Homework 3

Brandon Chung, Jiaxin Zheng, Andreina Arias, and Stephanie Chiang

Fall 2025

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)

I. Data Exploration:

There are two files provided:

- crime-training-data_modified.cvs
- crime-evolution-data_modified.cvs

The training data contains 13 variables and 466 observations, all with positive numeric values.

```r
str(train_df)
```

```
## '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 ...
```

1. Summary

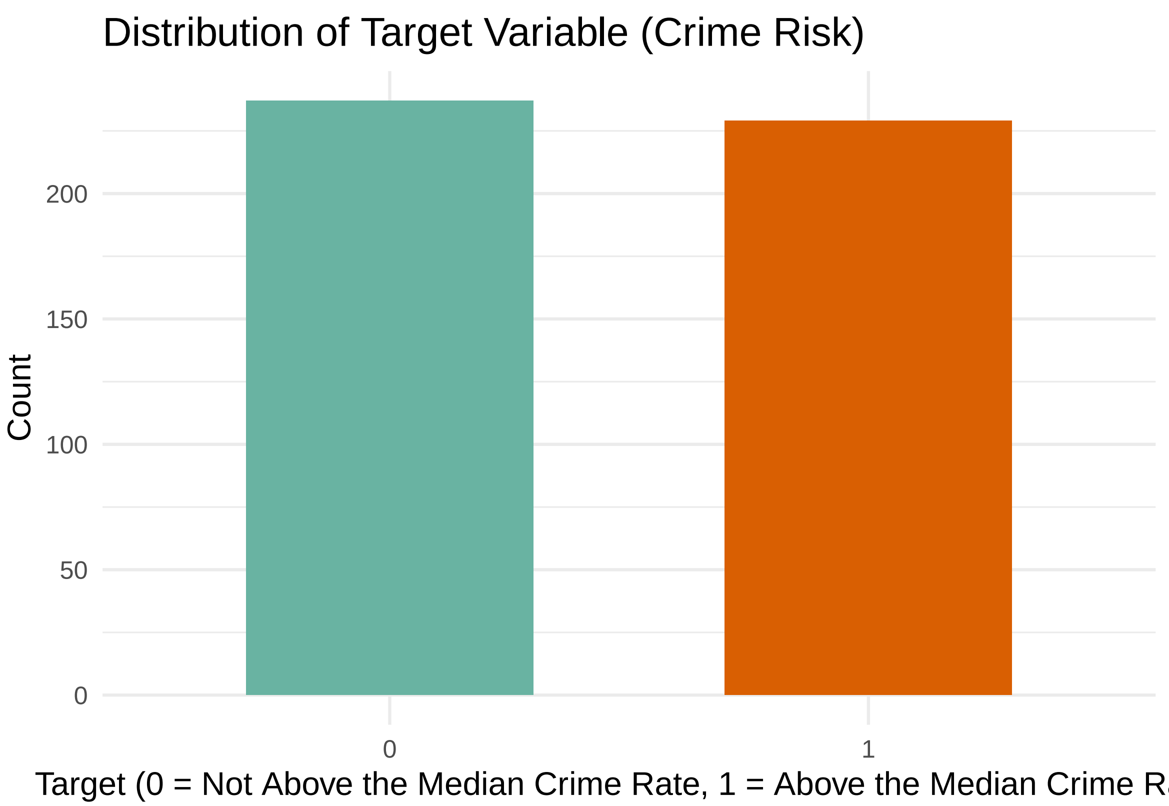
- We observe there is no missing value in the data (no NA's).
- Based on the summary statistics below, it appears we have many means that are far from the median, it indicating a skewed distribution.

```r
summary(train_df)
```

```
##        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  
```

