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

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

Last edited November 05, 2023

Contents

1 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)**

1.1 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.

1.2 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

2 Data Exploration

2.1 Load the data

```
df_crime_eval <-
  read.csv(paste0(git_url,"crime-evaluation-data_modified.csv"))
```

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

```
df_crime_train <-
  read.csv(paste0(git_url,"crime-training-data_modified.csv"))
```

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

2.2 Remove NA's

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

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

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

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

2.3 Summaries

```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
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
```

```
str(df_crime_train)
```

```
## 'data.frame':    466 obs. of  13 variables:
##  $ zn     : num  0 0 0 30 0 0 0 0 0 80 ...
##  $ indus  : num  19.58 19.58 18.1 4.93 2.46 ...
##  $ chas   : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.605 0.871 0.74 0.428 0.488 0.52 0.693 0.693 0.515 0.392 ...
##  $ rm     : num  7.93 5.4 6.49 6.39 7.16 ...
##  $ age    : num  96.2 100 100 7.8 92.2 71.3 100 100 38.1 19.1 ...
##  $ dis    : num  2.05 1.32 1.98 7.04 2.7 ...
##  $ rad    : int  5 5 24 6 3 5 24 24 5 1 ...
##  $ tax    : int  403 403 666 300 193 384 666 666 224 315 ...
##  $ ptratio: num  14.7 14.7 20.2 16.6 17.8 20.9 20.2 20.2 20.2 16.4 ...
##  $ lstat  : num  3.7 26.82 18.85 5.19 4.82 ...
##  $ medv   : num  50 13.4 15.4 23.7 37.9 26.5 5 7 22.2 20.9 ...
##  $ target : int  1 1 1 0 0 0 1 1 0 0 ...
```

Checking to see if data is categorical

```
# f<-xtabs(~ target + medv ,  data = df_crime_train)
# pander(f)
```

```
# Create a correlation matrix for all variables
(cor_matrix <- 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
```

The logistic regression model dependant variable target has

Changing categorical data into factors to ensure that the model can appropriately interpret and analyze categorical variables.Change after looking at collinearity.

```r
#Don't use this
#df_crime_train$chas <- as.factor(df_crime_train$chas)
#df_crime_train$rad <- as.factor(df_crime_train$rad)
#df_crime_train$target <- as.factor(df_crime_train$target)
```

```r
str(df_crime_train)
```

```
## 'data.frame':    466 obs. of  13 variables:
##  $ zn     : num  0 0 0 30 0 0 0 0 0 80 ...
##  $ indus  : num  19.58 19.58 18.1 4.93 2.46 ...
##  $ chas   : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.605 0.871 0.74 0.428 0.488 0.52 0.693 0.693 0.515 0.392 ...
##  $ rm     : num  7.93 5.4 6.49 6.39 7.16 ...
##  $ age    : num  96.2 100 100 7.8 92.2 71.3 100 100 38.1 19.1 ...
##  $ dis    : num  2.05 1.32 1.98 7.04 2.7 ...
##  $ rad    : int  5 5 24 6 3 5 24 24 5 1 ...
##  $ tax    : int  403 403 666 300 193 384 666 666 224 315 ...
##  $ ptratio: num  14.7 14.7 20.2 16.6 17.8 20.9 20.2 20.2 20.2 16.4 ...
##  $ lstat  : num  3.7 26.82 18.85 5.19 4.82 ...
##  $ medv   : num  50 13.4 15.4 23.7 37.9 26.5 5 7 22.2 20.9 ...
##  $ target : int  1 1 1 0 0 0 1 1 0 0 ...
```

```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)
```

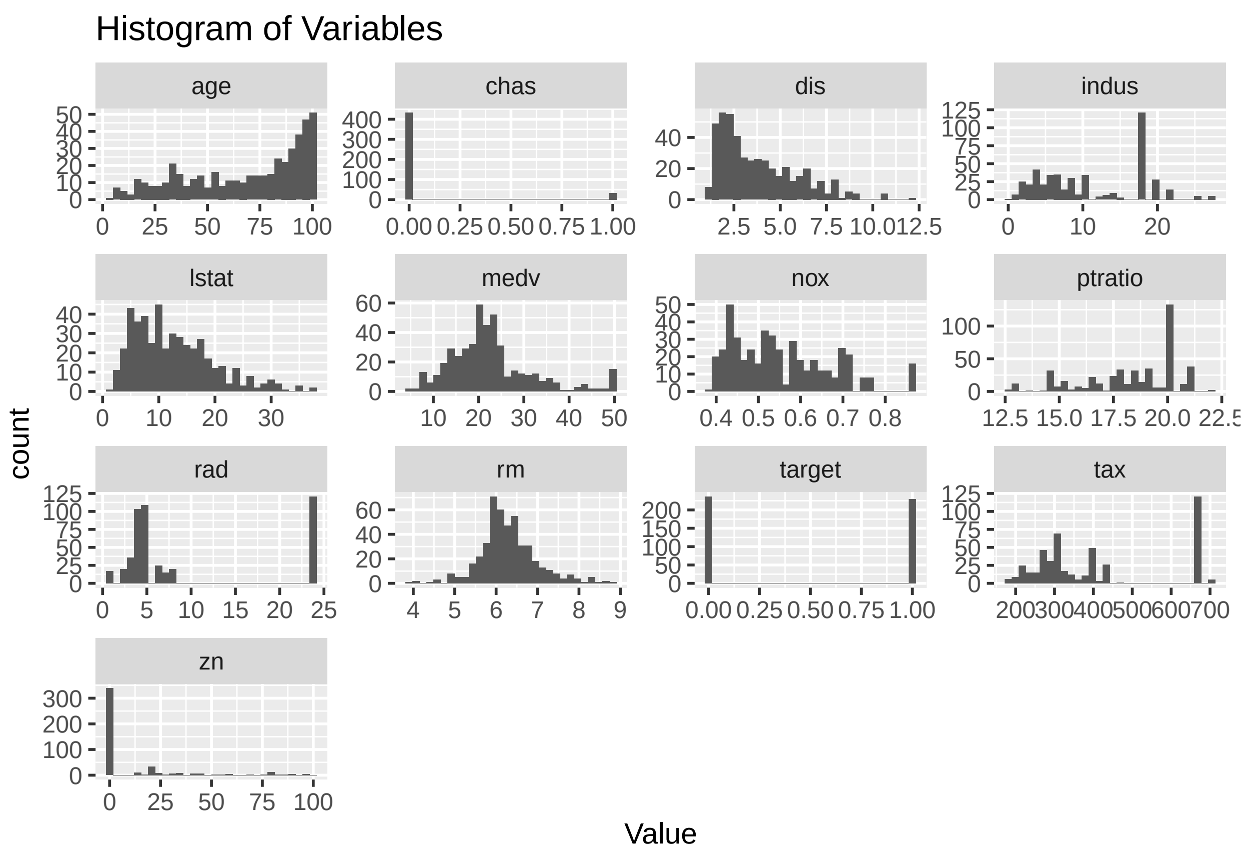
```
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -40.822934   6.632913  -6.155 7.53e-10 ***
## zn           -0.065946   0.034656  -1.903  0.05706 .
## indus        -0.064614   0.047622  -1.357  0.17485
## chas          0.910765   0.755546   1.205  0.22803
## 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
```

```r
print(xtable(model_1,type="latex", comment=FALSE))
```

```
## % latex table generated in R 4.3.1 by xtable 1.8-4 package
## % Sun Nov  5 19:35:44 2023
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrr}
##   \hline
##  & Estimate & Std. Error & z value & Pr($>$$|$z$|$) \\
##   \hline
## (Intercept) & -40.8229 & 6.6329 & -6.15 & 0.0000 \\
##   zn & -0.0659 & 0.0347 & -1.90 & 0.0571 \\
##   indus & -0.0646 & 0.0476 & -1.36 & 0.1748 \\
##   chas & 0.9108 & 0.7555 & 1.21 & 0.2280 \\
##   nox & 49.1223 & 7.9317 & 6.19 & 0.0000 \\
##   rm & -0.5875 & 0.7228 & -0.81 & 0.4164 \\
##   age & 0.0342 & 0.0138 & 2.47 & 0.0133 \\
##   dis & 0.7387 & 0.2303 & 3.21 & 0.0013 \\
##   rad & 0.6664 & 0.1632 & 4.08 & 0.0000 \\
##   tax & -0.0062 & 0.0030 & -2.09 & 0.0367 \\
##   ptratio & 0.4026 & 0.1266 & 3.18 & 0.0015 \\
##   lstat & 0.0459 & 0.0540 & 0.85 & 0.3961 \\
##   medv & 0.1808 & 0.0683 & 2.65 & 0.0081 \\
##    \hline
## \end{tabular}
## \end{table}
```

