Data621- HW3

Group: Abdelmalek Hajjam / Monu Chacko

3/18/2020

Contents

0.1 Overview

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

Your objective is to build a binary logistic regression model on the training data set to predict whether the neighborhood will be at risk for high crime levels.

You will provide classifications and probabilities for the evaluation data set using your binary logistic regression model. You can only use the variables given to you (or variables that you derive from the variables provided).

0.2 Description

Let's look at our variables of interest in our dataset are:

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.

Type	Variable	Description
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	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 Data Exploration

1.1 Reading the data

The crime dataset in composed of 2 csv files, one for training our data and the other one is for evaluation. We are reading them from our github repository.

```
training_data <- read.csv("https://raw.githubusercontent.com/theoracley/Data621/master/Homework3/crime-

evaluation_data <- read.csv("https://raw.githubusercontent.com/theoracley/Data621/master/Homework3/crim
```

1.2 Our pattern

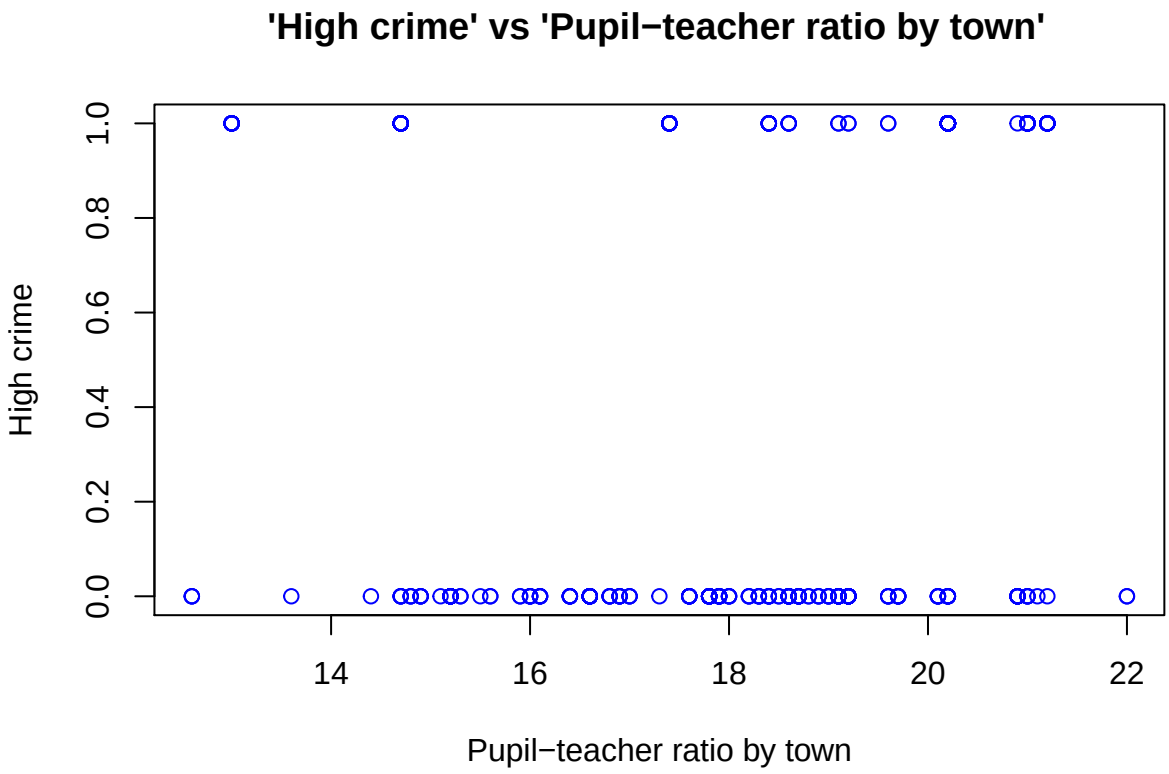
We will be guided by the following pattern all along, predicting the target variable using every explanatory variable. The following is one example of such a variable:

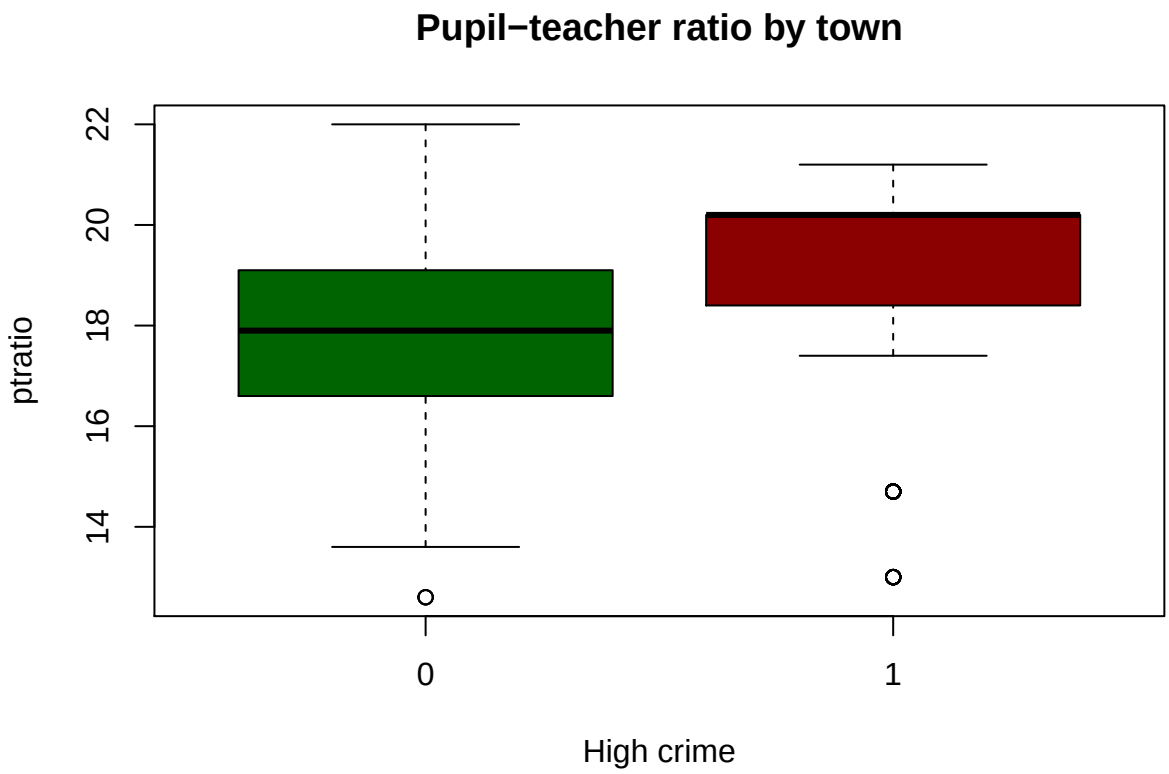
```
glm.tr <- glm(target ~ ptratio, data = training_data)
summary(glm.tr)
```

```
##
## Call:
## glm(formula = target ~ ptratio, data = training_data)
##
## 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
```

Not to say anything for this example output at this time, but just to mention that the predicted model will include $\beta_0 = -0.55998$ for the intercept and $\beta_1 = 0.05715$ for the rate of change.

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.

1.3 general exploration

Let's get deeper with our data and try to get any insights we can. So Let's go!

1.3.1 Dimensions

Our data has the folowing dimensions.

Records	Variables
466	13

It looks like our data has 466 records and 13 variables including the **target** variable corresponding to *high crime*.

1.3.2 Structure

Let's investigate our dataset and take a look at its structure.