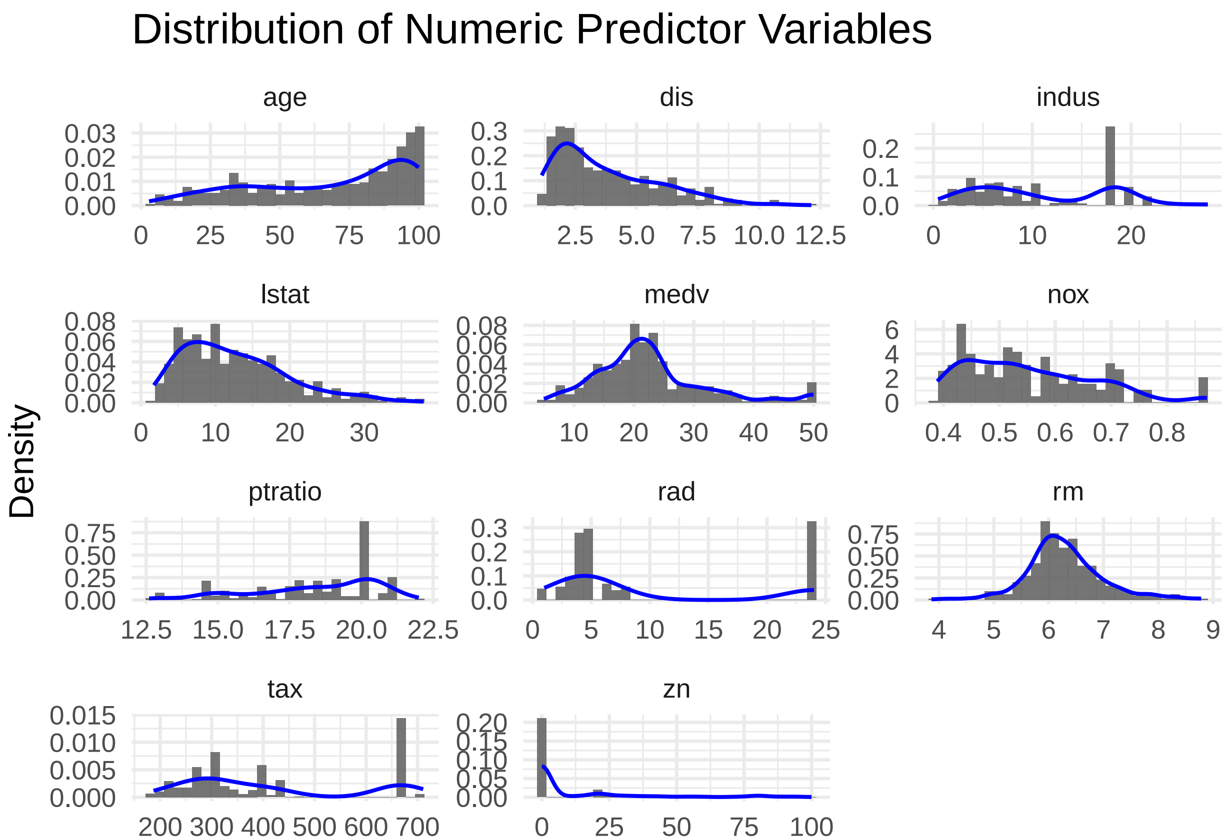
2. Distributions

- The bar chart shows that neighborhoods with low and high rime rate are nearly equal.