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

ggplot(data_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`.
```



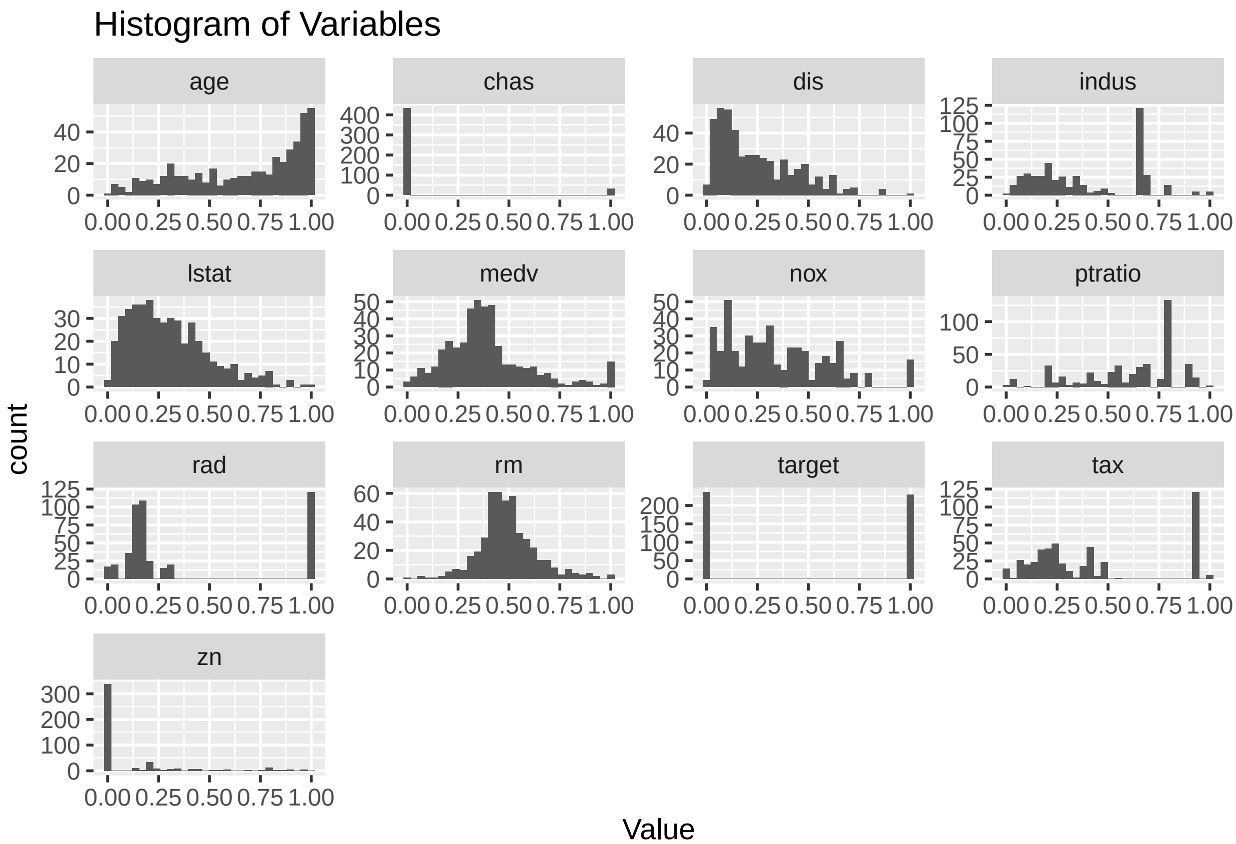
Variable here are not normalized and those normalized need to be on the same scale as the others to make data more interpretable.

******** I need to rescale after normalizing.

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

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

ggplot(data_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
(cor_matrix <- cor(data_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
```

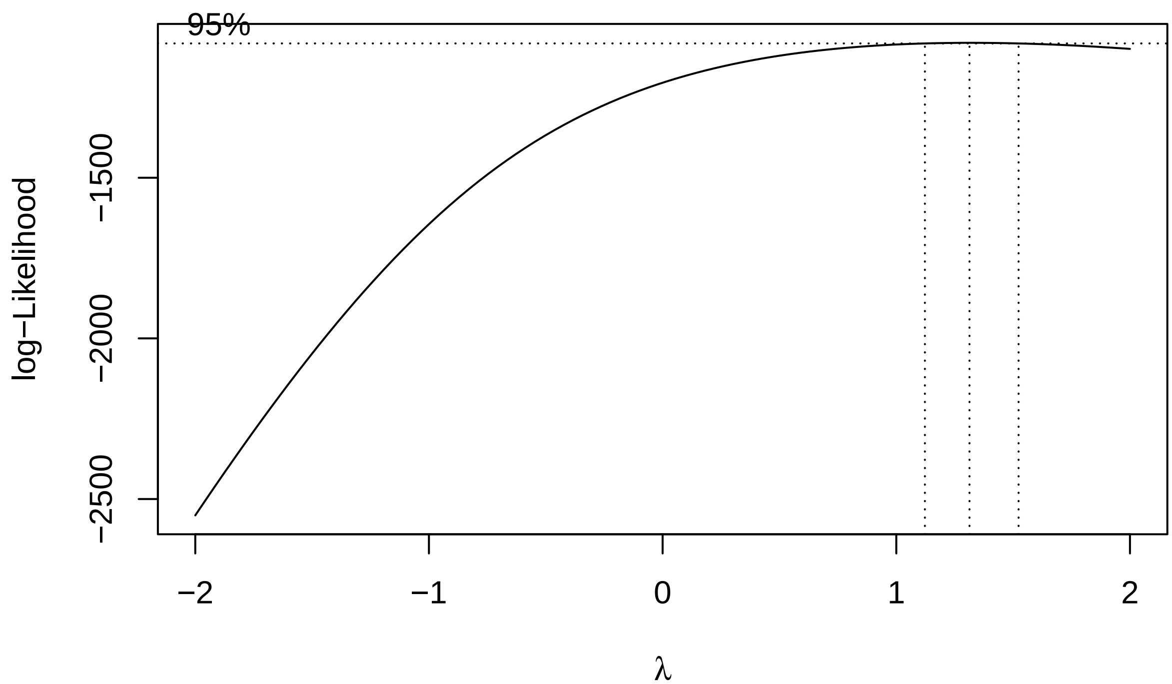
```
## rm       0.19901584 -0.20844570 -0.29693430 -0.3603471 -0.63202445  0.7053368
## age     -0.75089759  0.46031430  0.51212452  0.2554479  0.60562001 -0.3781560
## dis      1.00000000 -0.49499193 -0.53425464 -0.2333394 -0.50752800  0.2566948
## rad     -0.49499193  1.00000000  0.90646323  0.4714516  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
```

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

(summary=(model_2))
```

```
##
## Call:  glm(formula = target ~ ., family = binomial, data = data_scaled)
##
## Coefficients:
## (Intercept)           zn        indus         chas          nox           rm
##    -17.5119      -6.5946      -1.7627       0.9108      23.6769      -2.8887
##         age          dis          rad          tax      ptratio        lstat
##      3.3197       8.1230      15.3264      -3.2338       3.7841       1.6623
##        medv
##      8.1371
##
## Degrees of Freedom: 465 Total (i.e. Null);  453 Residual
## Null Deviance:       645.9
## Residual Deviance: 192   AIC: 218
```

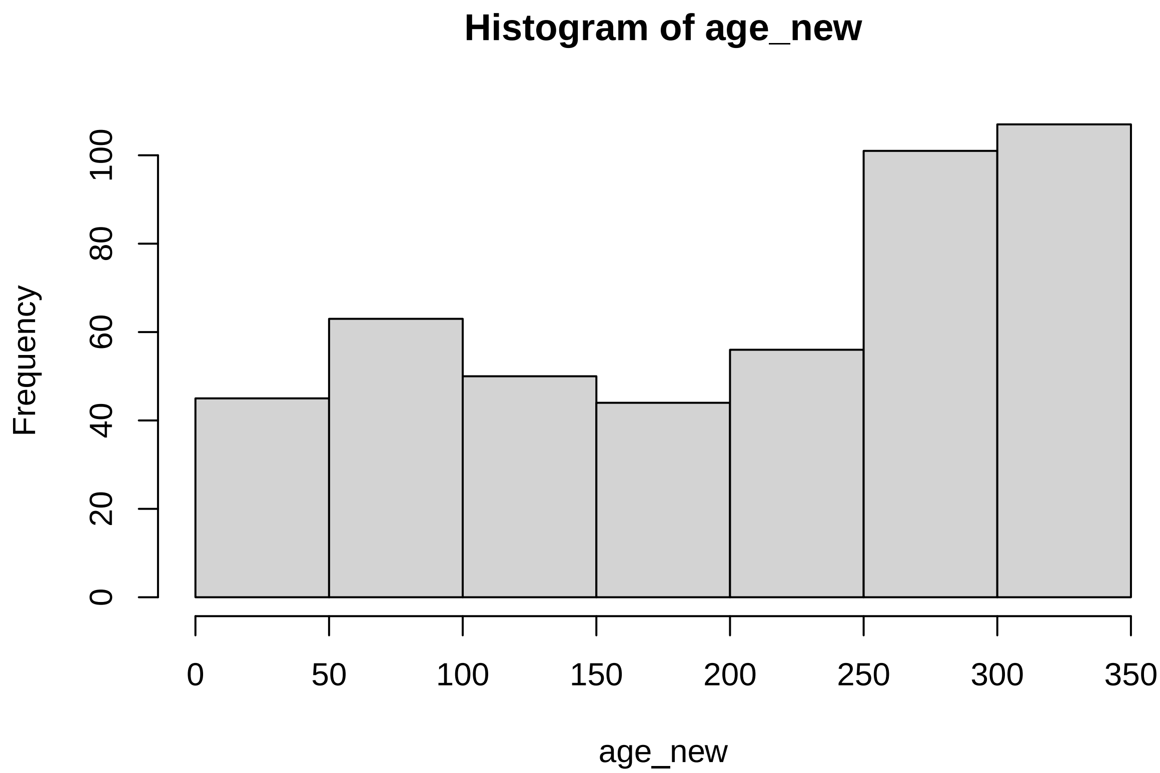
```r
df_crime_train$age <- as.numeric(df_crime_train$age)
```