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

1.3.3 Summary

Let's look at the summary statistics about our 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
lstat	466	-none-	numeric
medv	466	-none-	numeric
target	466	-none-	numeric

Now, let's look at all the variables closely,including the target variable, and try to get insight from them.

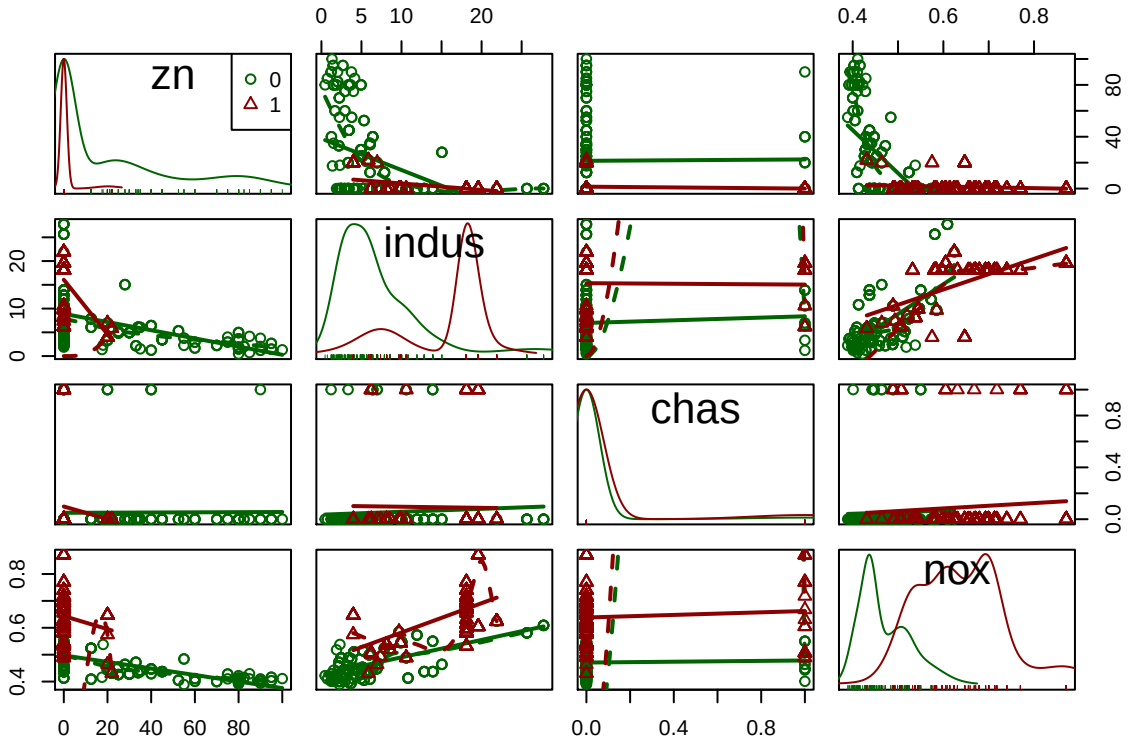
	Min	1st Qu	Median	Mean	3rd Qu	Max
zn	0.000	0.000	0.000	11.58000	16.250	100.000
indus	0.460	5.145	9.690	11.10500	18.100	27.740
chas	0.000	0.000	0.000	0.07082	0.000	1.000
nox	0.389	0.448	0.538	0.55430	0.624	0.871
rm	3.863	5.887	6.210	6.29100	6.630	8.780
age	2.900	43.880	77.150	68.37000	94.100	100.000
dis	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
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

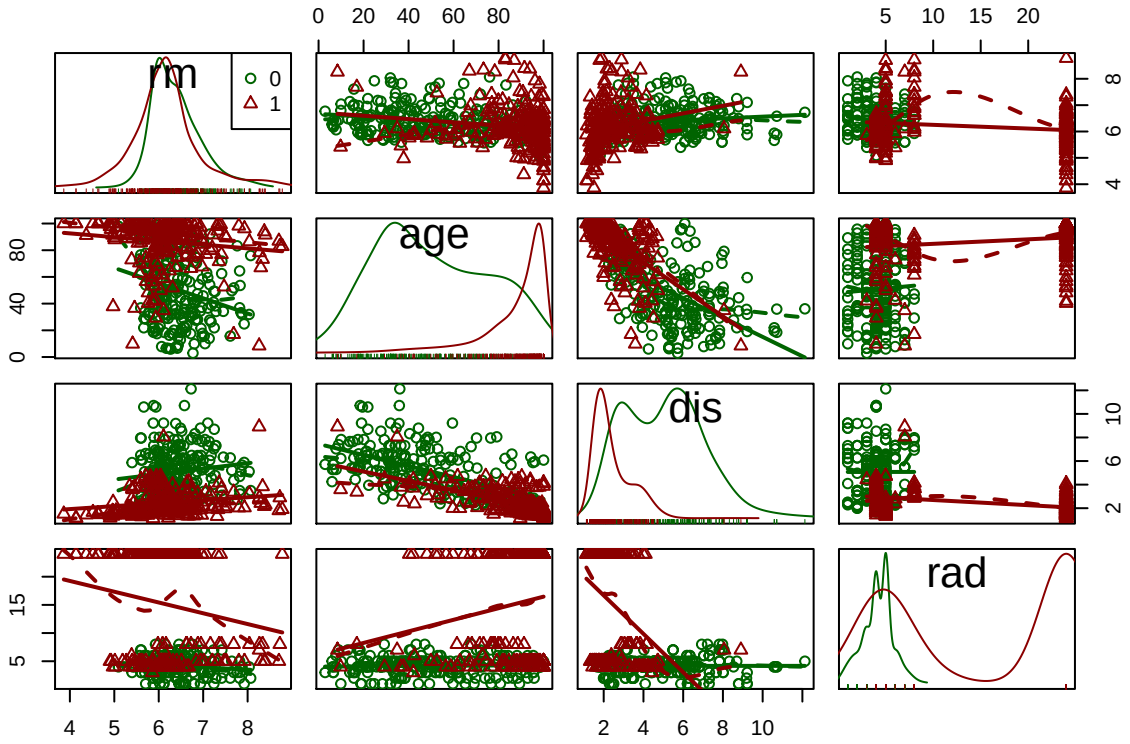
1.3.4 Missing data

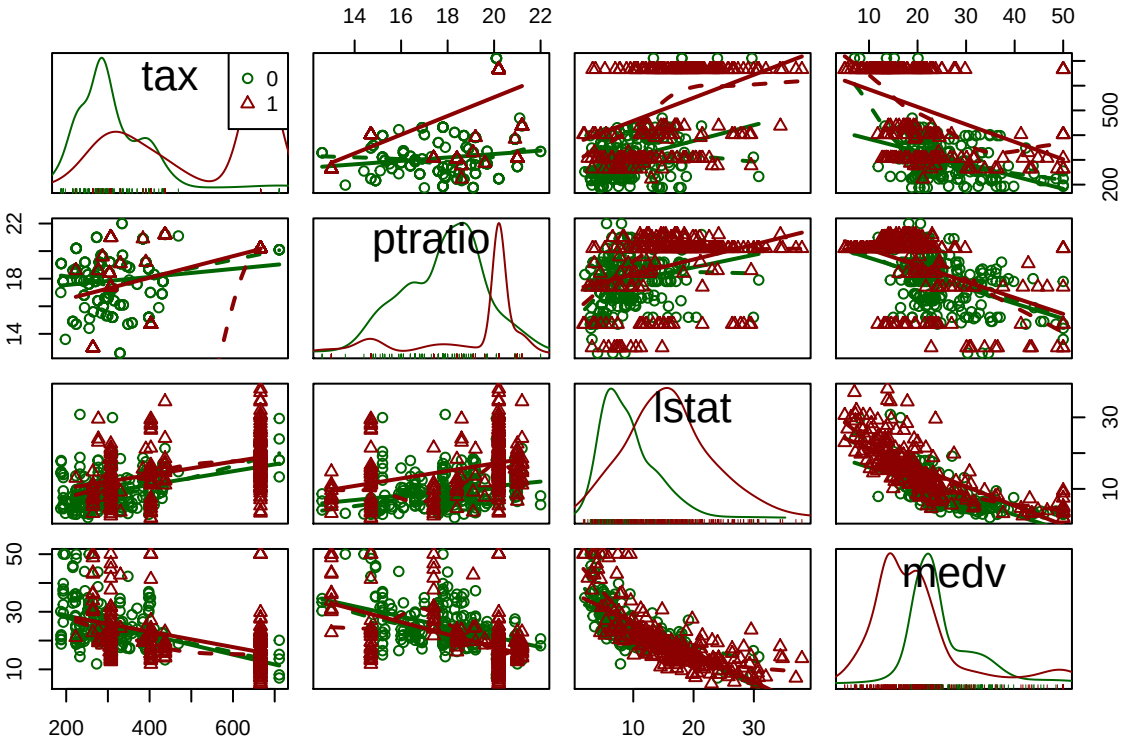
According to the statistics above, there are no missing values or **NA**, since missing data was not reported above.

