Homework 03

CUNY MSDS DATA 621

Duubar Villalobos Jimenez mydvtech@gmail.com

November 04, 2018

Contents

1 HOMEWORK #3

1.1 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).

1.2 Objective

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).

1.3 Description

Below is a short description of the variables of interest in the data set:

Type	Variable	Description
Predictor	zn	Proportion of residential land zoned for large lots (over 25000 square feet)
Predictor	indus	Proportion of non-retail business acres per suburb.
Predictor	chas	Dummy var. for whether the suburb borders the Charles River (1) or not (0).
Predictor	nox	Nitrogen oxides concentration (parts per 10 million).
Predictor	rm	Average number of rooms per dwelling.
Predictor	age	Proportion of owner-occupied units built prior to 1940.
Predictor	dis	Weighted mean of distances to five Boston employment centers.
Predictor	rad	Index of accessibility to radial highways.
Predictor	tax	Full-value property-tax rate per $10,000.
Predictor	ptratio	Pupil-teacher ratio by town.
Predictor	black	1000(B k - 0.63) 2 where B k is the proportion of blacks by town.
Predictor	lstat	Lower status of the population (percent).
Predictor	medv	Median value of owner-occupied homes in $1000s.
Response	target	Whether the crime rate is above the median crime rate (1) or not (0)

1.4 Deliverables

Upon following the instructions below, use your created R functions and the other packages to generate the classification metrics for the provided data set. A write-up of your solutions submitted in PDF format.

2 DATA EXPLORATION

2.1 Data acquisition

For reproducibility purposes, I have included the original data sets in my GitHub account, I will read it as a data frame from that location.

```
train.data <- paste(git_user, git_dir, "crime-training-data.csv", sep = "")
eval.data <- paste(git_user, git_dir, "crime-evaluation-data.csv", sep = "")
```

```
crime.train <- read.csv(train.data)
crime.eval <- read.csv(eval.data)
```

2.2 Simple Example

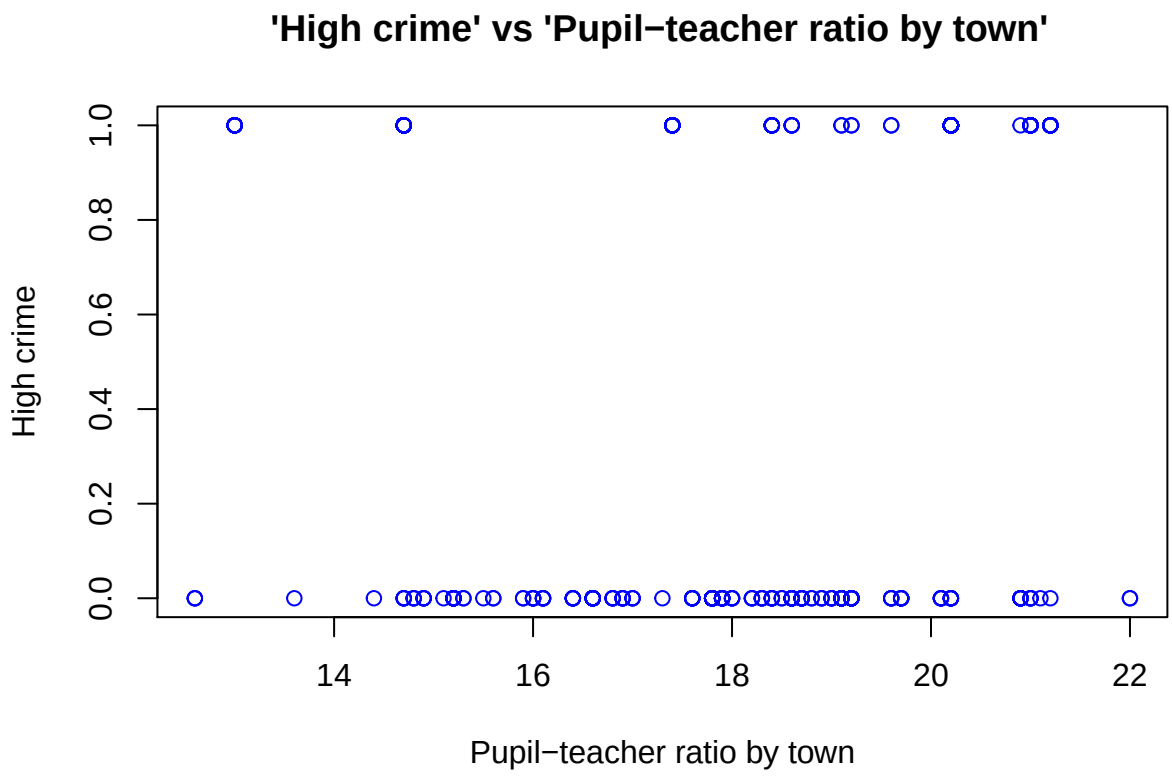
This example will help determine the ideas to follow in order to solve our problem; this is for explanatory purposes on how this problem will be approached. This example will predict the `target` based on the `ptratio`.

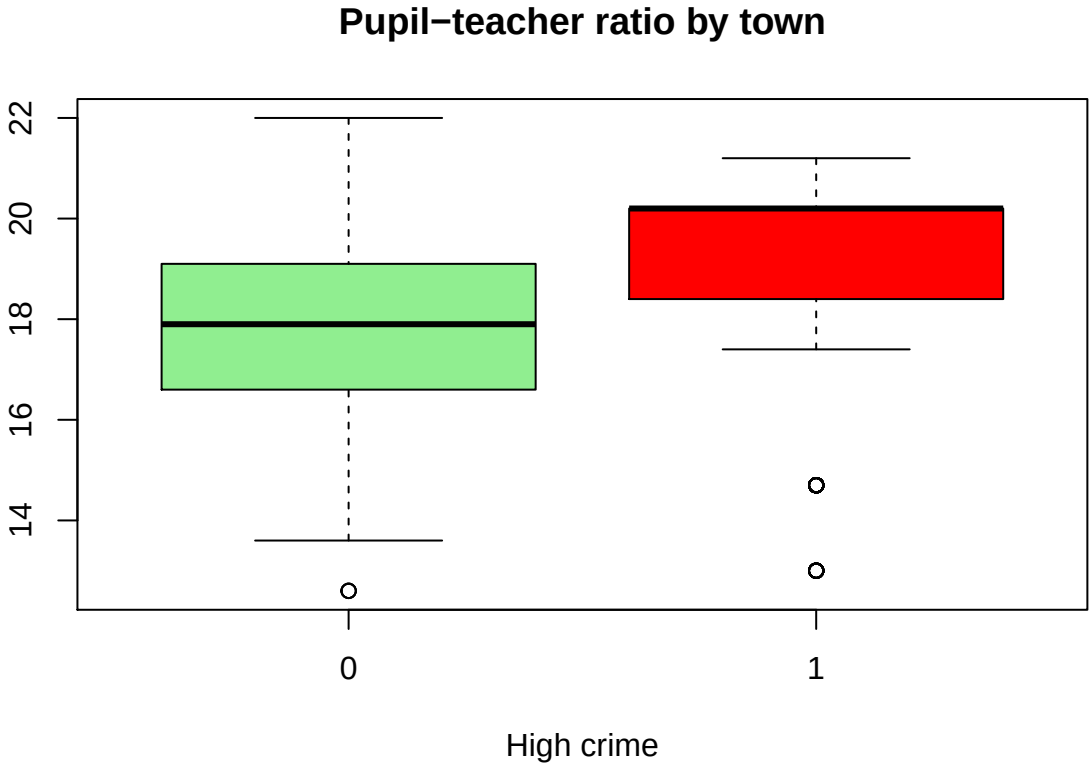
```
glm.tr <- glm(target ~ ptratio, data = crime.train)
summary(glm.tr)
```

```
##
## Call:
## glm(formula = target ~ ptratio, data = crime.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6972  -0.4629  -0.2401   0.4056   0.8171
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.55998    0.18969  -2.952  0.00332 **
## ptratio      0.05715    0.01024   5.582 4.05e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.2352091)
##
##     Null deviance: 116.47  on 465  degrees of freedom
## Residual deviance: 109.14  on 464  degrees of freedom
## AIC: 652.01
##
## Number of Fisher Scoring iterations: 2
```

Since this is just a very simple example, I would not describe much to it at this point in time; other that the predicted model will include $\beta_0 = -0.55998$ for the intercept and $\beta_1 = 0.05715$ for the rate of change.

Let's visualize this example:





From that simple example we could make some inferences such as it seems that the higher the *Pupil-teacher ratio by town* could influence in *High crime*; this could make sense in the real world since teachers aren't able to provide more individualized education techniques when group sizes are bigger, thus reducing quality education time per student. But yet again, this is just an example on how one predictor could influence in this particular case.

2.3 General exploration

The below process will help us obtain insights from the data.

2.3.1 Dimensions

Let's see the dimensions of our training data set.

Records	Variables
466	14

As we can notice, the training data set has a total of 466 different records and 14 variables including the **target** variable corresponding to *high crime*.

2.3.2 Structure

The below structure is currently present in the data, for simplicity purposes, I have previously loaded and treated this data set as a data frame in which all the variables with decimals are numeric.

```
## 'data.frame':    466 obs. of  14 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 ...
##  $ black  : num  369 397 387 375 394 ...
##  $ 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 ...
```

2.3.3 Summary

Let's find some summary statistics about our given data.

	Length	Class	Mode
zn	466	-none-	numeric
indus	466	-none-	numeric
chas	466	-none-	numeric
nox	466	-none-	numeric
rm	466	-none-	numeric
age	466	-none-	numeric
dis	466	-none-	numeric
rad	466	-none-	numeric
tax	466	-none-	numeric
ptratio	466	-none-	numeric

	Length	Class	Mode
black	466	-none-	numeric
lstat	466	-none-	numeric
medv	466	-none-	numeric
target	466	-none-	numeric

Let's get a little bit more insights for all the columns including the *target* variable.