```r
# Convert a DataFrame column to a list
age_list <- as.numeric(as.list(df_crime_train$age))

#find optimal lambda for Box-Cox transformation
bc <- boxcox(age_list~ 1, lambda = seq(-2,2,0.1))
```

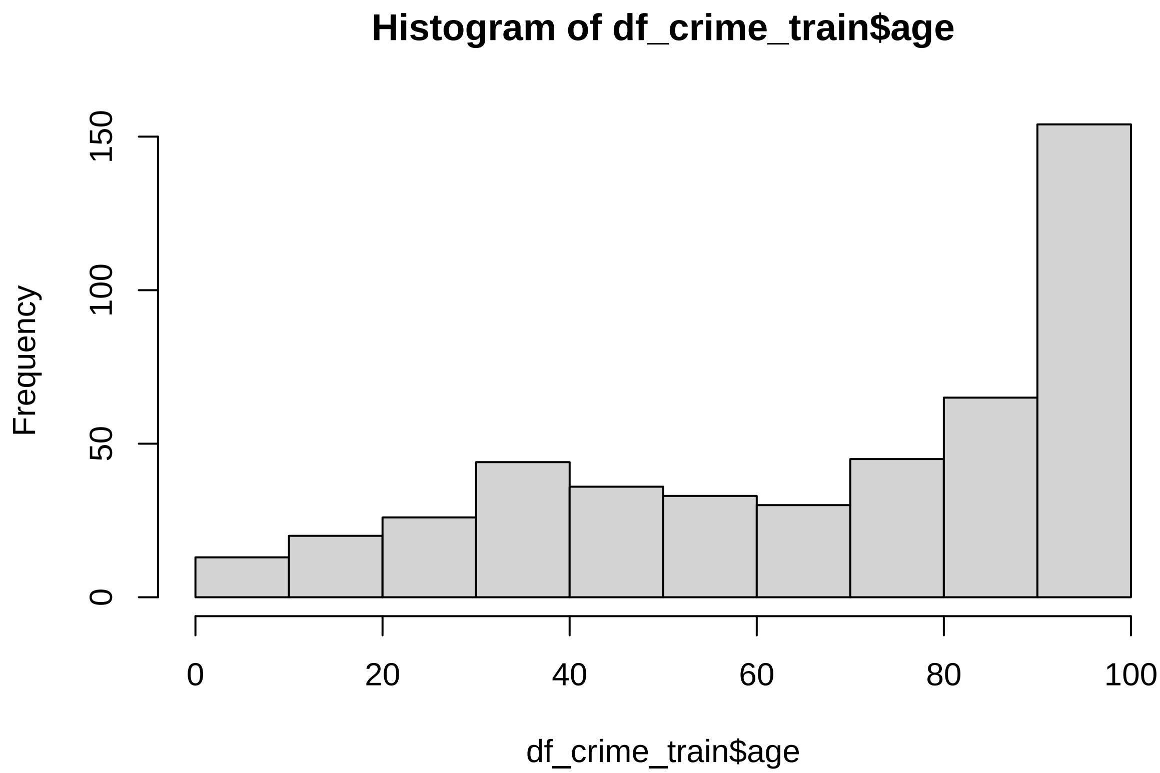



```r
lambda <- bc$x[which.max(bc$y)]

# Apply the Box-Cox transformation
age_new = (age_list^lambda-1)/lambda
```

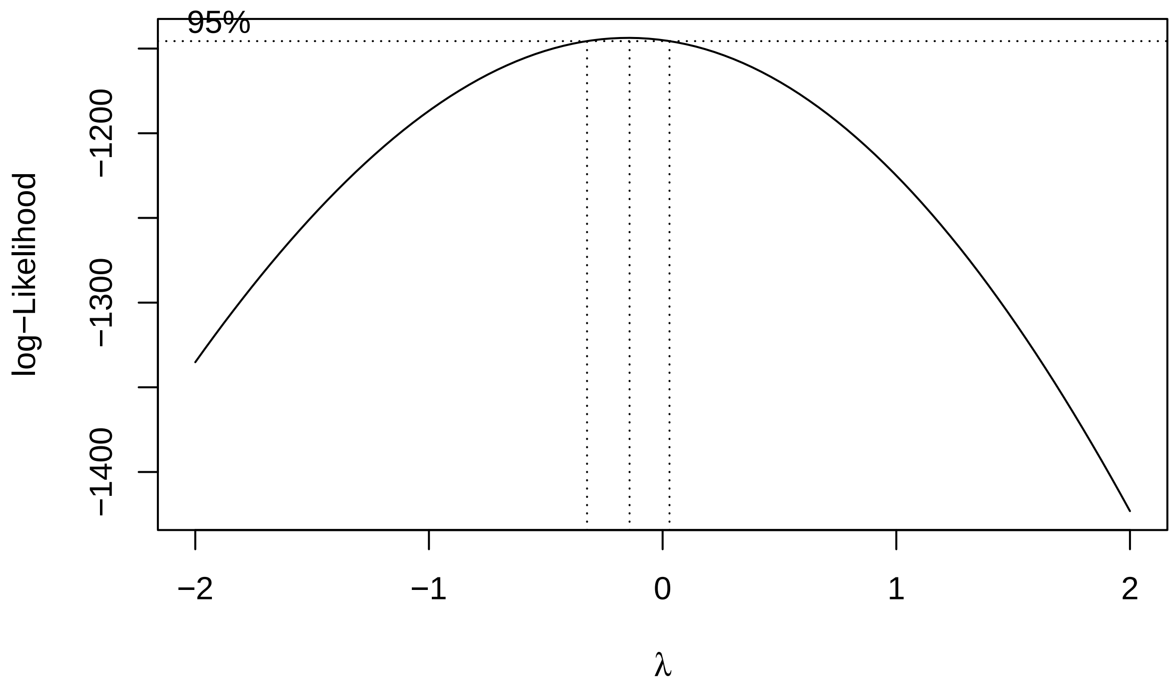
```r
hist(age_new)
```




```r
hist(df_crime_train$age)
```

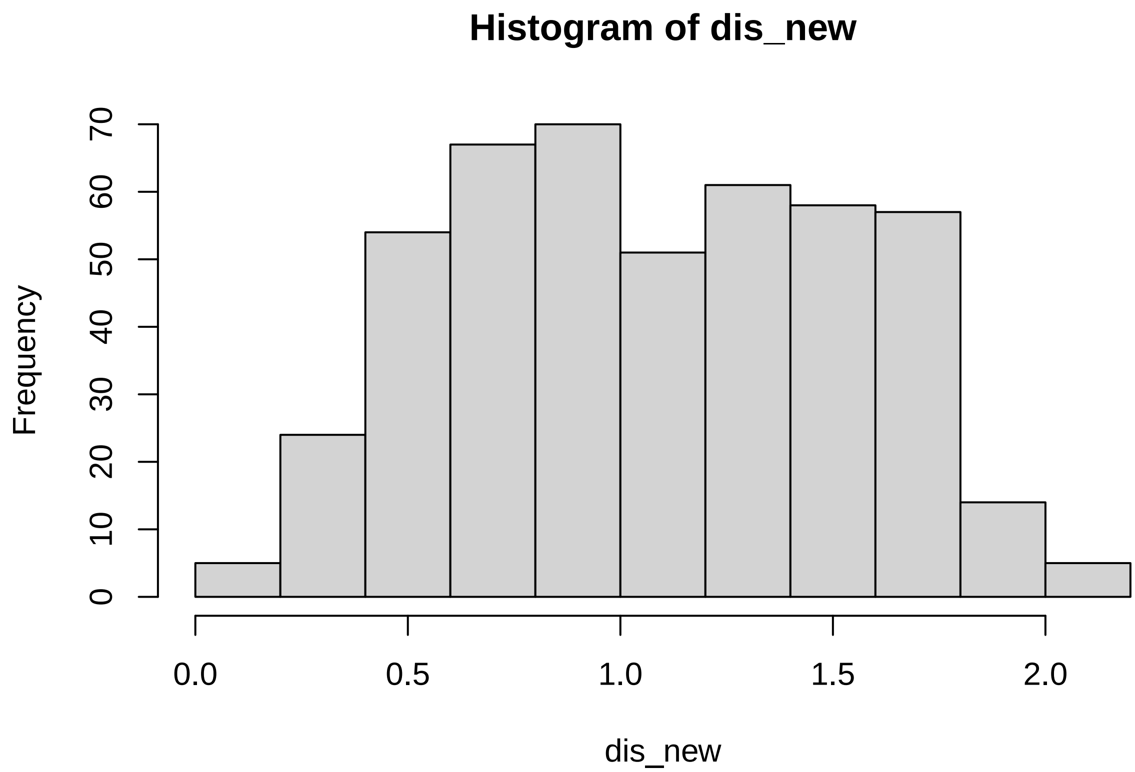



```
# Convert a DataFrame column to a list
dis_list <- as.numeric(as.list(df_crime_train$dis))

