Group#2 Homework# 3 - Logistic Regression

Group 2

3/29/2018

Introduction

This assignment explores, analyze and model a dataset containing information about crime in various neighborhoods in 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). The crime dataset contains 18 variables and 466 observations. #Objective and Requirements The objective is to build a binary logistic regression model on the training dataset to predict whether the neighborhood will be at risk for high crime levels. Classifications and probabilities for the evaluation dataset using the binary logistic regression model. #Approach The team met to discuss this assignment and an approach to plan, complete the assignment. Each of the 5 team members was assigned tasks. The following tasks were assigned: *Data Exploration* Data Preparation *Build Models* Select Models

Github was used to manage the project. Using Github helped with version control and ensured each team member had access to the latest version of the project documentation.

Slack was used to by the team to communicate during the project and for quick access to code and documentation.

Dataset

For reproducibility of the results, the data was loaded to and accessed from a Github repository. The age variable was rounded to a whole number.

Data Exploration

The following variables comprise the data set. The response variable (Target) is the variable of interest. The response variable is binary (0, 1) and identifies whether the crime rate is above the median crime rate. The remaining 12 variables are predictors. All variables are numeric.

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

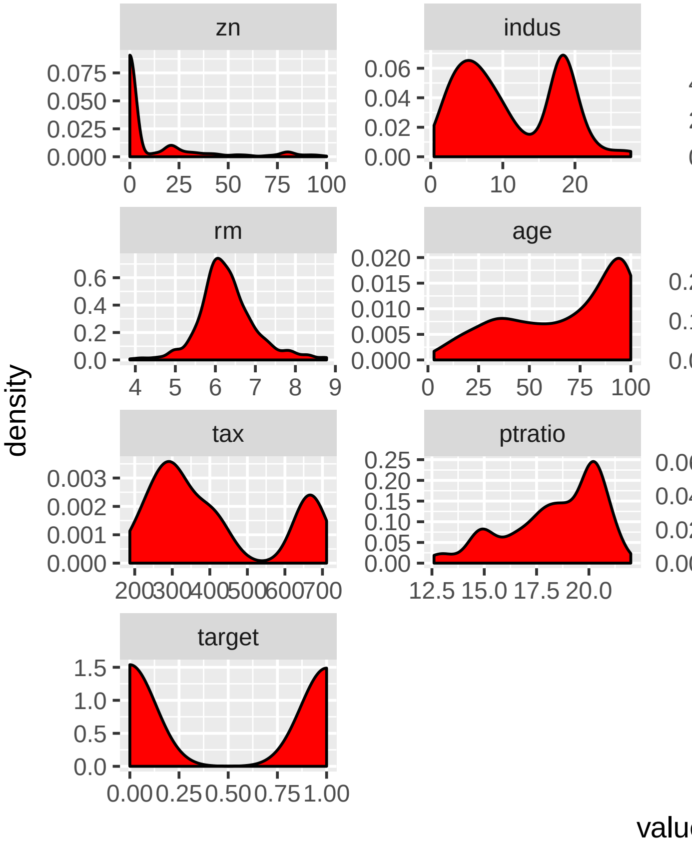
Descriptive statistics were calculated to examine the basic features of the data. Each variable has 466 observations.

```
##         vars   n   mean     sd median trimmed    mad    min    max  range
## zn         1 466  11.58  23.36   0.00    5.35   0.00   0.00 100.00 100.00
## indus      2 466  11.11   6.85   9.69   10.91   9.34   0.46  27.74  27.28
## chas       3 466   0.07   0.26   0.00    0.00   0.00   0.00   1.00   1.00
## nox        4 466   0.55   0.12   0.54    0.54   0.13   0.39   0.87   0.48
## rm         5 466   6.29   0.70   6.21    6.26   0.52   3.86   8.78   4.92
## age        6 466  68.35  28.32  77.00   70.93  29.65   3.00 100.00  97.00
## dis        7 466   3.80   2.11   3.19    3.54   1.91   1.13  12.13  11.00
## rad        8 466   9.53   8.69   5.00    8.70   1.48   1.00  24.00  23.00
## tax        9 466 409.50 167.90 334.50  401.51 104.52 187.00 711.00 524.00
## ptratio   10 466  18.40   2.20  18.90   18.60   1.93  12.60  22.00   9.40
## lstat     11 466  12.63   7.10  11.35   11.88   7.07   1.73  37.97  36.24
## medv      12 466  22.59   9.24  21.20   21.63   6.00   5.00  50.00  45.00
## target    13 466   0.49   0.50   0.00    0.49   0.00   0.00   1.00   1.00
##          skew kurtosis   se    IQR  Q0.1  Q0.25  Q0.75   Q0.9
## zn       2.18     3.81 1.08  16.25   0.00   0.00  16.25  45.00
## indus    0.29    -1.24 0.32  12.96   2.91   5.15  18.10  19.58
## chas     3.34     9.15 0.01   0.00   0.00   0.00   0.00   0.00
## nox      0.75    -0.04 0.01   0.18   0.43   0.45   0.62   0.71
## rm       0.48     1.54 0.03   0.74   5.57   5.89   6.63   7.17
## age     -0.58    -1.01 1.31  50.00  26.00  44.00  94.00  99.00
## dis      1.00     0.47 0.10   3.11   1.61   2.10   5.21   6.81
## rad      1.01    -0.86 0.40  20.00   3.00   4.00  24.00  24.00
## tax      0.66    -1.15 7.78 385.00 233.00 281.00 666.00 666.00
## ptratio -0.75    -0.40 0.10   3.30  14.70  16.90  20.20  20.90
## lstat    0.91     0.50 0.33   9.89   4.70   7.04  16.93  22.86
## medv     1.08     1.37 0.43   7.98  12.70  17.02  25.00  34.90
## target   0.03    -2.00 0.02   1.00   0.00   0.00   1.00   1.00
```

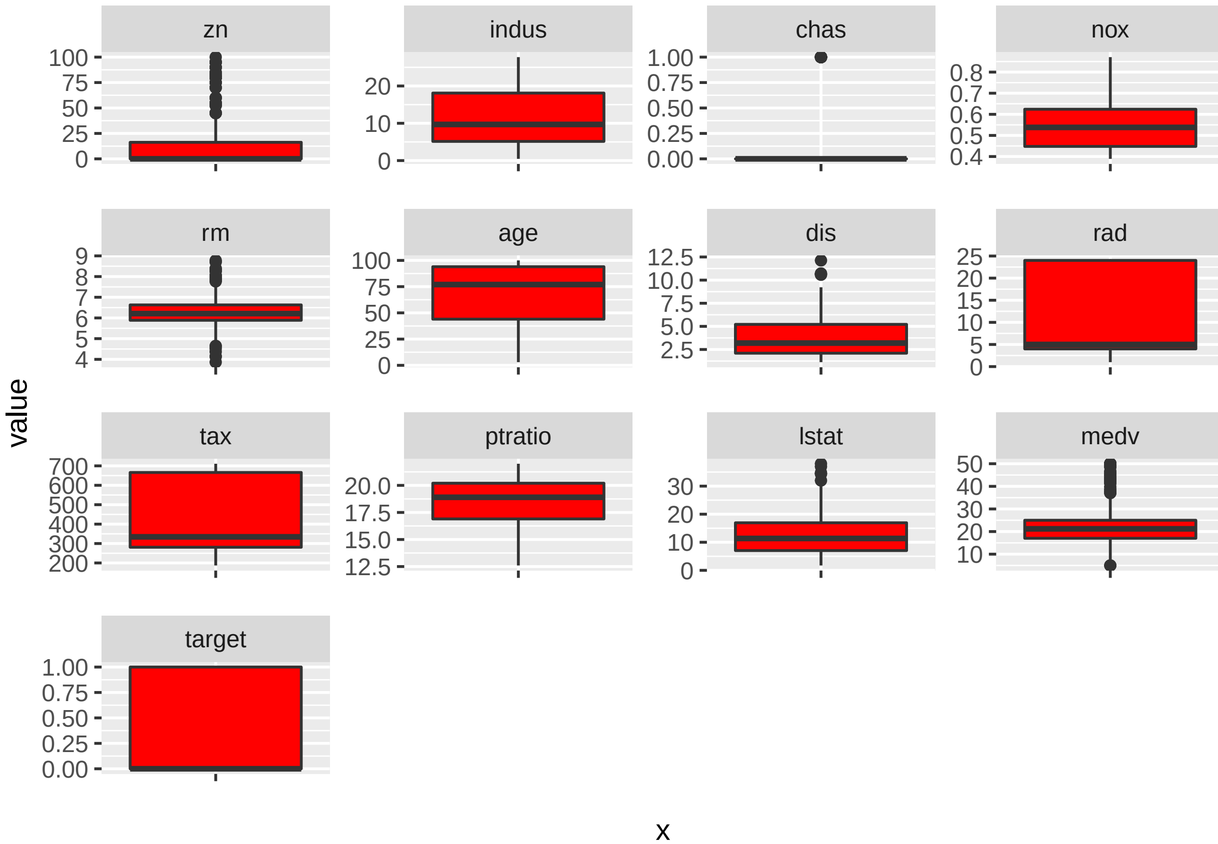
```
##   zn indus chas nox rm age dis rad tax ptratio lstat medv target
## 1  0     0    0   0  0   0   0   0   0       0     0    0      0
```