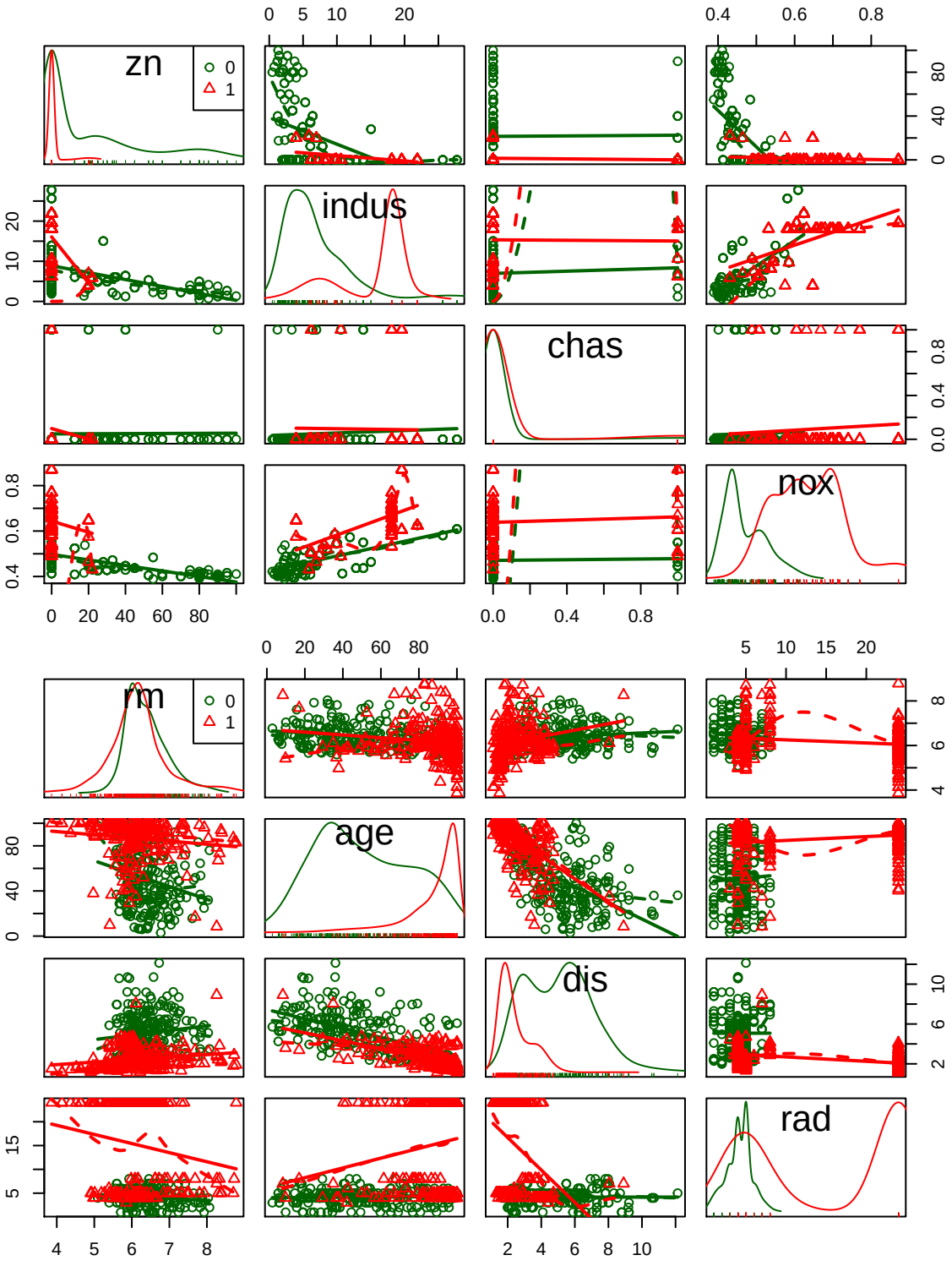
	Min	1st Qu	Median	Mean	3rd Qu	Max
zn	0.000	0.000	0.000	11.58000	16.250	100.000
indus	0.460	5.145	9.690	11.10500	18.100	27.740
chas	0.000	0.000	0.000	0.07082	0.000	1.000
nox	0.389	0.448	0.538	0.55430	0.624	0.871
rm	3.863	5.887	6.210	6.29100	6.630	8.780
age	2.900	43.880	77.150	68.37000	94.100	100.000
dis	1.130	2.101	3.191	3.79600	5.215	12.127
rad	1.000	4.000	5.000	9.53000	24.000	24.000
tax	187.000	281.000	334.500	409.50000	666.000	711.000
ptratio	12.600	16.900	18.900	18.40000	20.200	22.000
black	0.320	375.610	391.340	357.12000	396.240	396.900
lstat	1.730	7.043	11.350	12.63100	16.930	37.970
medv	5.000	17.020	21.200	22.59000	25.000	50.000
target	0.000	0.000	0.000	0.49140	1.000	1.000

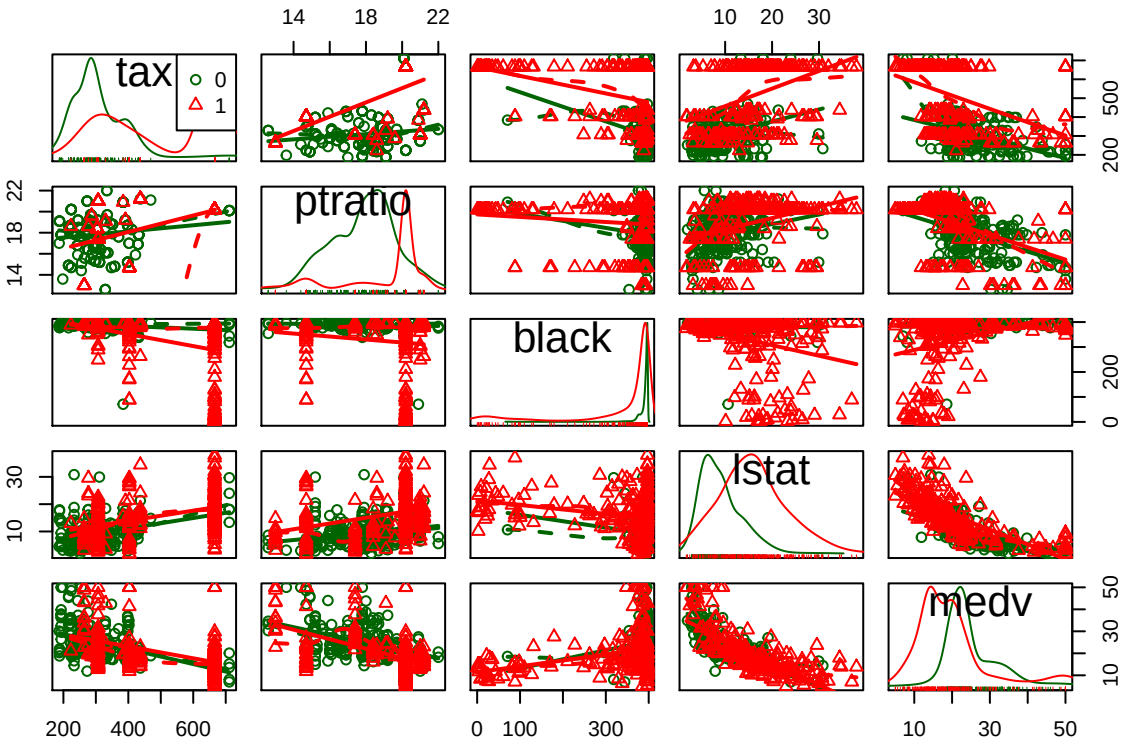
2.3.4 Missing data

Fortunately from the above statistics summary, it seems that we don't need to worry about missing values or **NA**, since no reports are given for that category.