#find optimal lambda for Box-Cox transformation
bc <- boxcox(dis_list~ 1, lambda = seq(-2,2,0.1))
```

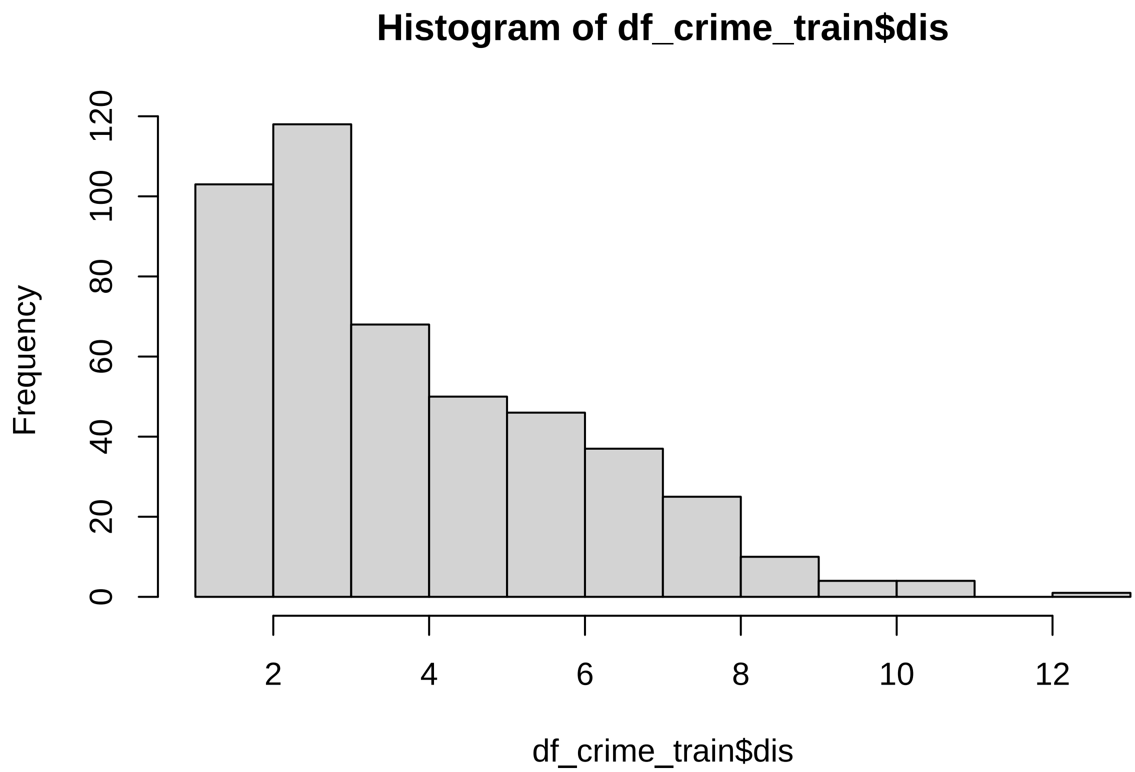



```
lambda_dis <- bc$x[which.max(bc$y)]

# Apply the Box-Cox transformation
dis_new = (dis_list^lambda_dis-1)/lambda_dis
```

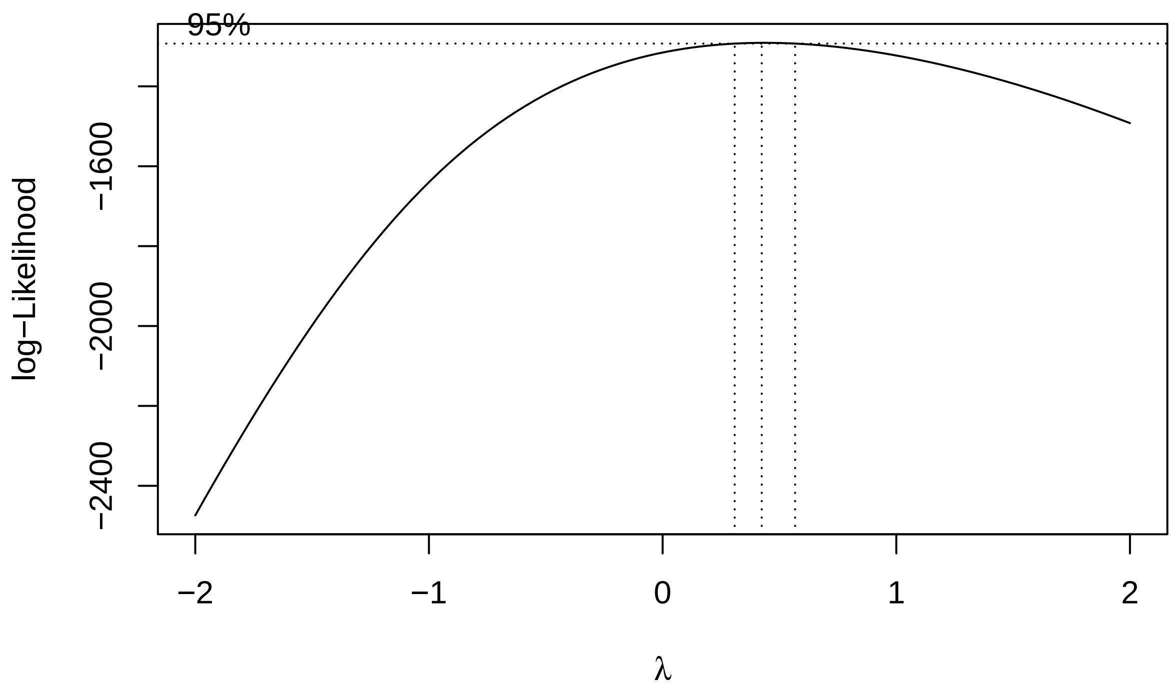
```
hist(dis_new)
```



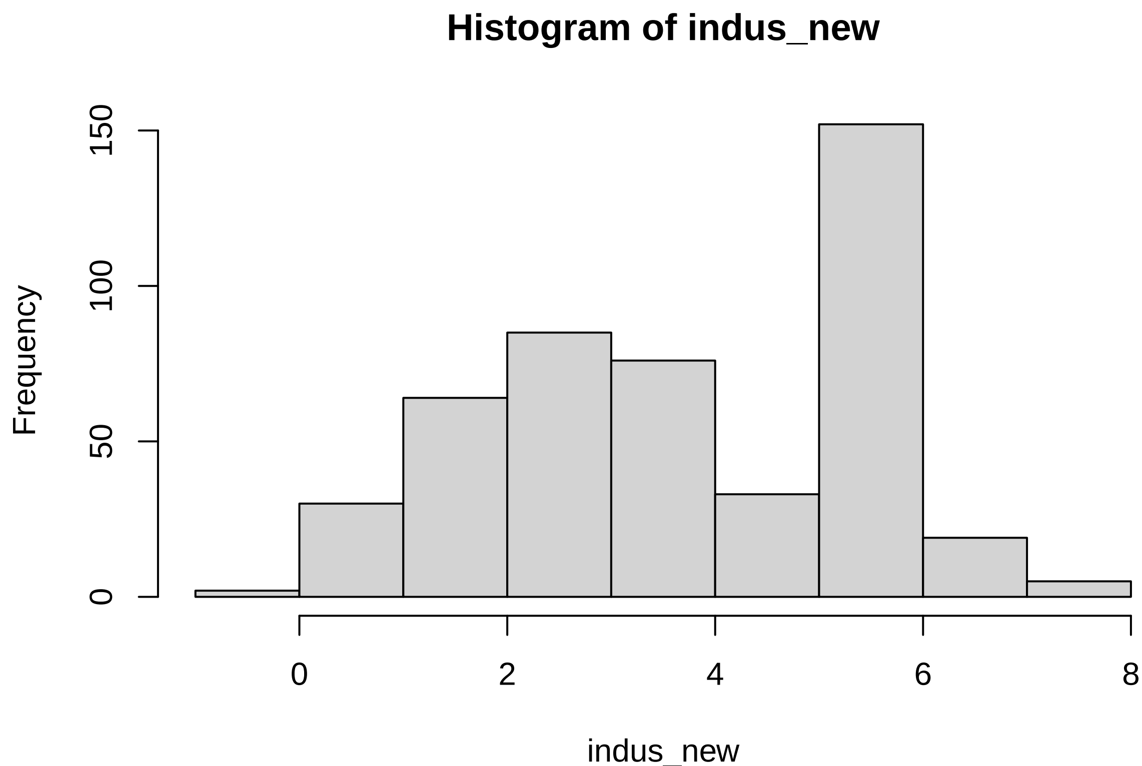