Density Plot

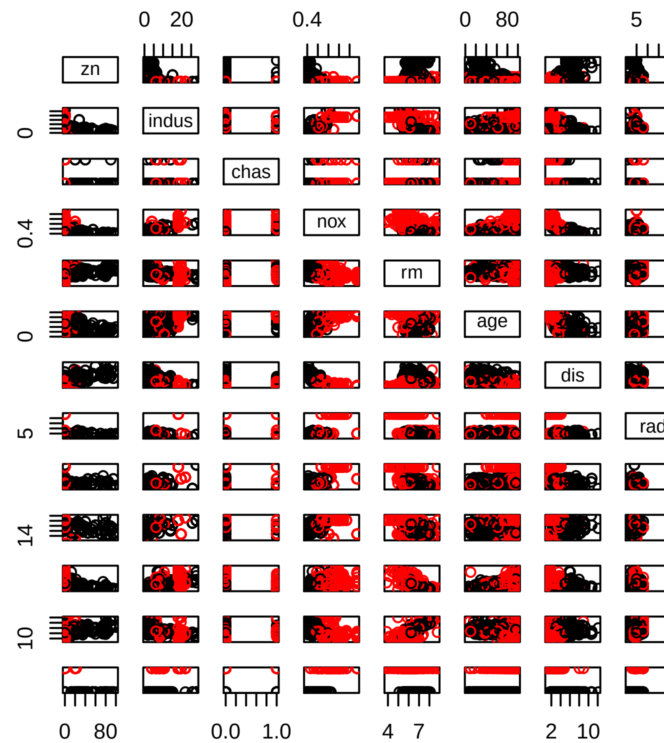
The density plot of predictor variables illustrates the zn, chas, dis, lstat predictor variales are hightly skewed.



The rm variable is the only predictor that is normally distributed.



The following looks at all of the predictor variables compared to each other and the response, with red values



showing observations where the crime rate exceeded the median.

zn

This variable is hightly skewed to the left. The range is from 85-100.

```
## $Obs
## [1] 100.0  95.0  90.0  85.0  82.5
##
## $Hi
## [1] 100  95
##
## $Low
## [1] 82.5 85.0
```

indus

This predictor variable is bi-modal.

```
## $Obs
## numeric(0)
##
## $Hi
## [1] NA
##
## $Low
```

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

nox

This variable is skewed to the left.

```
## $Obs
## numeric(0)
##
## $Hi
## [1] NA
##
## $Low
## numeric(0)
```

rm

```
## $Obs
## [1] 8.780 8.725 8.704 4.138 3.863
##
## $Hi
## [1] 8.780 8.725
##
## $Low
## [1] 3.863 4.138
```

age

```
## $Obs
## numeric(0)
##
## $Hi
## [1] NA
##
## $Low
## numeric(0)
```

dis

```
## $Obs
## [1] 12.1265 10.7103 10.5857
##
## $Hi
## [1] 12.1265
##
## $Low
## [1] 10.5857
```

rad

```
## $Obs
## integer(0)
##
## $Hi
## [1] NA
##
## $Low
## integer(0)
```

tax

```
## $Obs
## integer(0)
##
## $Hi
## [1] NA
##
## $Low
## integer(0)
```

ptratio

```
## $Obs
## numeric(0)
##
## $Hi
## [1] NA
##
## $Low
## numeric(0)
```

lstat

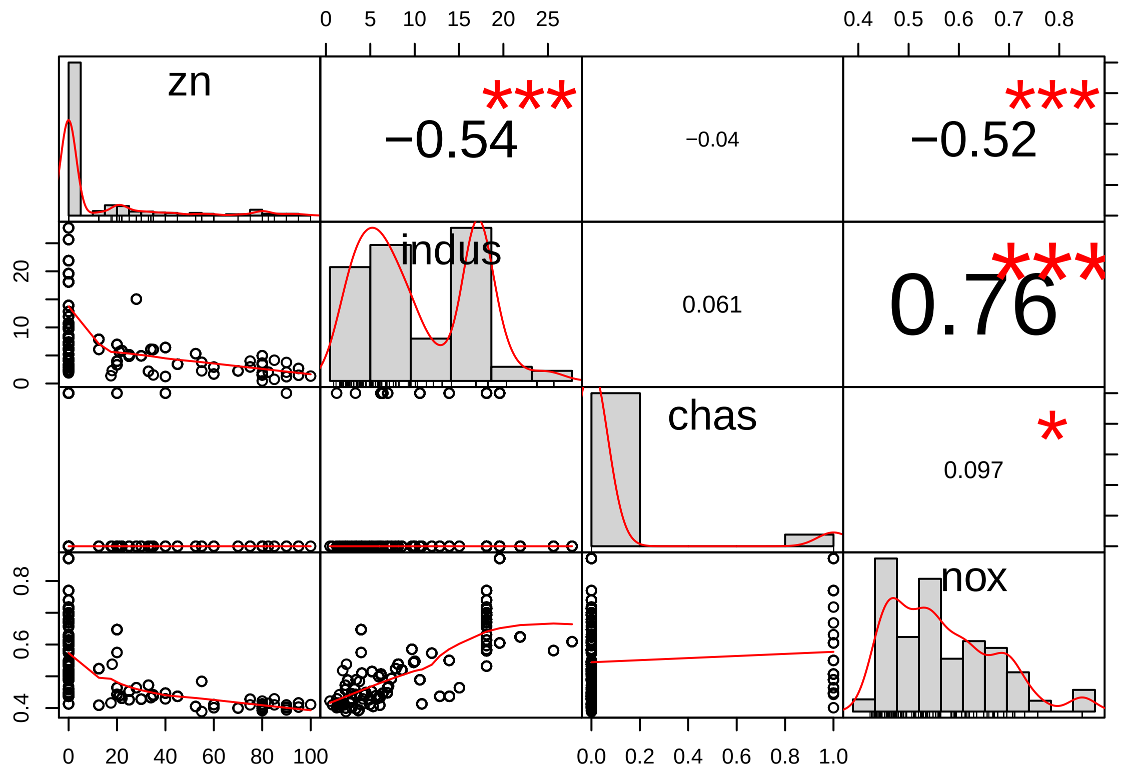
```
## $Obs
## [1] 37.97 36.98 34.77 34.41 34.37
##
## $Hi
## [1] 37.97 36.98
##
## $Low
## [1] 34.37 34.41
```