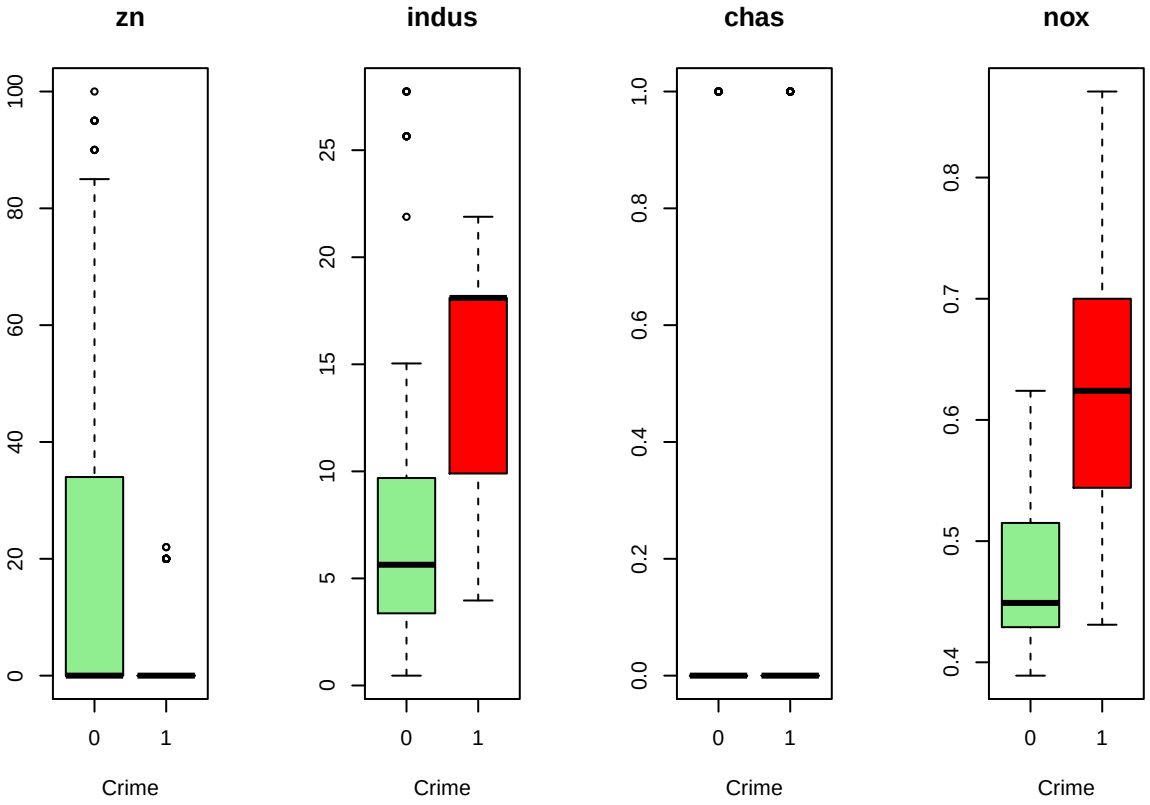
2.3.5 Visualizations

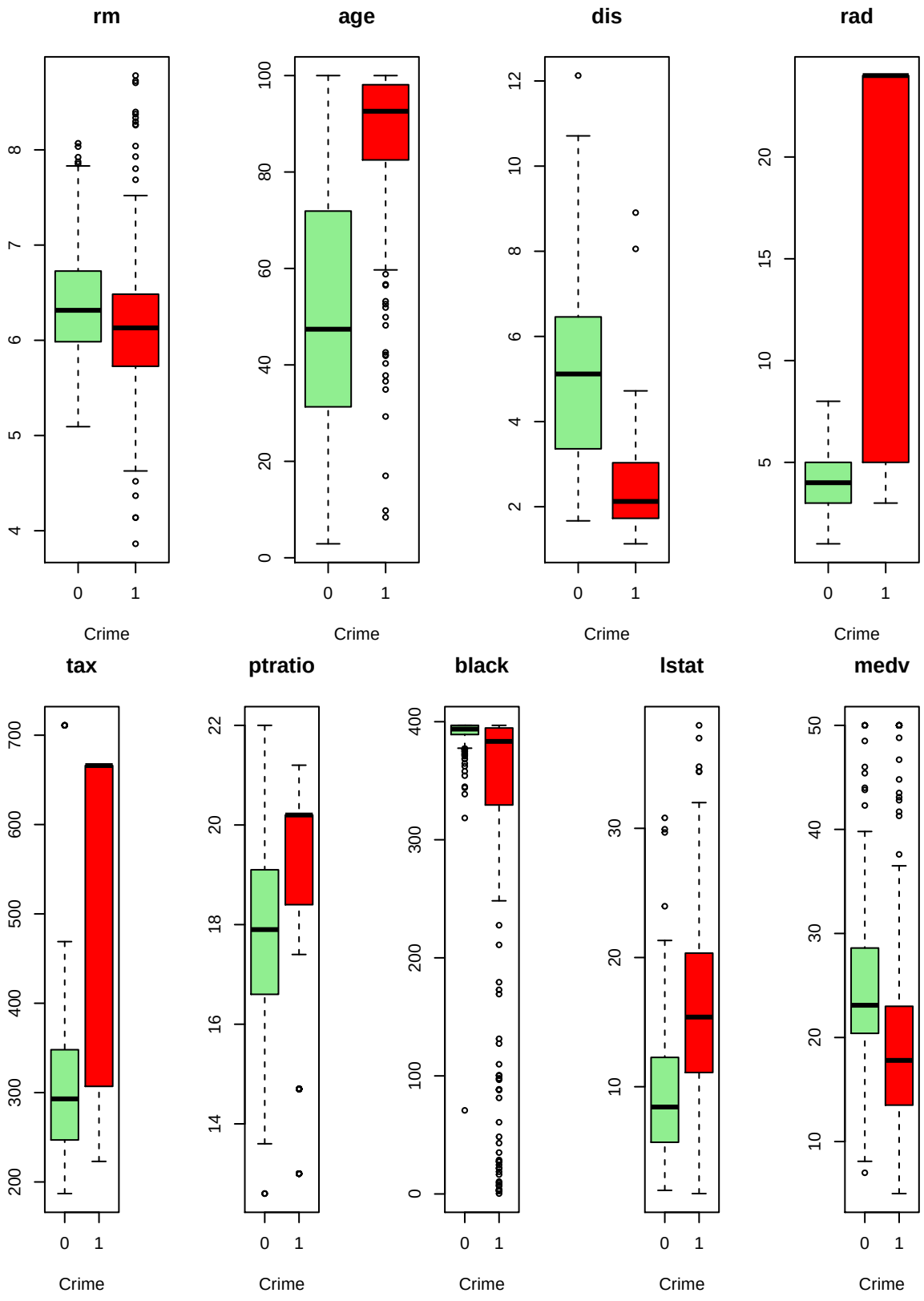
In the below graphs, the colors indicate that any record not including a high crime shows a green circle, while a record indicating a high crime has been plot in a red triangle. The diagonal plots the empirical distribution for both classes.


zn
indus
chas
nox
rm
age
dis
rad
0
1



Let's separate our data for visualization purposes.




rm
age
dis
rad
tax
ptratio
black
lstat
medv
Crime
0
1

2.3.6 Count values

Let's have a small understanding on how many records were categorized as 0 and how many as 1.

target	Counts	Percent
0	237	0.509
1	229	0.491

From the above results, we could assume that in effect the values seems to be uniformly distributed since almost half the data represent 0 and almost half represent 1.

2.3.7 Mean values

From the above graphs, we could notice how the means on both categories seem to have different values.

Let's try to calculate their respective mean values.

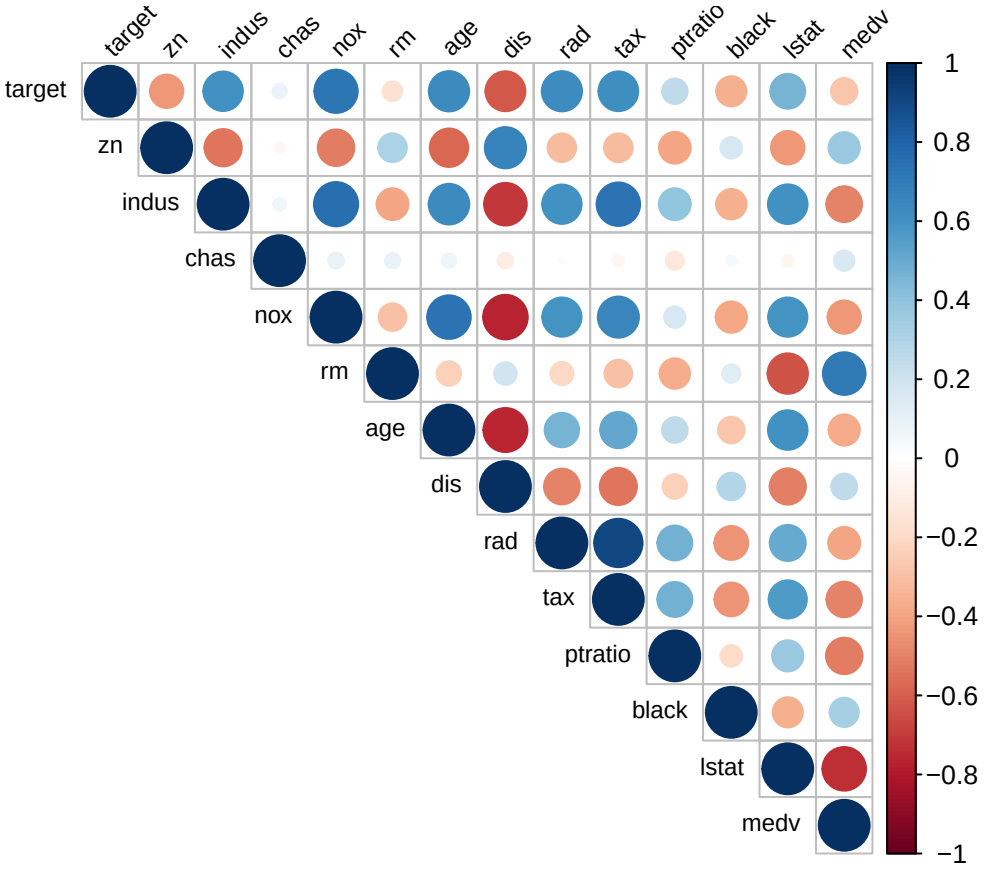
	Low Crime	High Crime	Insights
zn	21.48	1.33	94 % lower
indus	7.04	15.31	118 % higher
chas	0.05	0.09	81 % higher
nox	0.47	0.64	36 % higher
rm	6.40	6.18	3 % lower
age	50.84	86.50	70 % higher
dis	5.08	2.47	51 % lower
rad	4.17	15.07	261 % higher
tax	308.75	513.77	66 % higher
ptratio	17.86	18.96	6 % higher
black	388.77	324.36	17 % lower
lstat	9.36	16.02	71 % higher
medv	25.04	20.05	20 % lower

From the above table, we can easily identify how the mean values for the respective categories differ from one another and by how much. Also, we can quickly identify how the respective percentages compare to one another.

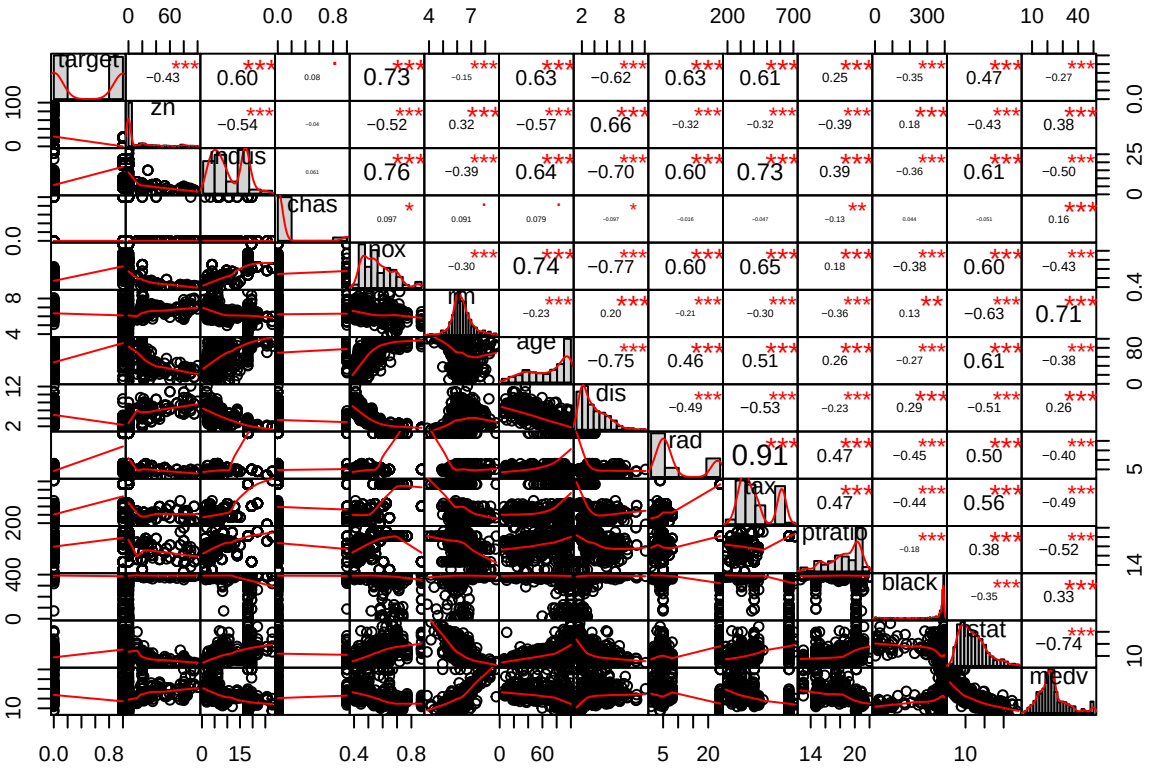
2.3.8 Correlations

Let's create some visualizations for the correlation matrix.