```
hist(df_crime_train$dis)
```



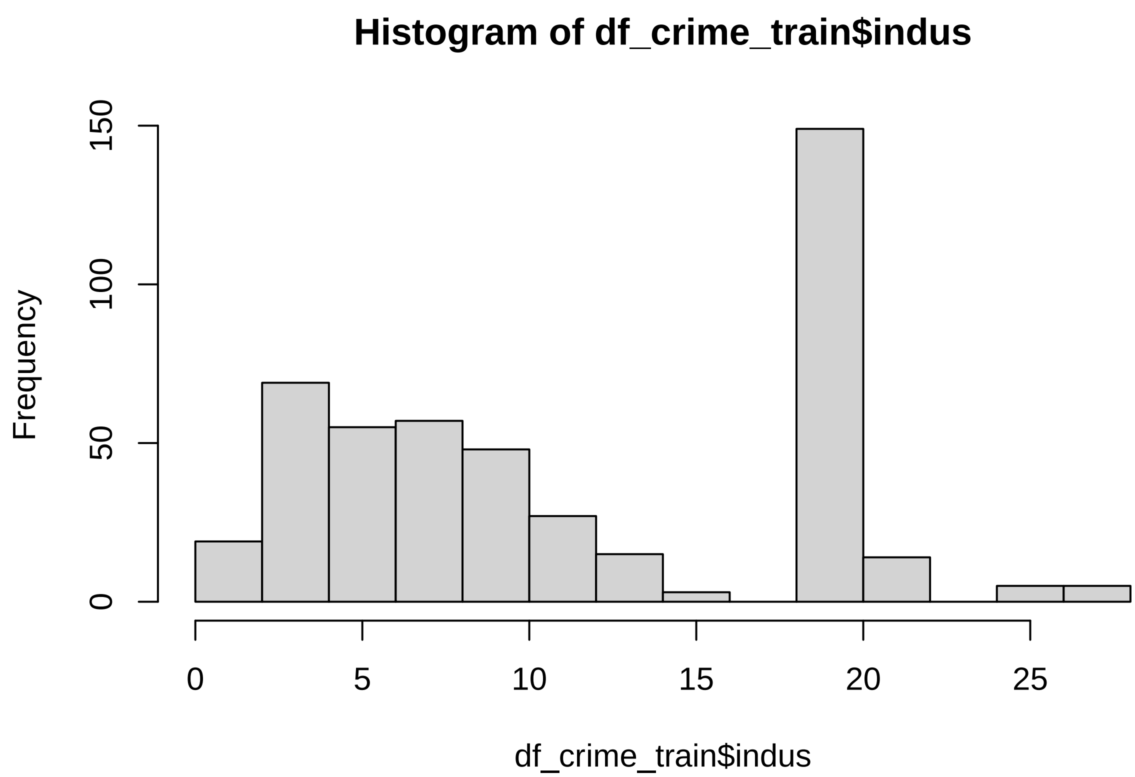

```r
# Convert a DataFrame column to a list
indus_list <- as.numeric(as.list(df_crime_train$indus))

#find optimal lambda for Box-Cox transformation
bc <- boxcox(indus_list~ 1, lambda = seq(-2,2,0.1))
```



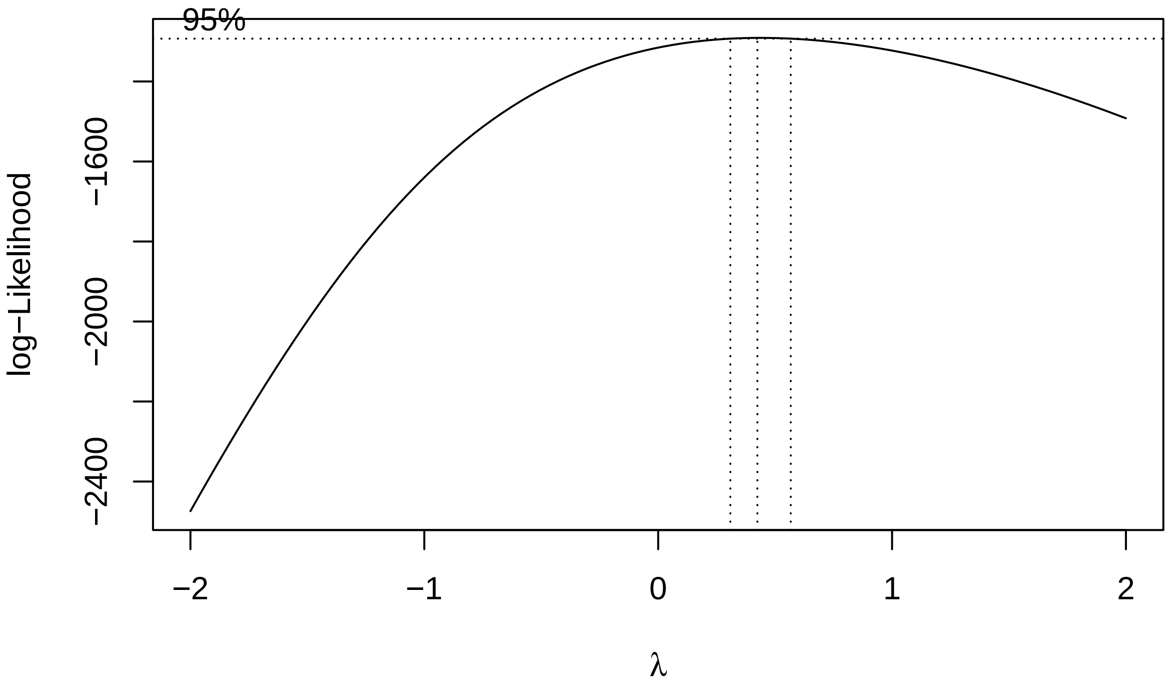

```
lambda_indus <- bc$x[which.max(bc$y)]

# Apply the Box-Cox transformation
indus_new = (indus_list^lambda_indus-1)/lambda_indus


hist(indus_new )
```

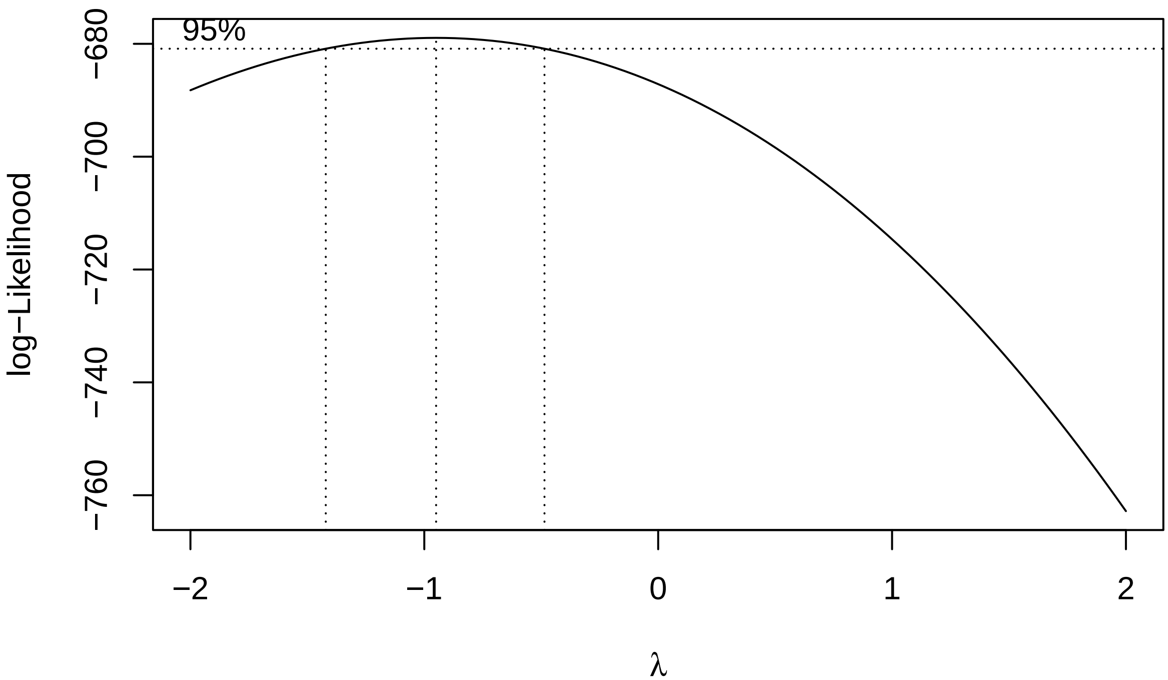


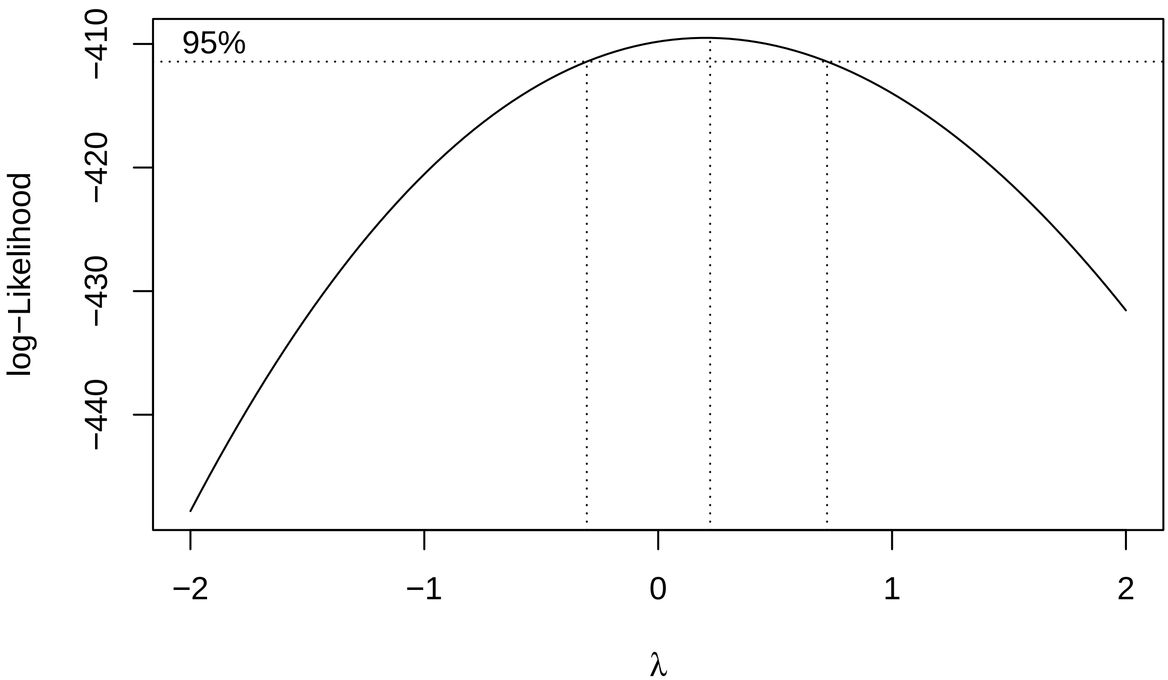