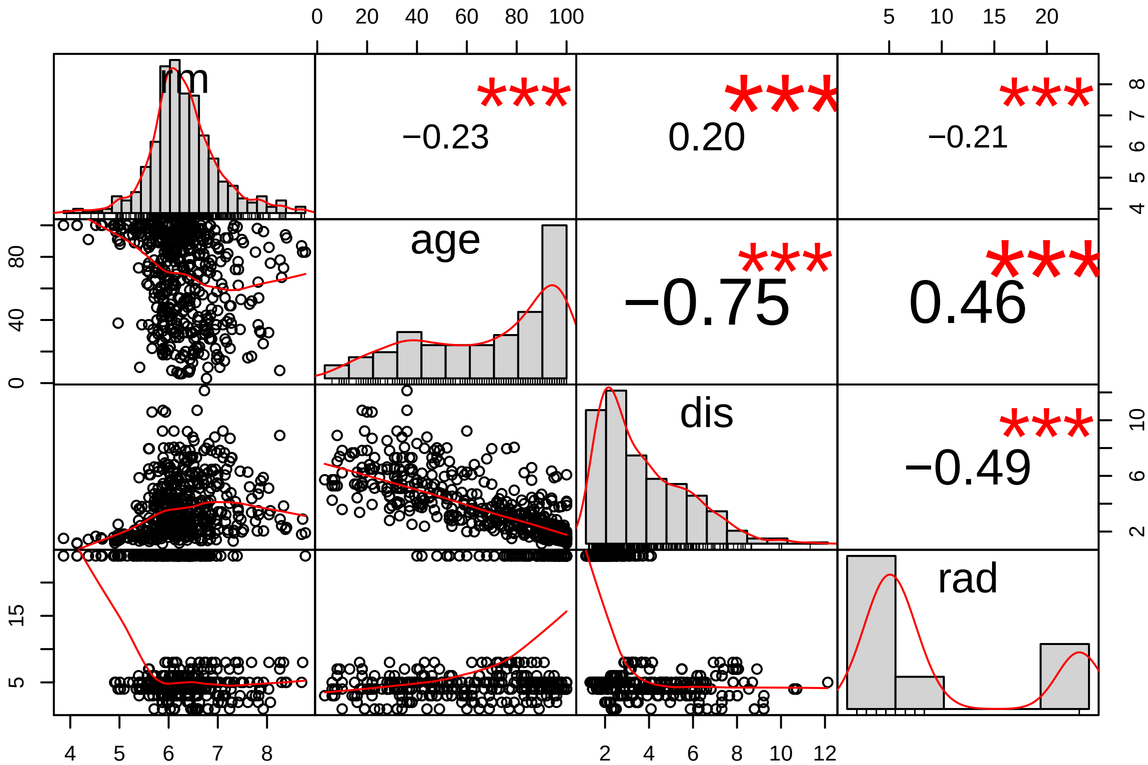
medv

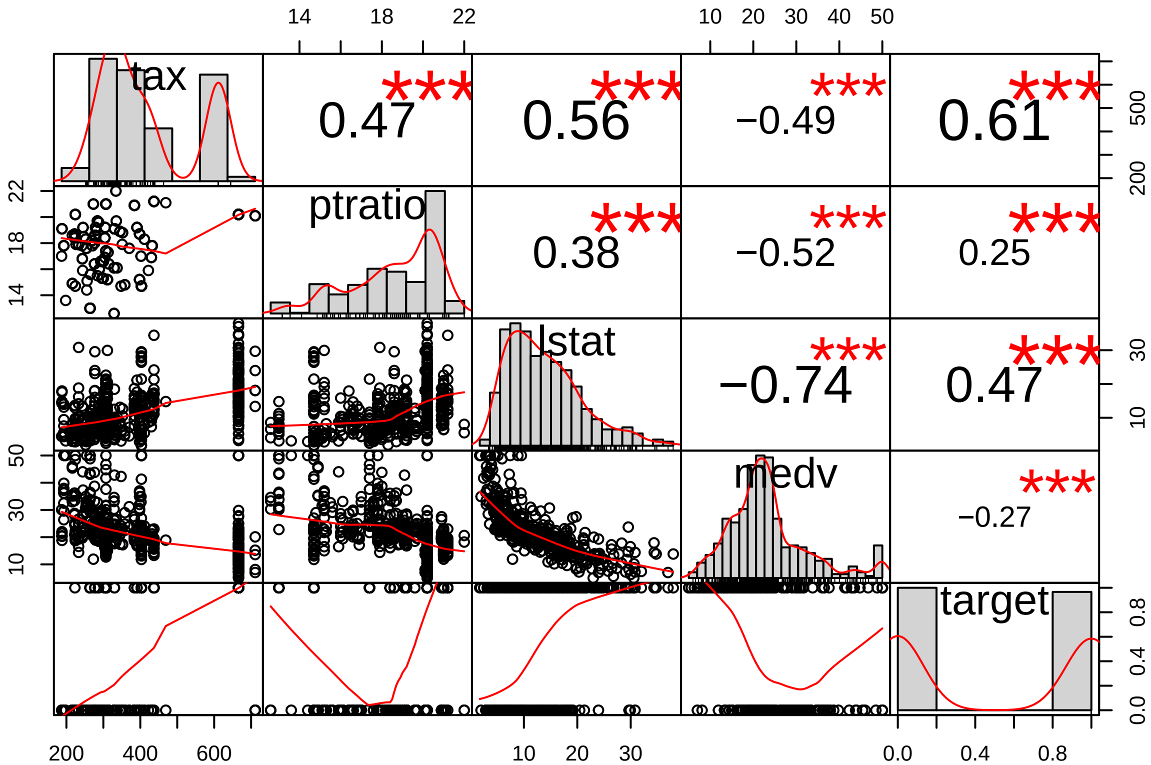
```
## $Obs
```

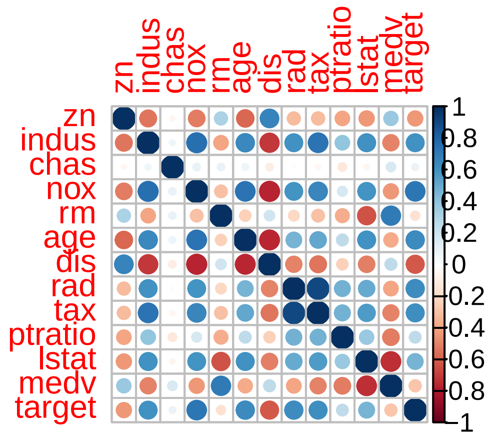
```
## numeric(0)
##
## $Hi
## [1] NA
##
## $Low
## numeric(0)
```

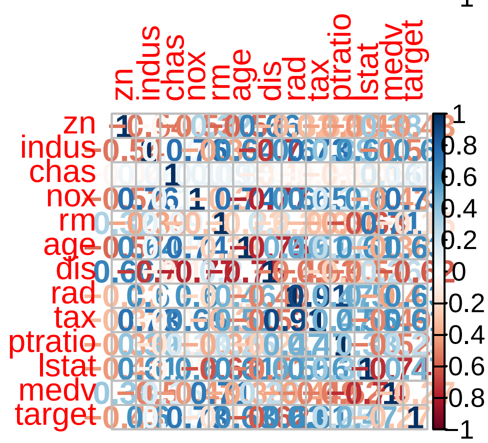
Correlation between variables




rm
age
dis
rad
−0.23
0.20
−0.21
−0.75
0.46
−0.49
***
0 20 40 60 80 100
5 10 15 20
4 5 6 7 8
2 4 6 8 10 12


tax
ptratio
lstat
medv
target
0.47
0.56
−0.49
0.61
0.38
−0.52
0.25
−0.74
0.47
−0.27
***
14
18
22
10
20
30
40
50
500
200
30
10
0.0
0.4
0.8
200
400
600
0.0
0.4
0.8





Multicollinearity

This section will test the predictor variables to determine if there is correlation among them. Variance inflaction factors (VIF) is used to detect multicollinearity, specifically among the entire set of predictors versus within pairs of variables.

Testing for Collinearity among the predictor variables, we see that the following variables may have a problem with collinearity:

```
vifcor(crime_trainData[, 1:12],th=0.4)
```

```
## 8 variables from the 12 input variables have collinearity problem:
##
## tax nox dis lstat medv indus age ptratio
##
## After excluding the collinear variables, the linear correlation coefficients ranges between:
## min correlation ( rad ~ chas ):  -0.01590037
## max correlation ( rm ~ zn ):  0.3198141
##
## ---------- VIFs of the remained variables --------
##   Variables      VIF
## 1        zn 1.207309
## 2      chas 1.014001
## 3        rm 1.143040
## 4       rad 1.126988
```