2.3.8.1 Graphical visualization



2.3.8.2 Numerical visualization



From the above graphs, we can easily identify some strong correlations in between the response variable `target` and other variables.

Let's read our correlations table to gain extra insights.

	target
target	1.0000000
zn	-0.4316818
indus	0.6048507
chas	0.0800419
nox	0.7261062
rm	-0.1525533
age	0.6301062
dis	-0.6186731
rad	0.6281049
tax	0.6111133
ptratio	0.2508489
black	-0.3529568
lstat	0.4691270
medv	-0.2705507

As we can easily check the above results, there seems to have considerable correlations in between our `target` variable among other given variables.

Something interesting to note from the above graph, is that we can easily visualize some sort of strong positive correlation in between variables; for example: `tax` seems to be strongly positively correlated to `ptratio`; in this case, their correlation values will be: 0.9064632; so I will keep this in mind in case of multivariate co-linearity.

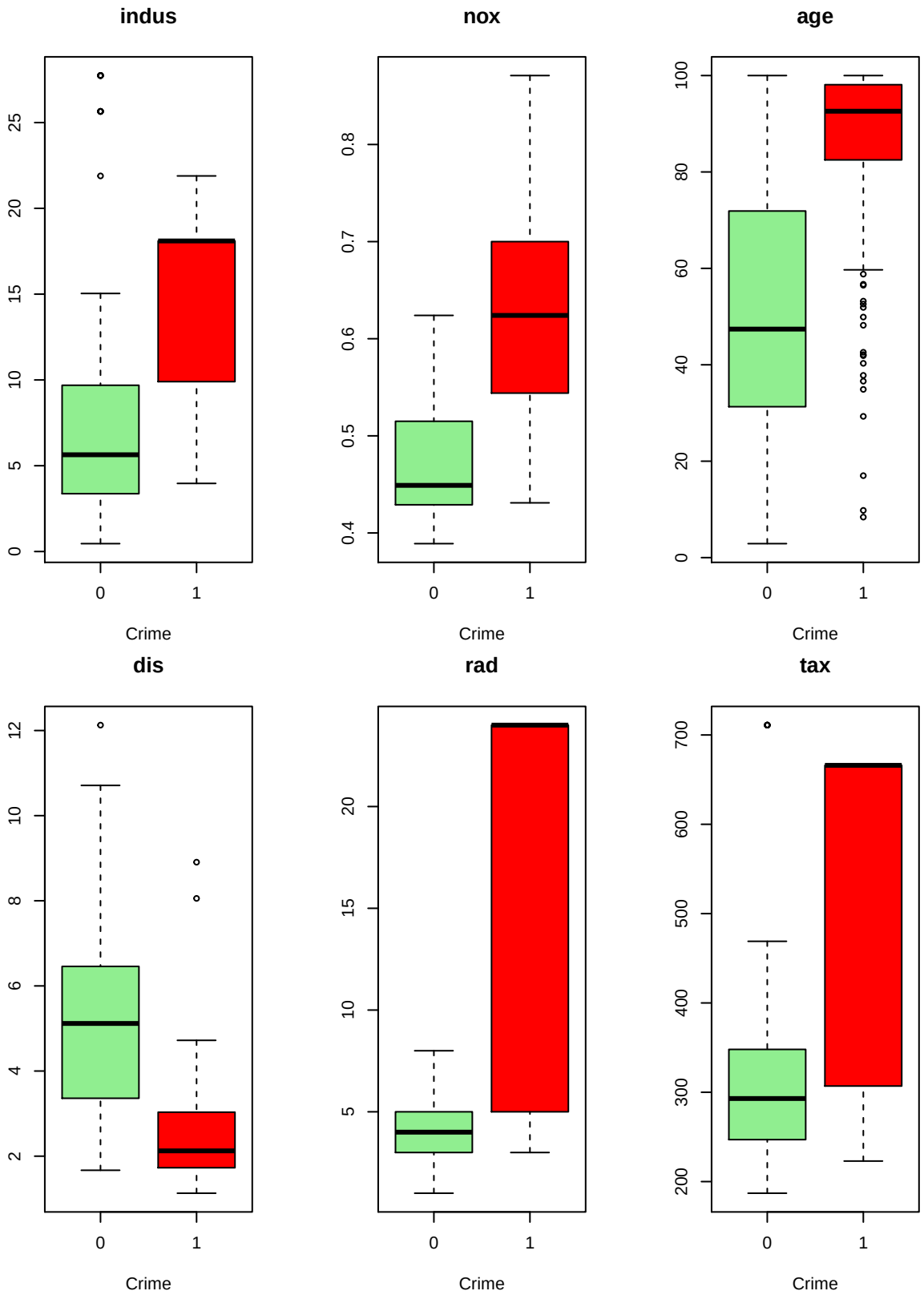
3 DATA PREPARATION

From the correlations table, we could focus on the variables that contain the strongest correlations related to our `target` variable; in this case, I will set my cut off at with any correlation in which the absolute value will be higher than 0.5.

	target
target	1.0000000
indus	0.6048507
nox	0.7261062
age	0.6301062
dis	-0.6186731
rad	0.6281049
tax	0.6111133

As we can see, we have reduced our number of possible predictor in half. From now on, I will focus on these variables only. Notice how in this smaller table `ptratio` is not part of it? In this case, I will assume this to be correct avoiding co-linearity problems further down.

Let's recap our previous plots for those variables.



Let's recap the structure of the remaining variables:

```
str(reduced.train)
```

```
## 'data.frame':    466 obs. of  7 variables:
##  $ target: int  1 1 1 0 0 0 1 1 0 0 ...
##  $ indus : num  19.58 19.58 18.1 4.93 2.46 ...
##  $ nox   : num  0.605 0.871 0.74 0.428 0.488 0.52 0.693 0.693 0.515 0.392 ...
##  $ 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 ...
```

At this point, we are getting ready to start building models, however I would like to point out that in this case is a little bit difficult to determine what data transformation could be used in order to refine our models.

3.1 Binary Logistic Regression

I would like to point that since this work requires **Binary Logistic Regression**, we are going to be using the **logit** function as our Likelihood link function for Logistic Regression by assuming that it follows a binomial distribution as follows:

$$y_i|x_i \sim Bin(m_i, \theta(x_i))$$

so that,

$$P(Y_i = y_i|x_i) = \binom{m_i}{y_i}\theta(x_i)^{y_i}(1-\theta(x_i))^{m_i-y_i}$$

Now, in order to solve our problem, we need to build a linear predictor model in which the individual predictors that compose the response Y_i are all subject to the same q predictors $(x_{i1}, \ldots, x_{iq})$. Please note that the group of predictors, are commonly known as **covariate classess**. In this case, we need a model that describes the relationship of $x_1, \ldots, x_q$ to p. In order to solve this problem, we will construct a linear predictor model as follows:

$$\mathfrak{N}_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_q x_{iq}$$

3.2 Logit link function

In this case, since we need to set $\mathfrak{N}_i = p_i$; with $0 \leq p_i \leq 1$, I will use the *link function* g such that $\mathfrak{N}_i = g(p_i)$ with $0 \leq g^{-1}(\mathfrak{N}) \leq 1$ for any $\mathfrak{N}$. In order to do so, I will pick the **Logit** link function $\mathfrak{N} = log(p/(1-p))$.

An alternate way will be by employing the χ^2 Chi square distribution; for the purposes of this project, I will employ the use of the binomial distribution or the χ^2 depending on which one is a better choice, also I will assume that all Y_i are all independent of each other.

4 BUILD MODELS

The following will be the methods employed in order to build our model.

4.1 NULL Model

In this section I will build a **Binary Logistic Regression** Null model utilizing all the variables and data, please note that I won't do any transformations. This model will be considered to be valid and will be considered as we advance.

```
Model_NULL <- glm(target ~ 1,
              data = crime.train,
              family = binomial(link ="logit"))

summary(Model_NULL)
```

```
##
## Call:
## glm(formula = target ~ 1, family = binomial(link = "logit"),
##     data = crime.train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.163  -1.163  -1.163   1.192   1.192
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.03434    0.09266  -0.371    0.711
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 645.88  on 465  degrees of freedom
## AIC: 647.88
##
## Number of Fisher Scoring iterations: 3
```

I will assume that this to be a valid model.