1.3.5 Visualizations

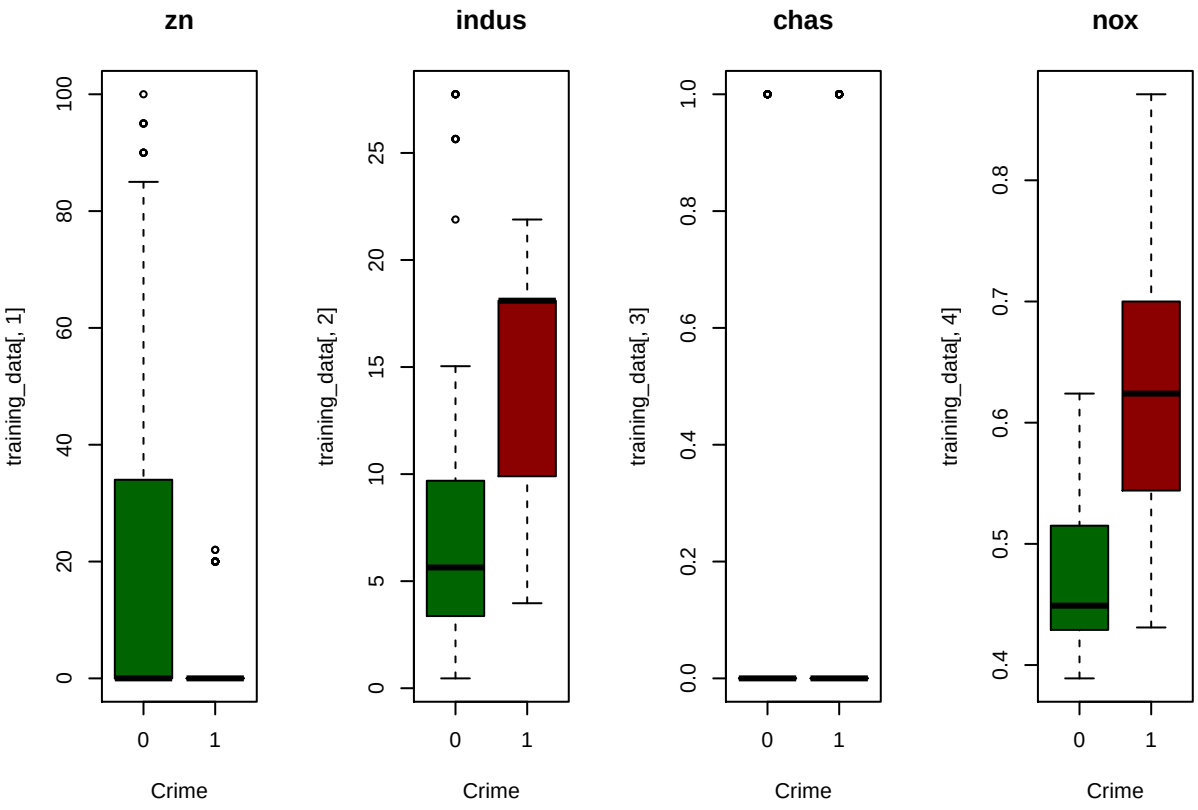
In the below graphs, the colors indicate that any record not including a high crime shows a green circle, while a record indicating a high crime has been plot in a red triangle. The diagonal plots the empirical distribution for both classes.