Variable Name
* tax
* nox
* dis
* lstat
* medv
* indus
* age
* ptratio

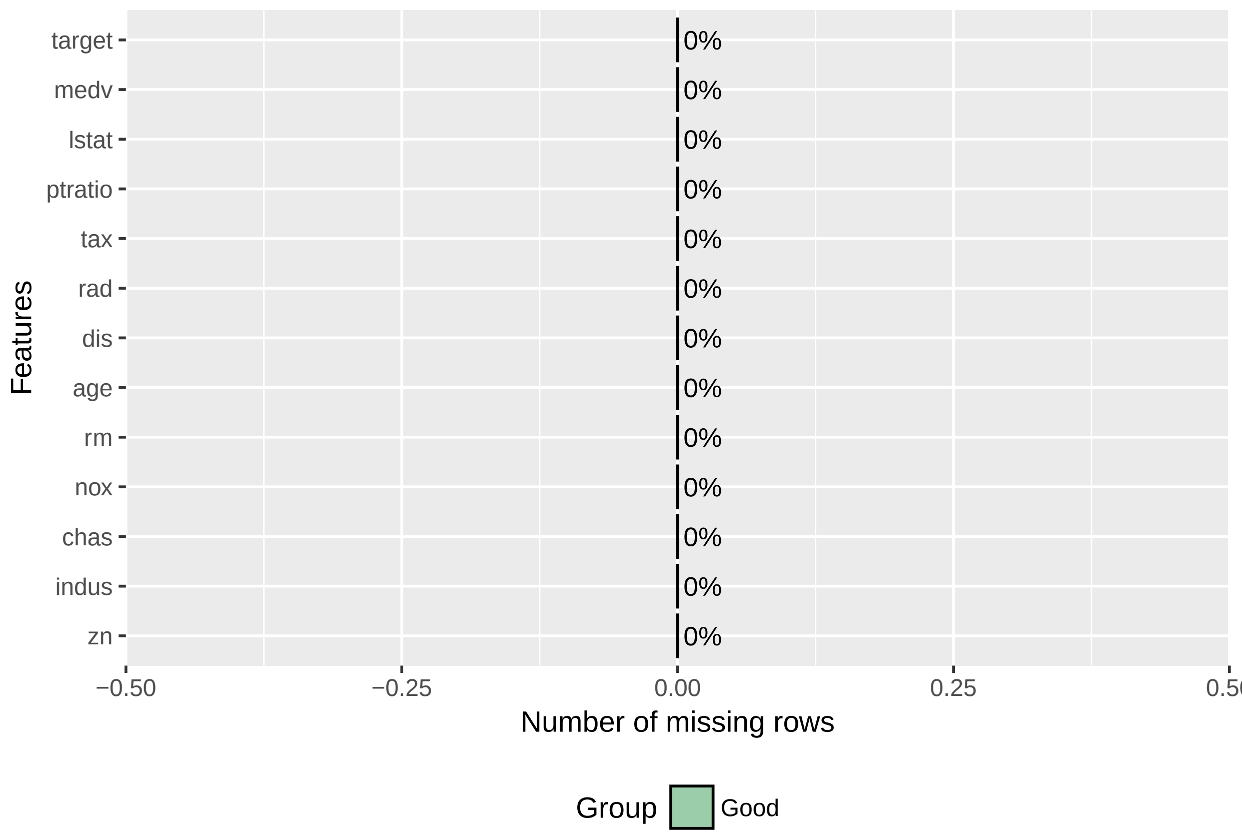
```
##    Variables      VIF
## 1         zn 2.323545
## 2      indus 4.120617
## 3       chas 1.090329
## 4        nox 4.504675
## 5         rm 2.354453
## 6        age 3.142118
## 7        dis 4.243532
## 8        rad 6.782250
## 9        tax 9.217602
## 10   ptratio 2.013194
## 11     lstat 3.650759
## 12      medv 3.667409
```

If we set our VIF threshold at 4, the following predictor variables are highly correlated.

Variable Name	VIF
indus	4.120617
dis	4.243532
nox	4.504675
rad	6.782250
tax	9.217602

Data Preparation

There are no NA values in the data; however, it is possible that zero values in a particular data set may be equivalent to missing information. For instance, we would not expect to see any observation where the average number of rooms per dwelling is equal to zero. We look at the dataset to determine if there are zero values for each variable and check for reasonableness.



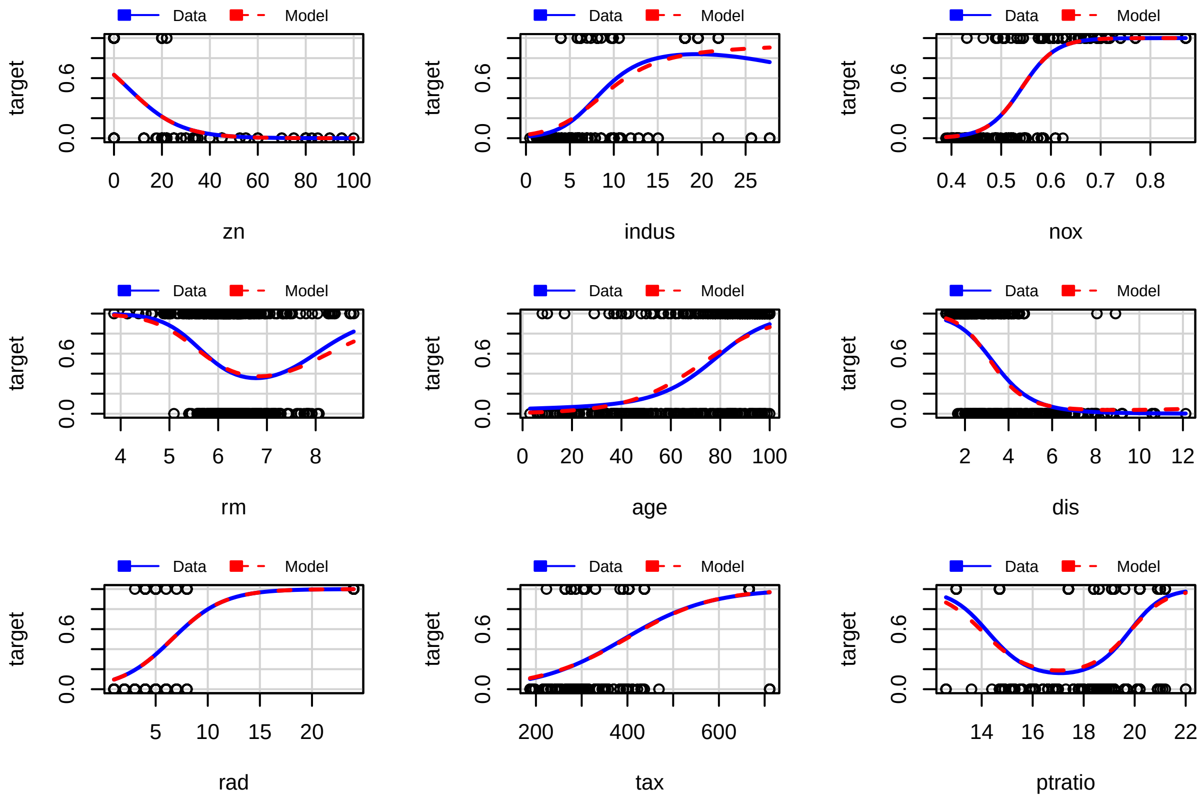
	x
zn	339
indus	0
chas	433
nox	0
rm	0
age	0
dis	0
rad	0
tax	0
ptratio	0
lstat	0
medv	0
target	237

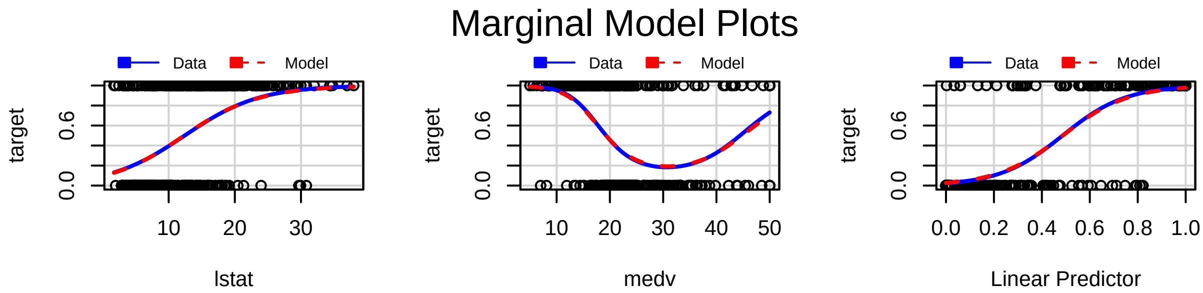
It is reasonable that there could be no land zoned for large lots (zn) in a particular suburb. The chas variable is a binary variable that tells us whether a suburb borders the Charles river, with zero meaning no, and the target variable is also binary. It is also feasible that the other variables would not necessarilly contain zero values. It appears that this data set did not contain any missing values.