4.2 FULL Model

In this section I will build a **Binary Logistic Regression** Full model utilizing all the variables and data, please note that I won't do any transformations. This model will be considered to be valid and will be considered as we advance.

```
Model_FULL <- glm(target ~ .,
              data = crime.train,
              family = binomial(link ="logit"))

summary(Model_FULL)
```

```
##
## Call:
## glm(formula = target ~ ., family = binomial(link = "logit"),
##     data = crime.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2854  -0.1372  -0.0017   0.0020   3.4721
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -36.839521   7.028726  -5.241 1.59e-07 ***
## zn           -0.061720   0.034410  -1.794 0.072868 .
## indus        -0.072580   0.048546  -1.495 0.134894
## chas          1.032352   0.759627   1.359 0.174139
## nox          50.159513   8.049503   6.231 4.62e-10 ***
## rm           -0.692145   0.741431  -0.934 0.350548
## age           0.034522   0.013883   2.487 0.012895 *
## dis           0.765795   0.234407   3.267 0.001087 **
## rad           0.663015   0.165135   4.015 5.94e-05 ***
## tax          -0.006593   0.003064  -2.152 0.031422 *
## ptratio       0.442217   0.132234   3.344 0.000825 ***
## black        -0.013094   0.006680  -1.960 0.049974 *
## lstat         0.047571   0.054508   0.873 0.382802
## medv          0.199734   0.071022   2.812 0.004919 **
## ---
## 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: 186.15  on 452  degrees of freedom
## AIC: 214.15
##
## Number of Fisher Scoring iterations: 9
```

In this particular case, we notice how some variables are not statistically significant; for study purposes, I will assume that this is a valid model.

4.3 STEP Procedure

In this case, I will create multiple models, let's see the results.

```r
Model_STEP <- step(Model_NULL,
                   scope = list(upper=Model_FULL),
                   direction="both",
                   test="Chisq",
                   data=crime.train)
```

```
## Start:  AIC=647.88
## target ~ 1
##
##           Df Deviance    AIC    LRT  Pr(>Chi)
## + nox      1   292.01 296.01 353.86 < 2.2e-16 ***
## + rad      1   404.16 408.16 241.71 < 2.2e-16 ***
## + dis      1   409.50 413.50 236.38 < 2.2e-16 ***
## + age      1   424.75 428.75 221.13 < 2.2e-16 ***
## + tax      1   442.38 446.38 203.50 < 2.2e-16 ***
## + indus    1   453.23 457.23 192.64 < 2.2e-16 ***
## + zn       1   518.46 522.46 127.41 < 2.2e-16 ***
## + lstat    1   528.01 532.01 117.87 < 2.2e-16 ***
## + black    1   554.19 558.19  91.69 < 2.2e-16 ***
## + medv     1   609.62 613.62  36.26 1.729e-09 ***
## + ptratio  1   615.64 619.64  30.24 3.823e-08 ***
## + rm       1   634.82 638.82  11.05 0.0008863 ***
## + chas     1   642.86 646.86   3.02 0.0824375 .
## <none>         645.88 647.88
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=296.01
## target ~ nox
##
##           Df Deviance    AIC    LRT  Pr(>Chi)
## + rad      1   239.51 245.51  52.50   4.3e-13 ***
## + black    1   284.50 290.50   7.51  0.006142 **
## + rm       1   284.63 290.63   7.38  0.006598 **
## + medv     1   285.86 291.86   6.16  0.013103 *
## + indus    1   288.11 294.11   3.90  0.048195 *
## + zn       1   288.29 294.29   3.73  0.053593 .
## + tax      1   288.40 294.40   3.61  0.057432 .
## + chas     1   288.47 294.47   3.54  0.059824 .
## <none>         292.01 296.01
## + ptratio  1   290.14 296.14   1.88  0.170676
## + age      1   290.63 296.63   1.39  0.238898
## + dis      1   290.91 296.91   1.10  0.293997
## + lstat    1   291.93 297.93   0.09  0.770159
## - nox      1   645.88 647.88 353.86 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=245.51
## target ~ nox + rad
##
##           Df Deviance    AIC     LRT  Pr(>Chi)
## + tax      1   224.47 232.47  15.039 0.0001053 ***
## + indus    1   233.09 241.09   6.418 0.0112991 *
```

```
## + zn       1   235.19 243.19   4.325 0.0375672 *
## + rm       1   236.61 244.61   2.906 0.0882694 .
## + black    1   236.65 244.65   2.863 0.0906389 .
## + age      1   236.76 244.76   2.748 0.0973934 .
## + medv     1   236.86 244.86   2.651 0.1035095
## + ptratio  1   237.33 245.33   2.180 0.1398571
## <none>         239.51 245.51
## + chas     1   237.64 245.64   1.871 0.1713327
## + dis      1   237.96 245.96   1.548 0.2134708
## + lstat    1   239.47 247.47   0.037 0.8472926
## - rad      1   292.01 296.01  52.501   4.3e-13 ***
## - nox      1   404.16 408.16 164.650 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=232.47
## target ~ nox + rad + tax
##
##           Df Deviance    AIC     LRT  Pr(>Chi)
## + ptratio  1   218.70 228.70   5.770 0.0162983 *
## + black    1   219.86 229.86   4.616 0.0316663 *
## + zn       1   219.94 229.94   4.530 0.0333117 *
## + age      1   220.44 230.44   4.027 0.0447786 *
## <none>         224.47 232.47
## + dis      1   223.30 233.30   1.169 0.2796213
## + indus    1   223.40 233.40   1.076 0.2996421
## + chas     1   223.63 233.63   0.841 0.3592167
## + lstat    1   223.71 233.71   0.760 0.3832294
## + rm       1   223.75 233.75   0.720 0.3960720
## + medv     1   224.27 234.27   0.205 0.6508862
## - tax      1   239.51 245.51  15.039 0.0001053 ***
## - rad      1   288.40 294.40  63.931 1.289e-15 ***
## - nox      1   395.48 401.48 171.012 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=228.7
## target ~ nox + rad + tax + ptratio
##
##           Df Deviance    AIC     LRT  Pr(>Chi)
## + black    1   214.36 226.36   4.337   0.03730 *
## + age      1   214.46 226.46   4.239   0.03949 *
## + medv     1   215.23 227.23   3.474   0.06233 .
## + rm       1   216.12 228.12   2.581   0.10815
## + zn       1   216.32 228.32   2.386   0.12246
## <none>         218.70 228.70
## + chas     1   216.81 228.81   1.888   0.16944
## + dis      1   217.79 229.79   0.907   0.34078
## + indus    1   217.82 229.82   0.885   0.34693
## + lstat    1   218.57 230.57   0.129   0.71931
## - ptratio  1   224.47 232.47   5.770   0.01630 *
## - tax      1   237.33 245.33  18.630 1.587e-05 ***
## - rad      1   287.59 295.59  68.885 < 2.2e-16 ***
## - nox      1   394.21 402.21 175.507 < 2.2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=226.36
## target ~ nox + rad + tax + ptratio + black
##
##           Df Deviance    AIC     LRT  Pr(>Chi)
## + medv     1   209.80 223.80   4.567   0.03260 *
## + age      1   210.54 224.54   3.821   0.05063 .
## + rm       1   211.16 225.16   3.205   0.07341 .
## + chas     1   212.27 226.27   2.094   0.14784
## + zn       1   212.32 226.32   2.041   0.15308
## <none>         214.36 226.36
## + indus    1   213.36 227.36   1.001   0.31710
## + dis      1   213.38 227.38   0.986   0.32075
## + lstat    1   214.33 228.33   0.036   0.85053
## - black    1   218.70 228.70   4.337   0.03730 *
## - ptratio  1   219.86 229.86   5.491   0.01912 *
## - tax      1   234.81 244.81  20.440 6.152e-06 ***
## - rad      1   282.28 292.28  67.920 < 2.2e-16 ***
## - nox      1   373.37 383.37 159.008 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=223.8
## target ~ nox + rad + tax + ptratio + black + medv
##
##           Df Deviance    AIC     LRT  Pr(>Chi)
## + age      1   204.54 220.54   5.255  0.021886 *
## + lstat    1   206.57 222.57   3.231  0.072269 .
## + dis      1   207.27 223.27   2.531  0.111626
## + zn       1   207.43 223.43   2.364  0.124152
## + chas     1   207.52 223.52   2.283  0.130833
## <none>         209.80 223.80
## + indus    1   208.81 224.81   0.993  0.319071
## + rm       1   209.80 225.80   0.001  0.975428
## - medv     1   214.36 226.36   4.567  0.032598 *
## - black    1   215.23 227.23   5.429  0.019801 *
## - ptratio  1   219.42 231.42   9.622  0.001922 **
## - tax      1   225.15 237.15  15.353 8.919e-05 ***
## - rad      1   265.34 277.34  55.539 9.164e-14 ***
## - nox      1   373.21 385.21 163.409 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=220.54
## target ~ nox + rad + tax + ptratio + black + medv + age
##
##           Df Deviance    AIC    LRT  Pr(>Chi)
## + dis      1   198.28 216.28  6.266 0.0123042 *
## <none>         204.54 220.54
## + zn       1   202.92 220.92  1.620 0.2030736
## + lstat    1   203.02 221.02  1.525 0.2169051
## + chas     1   203.11 221.11  1.437 0.2305799
```

```
## + indus    1   203.32 221.32  1.227 0.2680353
## + rm       1   203.64 221.64  0.905 0.3413990
## - black    1   209.55 223.55  5.009 0.0252201 *
## - age      1   209.80 223.80  5.255 0.0218863 *
## - medv     1   210.54 224.54  6.001 0.0142974 *
## - ptratio  1   215.54 229.54 10.998 0.0009123 ***
## - tax      1   221.06 235.06 16.516 4.823e-05 ***
## - rad      1   262.36 276.36 57.815 2.880e-14 ***
## - nox      1   282.74 296.74 78.196 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=216.28
## target ~ nox + rad + tax + ptratio + black + medv + age + dis
##
##           Df Deviance    AIC    LRT  Pr(>Chi)
## + zn       1   192.57 212.57  5.710 0.0168719 *
## + rm       1   195.77 215.77  2.503 0.1136254
## + chas     1   195.81 215.81  2.463 0.1165750
## <none>         198.28 216.28
## + lstat    1   196.67 216.67  1.609 0.2046672
## + indus    1   196.76 216.76  1.520 0.2175898
## - black    1   203.45 219.45  5.171 0.0229697 *
## - dis      1   204.54 220.54  6.266 0.0123042 *
## - age      1   207.27 223.27  8.990 0.0027143 **
## - medv     1   208.49 224.49 10.215 0.0013927 **
## - ptratio  1   211.65 227.65 13.370 0.0002557 ***
## - tax      1   212.36 228.36 14.083 0.0001749 ***
## - rad      1   251.68 267.68 53.405 2.715e-13 ***
## - nox      1   269.35 285.35 71.072 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=212.57
## target ~ nox + rad + tax + ptratio + black + medv + age + dis +
##     zn
##
##           Df Deviance    AIC    LRT  Pr(>Chi)
## + lstat    1   190.51 212.51  2.053 0.1519308
## + rm       1   190.56 212.56  2.003 0.1569938
## <none>         192.57 212.57
## + chas     1   190.94 212.94  1.631 0.2015480
## + indus    1   191.35 213.35  1.221 0.2692186
## - black    1   197.32 215.32  4.756 0.0291972 *
## - zn       1   198.28 216.28  5.710 0.0168719 *
## - age      1   201.78 219.78  9.215 0.0024005 **
## - ptratio  1   201.84 219.84  9.275 0.0023227 **
## - dis      1   202.92 220.92 10.356 0.0012905 **
## - tax      1   203.87 221.87 11.298 0.0007760 ***
## - medv     1   205.09 223.09 12.528 0.0004009 ***
## - rad      1   241.89 259.89 49.326 2.168e-12 ***
## - nox      1   264.68 282.68 72.115 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Step:  AIC=212.51
## target ~ nox + rad + tax + ptratio + black + medv + age + dis +
##     zn + lstat
##
##           Df Deviance    AIC    LRT  Pr(>Chi)
## <none>         190.51 212.51
## - lstat    1   192.57 212.57  2.053 0.1519308
## + indus    1   188.93 212.93  1.582 0.2084729
## + chas     1   189.38 213.38  1.137 0.2862408
## + rm       1   189.62 213.62  0.891 0.3452530
## - black    1   195.51 215.51  5.001 0.0253393 *
## - zn       1   196.67 216.67  6.154 0.0131141 *
## - age      1   196.98 216.98  6.464 0.0110079 *
## - ptratio  1   200.51 220.51  9.992 0.0015725 **
## - dis      1   201.23 221.23 10.713 0.0010638 **
## - tax      1   202.79 222.79 12.277 0.0004585 ***
## - medv     1   204.52 224.52 14.006 0.0001822 ***
## - rad      1   240.50 260.50 49.988 1.547e-12 ***
## - nox      1   261.63 281.63 71.118 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Model_STEP
```

```
##
## Call:  glm(formula = target ~ nox + rad + tax + ptratio + black + medv +
##     age + dis + zn + lstat, family = binomial(link = "logit"),
##     data = crime.train)
##
## Coefficients:
## (Intercept)          nox          rad          tax      ptratio
##  -34.831121    43.191700     0.743199    -0.008639     0.353646
##       black         medv          age          dis           zn
##   -0.011777     0.147677     0.027461     0.671699    -0.070361
##       lstat
##    0.068969
##
## Degrees of Freedom: 465 Total (i.e. Null);  455 Residual
## Null Deviance:       645.9
## Residual Deviance: 190.5     AIC: 212.5
```