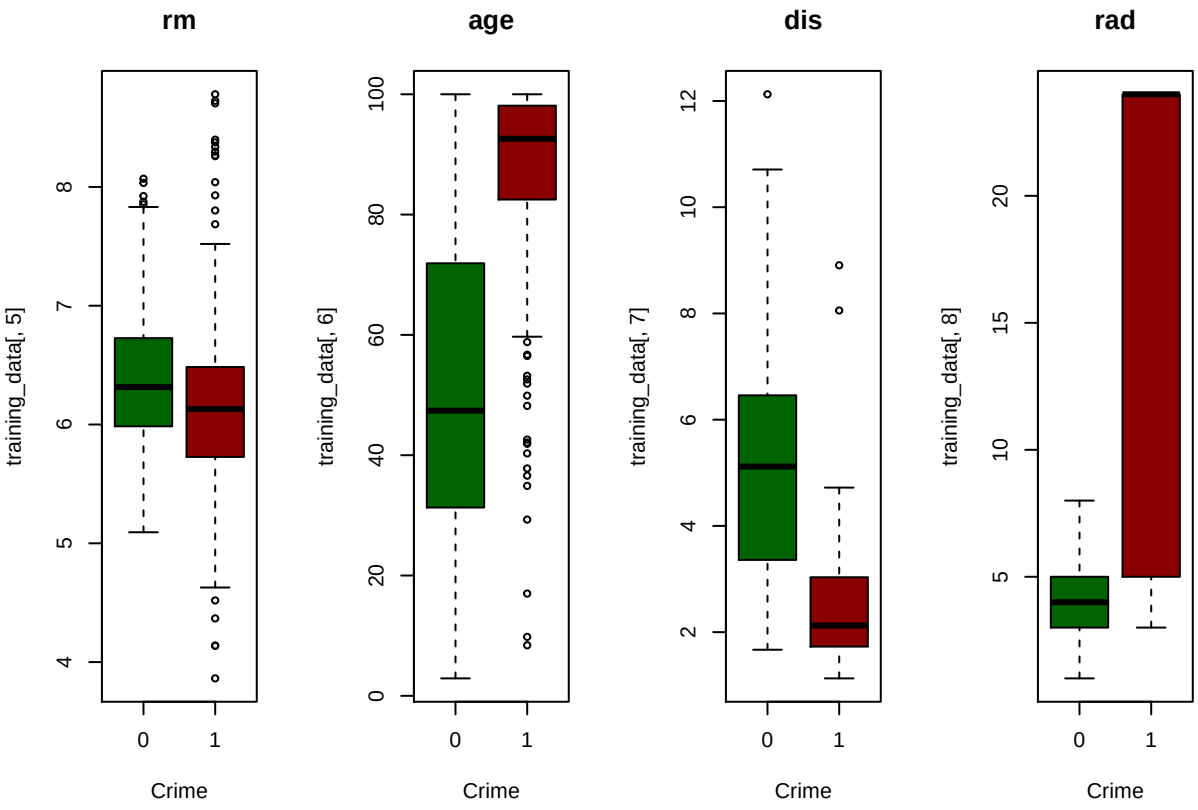


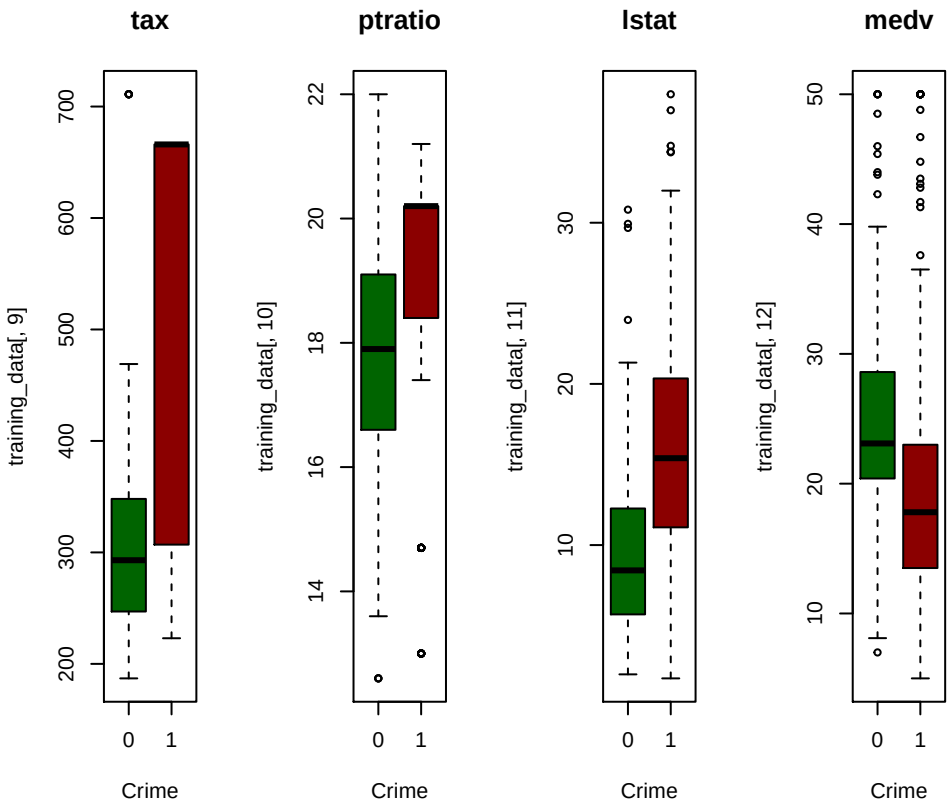

rm
age
dis
rad
0
1
0 20 40 60 80
5 10 15 20
4 6 8
0 40 80
2 6 10
5 15
4 5 6 7 8
2 4 6 8 10



Let's separate our data for visualization purposes.


zn
indus
chas
nox
training_data[, 1]
training_data[, 2]
training_data[, 3]
training_data[, 4]
0
20
40
60
80
100
0
5
10
15
20
25
0.0
0.2
0.4
0.6
0.8
1.0
0.4
0.5
0.6
0.7
0.8
0
1
Crime


rm
training_data[, 5]
4
5
6
7
8
0
1
Crime
age
training_data[, 6]
0
20
40
60
80
100
0
1
Crime
dis
training_data[, 7]
2
4
6
8
10
12
0
1
Crime
rad
training_data[, 8]
5
10
15
20
0
1
Crime



1.3.6 Count values

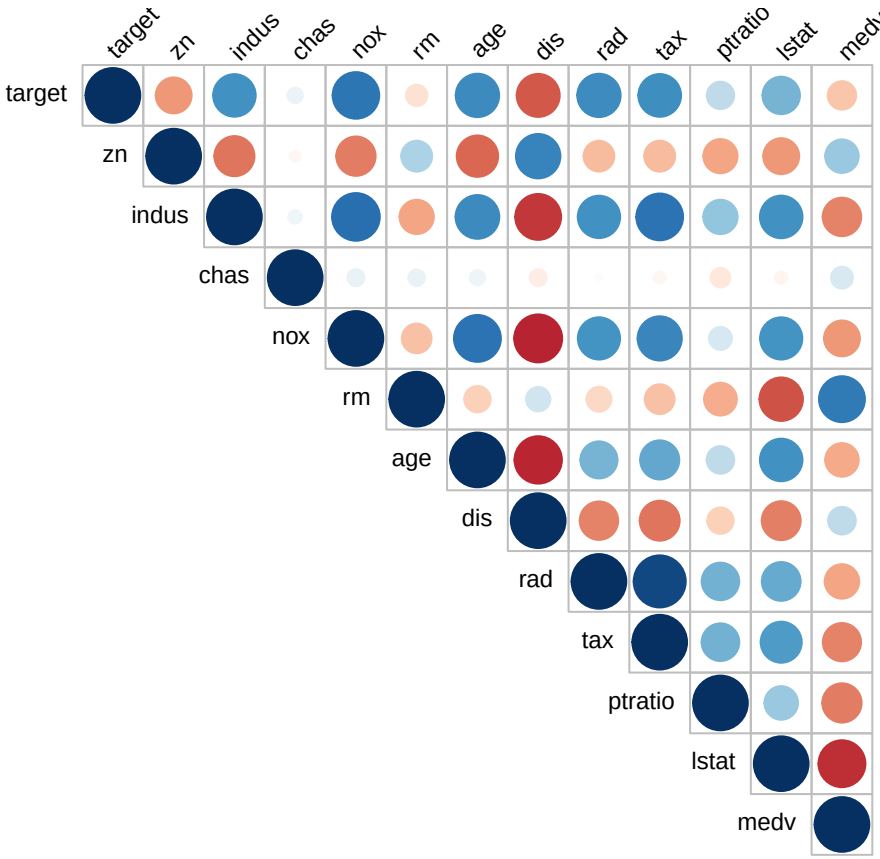
Let's have a small understanding on how many records were categorized as 0 and how many as 1.

target	Counts	Percent
0	237	0.509
1	229	0.491

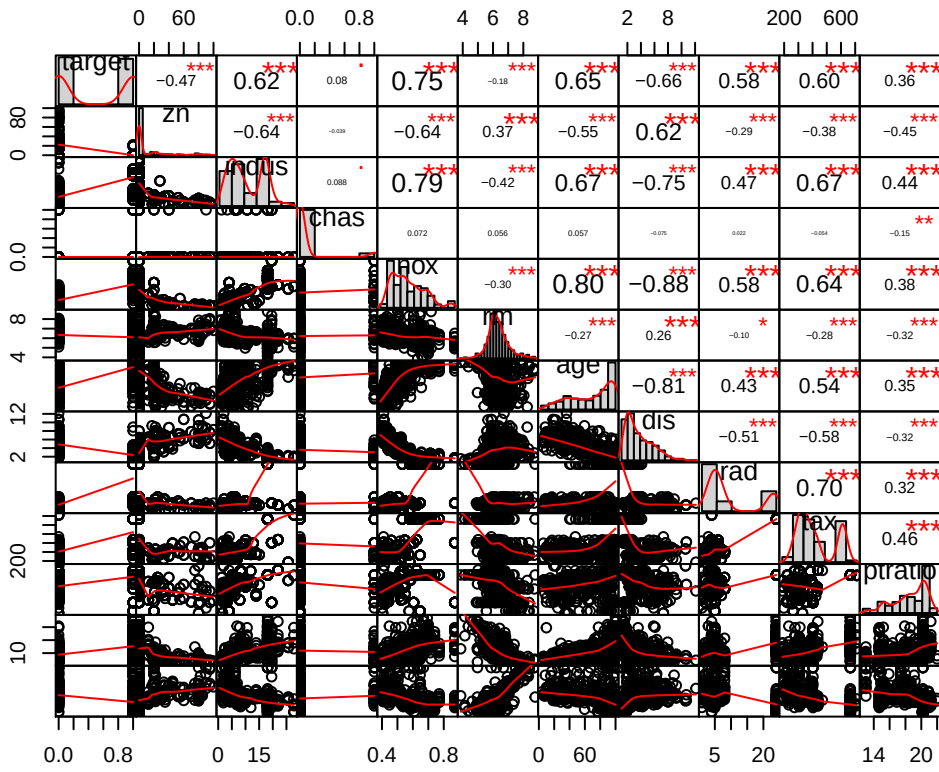
From the above results, we could assume that in effect the values seems to be uniformly distributed since almost half the data represent 0 and almost half represent 1.