Transformations

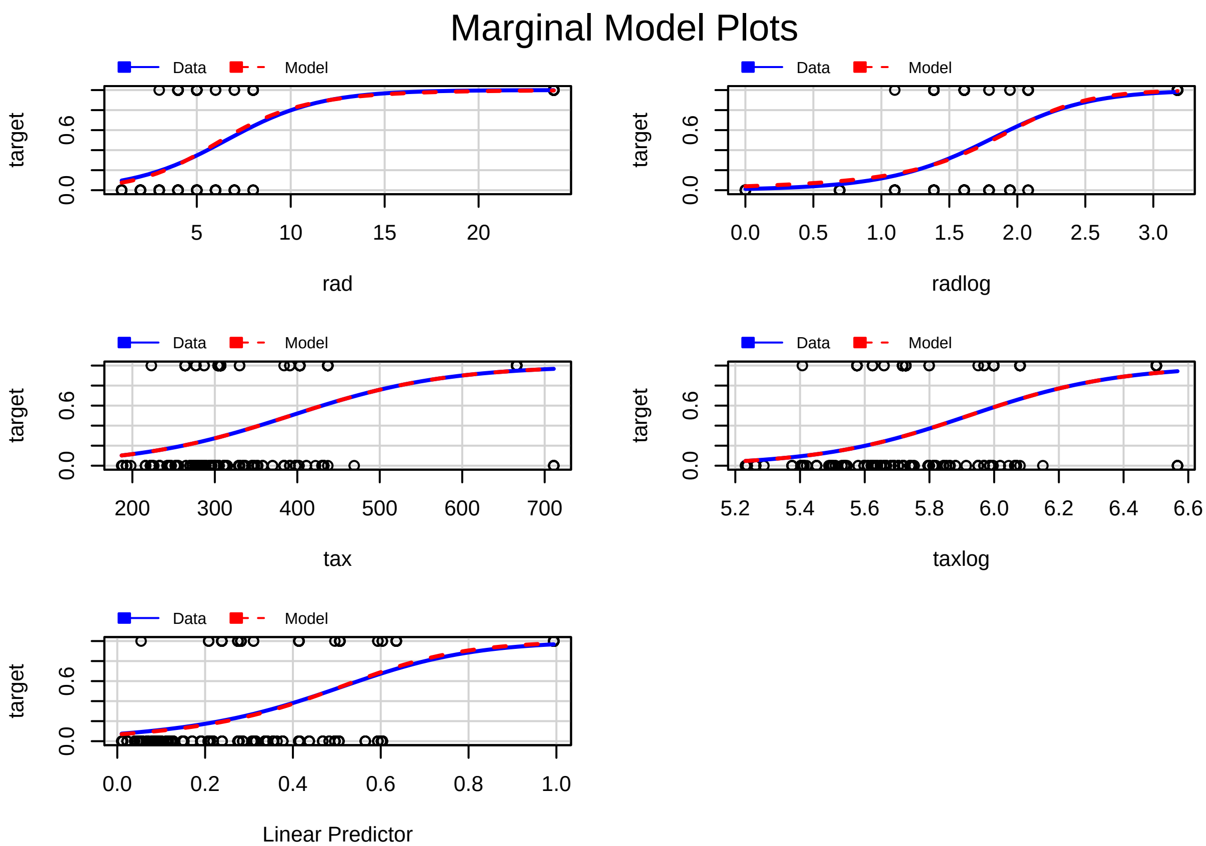
In the case of logistic regression, transformations are not necessary as normality of predictors is not required. We can compare the independent variable itself to the dependent variable using marginal model plots to help us determine if transformation improves the fit between the predictor and response.

https://www.researchgate.net/post/Should_I_transform_non-normal_independent_variables_in_logistic_regression





Two which stand out are rad (index of accessibility to radial highways) and tax (full-value property-tax rate per $10,000) which we can transform and then compare the use of the transformed variable and the original in our models.



It looks as though our fit has improved. We will determine if this improves the overall model in the next section.

Build Models

Model 1 : Baseline using all Predictor Variables

As a baseline, the first model build will be a logistic regression model using all predictor variables provided. No transformation has been performed on the predictor variables.

```
model1 <- glm(target ~ ., family=binomial(), data=dev_train)

summary(model1)
```

```
##
## Call:
## glm(formula = target ~ ., family = binomial(), data = dev_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.88220  -0.10094  -0.00031   0.00027   2.94182
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -49.184177   9.422981  -5.220 1.79e-07 ***
## zn           -0.076954   0.044453  -1.731 0.083427 .
```

```
## indus        -0.044532   0.064952  -0.686 0.492959
## chas          1.226574   1.082934   1.133 0.257366
## nox          52.509712  10.990233   4.778 1.77e-06 ***
## rm           -0.861188   0.913113  -0.943 0.345612
## age           0.064011   0.019536   3.277 0.001051 **
## dis           0.953227   0.295664   3.224 0.001264 **
## rad           0.976962   0.228521   4.275 1.91e-05 ***
## tax          -0.007383   0.003829  -1.928 0.053857 .
## ptratio       0.623825   0.180634   3.454 0.000553 ***
## lstat        -0.031103   0.064727  -0.481 0.630856
## medv          0.214037   0.088075   2.430 0.015091 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 118.33  on 314  degrees of freedom
## AIC: 144.33
##
## Number of Fisher Scoring iterations: 9
```

As we can see in our first model, `zn`, `indus`, `chas`, `rm`, `tax`, and `lstat` are not statistically significant. As for the statistically significant variables, `nox` and `rad` have the lowest p-values suggesting a strong association between nitrogen oxide concentration and accessibility to radial highways with the probability of crime rates above the median.

```r
exp(coef(model1))
```

```
##  (Intercept)           zn        indus         chas          nox
## 4.360972e-22 9.259326e-01 9.564453e-01 3.409529e+00 6.377907e+22
##           rm          age          dis          rad          tax
## 4.226597e-01 1.066104e+00 2.594068e+00 2.656375e+00 9.926441e-01
##      ptratio        lstat         medv
## 1.866052e+00 9.693761e-01 1.238669e+00
```

Recall that the estimates from logistic regression characterize the relationship between the predictor and response variable on a log-odds scale. This suggests that for every one unit increase in `nox`, the log-odds of the crime rate increases signficantly in magnitude. Access to radial highways, while not nearly to the same magnitude, also increases the the log-odds of crime above the median.

It is interesting to note that that `nox` is a significant predictor of crime by orders of magnitude when compared to the other significant predictors. NOx (nitrogen dioxide and nitric oxide) are typcially associated with smog and acid rain pollution. NOx has been linked to adverse health effects in humans.

AIC (Akaike Information Criterion) for Model 1 = 144.3266013
BIC (Bayesian Information Criterion) for Model 1 = 193.5960836

```
## Warning in Ops.factor(1, obs): '-' not meaningful for factors

## Warning in Ops.factor(1, obs): '-' not meaningful for factors
```

Model 2 : Baseline using Transformed Variables

In the data preparation section, the log transformation of the the `rad` and `tax` predictor variables where determined to be potentially beneficial transformations. This model will use those transformed variables and repeat the modeling process in Model 1.

```
model2 <- glm(target ~ .,  family=binomial(), data=dev_train_T)

summary(model2)
```

```
##
## Call:
## glm(formula = target ~ ., family = binomial(), data = dev_train_T)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2127  -0.0724   0.0000   0.0314   4.0565
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -32.82833   11.11838  -2.953  0.00315 **
## zn           -0.07365    0.04891  -1.506  0.13216
## indus        -0.12025    0.06622  -1.816  0.06939 .
## chas         -0.25948    0.98736  -0.263  0.79270
## nox          65.61314   12.79735   5.127 2.94e-07 ***
## rm           -0.45659    0.97326  -0.469  0.63898
## age           0.06588    0.02044   3.223  0.00127 **
## dis           0.71536    0.31357   2.281  0.02253 *
## rad           4.36904    1.05771   4.131 3.62e-05 ***
## tax          -3.94655    1.59989  -2.467  0.01363 *
## ptratio       0.44875    0.16876   2.659  0.00784 **
## lstat        -0.08538    0.08047  -1.061  0.28865
## medv          0.11886    0.09095   1.307  0.19123
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 117.83  on 314  degrees of freedom
## AIC: 143.83
##
## Number of Fisher Scoring iterations: 8
```

Contrasting against model 1, we now see that `nox`, `age`, and `rad` (log-transformed) are now the most statistically significant variables with `dis`, `tax` (log-transformed), and `ptratio` showing some significance but to a lesser degree.