```
hist(df_crime_train$indus)
```

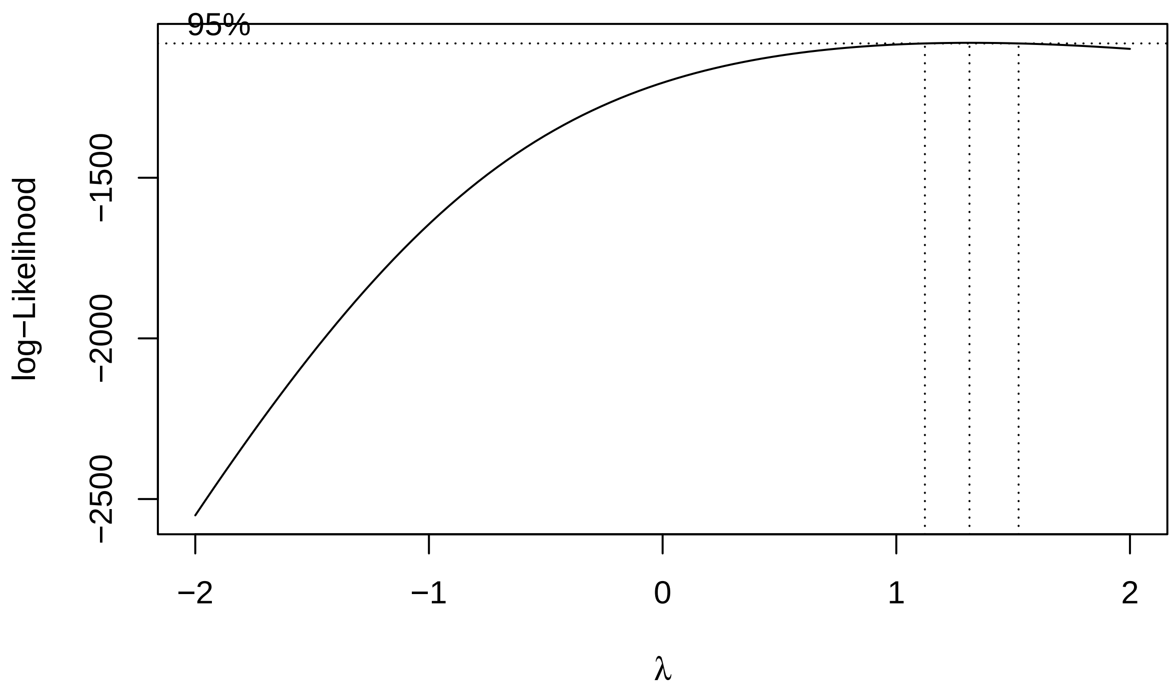


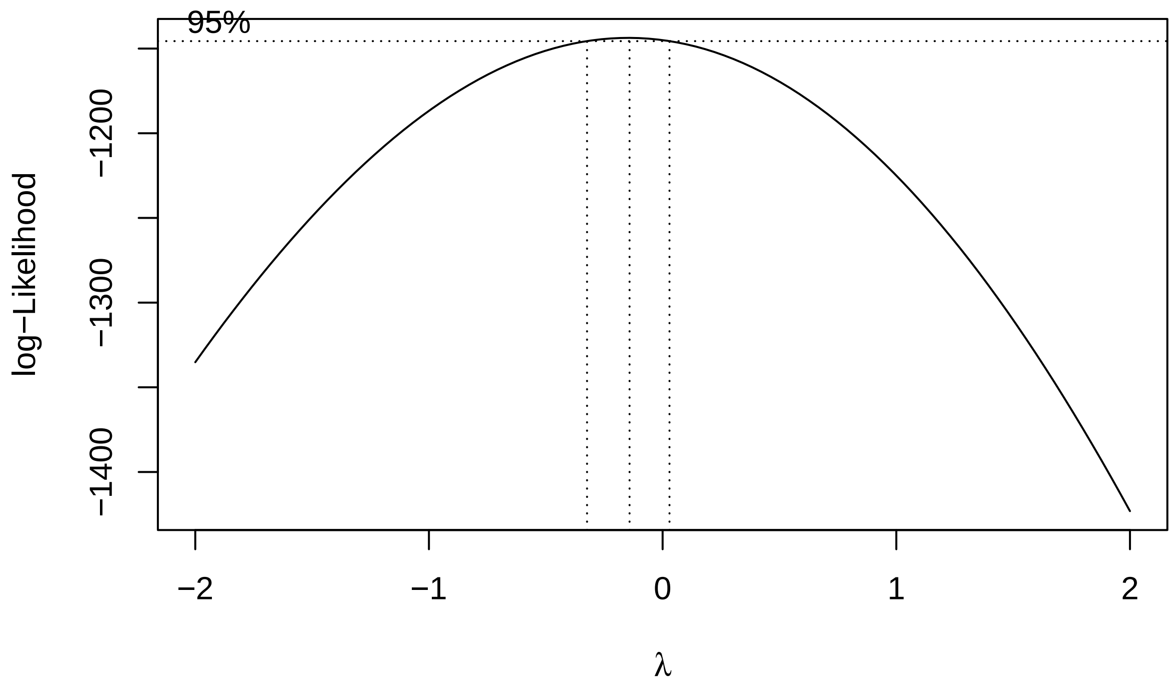

```r
# Create an empty list to store the transformed columns
transformed_columns <- list()

# Define the names of columns to exclude from transformation because there variables response must be p
exclude_columns <- 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% exclude_columns)) {
    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 <- (col_list^lambda_col - 1) / lambda_col