1.3.7 Correlations

Let's create some visualizations for the correlation matrix.



1.3.7.1 Graphical visualization



1.3.7.2 Numerical visualization

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

Get more insights from the Correlations table.

	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
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, something to keep in mind in case of

multivariate co-linearity.

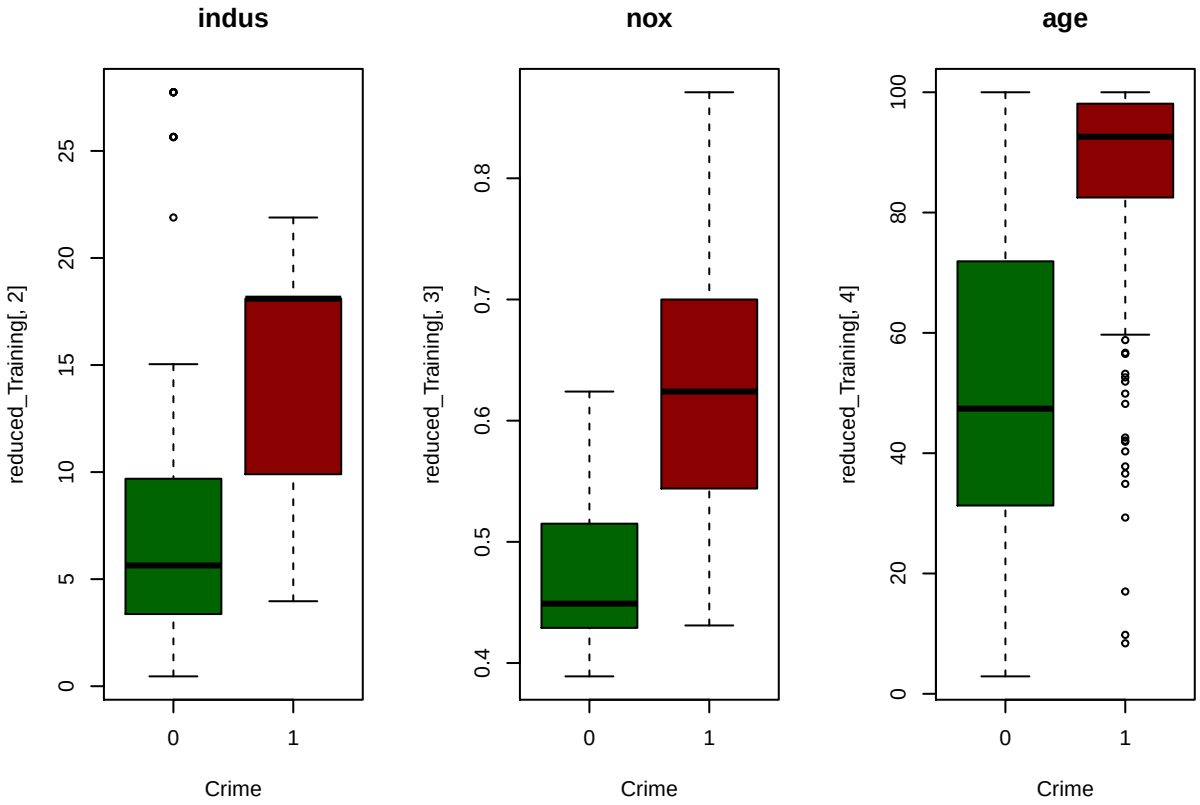
2 DATA PREPARATION

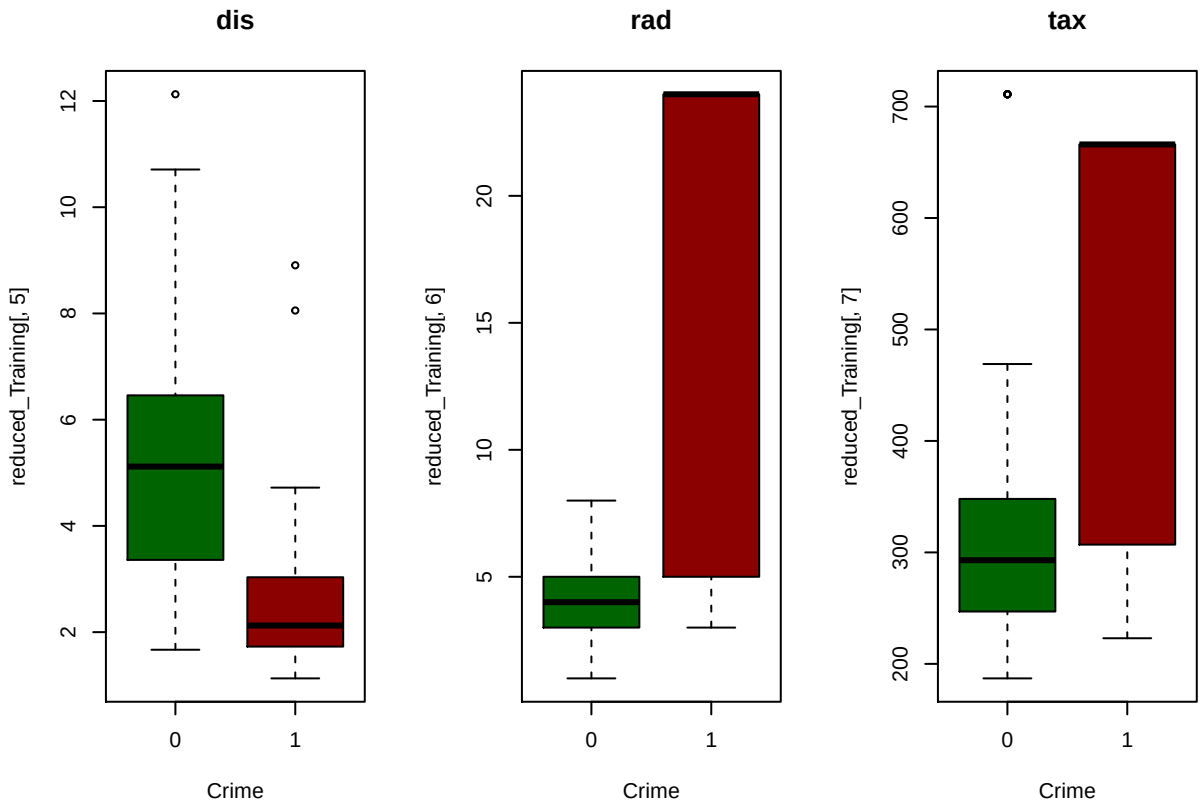
From the correlations table, we could focus on the variables that contain the strongest correlations related to our `target` variable; in this case, I will set my cut off at with any correlation in which the absolute value will be higher than 0.5.

	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_Training)
```

```
## '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 Binary Logistic Regression

We 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:

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

3.1 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

We will use the following methods in order to build our model.