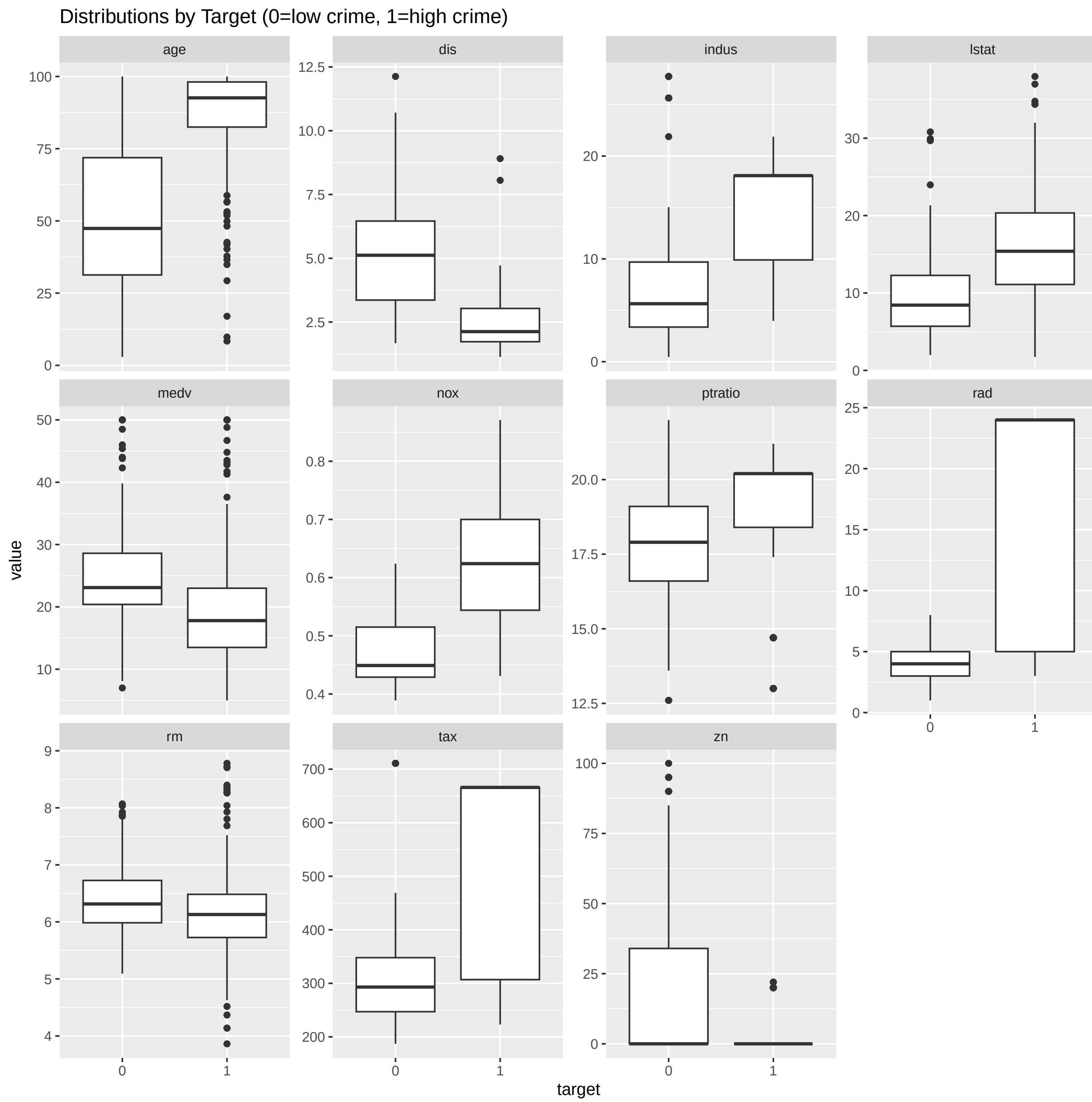


- Next, we visualize the distribution for each predictor variables.
- The distribution profiles show the dis, lstat, nox, rm, zn are right skewed, specially dis, and lstat.
- We also note that age and ptratio are left skewed

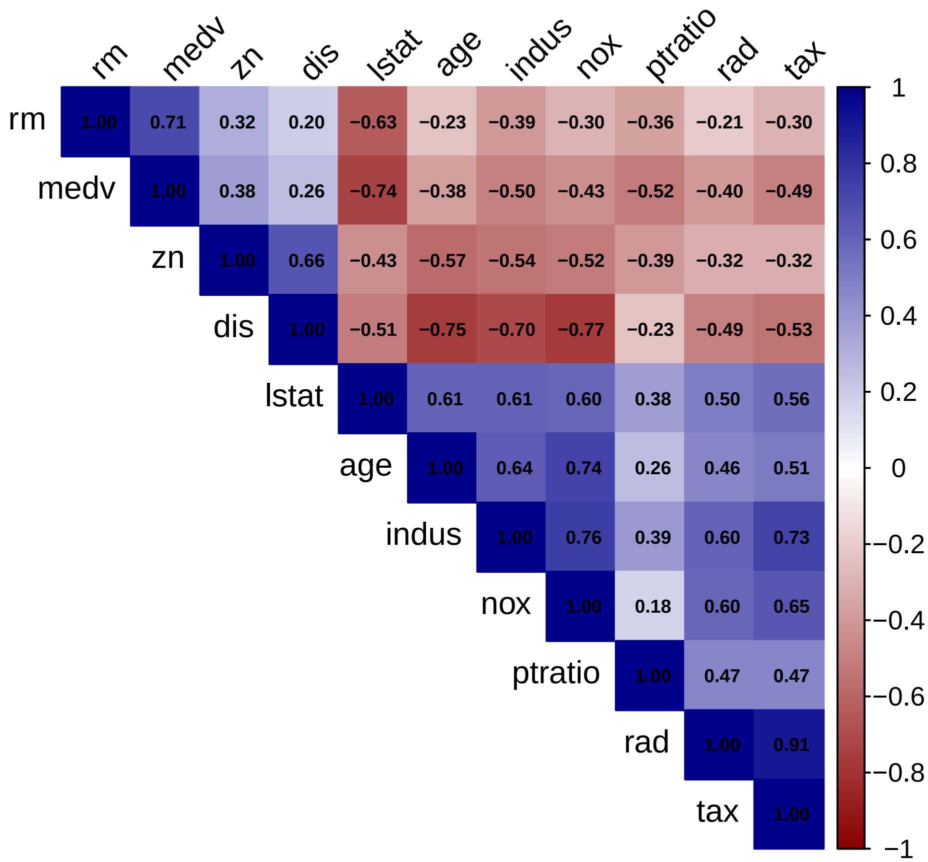
```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



- Looking at the box plots below, we can see there are significant outliers in agem dis, indus, lstat, medv, ptratio, rm, tax, and zn, they may need to be imputed if necessary.



- Going though the heatmap, we can see which variables are correlated to be included together in a model as predictor variables. This will help us later during the model selection process.
- With a threshold of 0.90 we can see that variables rad and tax are highly correlated, with a correlation 0.91.