Model 2 sees an uptick in significance in the tax variable, and the new taxlog variable has one of the lowest p-values suggesting a strong association between property tax rate and crime rates. Of interest here is that this is only predictor variable which is showing a log-odds decrease in crime for an unit increase in the tax rate.

`ptratio`, the pupil-teacher ratio by town, also saw an increase in significance when running model 2 with the transformed data.

```
exp(coef(model1))
```

```
##  (Intercept)           zn        indus         chas          nox
## 4.360972e-22 9.259326e-01 9.564453e-01 3.409529e+00 6.377907e+22
##           rm          age          dis          rad          tax
```

```
## 4.226597e-01 1.066104e+00 2.594068e+00 2.656375e+00 9.926441e-01
##      ptratio        lstat         medv
## 1.866052e+00 9.693761e-01 1.238669e+00
```

AIC (Akaike Information Criterion) for Model 2 = 143.8252129
BIC (Bayesian Information Criterion) for Model 2 = 193.0946951

```
## Warning in Ops.factor(1, obs): '-' not meaningful for factors

## Warning in Ops.factor(1, obs): '-' not meaningful for factors
```

Model 1 - Model 2 Comparison

Comparing the two models using a Chi-square test, there's no significance difference detected between the two. However, we do see that Model 2 resulted in a slightly lower AIC value. Consequently, further modeling will be based on the transformed dataset.

```r
anova(model1, model2,  test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ zn + indus + chas + nox + rm + age + dis + rad + tax +
##     ptratio + lstat + medv
## Model 2: target ~ zn + indus + chas + nox + rm + age + dis + rad + tax +
##     ptratio + lstat + medv
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       314     118.33
## 2       314     117.83  0  0.50139
```

Model 3 : AIC Stepwise Variable Selection

The third model used was a stepwise regression, and we chose to use both the "forward" and "backward" methods to obtain the optimal model. Since we chose to model forward with the transformed dataset we used it here as well.

After starting from nothing and adding variables one at a time, then repeating the process backwards starting with a full dataset and subracting variables one at a time, the ideal model chosen included `zn`, `indus`, `nox`, `age`, `dis`, `rad`, `tax`, `ptratio`, and `medv`, with `nox`, `age`, and `rad` having the most statistical significance as shown by the summary below.

```
## Start:  AIC=143.83
## target ~ zn + indus + chas + nox + rm + age + dis + rad + tax +
##     ptratio + lstat + medv
##
##           Df Deviance    AIC
## - chas     1   117.89 141.89
## - rm       1   118.05 142.04
## - lstat    1   118.96 142.96
## - medv     1   119.62 143.62
## <none>         117.83 143.82
## - zn       1   120.80 144.80
## - indus    1   121.47 145.47
## - dis      1   123.44 147.44
## - tax      1   123.91 147.91
## - ptratio  1   125.64 149.63
## - age      1   130.42 154.42
```

```
## - rad      1   155.39 179.39
## - nox      1   176.24 200.24
##
## Step:  AIC=141.89
## target ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio +
##     lstat + medv
##
##           Df Deviance    AIC
## - rm       1   118.16 140.16
## - lstat    1   119.14 141.14
## - medv     1   119.76 141.76
## <none>         117.89 141.89
## - zn       1   120.84 142.84
## + chas     1   117.83 143.82
## - indus    1   121.98 143.98
## - dis      1   123.57 145.57
## - tax      1   124.13 146.13
## - ptratio  1   126.51 148.51
## - age      1   130.48 152.48
## - rad      1   157.80 179.80
## - nox      1   177.45 199.45
##
## Step:  AIC=140.16
## target ~ zn + indus + nox + age + dis + rad + tax + ptratio +
##     lstat + medv
##
##           Df Deviance    AIC
## - lstat    1   119.15 139.15
## <none>         118.16 140.16
## - medv     1   120.34 140.34
## - zn       1   121.43 141.43
## + rm       1   117.89 141.89
## + chas     1   118.05 142.04
## - indus    1   122.16 142.16
## - dis      1   123.58 143.58
## - tax      1   124.79 144.79
## - ptratio  1   126.60 146.60
## - age      1   134.18 154.18
## - rad      1   157.88 177.88
## - nox      1   177.53 197.53
##
## Step:  AIC=139.15
## target ~ zn + indus + nox + age + dis + rad + tax + ptratio +
##     medv
##
##           Df Deviance    AIC
## <none>         119.15 139.15
## + lstat    1   118.16 140.16
## + chas     1   118.96 140.96
## - zn       1   123.07 141.07
## - indus    1   123.11 141.11
## + rm       1   119.14 141.14
## - medv     1   124.99 142.99
## - dis      1   125.25 143.25
```

```
## - tax      1   125.56 143.56
## - ptratio  1   127.11 145.11
## - age      1   134.18 152.18
## - rad      1   157.91 175.91
## - nox      1   177.58 195.58
```

```
##
## Call:
## glm(formula = target ~ zn + indus + nox + age + dis + rad + tax +
##     ptratio + medv, family = binomial(), data = dev_train_T)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1872  -0.0813  -0.0001   0.0296   3.9817
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -34.46743   10.67991  -3.227 0.001250 **
## zn           -0.08008    0.04784  -1.674 0.094155 .
## indus        -0.11946    0.06347  -1.882 0.059828 .
## nox          63.02414   12.02294   5.242 1.59e-07 ***
## age           0.05638    0.01594   3.537 0.000405 ***
## dis           0.70251    0.29494   2.382 0.017223 *
## rad           4.20004    0.97091   4.326 1.52e-05 ***
## tax          -3.79036    1.49790  -2.530 0.011391 *
## ptratio       0.41386    0.15243   2.715 0.006627 **
## medv          0.11537    0.05187   2.224 0.026132 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 119.15  on 317  degrees of freedom
## AIC: 139.15
##
## Number of Fisher Scoring iterations: 8
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: target
##
## Terms added sequentially (first to last)
##
##
##         Df Deviance Resid. Df Resid. Dev
## NULL                      326     453.24
## zn       1   83.250       325     369.99
## indus    1   53.372       324     316.62
## nox      1  128.592       323     188.03
## age      1    3.247       322     184.78
## dis      1    4.104       321     180.68
## rad      1   43.048       320     137.63
```

```
## tax      1    8.686       319     128.94
## ptratio  1    3.956       318     124.99
## medv     1    5.840       317     119.15
```

AIC (Akaike Information Criterion) for Model 2 = 139.1484546
BIC (Bayesian Information Criterion) for Model 2 = 177.0480563

```
## Warning in Ops.factor(1, obs): '-' not meaningful for factors

## Warning in Ops.factor(1, obs): '-' not meaningful for factors
```

Model 4 : Using VIF Reduction with Transformed Predictor Variables

Since multicollinearity was detected during the EDA phase, Model 4 will select meaningful variables using VIF reduction. The presence of multicollinearity among predictors can lead to overfitting so this modeling approach will attempt to limit that by reducing the predictor variables to those with lower magnitude VIF.

Calculating and reviewing VIF for the predictor variables (below):

```
##       zn    indus     chas      nox       rm      age      dis      rad
## 1.599503 2.968206 1.367180 4.525322 5.607293 2.596930 2.940839 2.975277
##      tax  ptratio    lstat     medv
## 3.592654 2.214175 2.932736 7.713861
```

We see that `nox`, `rm`, and `medv` have the high variance inflation factor. However, knowing the signifcance of nox, we'll keep this variable as a predictor and update the model to remove `rm` and `medv`.

In the summary of model 4, several variables are not statistically significant and will be dropped from the final model 4.

Dropped Variables
* zn
* chas
* dis
* ptratio
* lstat

```
##
## Call:
## glm(formula = target ~ indus + nox + age + rad + tax, family = binomial(),
##     data = dev_train_T)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2531  -0.1493  -0.0013   0.0376   3.2493
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -13.84525    6.08740  -2.274  0.02294 *
## indus        -0.12492    0.06029  -2.072  0.03828 *
## nox          50.04512   10.67525   4.688 2.76e-06 ***
## age           0.03943    0.01327   2.970  0.00297 **
## rad           3.66138    0.76378   4.794 1.64e-06 ***
## tax          -3.59863    1.24154  -2.899  0.00375 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 135.94  on 321  degrees of freedom
## AIC: 147.94
##
## Number of Fisher Scoring iterations: 8
```

AIC (Akaike Information Criterion) for Model 4 = 147.9402702
BIC (Bayesian Information Criterion) for Model 4 = 170.6800312

```
## Warning in Ops.factor(1, obs): '-' not meaningful for factors

## Warning in Ops.factor(1, obs): '-' not meaningful for factors
```

Model 5 : Using BestGlm using Transformed Predictors

In the final model build the `bestglm` R package is used to determine the best set of predictors using both AIC and BIC as selection criteria.