4.1 NULL Model

In this section, we will build a **Binary Logistic Regression** Null model utilizing all the variables and data. This model will be considered to be valid and will be modified as we advance.

```
Model_NULL <- glm(target ~ 1,
                  data = training_data,
                  family = binomial(link ="logit"))
summary(Model_NULL)
```

```
##
## Call:
## glm(formula = target ~ 1, family = binomial(link = "logit"),
##     data = training_data)
##
## 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
```

We consider 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 = training_data,
              family = binomial(link ="logit"))
summary(Model_FULL)
```

```
##
## Call:
## glm(formula = target ~ ., family = binomial(link = "logit"),
##     data = training_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8464  -0.1445  -0.0017   0.0029   3.4665
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -40.822934   6.632913  -6.155 7.53e-10 ***
## zn           -0.065946   0.034656  -1.903  0.05706 .
## indus        -0.064614   0.047622  -1.357  0.17485
## chas          0.910765   0.755546   1.205  0.22803
## nox          49.122297   7.931706   6.193 5.90e-10 ***
## rm           -0.587488   0.722847  -0.813  0.41637
## age           0.034189   0.013814   2.475  0.01333 *
## dis           0.738660   0.230275   3.208  0.00134 **
## rad           0.666366   0.163152   4.084 4.42e-05 ***
## tax          -0.006171   0.002955  -2.089  0.03674 *
## ptratio       0.402566   0.126627   3.179  0.00148 **
## lstat         0.045869   0.054049   0.849  0.39608
## medv          0.180824   0.068294   2.648  0.00810 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 192.05  on 453  degrees of freedom
## AIC: 218.05
##
## Number of Fisher Scoring iterations: 9
```

Some variables are not statistically significant. But, we will assume that this is a valid model.

4.3 STEP Procedure

Here, we will create multiple models. Here we go!

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

```
## 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 ***
## + 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 ***
## + 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 .
## + 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 *
## + 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)
## + 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=226.46
```

```
## target ~ nox + rad + tax + ptratio + age
##
##           Df Deviance    AIC    LRT  Pr(>Chi)
## + medv     1   209.55 223.55  4.910   0.02670 *
## + rm       1   212.31 226.31  2.154   0.14217
## + dis      1   212.40 226.40  2.061   0.15115
## <none>         214.46 226.46
## + zn       1   212.67 226.67  1.795   0.18037
## + chas     1   213.24 227.24  1.220   0.26945
## + indus    1   213.38 227.38  1.084   0.29775
## + lstat    1   214.35 228.35  0.113   0.73629
## - age      1   218.70 228.70  4.239   0.03949 *
## - ptratio  1   220.44 230.44  5.983   0.01445 *
## - tax      1   234.99 244.99 20.524 5.889e-06 ***
## - rad      1   286.00 296.00 71.540 < 2.2e-16 ***
## - nox      1   296.04 306.04 81.581 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=223.55
## target ~ nox + rad + tax + ptratio + age + medv
##
##           Df Deviance    AIC    LRT  Pr(>Chi)
## + dis      1   203.45 219.45  6.104  0.013484 *
## <none>         209.55 223.55
## + zn       1   207.64 223.64  1.909  0.167123
## + lstat    1   208.07 224.07  1.477  0.224216
## + chas     1   208.33 224.33  1.223  0.268838
## + indus    1   208.58 224.58  0.973  0.324036
## + rm       1   208.79 224.79  0.766  0.381415
## - medv     1   214.46 226.46  4.910  0.026698 *
## - age      1   215.23 227.23  5.675  0.017204 *
## - ptratio  1   219.94 231.94 10.394  0.001264 **
## - tax      1   224.71 236.71 15.159 9.885e-05 ***
## - rad      1   269.51 281.51 59.960 9.679e-15 ***
## - nox      1   294.08 306.08 84.529 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=219.45
## target ~ nox + rad + tax + ptratio + age + medv + dis
##
##           Df Deviance    AIC    LRT  Pr(>Chi)
## + zn       1   197.32 215.32  6.124 0.0133321 *
## + chas     1   201.29 219.29  2.157 0.1419100
## + rm       1   201.35 219.35  2.093 0.1480183
## <none>         203.45 219.45
## + lstat    1   202.05 220.05  1.393 0.2378583
## + indus    1   202.23 220.23  1.220 0.2693725
## - dis      1   209.55 223.55  6.104 0.0134845 *
## - medv     1   212.40 226.40  8.954 0.0027685 **
## - age      1   212.97 226.97  9.519 0.0020335 **
## - tax      1   216.21 230.21 12.760 0.0003541 ***
## - ptratio  1   216.35 230.35 12.907 0.0003274 ***
```

```
## - rad      1   259.98 273.98 56.530 5.534e-14 ***
## - nox      1   278.84 292.84 75.390 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=215.32
## target ~ nox + rad + tax + ptratio + age + medv + dis + zn
##
##           Df Deviance    AIC    LRT  Pr(>Chi)
## <none>         197.32 215.32
## + lstat    1   195.51 215.51  1.808 0.1787290
## + rm       1   195.75 215.75  1.575 0.2094316
## + chas     1   195.97 215.97  1.349 0.2454148
## + indus    1   196.33 216.33  0.995 0.3185882
## - zn       1   203.45 219.45  6.124 0.0133321 *
## - ptratio  1   206.27 222.27  8.948 0.0027770 **
## - age      1   207.13 223.13  9.810 0.0017361 **
## - tax      1   207.62 223.62 10.293 0.0013356 **
## - dis      1   207.64 223.64 10.320 0.0013157 **
## - medv     1   208.65 224.65 11.326 0.0007644 ***
## - rad      1   250.98 266.98 53.659 2.385e-13 ***
## - nox      1   273.18 289.18 75.852 < 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 + age + medv +
##     dis + zn, family = binomial(link = "logit"), data = training_data)
##
## Coefficients:
## (Intercept)          nox          rad          tax      ptratio
##  -37.415922    42.807768     0.725109    -0.007756     0.323628
##         age         medv          dis           zn
##    0.032950     0.110472     0.654896    -0.068648
##
## Degrees of Freedom: 465 Total (i.e. Null);  457 Residual
## Null Deviance:       645.9
## Residual Deviance: 197.3     AIC: 215.3
```

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, age, medv, dis, zn**.

4.3.1 ANOVA results

From the results above, we can see the ANOVA table.

```
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
+ age	-1	4.239474	460	214.4620	226.4620
+ medv	-1	4.910242	459	209.5518	223.5518
+ dis	-1	6.104410	458	203.4473	219.4473
+ zn	-1	6.124494	457	197.3229	215.3229

Nota Bene:

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 that just by adding certain variables, our AIC values decrease making it a better model.