```
##   Var1 Var2 Correlation
## 1  tax  rad   0.9064632
```

II. Data Preparation:

a. Missing Data There is no missing values in our predictors, so there will be no need to impute any variables.

```
##      zn   indus    chas     nox      rm     age     dis     rad     tax ptratio
##       0       0       0       0       0       0       0       0       0       0
##   lstat    medv  target
##       0       0       0
```

b. Near Zero Variance There are no predictors with near zero variance, so there is no need to remove predictors based on non significance / noise reduction.

```
##         freqRatio percentUnique zeroVar   nzv
## zn      16.142857     5.5793991   FALSE FALSE
## indus    4.321429    15.6652361   FALSE FALSE
## chas    13.121212     0.4291845   FALSE FALSE
## nox      1.176471    16.9527897   FALSE FALSE
## rm       1.000000    89.9141631   FALSE FALSE
## age     10.500000    71.4592275   FALSE FALSE
## dis      1.000000    81.5450644   FALSE FALSE
```

```
## rad      1.110092     1.9313305   FALSE FALSE
## tax      3.457143    13.5193133   FALSE FALSE
## ptratio  4.000000     9.8712446   FALSE FALSE
## lstat    1.000000    90.9871245   FALSE FALSE
## medv     2.142857    46.7811159   FALSE FALSE
## target   1.034934     0.4291845   FALSE FALSE
```

c. Outliers

- In our analysis we chose keep the outliers. Several predictors such as nox, lstat, and dis exhibited noticeable skewness and data points far from their IQRs, but it seems the values represent real neighborhood's data and are not structural or data entry errors. After reviewing the stated variables, the data points will be treated as leverage points and add generalizability to our resulting model.

d. Multicolinearity

- Through our variable correlation heatmap and correlation check we discovered that rad and tax have a correlation of 0.91. For one of our models to be tested, from this highly correlated pair we will drop rad to reduce multicolinearity based on domain relevance of tax. Tax is a more direct indicator of social economic status and likely has a greater impact on crime.

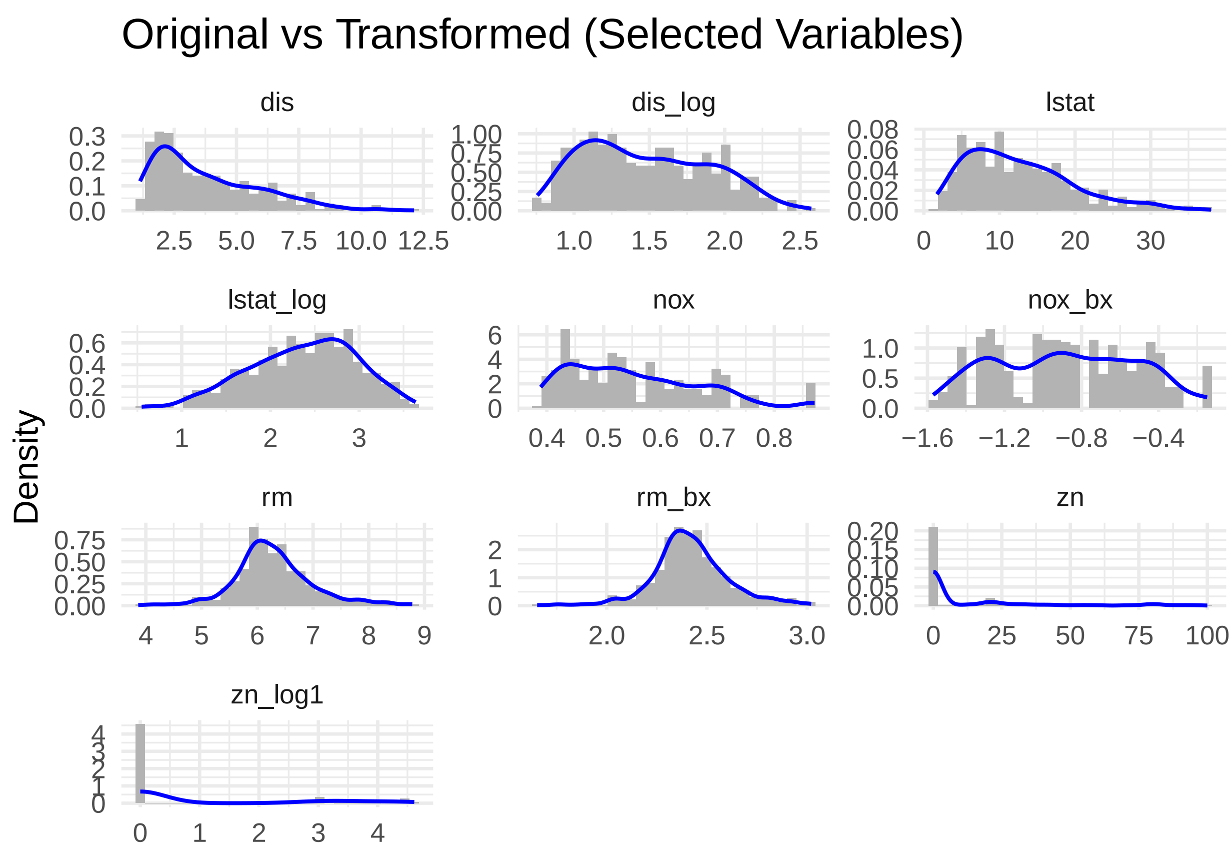
e. Transform Skewed Variables

- Some of the variables in our data display skew and non-constant variance. To combat this we applied either box-cox or logarithmic transformation. Box-cox transformation was used on rm and nox. For our variables dis, zn, and lstat there is noticable right sided skew so log transformation was applied.

