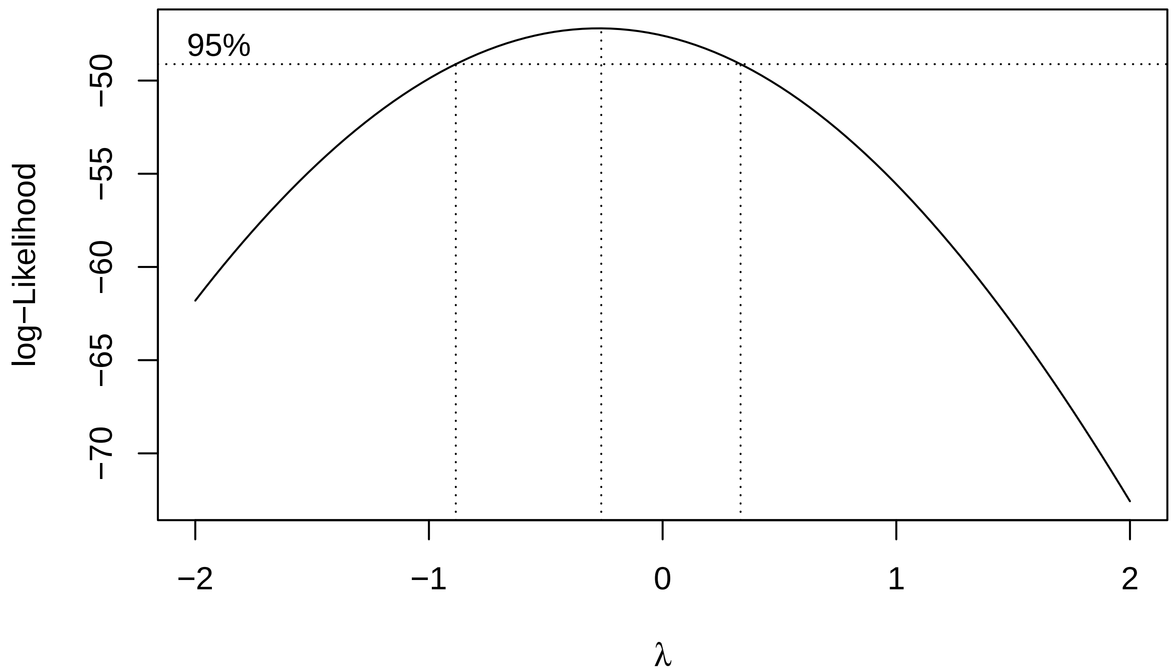
```r
# Create an empty list to store the transformed columns
col_transformed_eval <- list()

# Define the names of columns to exclude from transformation because there variables response must be p
col_exclude <- c("zn", "chas")

# Iterate through the columns in df_crime_eval
for (col_name in names(df_crime_eval)) {
  # Convert the column to a list and check if it's numeric and not in the exclude list
  if (is.numeric(df_crime_eval[[col_name]]) && !(col_name %in% col_exclude)) {
    col_list <- as.numeric(as.list(df_crime_eval[[col_name]]))

    # Find optimal lambda for Box-Cox transformation
    bc <- boxcox(col_list ~ 1, lambda = seq(-2, 2, 0.1))
    lambda_col <- bc$x[which.max(bc$y)]

    # Apply the Box-Cox transformation
    col_new <- ifelse(col_list==0, log(col_list), (col_list^lambda_col - 1) / lambda_col)

    # Store the transformed column in the list
    col_transformed_eval[[col_name]] <- col_new
  }
}
```