Using Alkaike Information Criterion (AIC)

```
## Morgan-Tatar search since family is non-gaussian.
```

Looking at the top 5 best models based on lowest AIC, the variables `zn`, `indus`, `nox`, `age`, `dis`, `rad`, `tax`, `ptratio`, and `medv` are selected. Top 5 models are shown below:

```
##      zn indus  chas  nox    rm  age  dis  rad  tax ptratio lstat  medv
## 1  TRUE  TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE    TRUE FALSE  TRUE
## 2  TRUE  TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE    TRUE  TRUE  TRUE
## 3  TRUE  TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE    TRUE  TRUE FALSE
## 4 FALSE  TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE    TRUE  TRUE FALSE
## 5  TRUE  TRUE  TRUE TRUE FALSE TRUE TRUE TRUE TRUE    TRUE FALSE  TRUE
##   Criterion
## 1  137.1485
## 2  138.1581
## 3  138.3375
## 4  138.5338
## 5  138.9625
```

The resulting model based on lowest AIC is not dissimilar from previous models. We see `nox`, `age`, and `rad` (again log-transformed) as the most significatn predictors.

```
##
## Call:
## glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1872  -0.0813  -0.0001   0.0296   3.9817
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -34.46743   10.67991  -3.227 0.001250 **
## zn           -0.08008    0.04784  -1.674 0.094155 .
## indus        -0.11946    0.06347  -1.882 0.059828 .
```

```
## nox          63.02414   12.02294   5.242 1.59e-07 ***
## age           0.05638    0.01594   3.537 0.000405 ***
## dis           0.70251    0.29494   2.382 0.017223 *
## rad           4.20004    0.97091   4.326 1.52e-05 ***
## tax          -3.79036    1.49790  -2.530 0.011391 *
## ptratio       0.41386    0.15243   2.715 0.006627 **
## medv          0.11537    0.05187   2.224 0.026132 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 119.15  on 317  degrees of freedom
## AIC: 139.15
##
## Number of Fisher Scoring iterations: 8
```

Using Bayesian Information Criterion (BIC)

Calculate the best set of predictors using Bayesian Information Criterion (BIC). The model with the loweset BIC will be selected.

```
## Morgan-Tatar search since family is non-gaussian.
```

Looking at the top 5 best models based on lowest BIC, the variables `indus`, `nox`, `age`, `rad`, and `tax` are selected. Top 5 models are shown below:

```
##      zn indus  chas  nox    rm  age   dis  rad  tax ptratio lstat  medv
## 1 FALSE FALSE FALSE TRUE FALSE TRUE FALSE TRUE TRUE    TRUE FALSE FALSE
## 2 FALSE  TRUE FALSE TRUE FALSE TRUE FALSE TRUE TRUE    TRUE FALSE FALSE
## 3 FALSE FALSE FALSE TRUE FALSE TRUE FALSE TRUE TRUE   FALSE FALSE FALSE
## 4 FALSE  TRUE FALSE TRUE FALSE TRUE FALSE TRUE TRUE   FALSE FALSE FALSE
## 5 FALSE FALSE FALSE TRUE FALSE TRUE FALSE TRUE TRUE    TRUE  TRUE FALSE
##   Criterion
## 1  163.5534
## 2  163.8524
## 3  163.9032
## 4  164.8901
## 5  165.3483
```

It should be noted that this model based on BIC uses the fewest number of predictors compared to the other model builds. The inclusion of the `indus` variable has a marginal affect on BIC so for simplicity of the second best model will be used.

```r
model5.bic <- glm(target ~ nox + age + rad + tax, family=binomial(), data=dev_train_T)

summary(model5.bic)
```

```
##
## Call:
## glm(formula = target ~ nox + age + rad + tax, family = binomial(),
##     data = dev_train_T)
##
## Deviance Residuals:
```

```
##      Min        1Q    Median        3Q       Max
## -1.93712  -0.17367  -0.00171   0.06190   3.05710
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.43708    4.21211  -1.291  0.19677
## nox         36.91089    6.98000   5.288 1.24e-07 ***
## age          0.03843    0.01301   2.954  0.00313 **
## rad          3.99417    0.77465   5.156 2.52e-07 ***
## tax         -4.13436    1.09974  -3.759  0.00017 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 140.74  on 322  degrees of freedom
## AIC: 150.74
##
## Number of Fisher Scoring iterations: 8
```

The resulting BIC model uses `nox`, `age`, `rad`, and `tax` as the final set of predictors. All are statistically significant.

```
## Warning in Ops.factor(1, obs): '-' not meaningful for factors

## Warning in Ops.factor(1, obs): '-' not meaningful for factors

## Warning in Ops.factor(1, obs): '-' not meaningful for factors

## Warning in Ops.factor(1, obs): '-' not meaningful for factors
```

Model Selection and Evaluation

Model Selection

We will use a structured evaluation of the models on validation data set (we split our training data set between a training set and a model evaluation set) with regards to:
* (i) parsimonious fit,
* (ii) goodness-of-fit,
* (iii) predictive accuracy, and
* (iv) more subjectively satisfying business requirements

(i) Parsimony

Parismonous models have optimal parsimony, or just the right amount of predictors needed to explain the model well. There is generally a tradeoff between goodness-of-fit and parsimony: low parsimony models then to have a better fit than high parsimony models.

We will use Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC)

BIC = LN(number of observations) * number of variables in your model- 2 Log Likelihood
AIC = 2*number of variables in your model = 2 Log Likelihood

(ii) Goodness-of-fit

the Goodness-of-fit of a model describes how well it fits a set of observations. Measures of goodness of fit typically summarize the discrepancy between observed values and the values expected under the model in question.

We will use McFadden's R^2 and the Hosmer-Lemeshow test

McFadded's R^2: Higher value (0.2 to 0.4) indicates a good fit

Hosmer_Lemeshow Test: Small values with large p-values indicate a good fit to the data while large values with p-values below 0.05 indicate a poor fit.

(iii) Predictive accuracy

Predictive accuracy of a model is how well a model is predicting correctly the outcome and also a measure of the incorrect predictions.

We will use Cohen's Kappa (or Kappa), Youden's Index, F1_Score, Percentage of False Positive, and AUC/ROC Curves

Kappa
Kappa takes into account the accuracy that would be generated purely by chance. The form of the measure is:
$Kappa = \frac{Total \quad Accuracy \quad - \quad Random \quad Accuracy}{1 \quad - \quad Random \quad Accuracy}$ where,
$Total \quad Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
and
$Random \quad Accuracy = \frac{(TN+FP)(TN+FN) \quad + \quad (FN+TP)(FP+TP)}{(TP+TN+FP+FN)^2}$

Kappa takes on values from -1 to +1, with a value of 0 meaning there is no agreement between the actual and classified classes. A value of 1 indicates perfect concordance of the model prediction and the actual classes and a value of ???1 indicates total disagreement between prediction and the actual

Youden's Index

Youden's index evaluates the ability of a classifier to avoid misclassifications. This index puts equal weights on a classifier's performance on both the positive and negative cases.
Thus:
$Youden's \quad Index \quad (\gamma) \quad = \quad Sensitivity \quad - \quad (1 - Specificity)$

We selected to look at False Positive instead of classification error rate since we think this measure is better aligned with the business requirements.

```
kable(all_model_metrics)
```