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


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

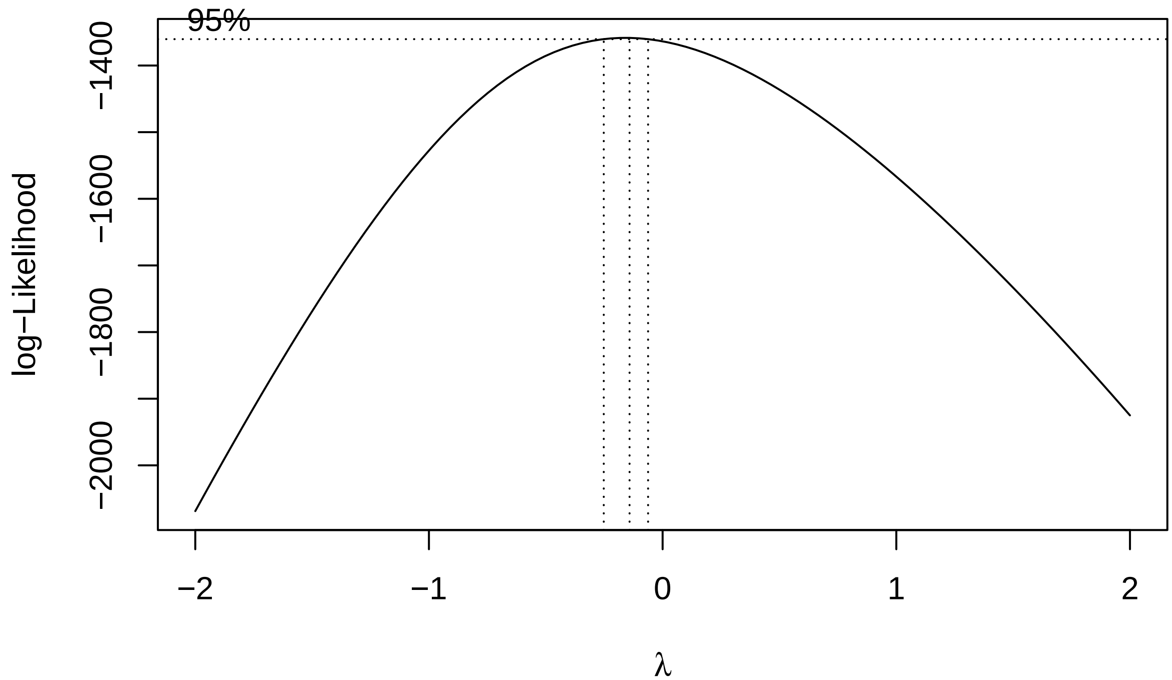

95%
log−Likelihood
−680
−700
−720
−740
−760
−2
−1
0
1
2
λ

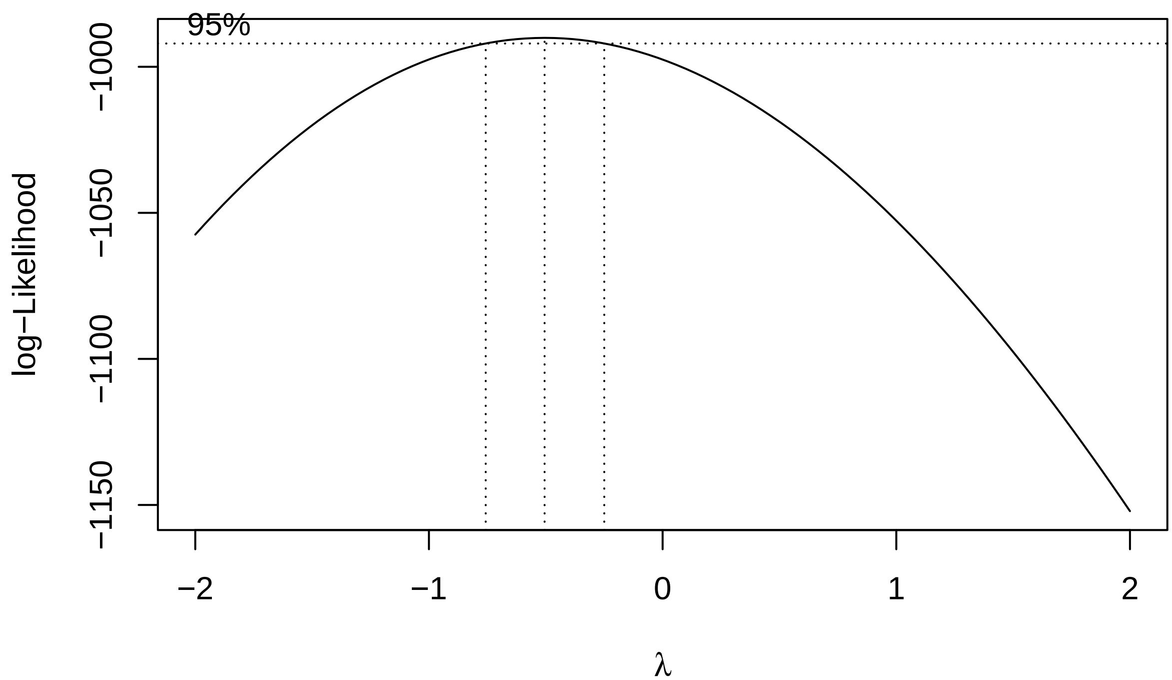

95%
−410
−420
−430
−440
log−Likelihood
−2
−1
0
1
2
λ

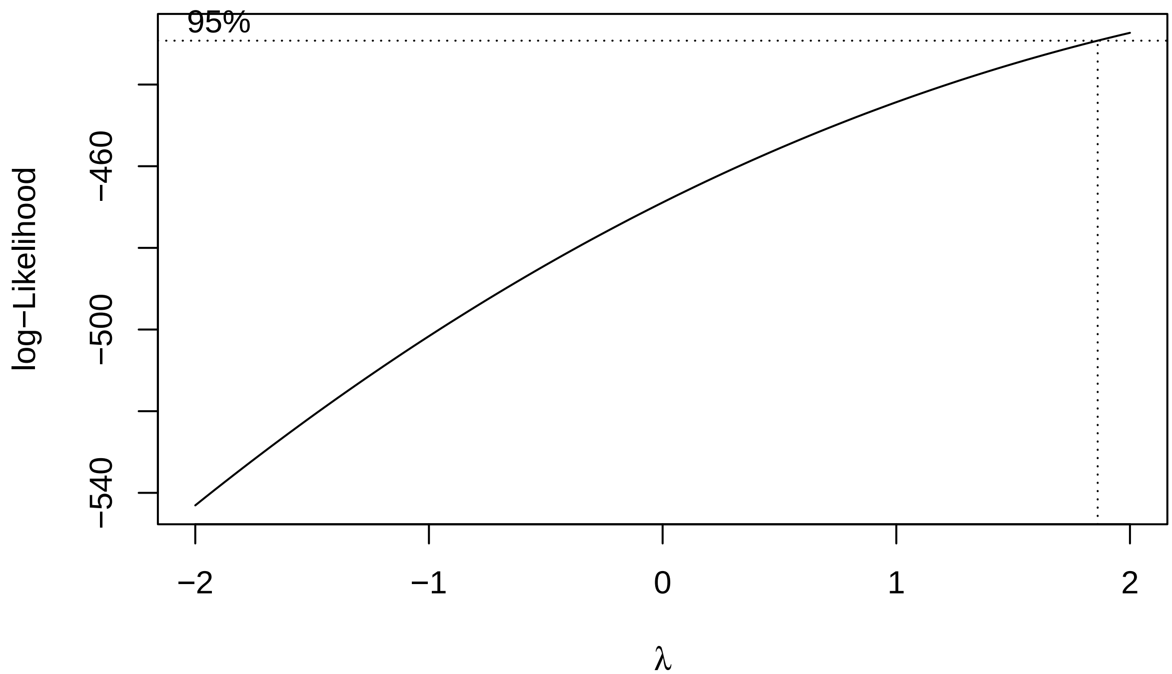

95%
log−Likelihood
−1500
−2000
−2500
−2
−1
0
1
2
λ

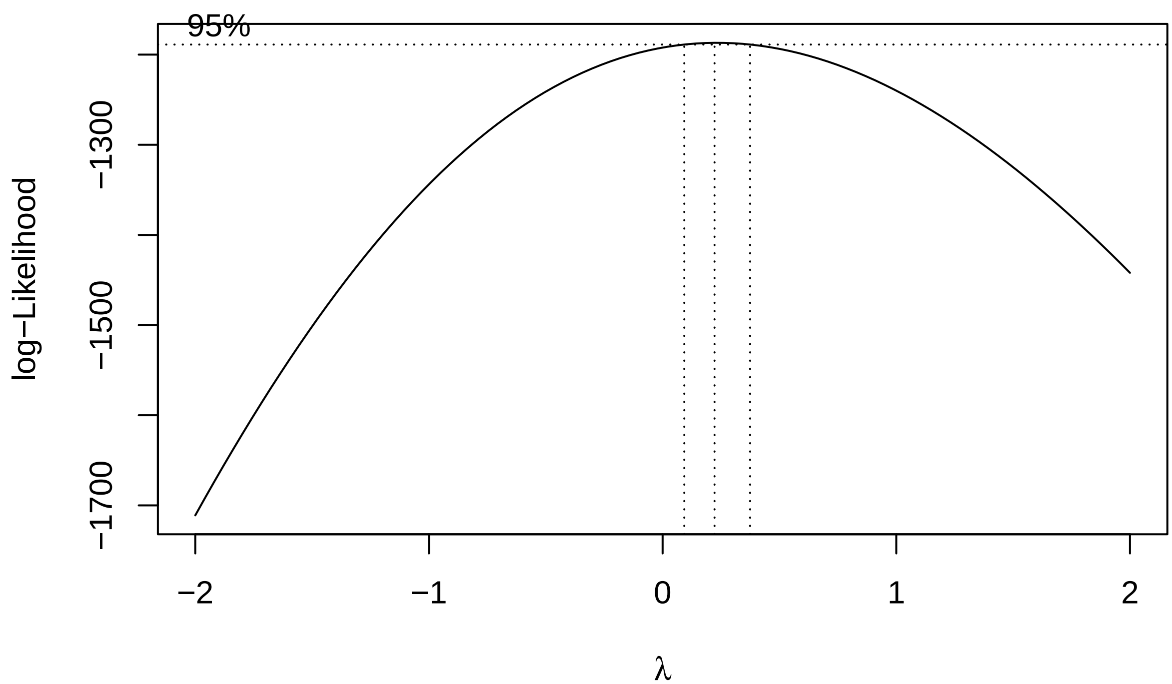


95%
−1200
−1300
−1400
log−Likelihood
−2
−1
0
1
2
λ

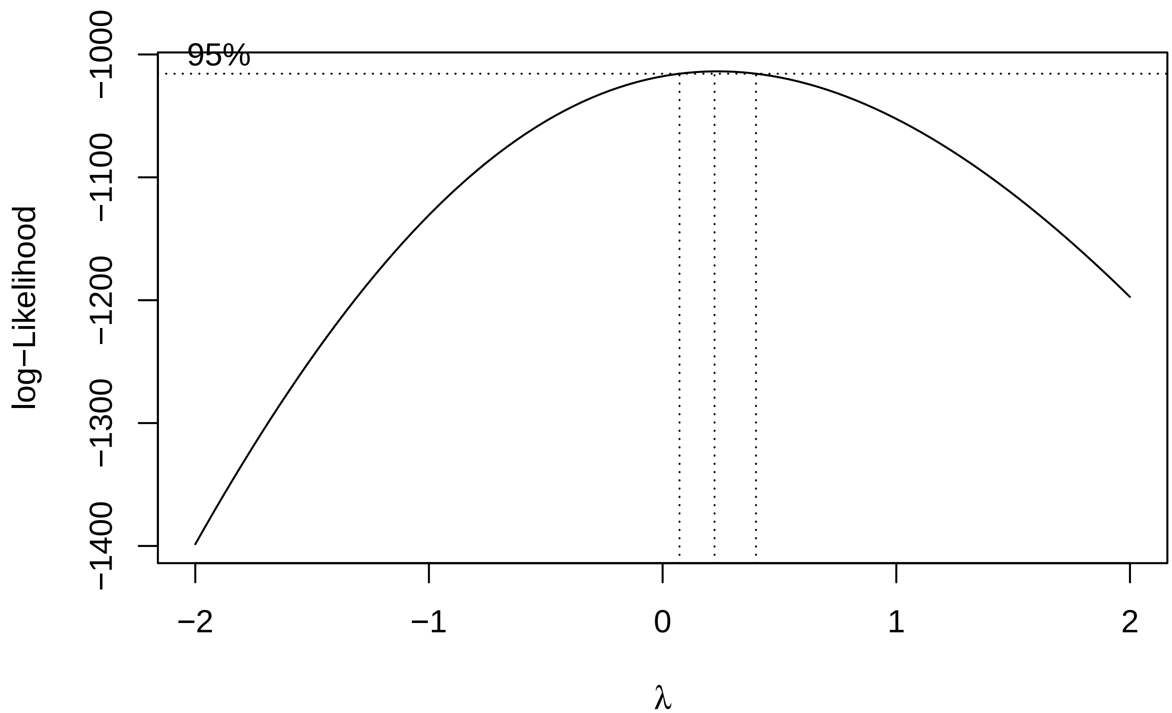

95%
log−Likelihood
−1400
−1600
−1800
−2000
−2
−1
0
1
2
λ


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