From the above possible models, it was concluded that the Model with the lowest **Akaike's Information Criterion (AIC)** is the one containing the following variables: **nox, rad, tax, ptratio, black, medv, age, dis, zn, lstat**.

4.3.1 ANOVA results

Let's check an ANOVA table based on the above testing results.

```
Model_STEP$anova
```

Step	Df	Deviance	Resid. Df	Resid. Dev	AIC
	NA	NA	465	645.8758	647.8758
+ nox	-1	353.863406	464	292.0124	296.0124
+ rad	-1	52.501302	463	239.5111	245.5111
+ tax	-1	15.039248	462	224.4719	232.4719
+ ptratio	-1	5.770398	461	218.7015	228.7015
+ black	-1	4.336863	460	214.3646	226.3646
+ medv	-1	4.566763	459	209.7978	223.7978
+ age	-1	5.254798	458	204.5430	220.5430
+ dis	-1	6.266495	457	198.2766	216.2766
+ zn	-1	5.709651	456	192.5669	212.5669
+ lstat	-1	2.052758	455	190.5141	212.5141

IMPORTANT

If we check our theory, the **AIC** defines as follows: *the smaller the value for AIC the better the model*; in this case, we can easily observe how the by adding certain variables, our AIC values decrease making it a better model.

4.4 AIC Model

From the above results and calculations, it was concluded that the best model is as follows:

```
Model_AIC = glm(formula = target ~
                nox + rad + tax + ptratio + black + medv + age + dis + zn + lstat,
                family = binomial(link = "logit"),
                data = crime.train)

summary(Model_AIC)
```

```
##
## Call:
## glm(formula = target ~ nox + rad + tax + ptratio + black + medv +
##     age + dis + zn + lstat, family = binomial(link = "logit"),
##     data = crime.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1610  -0.1556  -0.0017   0.0019   3.3921
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -34.831121   6.585833  -5.289 1.23e-07 ***
## nox          43.191700   6.752210   6.397 1.59e-10 ***
## rad           0.743199   0.154146   4.821 1.43e-06 ***
## tax          -0.008639   0.002758  -3.133 0.001732 **
## ptratio       0.353646   0.115275   3.068 0.002156 **
```

```
## black        -0.011777   0.006317  -1.864 0.062280 .
## medv          0.147677   0.042484   3.476 0.000509 ***
## age           0.027461   0.011238   2.444 0.014544 *
## dis           0.671699   0.216605   3.101 0.001929 **
## zn           -0.070361   0.032981  -2.133 0.032891 *
## lstat         0.068969   0.048236   1.430 0.152769
## ---
## 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: 190.51  on 455  degrees of freedom
## AIC: 212.51
##
## Number of Fisher Scoring iterations: 9
```

From the above model, it is interesting to note how all of the predictor variables but `lstat` are statistically significant; also, we can notice how the Median is near zero and how the standard error could be considered low.

4.5 Modified AIC

From the above results, i will create a new modified model by excluding `lstat` from the previous model.

```
Model_AIC = glm(formula = target ~
                nox + rad + tax + ptratio + black + medv + age + dis + zn,
                family = binomial(link = "logit"),
                data = crime.train)

summary(Model_AIC)
```

```
##
## Call:
## glm(formula = target ~ nox + rad + tax + ptratio + black + medv +
##     age + dis + zn, family = binomial(link = "logit"), data = crime.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2719  -0.1695  -0.0022   0.0022   3.4083
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -33.252218   6.510006  -5.108 3.26e-07 ***
## nox          42.893366   6.744624   6.360 2.02e-10 ***
## rad           0.724580   0.150914   4.801 1.58e-06 ***
## tax          -0.008216   0.002731  -3.009  0.00262 **
## ptratio       0.339874   0.114950   2.957  0.00311 **
## black        -0.011726   0.006535  -1.794  0.07276 .
## medv          0.117392   0.036009   3.260  0.00111 **
## age           0.031946   0.010928   2.923  0.00346 **
## dis           0.661897   0.216100   3.063  0.00219 **
## zn           -0.065747   0.031905  -2.061  0.03933 *
## ---
```

```
## 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: 192.57  on 456  degrees of freedom
## AIC: 212.57
##
## Number of Fisher Scoring iterations: 9
```

Is interesting to note that now, all predictors are statistically significant, the standard errors and the median are still small but it seems that actually increased alongside the AIC with a slight increase.

4.6 Intuition Model

From the correlations analysis table, I concluded that some variables were more correlated to `target` than others. In this section, I will create a model based on that output by including the following variables only and I will use it in order to choose my best selected model.

Variables
target
indus
nox
age
dis
rad
tax

In this case, I will employ the following variables: **indus, nox, age, dis, rad, tax**.

```
Model_INTUITION <- glm(target ~ indus + nox + age + dis + rad + tax,
                       data = crime.train,
                       family = binomial(link = logit))

summary(Model_INTUITION)
```

```
##
## Call:
## glm(formula = target ~ indus + nox + age + dis + rad + tax, family = binomial(link = logit),
##     data = crime.train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.94477  -0.26091  -0.02967   0.00597   2.79697
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -24.274103   3.942812  -6.157 7.43e-10 ***
## indus        -0.052082   0.046221  -1.127  0.25982
## nox          40.156934   7.149753   5.617 1.95e-08 ***
## age           0.021947   0.009674   2.269  0.02328 *
## dis           0.241860   0.156349   1.547  0.12188
## rad           0.615435   0.126554   4.863 1.16e-06 ***
```

```
## tax          -0.007753   0.002595  -2.988  0.00281 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 216.88  on 459  degrees of freedom
## AIC: 230.88
##
## Number of Fisher Scoring iterations: 8
```

From the above results, we can quickly identify the non statistical significance of `indus` and `dis`. Also, we notice how the AIC value has increased in a moderate way, along side the Residual Deviance.

From here moving forward, I will try to "refine" this model.

4.7 Intuition Model Refined

From the previous results, I will proceed to do backward elimination; in this case, I will exclude the variables `indus` and `dis`.

```
Model_Refined <- glm(target ~ nox + age + rad + tax,
              data = crime.train,
              family = binomial(link = logit))

summary(Model_Refined)
```

```
##
## Call:
## glm(formula = target ~ nox + age + rad + tax, family = binomial(link = logit),
##     data = crime.train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.84487  -0.28103  -0.03058   0.00821   2.65935
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -18.962071   2.427756  -7.811 5.69e-15 ***
## nox          31.611303   4.924409   6.419 1.37e-10 ***
## age           0.018315   0.009246   1.981 0.047595 *
## rad           0.649578   0.122558   5.300 1.16e-07 ***
## tax          -0.008663   0.002420  -3.580 0.000344 ***
## ---
## 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: 220.44  on 461  degrees of freedom
## AIC: 230.44
##
## Number of Fisher Scoring iterations: 8
```

Finally, we notice how all the given predictors are statistically significant but also, we can notice how the AIC increased, the Median is higher than before and how the residual deviance increased as well.