```r
# Box-Cox lambdas learned on train_df
rm_lambda  <- BoxCox.lambda(train_df$rm)
nox_lambda <- BoxCox.lambda(train_df$nox)

# Append transformed columns to train_df
train_df_transformed <- train_df %>%
  mutate(
    rm_bx     = BoxCox(rm,  rm_lambda),
    nox_bx    = BoxCox(nox, nox_lambda),
    dis_log   = log(dis + 1),
    zn_log1   = log(zn + 1),
    lstat_log = log(lstat)
  )

# Apply same to test_df
test_df_transformed <- test_df %>%
  mutate(
    rm_bx     = BoxCox(rm,  rm_lambda),
    nox_bx    = BoxCox(nox, nox_lambda),
    dis_log   = log(dis + 1),
    zn_log1   = log(zn + 1),
    lstat_log = log(lstat)
  )
```



III. Build Models:

```r
# Model A: Baseline
modA <- glm(target ~ ., data = train_df, family = binomial())
summary(modA)
```

Model A: Original data baseline

```
##
## Call:
## glm(formula = target ~ ., family = binomial(), data = train_df)
##
## 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
## chas1         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
```

```
modB <- glm(target ~ chas + zn_log1 + indus + nox_bx + rm_bx + age +
              dis_log + rad + tax + ptratio + lstat_log + medv,
            data = train_df_transformed, family = binomial())
summary(modB)
```

Model B: Transformed data — check improvement

```
##
## Call:
## glm(formula = target ~ chas + zn_log1 + indus + nox_bx + rm_bx +
##     age + dis_log + rad + tax + ptratio + lstat_log + medv, family = binomial(),
##     data = train_df_transformed)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.359593   6.353986  -0.214 0.830567
## chas1        0.875589   0.762164   1.149 0.250630
## zn_log1     -0.215989   0.228995  -0.943 0.345578
## indus       -0.019205   0.045516  -0.422 0.673070
## nox_bx      14.413662   2.202193   6.545 5.94e-11 ***
## rm_bx       -4.518450   2.778294  -1.626 0.103877
## age          0.045181   0.014186   3.185 0.001448 **
## dis_log      4.556856   1.216568   3.746 0.000180 ***
## rad          0.675524   0.168261   4.015 5.95e-05 ***
## tax         -0.004798   0.002888  -1.661 0.096627 .
## ptratio      0.443830   0.134116   3.309 0.000935 ***
## lstat_log   -0.148284   0.718943  -0.206 0.836593
## medv         0.223352   0.069696   3.205 0.001352 **
## ---
## 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: 191.47  on 453  degrees of freedom
## AIC: 217.47
##
## Number of Fisher Scoring iterations: 9
```

```r
modC <- stepAIC(modB, direction = "both")
```

Model C: Stepwise selection with transformed data — check if a smaller model can perform equally well