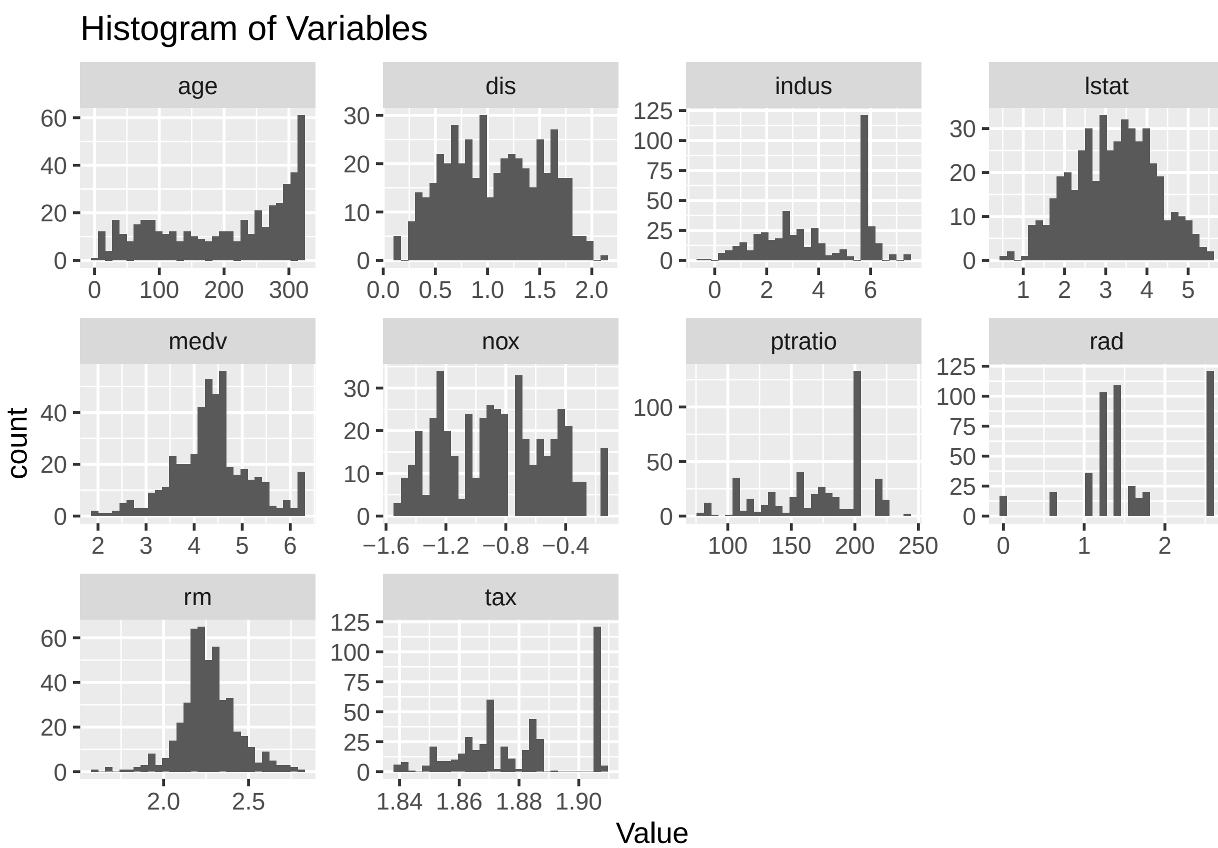
95%
log−Likelihood
−1300
−1500
−1700
−2
−1
0
1
2
λ




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

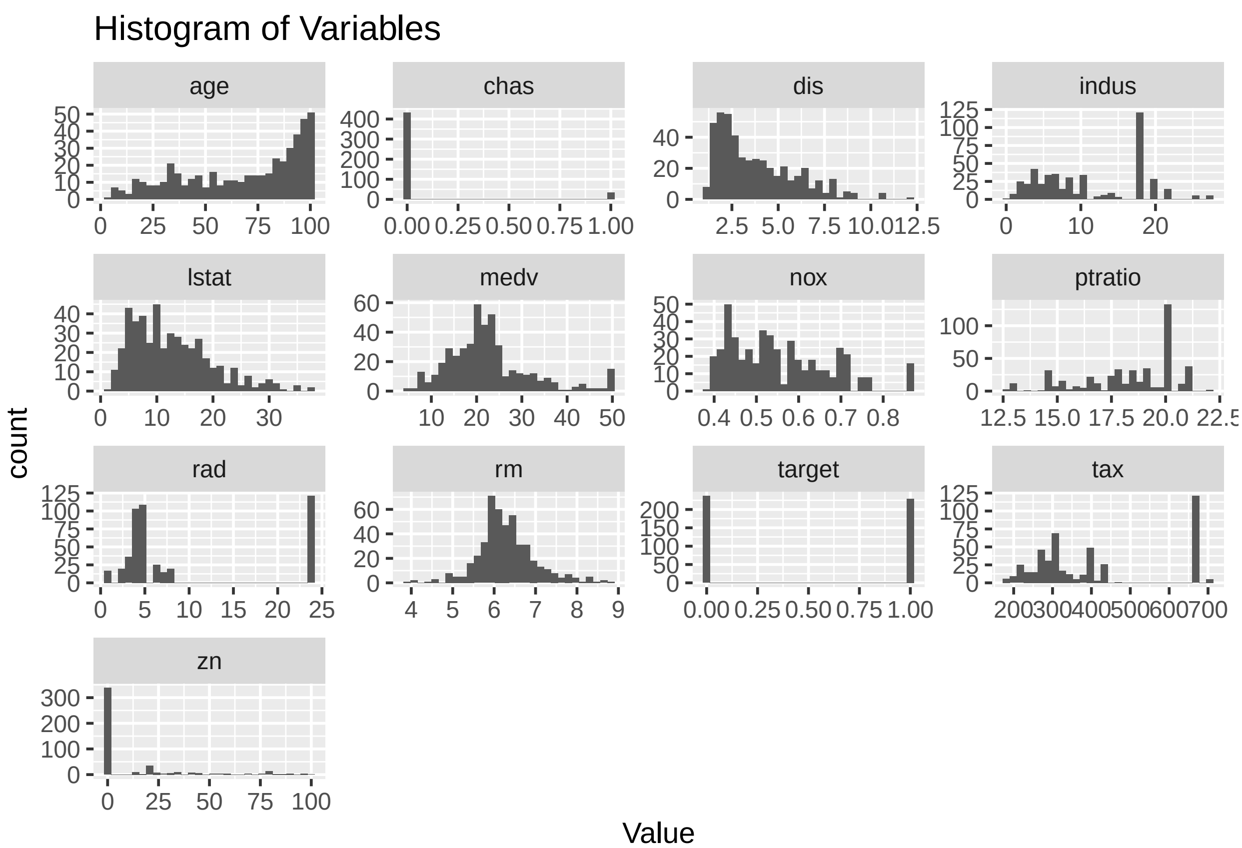
```r
# Gather the data into a long format
data_transformed_long <- gather(df_transformed, key = "Variable", value = "Value")

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`.
```




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

ggplot(data_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`.
```


Histogram of Variables
age
chas
dis
indus
lstat
medv
nox
ptratio
rad
rm
target
tax
zn
count
Value