4.4 AIC Model

From the above, we conclude that the best model is as follows:

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

```
##
## Call:
## glm(formula = target ~ nox + rad + tax + ptratio + age + medv +
##     dis + zn + lstat, family = binomial(link = "logit"), data = training_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9279  -0.1640  -0.0016   0.0027   3.3989
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -38.786168   6.161170  -6.295 3.07e-10 ***
## nox          43.014846   6.689945   6.430 1.28e-10 ***
## rad           0.739415   0.152375   4.853 1.22e-06 ***
## tax          -0.008045   0.002651  -3.035 0.002408 **
## ptratio       0.334196   0.111780   2.990 0.002792 **
## age           0.028379   0.011401   2.489 0.012809 *
## medv          0.138257   0.041630   3.321 0.000897 ***
```

```
## dis           0.658786   0.214787   3.067 0.002161 **
## zn           -0.072933   0.033022  -2.209 0.027199 *
## lstat         0.064814   0.048427   1.338 0.180769
## ---
## 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: 195.51  on 456  degrees of freedom
## AIC: 215.51
##
## 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, we will create a new modified model by excluding `lstat` from the previous model.

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

```
##
## Call:
## glm(formula = target ~ nox + rad + tax + ptratio + age + medv +
##     dis + zn, family = binomial(link = "logit"), data = training_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8295  -0.1752  -0.0021   0.0032   3.4191
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -37.415922   6.035013  -6.200 5.65e-10 ***
## nox          42.807768   6.678692   6.410 1.46e-10 ***
## rad           0.725109   0.149788   4.841 1.29e-06 ***
## tax          -0.007756   0.002653  -2.924  0.00346 **
## ptratio       0.323628   0.111390   2.905  0.00367 **
## age           0.032950   0.010951   3.009  0.00262 **
## medv          0.110472   0.035445   3.117  0.00183 **
## dis           0.654896   0.214050   3.060  0.00222 **
## zn           -0.068648   0.032019  -2.144  0.03203 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 197.32  on 457  degrees of freedom
## AIC: 215.32
##
## Number of Fisher Scoring iterations: 9
```

Worthy to note that 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

According to the correlations table, some variables are more correlation to `target` than others. In this section, we will create a model based on that output by including the following variables only and we will use it in order to choose our best selected model.

Variables
target
indus
nox
age
dis
rad
tax

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

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

```
##
## Call:
## glm(formula = target ~ indus + nox + age + dis + rad + tax, family = binomial(link = logit),
##     data = training_data)
##
## 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, we can see that `indus` and `dis` are not statistically significant. Also, we notice how the AIC value has increased in a moderate way, along side the Residual Deviance, which is not good. So let's **refine** this model.

4.7 Intuition Model Refined

Here, we'll conside backward elimination. So let's exclude the variables `indus` and `dis`.

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

```
##
## Call:
## glm(formula = target ~ nox + age + rad + tax, family = binomial(link = logit),
##     data = training_data)
##
## 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
```

We notice how all the given predictors are statistically significant but the AIC has increased, the Median is higher than before and the residual deviance also increased.

5 MODEL SELECTION

From the above possible models, we 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, medv, age, dis, zn**.

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

```
##
## Call:
## glm(formula = target ~ nox + rad + tax + ptratio + age + medv +
##     dis + zn, family = binomial(link = "logit"), data = training_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8295  -0.1752  -0.0021   0.0032   3.4191
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -37.415922   6.035013  -6.200 5.65e-10 ***
## nox          42.807768   6.678692   6.410 1.46e-10 ***
## rad           0.725109   0.149788   4.841 1.29e-06 ***
## tax          -0.007756   0.002653  -2.924  0.00346 **
## ptratio       0.323628   0.111390   2.905  0.00367 **
## age           0.032950   0.010951   3.009  0.00262 **
## medv          0.110472   0.035445   3.117  0.00183 **
## dis           0.654896   0.214050   3.060  0.00222 **
## zn           -0.068648   0.032019  -2.144  0.03203 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 197.32  on 457  degrees of freedom
## AIC: 215.32
##
## Number of Fisher Scoring iterations: 9
```

How we choose our final model this way, because:

- 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	41.083014	0.0000000
rad	1	23.434261	0.0000013
tax	1	8.548142	0.0034588
ptratio	1	8.441155	0.0036682
age	1	9.053579	0.0026218
medv	1	9.713774	0.0018289
dis	1	9.360848	0.0022167
zn	1	4.596726	0.0320331

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',
                'medv', 'age', 'dis', 'zn', 'lstat')
training_data.final <- training_data[select_var]
```

5.1.1 Final Model Comparisons

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

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

```
##
## Call:
## glm(formula = target ~ 1, family = binomial(link = "logit"),
##     data = training_data.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

Let's 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)
457	197.3229	NA	NA	NA
465	645.8758	-8	-448.553	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 units compared to our final model.

5.1.3 Likelihood ratio test

In order to do so, we 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)
```

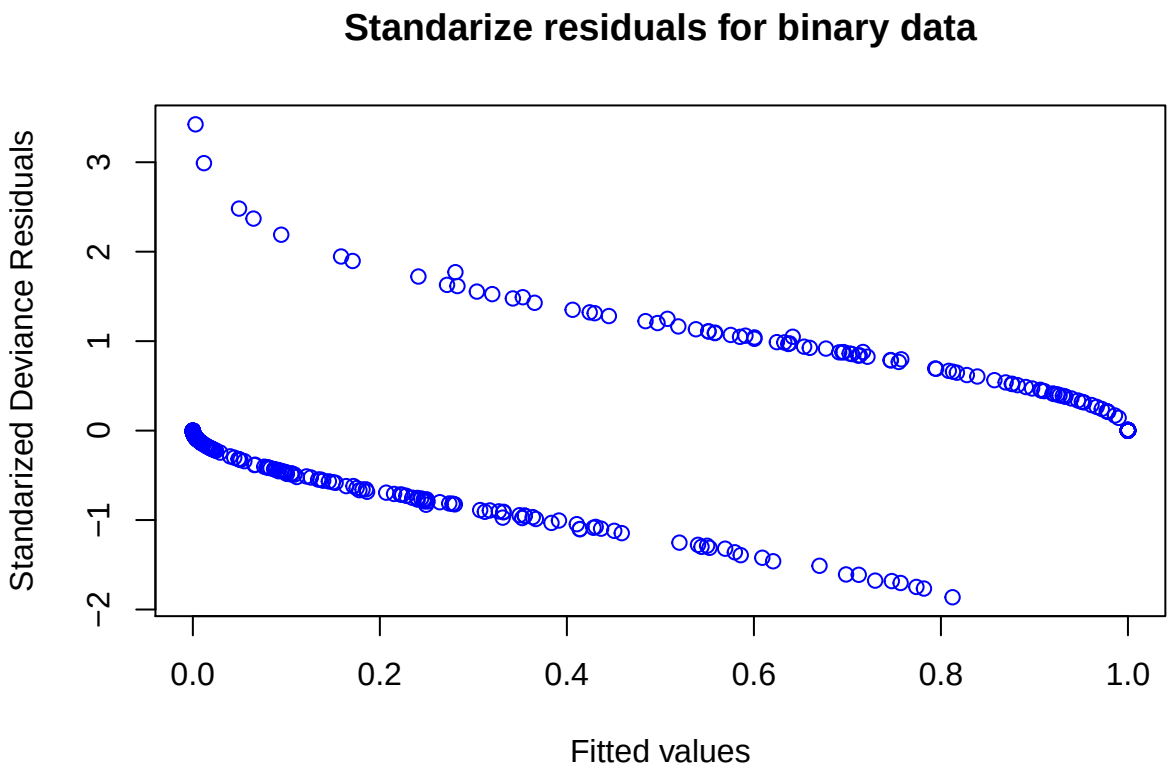
#Df	LogLik	Df	Chisq	Pr(>Chisq)
9	-98.66143	NA	NA	NA
1	-322.93791	-8	448.553	0

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.

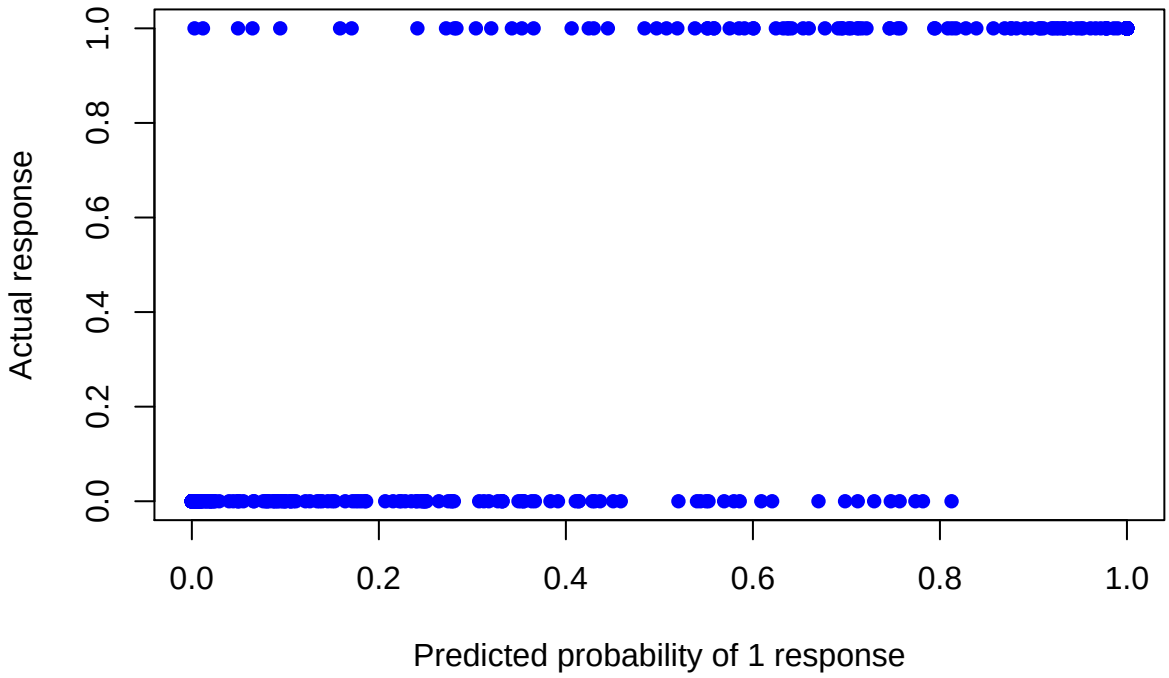
```
plot(fitted(Model_FINAL),
     rstandard(Model_FINAL),
     main = 'Standarize residuals for binary data',
     xlab = 'Fitted values',
     ylab = 'Standarized Deviance Residuals',
     col = 'blue')
```



5.1.5 Simple plot of predictions

This visual is a representation of the predicted values versus the given values aka `target`.

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



5.2 Evaluations

In this section, we will proceed to evaluate our 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, we will need to perform some transformations to round the given probabilities to zero decimals.