```
## Start:  AIC=217.47
## target ~ chas + zn_log1 + indus + nox_bx + rm_bx + age + dis_log +
##     rad + tax + ptratio + lstat_log + medv
##
##             Df Deviance    AIC
## - lstat_log  1   191.51 215.51
## - indus      1   191.65 215.65
## - zn_log1    1   192.39 216.39
## - chas       1   192.81 216.81
## <none>           191.47 217.47
## - rm_bx      1   194.20 218.20
## - tax        1   194.27 218.27
## - age        1   202.81 226.81
## - medv       1   203.33 227.33
## - ptratio    1   203.67 227.67
## - dis_log    1   207.76 231.76
## - rad        1   231.61 255.61
## - nox_bx     1   268.87 292.87
##
## Step:  AIC=215.51
## target ~ chas + zn_log1 + indus + nox_bx + rm_bx + age + dis_log +
##     rad + tax + ptratio + medv
##
##             Df Deviance    AIC
## - indus      1   191.70 213.70
## - zn_log1    1   192.61 214.61
## - chas       1   192.81 214.81
## <none>           191.51 215.51
## - tax        1   194.30 216.30
## - rm_bx      1   194.54 216.54
## + lstat_log  1   191.47 217.47
## - ptratio    1   203.77 225.77
## - medv       1   203.81 225.81
## - age        1   204.51 226.51
## - dis_log    1   207.95 229.95
## - rad        1   231.99 253.99
## - nox_bx     1   269.04 291.04
##
## Step:  AIC=213.7
## target ~ chas + zn_log1 + nox_bx + rm_bx + age + dis_log + rad +
##     tax + ptratio + medv
##
##             Df Deviance    AIC
## - zn_log1    1   192.81 212.81
## - chas       1   192.85 212.85
## <none>           191.70 213.70
## - rm_bx      1   194.67 214.67
```

```
## + indus       1   191.51 215.51
## + lstat_log   1   191.65 215.65
## - tax         1   195.91 215.91
## - ptratio     1   203.79 223.79
## - medv        1   203.97 223.97
## - age         1   204.63 224.63
## - dis_log     1   208.75 228.75
## - rad         1   238.84 258.84
## - nox_bx      1   273.78 293.78
##
## Step:  AIC=212.81
## target ~ chas + nox_bx + rm_bx + age + dis_log + rad + tax +
##     ptratio + medv
##
##             Df Deviance    AIC
## - chas       1   194.56 212.56
## <none>           192.81 212.81
## + zn_log1    1   191.70 213.70
## - rm_bx      1   196.33 214.33
## + lstat_log  1   192.58 214.58
## + indus      1   192.61 214.61
## - tax        1   197.58 215.58
## - medv       1   205.59 223.59
## - age        1   206.00 224.00
## - dis_log    1   208.76 226.76
## - ptratio    1   211.27 229.27
## - rad        1   240.97 258.97
## - nox_bx     1   276.94 294.94
##
## Step:  AIC=212.56
## target ~ nox_bx + rm_bx + age + dis_log + rad + tax + ptratio +
##     medv
##
##             Df Deviance    AIC
## <none>           194.56 212.56
## + chas       1   192.81 212.81
## + zn_log1    1   192.85 212.85
## + lstat_log  1   194.45 214.45
## + indus      1   194.52 214.52
## - rm_bx      1   198.61 214.61
## - tax        1   200.29 216.29
## - medv       1   207.73 223.73
## - age        1   209.13 225.13
## - dis_log    1   209.67 225.67
## - ptratio    1   211.87 227.87
## - rad        1   247.72 263.73
## - nox_bx     1   277.24 293.24
```

```r
summary(modC)
```

```
##
## Call:
## glm(formula = target ~ nox_bx + rm_bx + age + dis_log + rad +
##     tax + ptratio + medv, family = binomial(), data = train_df_transformed)
```

```
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.497921   4.612911  -0.325 0.745390
## nox_bx      13.896572   2.075127   6.697 2.13e-11 ***
## rm_bx       -4.922569   2.485988  -1.980 0.047689 *
## age          0.046136   0.012937   3.566 0.000362 ***
## dis_log      4.112507   1.126810   3.650 0.000263 ***
## rad          0.730035   0.155310   4.701 2.60e-06 ***
## tax         -0.005908   0.002599  -2.274 0.022985 *
## ptratio      0.473243   0.122018   3.878 0.000105 ***
## medv         0.232578   0.069649   3.339 0.000840 ***
## ---
## 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: 194.56  on 457  degrees of freedom
## AIC: 212.56
##
## Number of Fisher Scoring iterations: 9
```

```r
# Model D: Original with rad removed
train_df_rad <- subset(train_df, select = -rad)

modD <- glm(target ~ ., data = train_df_rad, family = binomial())
summary(modD)
```

Model D: Original data - rad removed