5 MODEL SELECTION

From the above possible models, I will select the model given with the lowest AIC; if it is true, it includes the highest number of variables, it is the model that provides better possible outcome in this particular case; hence my selected model will be the one containing the following variables: **nox, rad, tax, ptratio, black, medv, age, dis, zn, lstat**.

```
Model_FINAL <- Model_AIC
summary(Model_FINAL)
```

```
##
## Call:
## glm(formula = target ~ nox + rad + tax + ptratio + black + medv +
##     age + dis + zn, family = binomial(link = "logit"), data = crime.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2719  -0.1695  -0.0022   0.0022   3.4083
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -33.252218   6.510006  -5.108 3.26e-07 ***
## nox          42.893366   6.744624   6.360 2.02e-10 ***
## rad           0.724580   0.150914   4.801 1.58e-06 ***
## tax          -0.008216   0.002731  -3.009  0.00262 **
## ptratio       0.339874   0.114950   2.957  0.00311 **
## black        -0.011726   0.006535  -1.794  0.07276 .
## medv          0.117392   0.036009   3.260  0.00111 **
## age           0.031946   0.010928   2.923  0.00346 **
## dis           0.661897   0.216100   3.063  0.00219 **
## zn           -0.065747   0.031905  -2.061  0.03933 *
## ---
## 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: 192.57  on 456  degrees of freedom
## AIC: 212.57
##
## Number of Fisher Scoring iterations: 9
```

The reasons are explained below:

- This model returned the lowest **Akaike's Information Criterion** AIC.
- This model returned the nearest to zero median value.
- This model included the most number of significant statistically predictive values.
- This model displayed the smallest standard errors for the considered predictor variables.
- This model present the smallest rate of change for all predictor variables.

- This model returned the lowest residual deviance.
- From the below table we can see how the probability of being higher than the χ^2 are very low.

```
Anova(Model_FINAL, type="II", test="Wald")
```

	Df	Chisq	Pr(>Chisq)
nox	1	40.444992	0.0000000
rad	1	23.052204	0.0000016
tax	1	9.053076	0.0026225
ptratio	1	8.742223	0.0031093
black	1	3.219696	0.0727571
medv	1	10.628153	0.0011138
age	1	8.546196	0.0034624
dis	1	9.381495	0.0021919
zn	1	4.246463	0.0393322

5.1 Test model

From the above chosen model, I will create a reduced data frame containing only the variables needed in order to run our model.

```
select_var <- c('target', 'nox', 'rad', 'tax', 'ptratio', 'black',
                'medv', 'age', 'dis', 'zn', 'lstat')

crime.train.final <- crime.train[select_var]
```

5.1.1 Final Model Comparisons

From here, I will define a null model with the chosen variables in order to compare results with the final model.

```
Model_NULL = glm(target ~ 1,
                 data=crime.train.final,
                 family = binomial(link="logit"))

summary(Model_NULL)
```

```
##
## Call:
## glm(formula = target ~ 1, family = binomial(link = "logit"),
##     data = crime.train.final)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.163  -1.163  -1.163   1.192   1.192
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.03434    0.09266  -0.371    0.711
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 645.88  on 465  degrees of freedom
## AIC: 647.88
##
## Number of Fisher Scoring iterations: 3
```

5.1.2 Analysis of Deviance Table

The below table, will display a Deviance analysis by employing the χ^2 test.

```
anova(Model_FINAL,
      Model_NULL,
      test="Chisq")
```

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
456	192.5669	NA	NA	NA
465	645.8758	-9	-453.3089	0

In the above results, we can easily compare our Residual Deviance in which our model has better results compared to the null model since the null model's deviance will increase in 453.31 units compared to our final model.

5.1.3 Likelihood ratio test

In order to do so, I will employ the **lrtest** function from the **lmtest** library; this is a generic function for carrying out likelihood ratio tests. The default method can be employed for comparing nested (generalized) linear models.

```
lrtest(Model_FINAL)
```

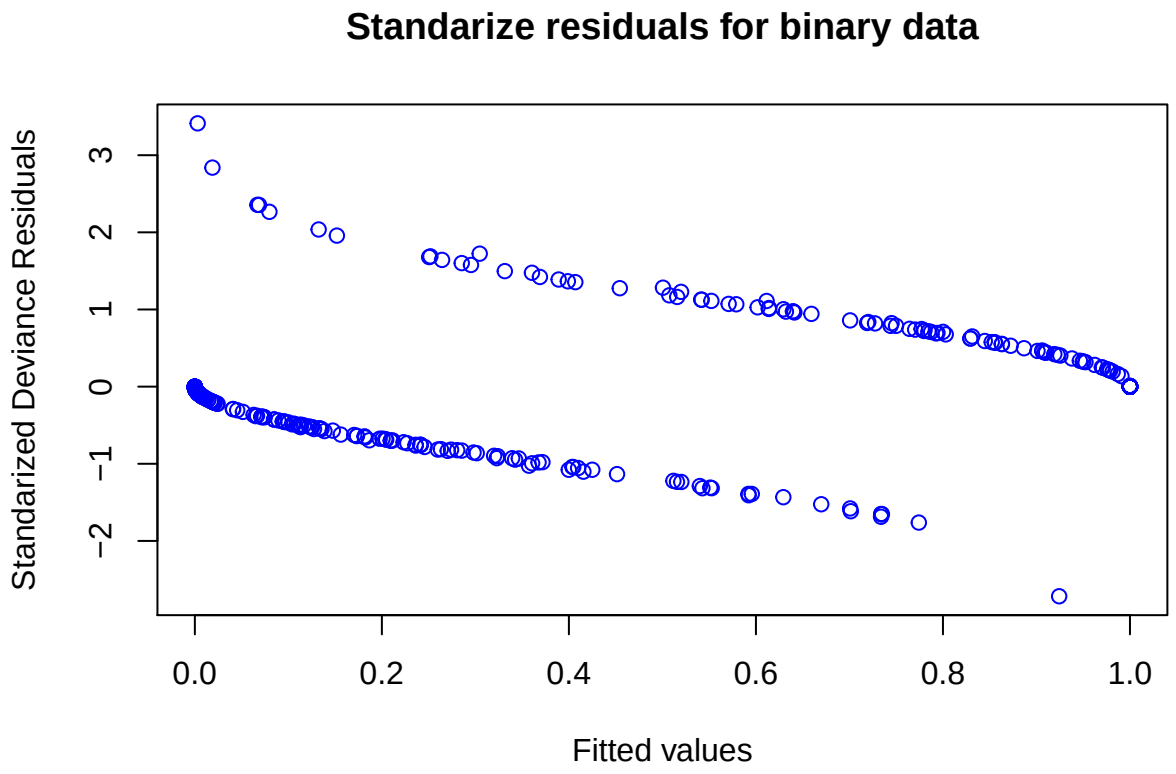
#Df	LogLik	Df	Chisq	Pr(>Chisq)
10	-96.28345	NA	NA	NA
1	-322.93791	-9	453.3089	0

As you can see, in our Final Model, we obtain much better results compared to our NULL model, hence this corroborates that our Final Model has a much better Likelihood ratio compared to the NULL Model.

5.1.4 Plot of standardized residuals

The below plot shows our fitted models vs the deviance r standardized residuals.

```
plot(fitted(Model_FINAL),
     rstandard(Model_FINAL),
     main = 'Standarize residuals for binary data',
     xlab = 'Fitted values',
     ylab = 'Standarized Deviance Residuals',
     col = 'blue')
```

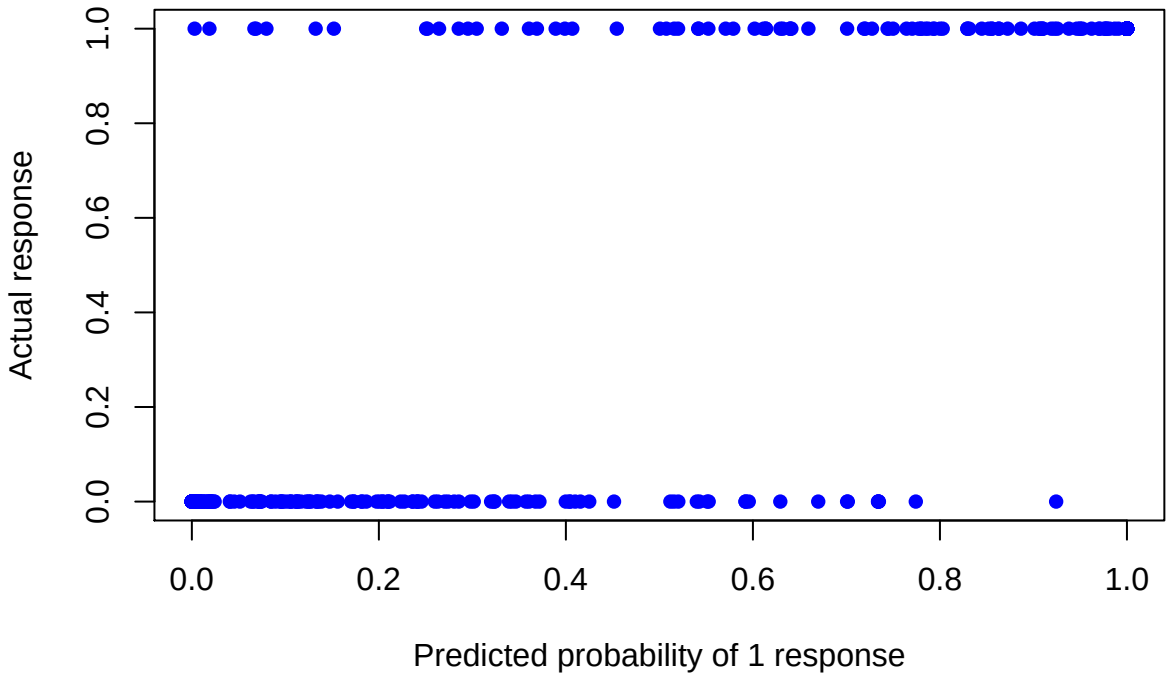


5.1.5 Simple plot of predictions

The below plot is a visual representation of the predicted values versus the given values aka `target`.

```
crime.train.final$predict = predict(Model_FINAL,
                                    type="response")

plot(target ~ predict,
     data = crime.train.final,
     pch = 16,
     xlab="Predicted probability of 1 response",
     ylab="Actual response",
     col = 'blue')
```



5.2 Evaluations

In this section, I will proceed to evaluate my chosen final model in terms of (a) accuracy, (b) classification error rate, (c) precision, (d) sensitivity, (e) specificity, (f) F1 score, (g) AUC, and (h) confusion matrix.