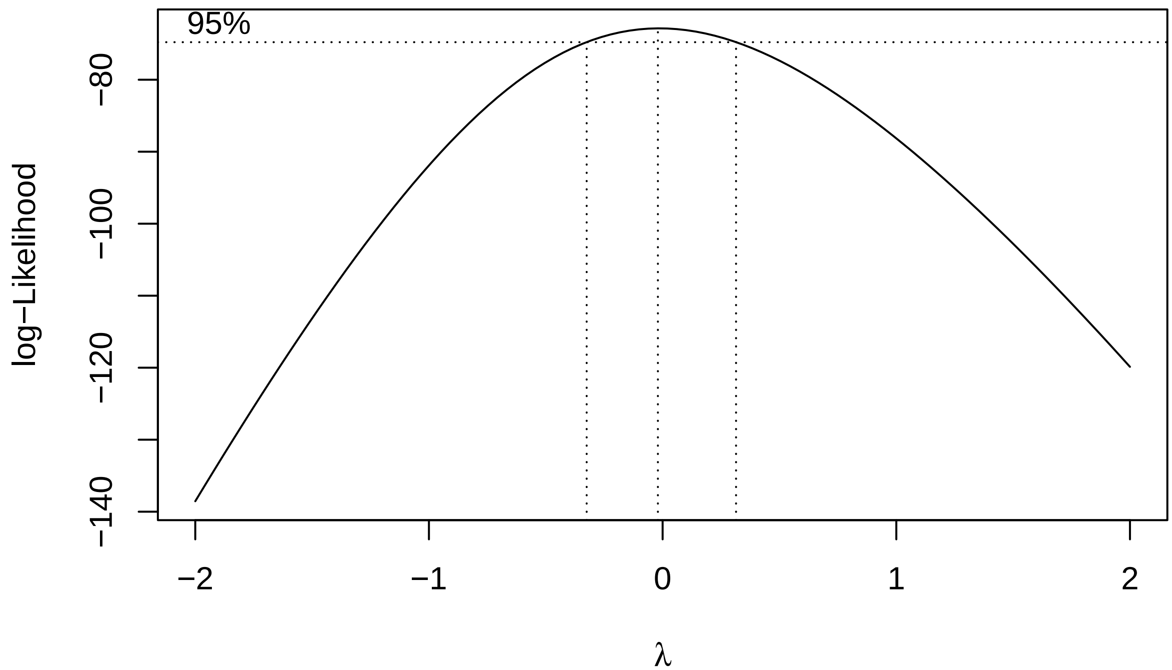

95%
log−Likelihood
−60
−70
−80
−90
−110
−2
−1
0
1
2
λ

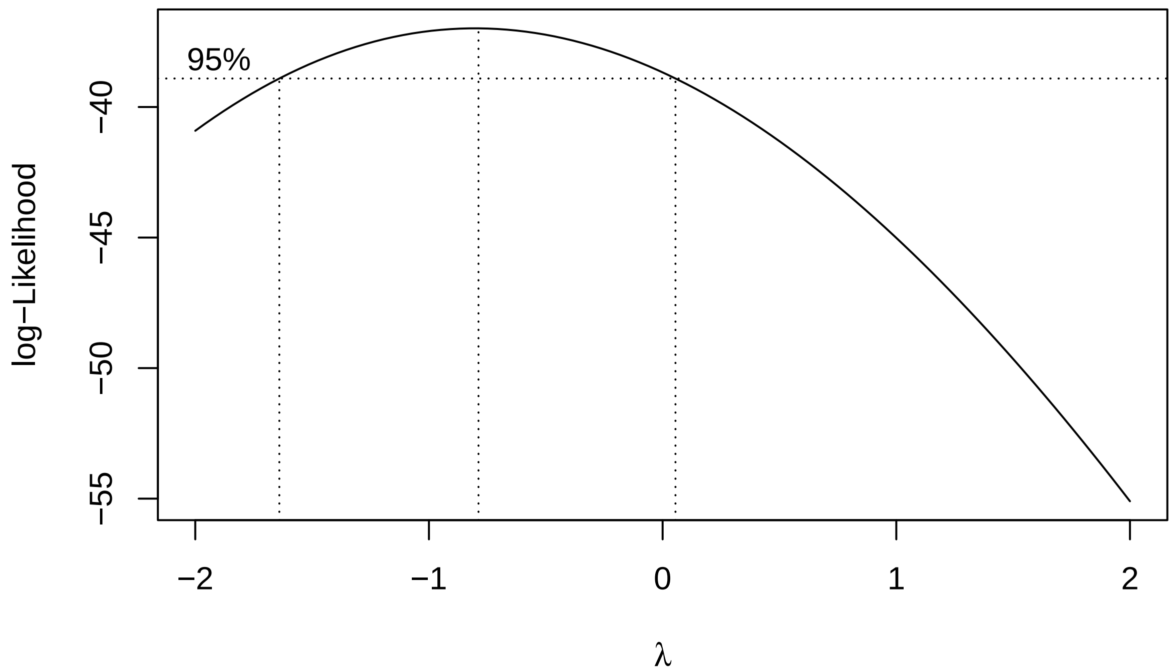

log−Likelihood
−7
−8
−9
−10
95%
−2
−1
0
1
2
λ

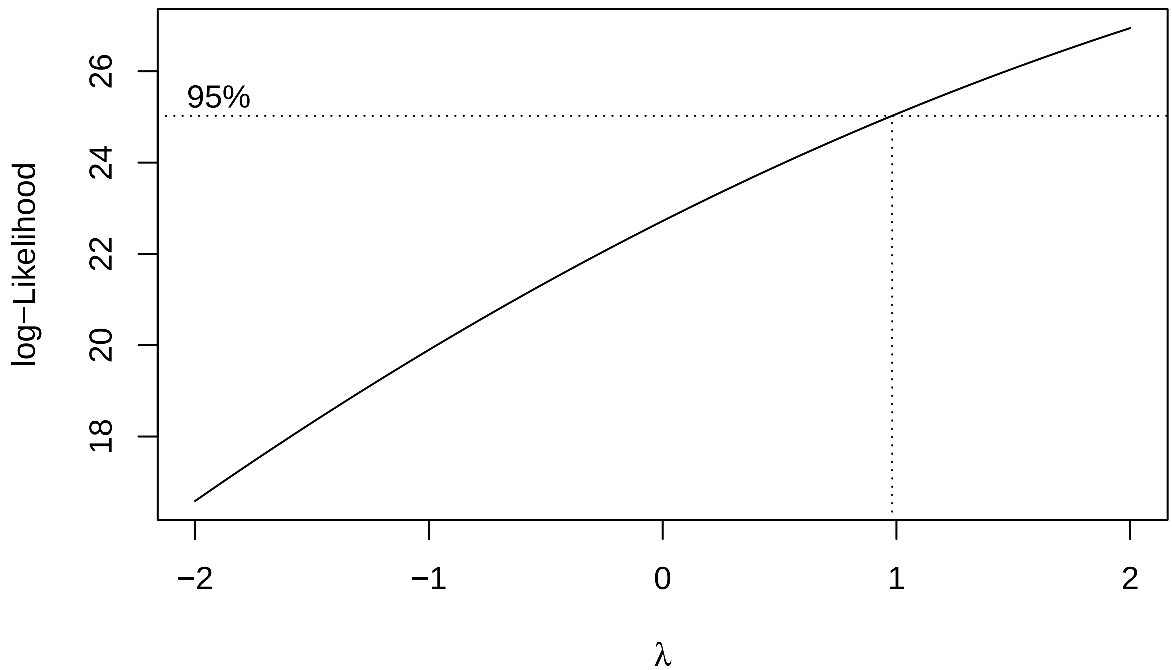

95%
log−Likelihood
15
10
5
0
−2
−1
0
1
2
λ


95%
−40
−60
−100
−140
log−Likelihood
−2
−1
0
1
2
λ

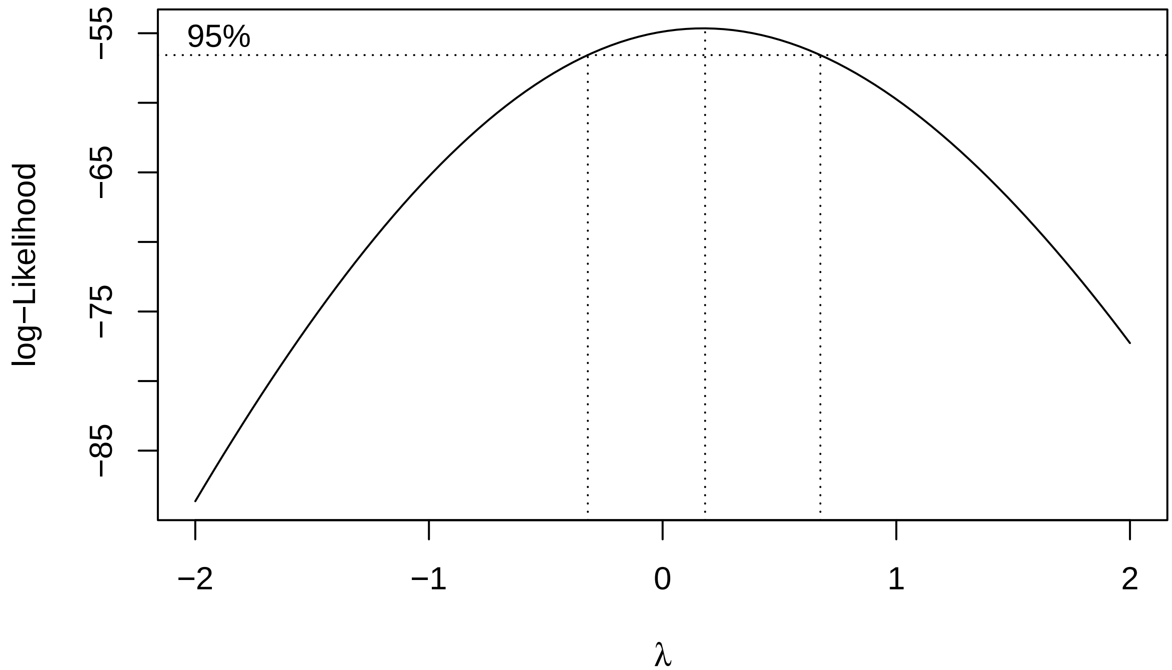

95%
log−Likelihood
−50
−55
−60
−65
−70
−2
−1
0
1
2
λ

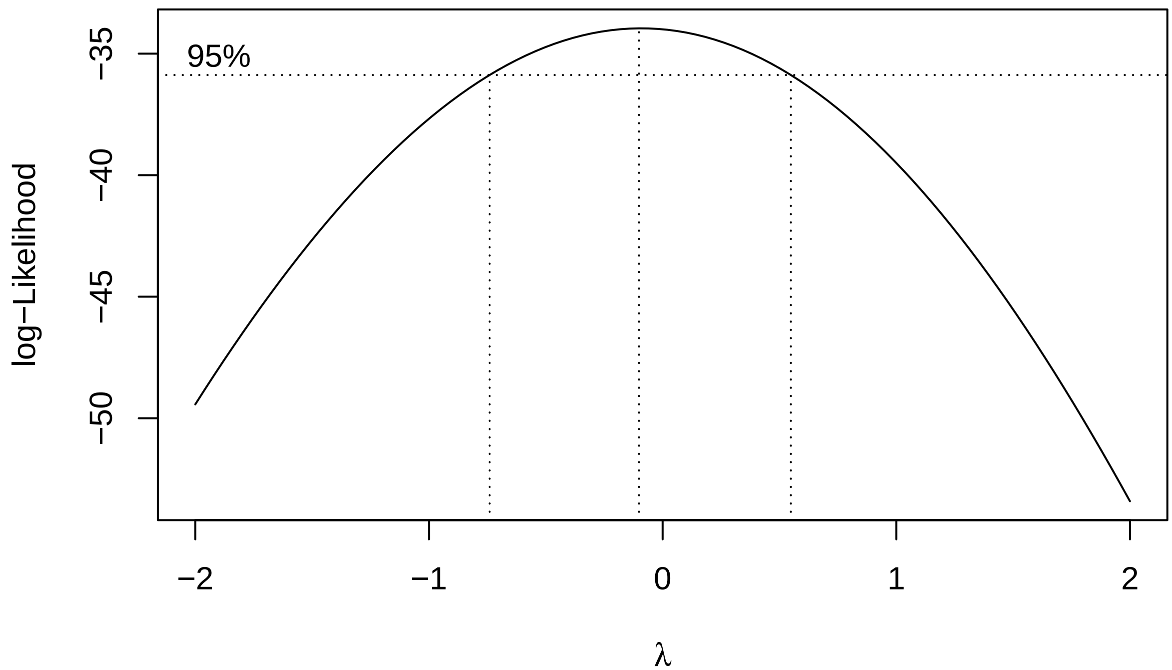

95%
log−Likelihood
−80
−100
−120
−140
−2
−1
0
1
2
λ


95%
log−Likelihood
−40
−45
−50
−55
−2
−1
0
1
2
λ



95%
log−Likelihood
18
20
22