```
##
## Call:
## glm(formula = target ~ ., family = binomial(), data = train_df_rad)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -40.033181   5.779572  -6.927 4.31e-12 ***
## zn           -0.070824   0.031059  -2.280 0.022588 *
## indus        -0.143371   0.049590  -2.891 0.003839 **
## chas1         1.796276   0.657480   2.732 0.006294 **
## nox          46.965637   6.938284   6.769 1.30e-11 ***
## rm           -0.150797   0.566111  -0.266 0.789951
## age           0.022903   0.011773   1.945 0.051718 .
## dis           0.786125   0.201883   3.894 9.86e-05 ***
## tax           0.006441   0.001935   3.329 0.000872 ***
## ptratio       0.315392   0.106237   2.969 0.002990 **
## lstat         0.043852   0.048531   0.904 0.366210
## medv          0.174600   0.053456   3.266 0.001090 **
## ---
## 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: 233.74  on 454  degrees of freedom
## AIC: 257.74
##
## Number of Fisher Scoring iterations: 8
```

```r
# Model E: Transformed data with rad removed
modE <- glm(target ~ chas + zn_log1 + indus + nox_bx + rm_bx + age +
              dis_log + tax + ptratio + lstat_log + medv,
            data = train_df_transformed, family = binomial())
summary(modE)
```

Model E: Transformed data - rad removed

```
##
## Call:
## glm(formula = target ~ chas + zn_log1 + indus + nox_bx + rm_bx +
##     age + dis_log + tax + ptratio + lstat_log + medv, family = binomial(),
##     data = train_df_transformed)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.914607   5.281067  -1.309 0.190426
## chas1        1.718950   0.669774   2.566 0.010274 *
## zn_log1     -0.350112   0.222255  -1.575 0.115193
## indus       -0.116740   0.047375  -2.464 0.013733 *
## nox_bx      14.428433   1.986178   7.264 3.75e-13 ***
## rm_bx       -1.178498   1.982187  -0.595 0.552148
## age          0.028655   0.011832   2.422 0.015440 *
## dis_log      4.821978   1.091374   4.418 9.95e-06 ***
## tax          0.007570   0.001975   3.832 0.000127 ***
## ptratio      0.325298   0.112107   2.902 0.003712 **
## lstat_log    0.393866   0.641298   0.614 0.539103
## medv         0.203550   0.052781   3.857 0.000115 ***
## ---
## 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: 231.61  on 454  degrees of freedom
## AIC: 255.61
##
## Number of Fisher Scoring iterations: 7
```

```r
modF <- stepAIC(modA, direction = "both")
```

Model F: Stepwise selection with original data

```
## Start:  AIC=218.05
## target ~ zn + indus + chas + nox + rm + age + dis + rad + tax +
##     ptratio + lstat + medv
##
##           Df Deviance    AIC
## - rm       1   192.71 216.71
## - lstat    1   192.77 216.77
## - chas     1   193.53 217.53
## - indus    1   193.99 217.99
## <none>         192.05 218.05
## - tax      1   196.59 220.59
## - zn       1   196.89 220.89
## - age      1   198.73 222.73
## - medv     1   199.95 223.95
## - ptratio  1   203.32 227.32
## - dis      1   203.84 227.84
## - rad      1   233.74 257.74
## - nox      1   265.05 289.05
##
## Step:  AIC=216.71
## target ~ zn + indus + chas + nox + age + dis + rad + tax + ptratio +
##     lstat + medv
##
##           Df Deviance    AIC
## - chas     1   194.24 216.24
## - lstat    1   194.32 216.32
## - indus    1   194.58 216.58
## <none>         192.71 216.71
## + rm       1   192.05 218.05
## - tax      1   197.59 219.59
## - zn       1   198.07 220.07
## - age      1   199.11 221.11
## - ptratio  1   203.53 225.53
## - dis      1   203.85 225.85
## - medv     1   205.35 227.35
## - rad      1   233.81 255.81
## - nox      1   265.14 287.14
##
## Step:  AIC=216.24
## target ~ zn + indus + nox + age + dis + rad + tax + ptratio +
##     lstat + medv
##
##           Df Deviance    AIC
## - indus    1   195.51 215.51
## <none>         194.24 216.24
## - lstat    1   196.33 216.33
## + chas     1   192.71 216.71
## + rm       1   193.53 217.53
```