In order to do so, I will need to perform a couple of "transformations"; that is to round the given probabilities to zero decimals.

```
crime.train.final$predicted_target <- round(crime.train.final$predict,0)

crime.train.table <- table(crime.train.final$predicted_target,
                           crime.train.final$target,
                           dnn = c("Predicted", "Target"))

data.frame(crime.train.table)
```

Predicted	Target	Freq
0	0	217
1	0	20
0	1	20
1	1	209

5.2.1 Confusion Matrix

Let's start by building a confusion matrix in order to obtain valuable insights.

```
cMatrix <- confusionMatrix(data = as.factor(crime.train.final$predicted_target),
                           reference = as.factor(crime.train.final$target),
                           positive = '1')
cMatrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 217  20
##          1  20 209
##
##                Accuracy : 0.9142
##                  95% CI : (0.8849, 0.938)
##     No Information Rate : 0.5086
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8283
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9127
##             Specificity : 0.9156
##          Pos Pred Value : 0.9127
##          Neg Pred Value : 0.9156
##              Prevalence : 0.4914
##          Detection Rate : 0.4485
##    Detection Prevalence : 0.4914
##       Balanced Accuracy : 0.9141
##
##        'Positive' Class : 1
##
```

From the above results, we obtain as follows:

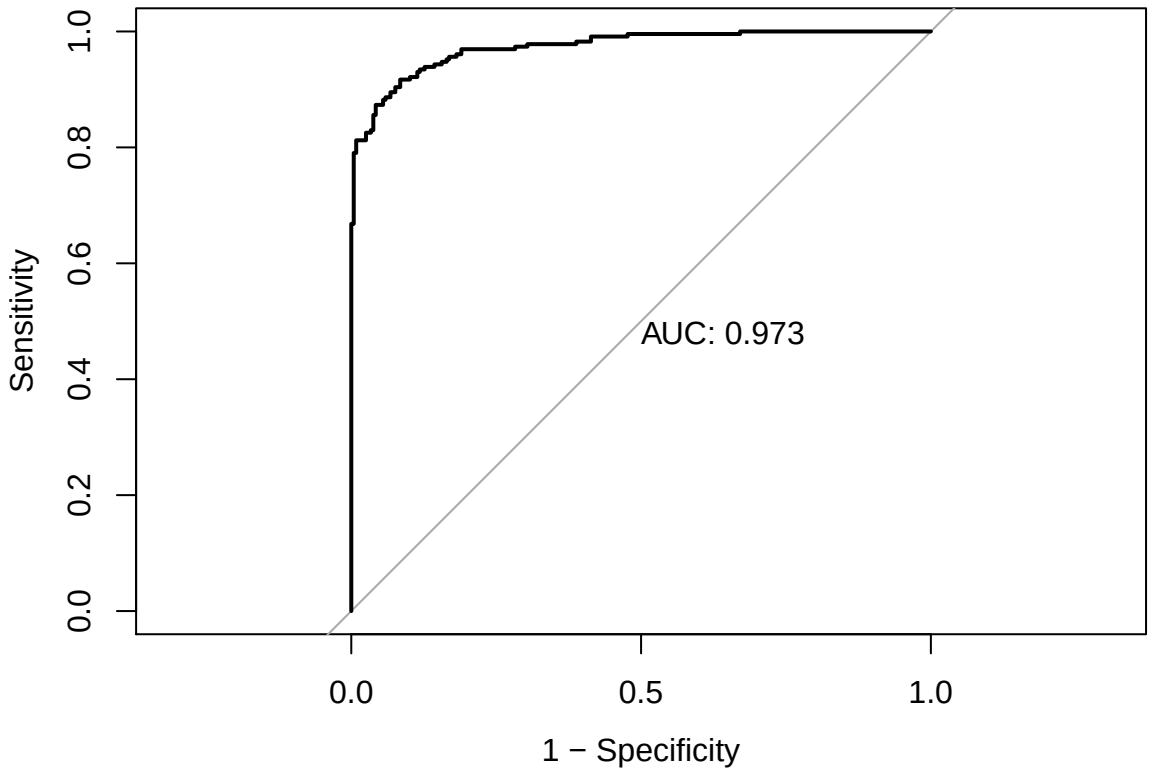
	Value
Sensitivity	0.9126638
Specificity	0.9156118
Pos Pred Value	0.9126638
Neg Pred Value	0.9156118
Precision	0.9126638
Recall	0.9126638
F1	0.9126638
Prevalence	0.4914163
Detection Rate	0.4484979
Detection Prevalence	0.4914163
Balanced Accuracy	0.9141378

5.2.2 ROC and AUC

As we know, the **Receiver Operating Characteristic Curves** (ROC) is a great quantitative assessment tool of the model. In order to quantify our model, I will employ as follows:

```
# First, let's prepare our function
rocCurve <- roc(target ~ predict, data = crime.train.final)

# Let's plot our RCO curve.
plot(rocCurve, print.auc=TRUE, legacy.axes = TRUE)
```



Let's see our confidence intervals.

	AUC
Lower bound	0.9612813
Estimated value	0.9733938
Higher bound	0.9855062

6 PREDICTIONS

6.1 Table

In this section, I will predict the values on the **evaluation** data set employing the **training** data set.

predicted	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
1	0	8.14	0	0.538	6.096	84.5	4.4619	4	307	21.0	380.02	10.26	18.2
1	0	8.14	0	0.538	6.495	94.4	4.4547	4	307	21.0	387.94	12.80	18.4
1	0	8.14	0	0.538	5.950	82.0	3.9900	4	307	21.0	232.60	27.71	13.2
0	0	5.96	0	0.499	5.850	41.5	3.9342	5	279	19.2	396.90	8.77	21.0
0	25	5.13	0	0.453	5.741	66.2	7.2254	8	284	19.7	395.11	13.15	18.7
0	25	5.13	0	0.453	5.966	93.4	6.8185	8	284	19.7	378.08	14.44	16.0
0	0	4.49	0	0.449	6.630	56.1	4.4377	3	247	18.5	392.30	6.53	26.6
0	0	4.49	0	0.449	6.121	56.8	3.7476	3	247	18.5	395.15	8.44	22.2
0	0	2.89	0	0.445	6.163	69.6	3.4952	2	276	18.0	391.83	11.34	21.4
1	0	25.65	0	0.581	5.856	97.0	1.9444	2	188	19.1	370.31	25.41	17.3
0	0	25.65	0	0.581	5.613	95.6	1.7572	2	188	19.1	359.29	27.26	15.7
1	0	21.89	0	0.624	5.637	94.7	1.9799	4	437	21.2	396.90	18.34	14.3
1	0	19.58	0	0.605	6.101	93.0	2.2834	5	403	14.7	240.16	9.81	25.0
1	0	19.58	0	0.605	5.880	97.3	2.3887	5	403	14.7	348.13	12.03	19.1
0	0	10.59	1	0.489	5.960	92.1	3.8771	4	277	18.6	393.25	17.27	21.7
0	0	6.20	0	0.504	6.552	21.4	3.3751	8	307	17.4	380.34	3.76	31.5
1	0	6.20	0	0.507	8.247	70.4	3.6519	8	307	17.4	378.95	3.95	48.3
0	22	5.86	0	0.431	6.957	6.8	8.9067	7	330	19.1	386.09	3.53	29.6
0	90	2.97	0	0.400	7.088	20.8	7.3073	1	285	15.3	394.72	7.85	32.2
0	80	1.76	0	0.385	6.230	31.5	9.0892	1	241	18.2	341.60	12.93	20.1
0	33	2.18	0	0.472	6.616	58.1	3.3700	7	222	18.4	393.36	8.93	28.4
0	0	9.90	0	0.544	6.122	52.8	2.6403	4	304	18.4	396.90	5.98	22.1
0	0	7.38	0	0.493	6.415	40.1	4.7211	5	287	19.6	396.90	6.12	25.0
0	0	7.38	0	0.493	6.312	28.9	5.4159	5	287	19.6	396.90	6.15	23.0
1	0	5.19	0	0.515	5.895	59.6	5.6150	5	224	20.2	394.81	10.56	18.5
0	80	2.01	0	0.435	6.635	29.7	8.3440	4	280	17.0	390.94	5.99	24.5
1	0	18.10	0	0.718	3.561	87.9	1.6132	24	666	20.2	354.70	7.12	27.5
1	0	18.10	1	0.631	7.016	97.5	1.2024	24	666	20.2	392.05	2.96	50.0
1	0	18.10	0	0.584	6.348	86.1	2.0527	24	666	20.2	83.45	17.64	14.5
1	0	18.10	0	0.740	5.935	87.9	1.8206	24	666	20.2	68.95	34.02	8.4
1	0	18.10	0	0.740	5.627	93.9	1.8172	24	666	20.2	396.90	22.88	12.8
1	0	18.10	0	0.740	5.818	92.4	1.8662	24	666	20.2	391.45	22.11	10.5
1	0	18.10	0	0.740	6.219	100.0	2.0048	24	666	20.2	395.69	16.59	18.4
1	0	18.10	0	0.740	5.854	96.6	1.8956	24	666	20.2	240.52	23.79	10.8
1	0	18.10	0	0.713	6.525	86.5	2.4358	24	666	20.2	50.92	18.13	14.1
1	0	18.10	0	0.713	6.376	88.4	2.5671	24	666	20.2	391.43	14.65	17.7
1	0	18.10	0	0.655	6.209	65.4	2.9634	24	666	20.2	396.90	13.22	21.4
1	0	9.69	0	0.585	5.794	70.6	2.8927	6	391	19.2	396.90	14.10	18.3
0	0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5.64	23.9

6.2 Classification and probability

In this section, I will provide a table in which the classification is reported alongside the probability for it.

predicted	probability
0	0.051
1	0.662
1	0.714
1	0.805
0	0.095
0	0.300
0	0.410
0	0.013
0	0.005
0	0.002
1	0.535
0	0.478
1	0.800
1	0.931
1	0.702
0	0.131
0	0.428
1	0.973
0	0.091
0	0.000
0	0.000
0	0.054
0	0.131
0	0.183
0	0.164
1	0.657
0	0.000
1	1.000
1	1.000
1	1.000
1	1.000
1	1.000
1	1.000
1	1.000
1	1.000
1	1.000
1	1.000
1	1.000
1	1.000
1	0.762
0	0.363