24
26
−2
−1
0
1
2
λ


95%
−55
−65
−75
−85
log−Likelihood
−2
−1
0
1
2
λ




```
# Convert the list of transformed columns into a DataFrame
df_transformed_eval <- as.data.frame(col_transformed_eval)
```

```
df_crime_train_with_transformed_eval <- df_transformed_eval %>%
  dplyr::select(dis, lstat, medv, nox) %>%
  mutate(dis_t = dis, lstat_t = lstat, medv_t = medv, nox_t = nox)%>%
              dplyr::select(-c(dis, lstat, medv, nox))
```

```
# Combine data frames by adding columns
result_eval <- cbind(df_crime_train_with_transformed_eval, df_crime_eval %>%
              dplyr::select(-c(dis, lstat, medv, nox)))
```

```
# Apply min-max scaling to all three variables
df_scaled_eval <- result_eval
df_scaled_eval[] <- lapply(result_eval, rescale)
```

4.4.2 Make Prediction on transformed dataset using selected model from training dataset - Backwards Elimination model A6

```
df_scaled_eval$PRED = predict(A6_back_elim, new = df_scaled_eval, type="response")

df_scaled_eval$PRED_CLASS <- ifelse(df_scaled_eval$PRED > 0.5, 1, 0)
```

```r
table(df_scaled_eval$PRED_CLASS)
```

```
##
##  0  1
## 15 25
```

As shown above, we predict that 25 cases with crime above the median rate.