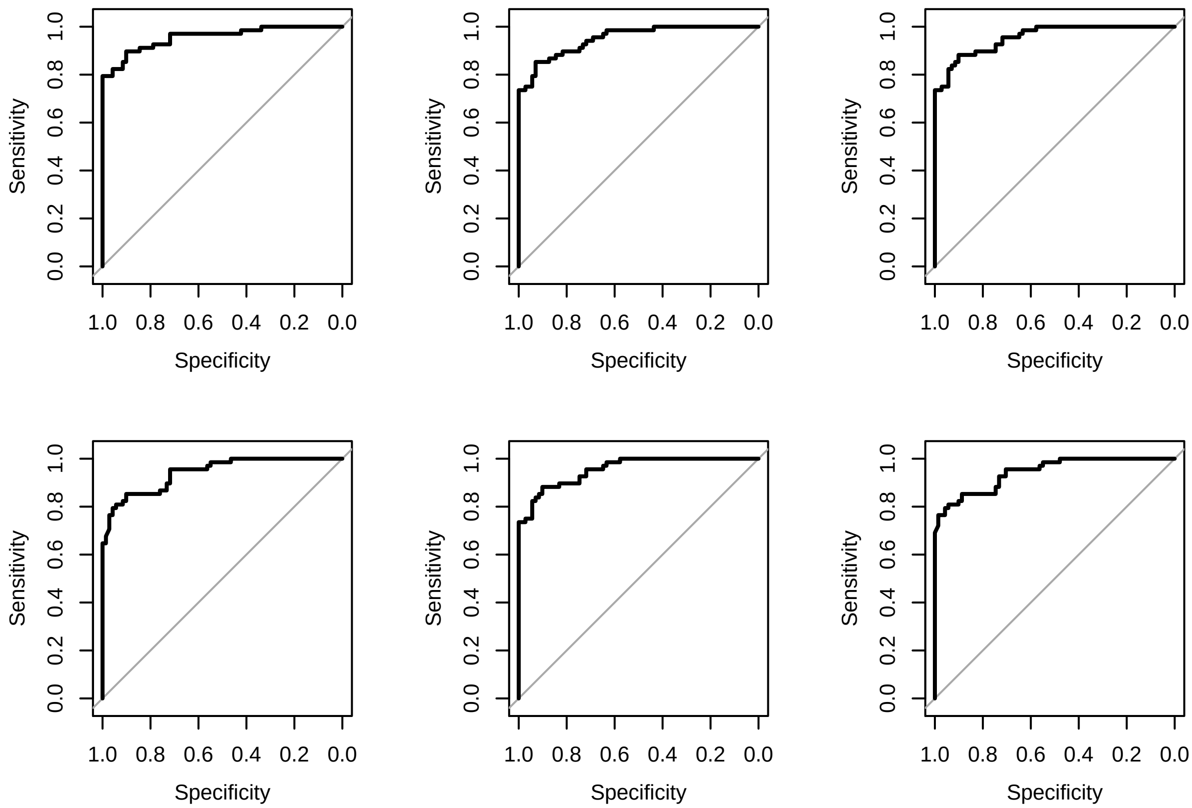
Model	AIC	BIC	McFadenR2	HL_Chi	HL_p	X.	Kappa	Youden	F1Score	FPPrct	AUC
Model1	144.327	193.596	0.739	327	0	*	0.755	0.757	0.872	7.19	0.877
Model2	143.825	193.095	0.739	327	0	*	0.755	0.757	0.872	7.19	0.877
Model3	139.148	177.048	0.739	327	0	*	0.755	0.757	0.872	7.19	0.877
Model4	147.940	170.680	0.739	327	0	*	0.740	0.747	0.862	8.63	0.870
Model5.AIC	139.148	177.048	0.739	327	0	*	0.755	0.757	0.872	7.19	0.877
Model5.BIC	150.743	169.693	0.739	327	0	*	0.712	0.718	0.846	9.35	0.855

From the various measurements matrix, we noticed that some of the measures do not come into play since they do not diffrentiate any of our models: McFaren R^2 and Hosmer-Lemeshow test.
The remaining measures clearly indicate that Model3 and Model5.AIC are superior models.

Let us now consider the ROC curves for all the models.

```
par(mfrow=c(2,3))

plot.roc(all_roc_curves[[1]])
plot.roc(all_roc_curves[[2]])
plot.roc(all_roc_curves[[3]])
plot.roc(all_roc_curves[[4]])
plot.roc(all_roc_curves[[5]])
plot.roc(all_roc_curves[[6]])
```



The side by side comparaison of the ROC curve is showing the tread-off between Sensitivity and Specificity. The closer the area under to 1, the better fit of the model. The ROC Curves plot support our selection of Model or Model 5

We will compare the 2 models.

```
summary(model3)
```

```
##
## Call:
## glm(formula = target ~ zn + indus + nox + age + dis + rad + tax +
##     ptratio + medv, family = binomial(), data = dev_train_T)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1872  -0.0813  -0.0001   0.0296   3.9817
##
```

```
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -34.46743   10.67991  -3.227 0.001250 **
## zn           -0.08008    0.04784  -1.674 0.094155 .
## indus        -0.11946    0.06347  -1.882 0.059828 .
## nox          63.02414   12.02294   5.242 1.59e-07 ***
## age           0.05638    0.01594   3.537 0.000405 ***
## dis           0.70251    0.29494   2.382 0.017223 *
## rad           4.20004    0.97091   4.326 1.52e-05 ***
## tax          -3.79036    1.49790  -2.530 0.011391 *
## ptratio       0.41386    0.15243   2.715 0.006627 **
## medv          0.11537    0.05187   2.224 0.026132 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 119.15  on 317  degrees of freedom
## AIC: 139.15
##
## Number of Fisher Scoring iterations: 8
```

```r
summary(model5.aic)
```

```
##
## Call:
## glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1872  -0.0813  -0.0001   0.0296   3.9817
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -34.46743   10.67991  -3.227 0.001250 **
## zn           -0.08008    0.04784  -1.674 0.094155 .
## indus        -0.11946    0.06347  -1.882 0.059828 .
## nox          63.02414   12.02294   5.242 1.59e-07 ***
## age           0.05638    0.01594   3.537 0.000405 ***
## dis           0.70251    0.29494   2.382 0.017223 *
## rad           4.20004    0.97091   4.326 1.52e-05 ***
## tax          -3.79036    1.49790  -2.530 0.011391 *
## ptratio       0.41386    0.15243   2.715 0.006627 **
## medv          0.11537    0.05187   2.224 0.026132 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 119.15  on 317  degrees of freedom
## AIC: 139.15
##
## Number of Fisher Scoring iterations: 8
```

Side to side comparaison relevals that these two model are actually the same. Since both model were built based on best AIC score, this is understandable.

We will recommand one of them as our best model; model5.AIC.

Evaluation

We will now run our model against our evaluation data set. However, before we can do so, we need to transform our evaluation data set since Model5.AIC

Load & Transformation of Data Set

```
# Loading and transforming Evaluation Data Set
crime_EvalData <- read.csv("https://raw.githubusercontent.com/621-Group2/HW3/master/crime-evaluation-da

crime_EvalData$age <- round(crime_EvalData[,6], digits = 0)

summary(crime_EvalData)
```

```
##        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.   :  7.00   Min.   :1.202   Min.   : 1.000
##  1st Qu.:5.874   1st Qu.: 56.75   1st Qu.:2.041   1st Qu.: 4.000
##  Median :6.143   Median : 83.00   Median :3.373   Median : 5.000
##  Mean   :6.214   Mean   : 71.00   Mean   :3.787   Mean   : 9.775
##  3rd Qu.:6.532   3rd Qu.: 93.00   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
```

```
#copy Evaluation Data Set prior to transformation
crime_EvalDataT <- crime_EvalData

# Apply Log Transform
crime_EvalDataT$radlog <- log(crime_EvalDataT$rad)
crime_EvalDataT$taxlog <- log(crime_EvalDataT$tax)
```

We will now run the prediction on our transformed evaluation data set. We will write the results to a .csv file.

```
pred_model_final <- predict(model5.aic, crime_EvalDataT, type = 'response')
y_pred_model_final <- as.factor(ifelse(pred_model_final > 0.5, 1, 0))
```

Our predictions indicates that all the neighboord reprensented in the evaluation set would be flag with low crime rate (below the median crime rate).

```
write.csv(as.data.frame(y_pred_model_final), file = "group2_project3_results.csv", row.names=FALSE)
```

Conclusion

As we approach this problem and explore the data and relationships between predictors, we did not think that there were any variables that could be derived to be used as additional predictors. Neither the training or evaluation data set had any missing data and we applyed a few transformation to improve the distrubtion of the most skewed predictors without making the final model to difficult to interpret.

We are confident in our approach to split the trainning data set to reserve an subset to evaluate each model and use predictive measures to help select the best model. We are confident that we have done so, in spite of the results of the final prediction.

We feel that possible overfitting has been balance with including parsimonous measures in the model selection process and is aliviated by knowing that our final model used AIC to guide the predictors inclusion process.

One area of conern, is that our test-evaluation data set happen to provide results that are not applicable to another evaluation set. This could have been aliviate by adopting a K-Fold Cross Validation method with randomization to prevent overfitting.