```
training_data.final$predicted_target <- round(training_data.final$predict,0)
training_data_table <- table(training_data.final$predicted_target,
                             training_data.final$target,
                             dnn = c("Predicted", "Target"))
data.frame(training_data_table)
```

Predicted	Target	Freq
0	0	218
1	0	19
0	1	22
1	1	207

5.2.1 Confusion Matrix

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

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

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 218  22
##          1  19 207
##
##                Accuracy : 0.912
##                  95% CI : (0.8825, 0.9361)
##     No Information Rate : 0.5086
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8239
##
##  Mcnemar's Test P-Value : 0.7548
##
##             Sensitivity : 0.9039
##             Specificity : 0.9198
##          Pos Pred Value : 0.9159
##          Neg Pred Value : 0.9083
##              Prevalence : 0.4914
##          Detection Rate : 0.4442
##    Detection Prevalence : 0.4850
##       Balanced Accuracy : 0.9119
##
##        'Positive' Class : 1
##
```

From the above results, we obtain as follows:

	Value
Sensitivity	0.9039301
Specificity	0.9198312
Pos Pred Value	0.9159292
Neg Pred Value	0.9083333
Precision	0.9159292
Recall	0.9039301
F1	0.9098901
Prevalence	0.4914163
Detection Rate	0.4442060
Detection Prevalence	0.4849785
Balanced Accuracy	0.9118807

5.2.2 ROC and AUC

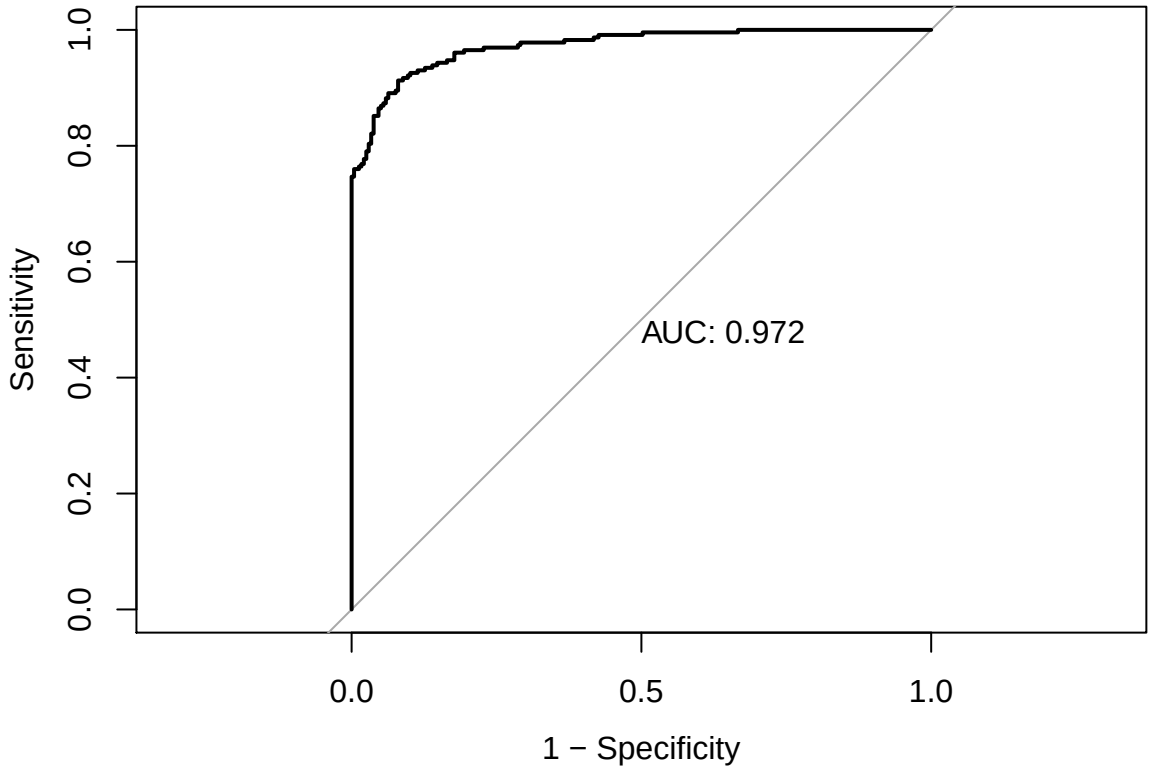
As we know, the **Receiver Operating Characteristic Curves** (ROC) is a great quantitative assessment tool of the model. In order to quantify our model, we will employ as follows:

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

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

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

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



Let's see our confidence intervals.

	AUC
Lower bound	0.9595128
Estimated value	0.9719382
Higher bound	0.9843635

6 PREDICTIONS

6.1 Table

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

6.2 Classification and probability

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

predicted	probability
0	0.052
1	0.656
1	0.729
0	0.426
0	0.108
0	0.313
0	0.388
0	0.014
0	0.006
0	0.002
1	0.502
0	0.417
1	0.841
1	0.743
1	0.650
0	0.149
0	0.403
1	0.967
0	0.079
0	0.000
0	0.000
0	0.052
0	0.152
0	0.199
0	0.178
1	0.677
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	0.802
0	0.395

7 APPENDIX

Code and markdown are at:

https://github.com/theoracley/Data621/blob/master/Homework3/Abdelmalek_Hajjam_HW3.Rmd