```
## - zn       1   200.59 220.59
## - tax      1   200.75 220.75
## - age      1   201.00 221.00
## - ptratio  1   203.94 223.94
## - dis      1   204.83 224.83
## - medv     1   207.12 227.12
## - rad      1   241.41 261.41
## - nox      1   265.19 285.19
##
## Step:  AIC=215.51
## target ~ zn + nox + age + dis + rad + tax + ptratio + lstat +
##     medv
##
##           Df Deviance    AIC
## - lstat    1   197.32 215.32
## <none>         195.51 215.51
## + indus    1   194.24 216.24
## + chas     1   194.58 216.58
## + rm       1   194.86 216.86
## - zn       1   202.05 220.05
## - age      1   202.23 220.23
## - ptratio  1   205.01 223.01
## - dis      1   205.96 223.96
## - tax      1   206.60 224.60
## - medv     1   208.13 226.13
## - rad      1   249.55 267.55
## - nox      1   270.59 288.59
##
## Step:  AIC=215.32
## target ~ zn + nox + age + dis + rad + tax + ptratio + medv
##
##           Df Deviance    AIC
## <none>         197.32 215.32
## + lstat    1   195.51 215.51
## + rm       1   195.75 215.75
## + chas     1   195.97 215.97
## + indus    1   196.33 216.33
## - zn       1   203.45 219.45
## - ptratio  1   206.27 222.27
## - age      1   207.13 223.13
## - tax      1   207.62 223.62
## - dis      1   207.64 223.64
## - medv     1   208.65 224.65
## - rad      1   250.98 266.98
## - nox      1   273.18 289.18
```

```r
summary(modC)
```

```
##
## Call:
## glm(formula = target ~ nox_bx + rm_bx + age + dis_log + rad +
##     tax + ptratio + medv, family = binomial(), data = train_df_transformed)
##
## Coefficients:
```

```
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.497921   4.612911  -0.325 0.745390
## nox_bx      13.896572   2.075127   6.697 2.13e-11 ***
## rm_bx       -4.922569   2.485988  -1.980 0.047689 *
## age          0.046136   0.012937   3.566 0.000362 ***
## dis_log      4.112507   1.126810   3.650 0.000263 ***
## rad          0.730035   0.155310   4.701 2.60e-06 ***
## tax         -0.005908   0.002599  -2.274 0.022985 *
## ptratio      0.473243   0.122018   3.878 0.000105 ***
## medv         0.232578   0.069649   3.339 0.000840 ***
## ---
## 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: 194.56  on 457  degrees of freedom
## AIC: 212.56
##
## Number of Fisher Scoring iterations: 9
```

IV: Model Selection:

- Model B (transformed data including rad) performed best overall and is the model we will use to make predictions on the test data. Model B provided the highest accuracy (0.918), F1 (0.916), and Rsquared (0.704) while also providing the lowest deviance (191.47) and decent AIC (217.47) (less than the original model but greater than the stepwise models). These results suggest that the variable rad was an impactful predictor, and that transforming specific predictors via box-cox and log improved linearity in the logit and stabilized variance for the model. The stepwise AIC model C performed slightly worse in accuracy and F1 although provided a tradeoff with its lower AIC.

```
## Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

```
## Warning in attr(x, "format"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

Table 1: Model Comparison Summary (Training Set)

Model	Accuracy	F1	Deviance	R2	AIC
Model A: Original	0.916	0.914	192.05	0.703	218.05
Model B: Transformed	0.918	0.916	191.47	0.704	217.47
Model C: Stepwise with Transformed	0.914	0.912	194.56	0.699	212.56
Model D: Original - rad Removed	0.891	0.888	233.74	0.638	257.74
Model E: Transformed - rad Removed	0.893	0.892	231.61	0.641	255.61
Model F: Stepwise with Original	0.912	0.910	197.32	0.694	215.32

Evaluating Model B wtih accuracy, classification error rate, precision, sensitivity, specificity, f1 score, AUC and confusion matrix

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## $ConfusionMatrix
##           Reference
## Prediction   0   1
##          0 220  21
##          1  17 208
##
## $Accuracy
##  Accuracy
## 0.9184549
##
## $ClassificationErrorRate
##   Accuracy
## 0.08154506
##
## $Precision
## Precision
## 0.9244444
##
## $Sensitivity
## Sensitivity
##   0.9082969
##
## $Specificity
## Specificity
##     0.92827
##
## $F1_Score
##        F1
## 0.9162996
##
## $AUC
## Area under the curve: 0.975
```

Making Predicitons on the test_df_transformed data frame using Model B

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
##  0  1  1  0  0  0  0  0  0  0  0  0  1  1  1  0  0  1  0  0  0  0  0  0  0  1
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40
##  0  1  1  1  1  1  1  1  1  1  1  1  1  0
```