

Data 621 Homework 3: Boston Crime Rates

*Tommy Jenkins, Violeta Stoyanova, Todd Weigel, Peter Kowalchuk, Eleanor R-Secoquian
October, 2019*

OVERVIEW

In this homework assignment, we 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).

Objective:

The 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.

DATA EXPLORATION

Data Summary

The dataset consists of two data files: training and evaluation. The training dataset contains 13 columns, while the evaluation dataset contains 12. The evaluation dataset is missing column target which represent our response variable and defines whether the crime rate is above the median crime rate (1) or not (0). We will start by exploring the training data set since it will be the one used to generate the regression model.

First we see that all data is numeric. The dataset does contain one dummy variable to identify if the property borders the Charles River (1) or not (0).

An important aspect of any dataset is to determine how much, if any, data is missing. We look at all the variables to see which if any have missing data. We look at the basic descriptive statistics as well as the missing data and their percentages:

```
vars
n
mean
sd
median
trimmed
mad
min
max
range
skew
kurtosis
se
na_count
```

na_count_perc

zn

1

466

11.5772532

23.3646511

0.00000

5.3542781

0.0000000

0.0000

100.0000

100.0000

2.1768152

3.8135765

1.0823466

0

0

indus

2

466

11.1050215

6.8458549

9.69000

10.9082353

9.3403800

0.4600

27.7400

27.2800

0.2885450

-1.2432132

0.3171281

0

0

chas

3

466

0.0708155
0.2567920
0.00000
0.0000000
0.0000000
0.0000
1.0000
1.0000
3.3354899
9.1451313
0.0118957
0
0
nox
4
466
0.5543105
0.1166667
0.53800
0.5442684
0.1334340
0.3890
0.8710
0.4820
0.7463281
-0.0357736
0.0054045
0
0
rm
5
466
6.2906738
0.7048513
6.21000
6.2570615

0.5166861
3.8630
8.7800
4.9170
0.4793202
1.5424378
0.0326516
0
0
age
6
466
68.3675966
28.3213784
77.15000
70.9553476
30.0226500
2.9000
100.0000
97.1000
-0.5777075
-1.0098814
1.3119625
0
0
dis
7
466
3.7956929
2.1069496
3.19095
3.5443647
1.9144814
1.1296
12.1265
10.9969

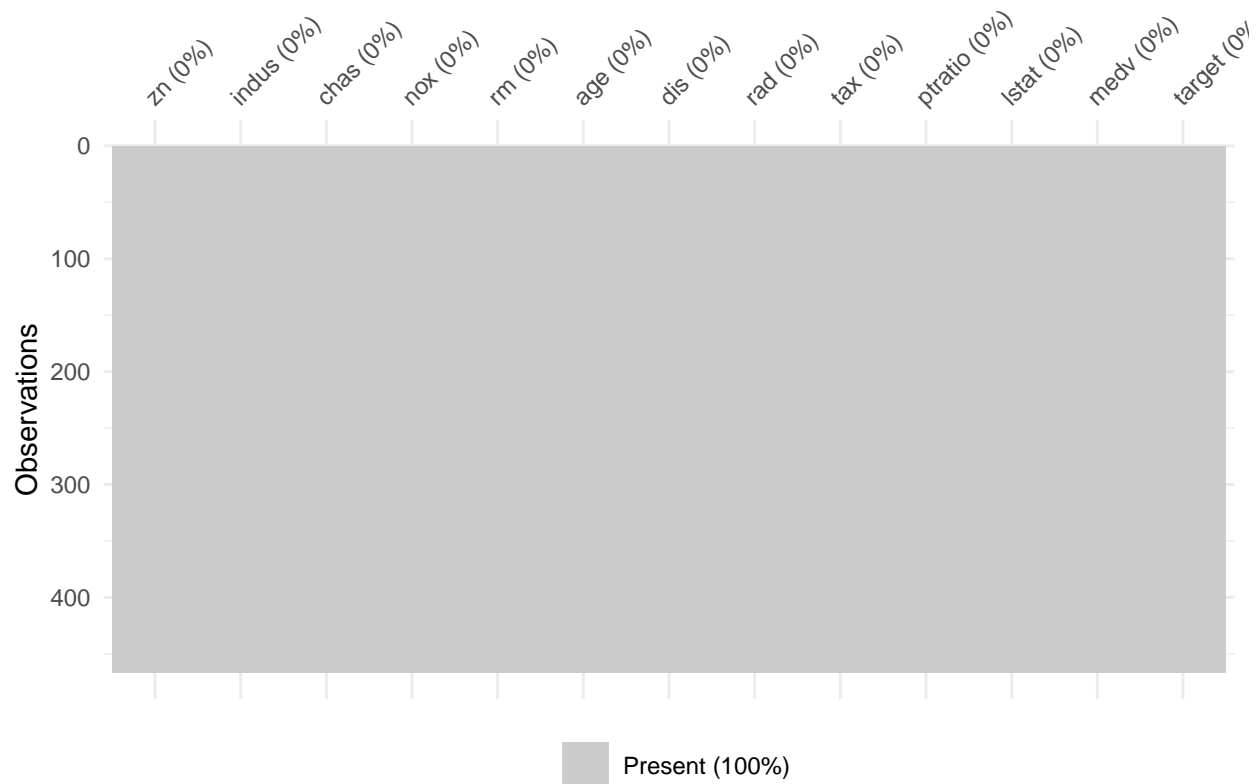
0.9988926
0.4719679
0.0976026
0
0
rad
8
466
9.5300429
8.6859272
5.00000
8.6978610
1.4826000
1.0000
24.0000
23.0000
1.0102788
-0.8619110
0.4023678
0
0
tax
9
466
409.5021459
167.9000887
334.50000
401.5080214
104.5233000
187.0000
711.0000
524.0000
0.6593136
-1.1480456
7.7778214
0

0
ptratio
10
466
18.3984979
2.1968447
18.90000
18.5970588
1.9273800
12.6000
22.0000
9.4000
-0.7542681
-0.4003627
0.1017669
0
0
lstat
11
466
12.6314592
7.1018907
11.35000
11.8809626
7.0720020
1.7300
37.9700
36.2400
0.9055864
0.5033688
0.3289887
0
0
medv
12
466

```

22.5892704
9.2396814
21.20000
21.6304813
6.0045300
5.0000
50.0000
45.0000
1.0766920
1.3737825
0.4280200
0
0
target
13
466
0.4914163
0.5004636
0.00000
0.4893048
0.0000000
0.0000
1.0000
1.0000
0.0342293
-2.0031131
0.0231835
0
0
##      zn      indus      chas      nox      rm      age      dis      rad      tax
##      0         0         0         0         0         0         0         0         0
## ptratio  lstat    medv  target
##      0         0         0         0

```

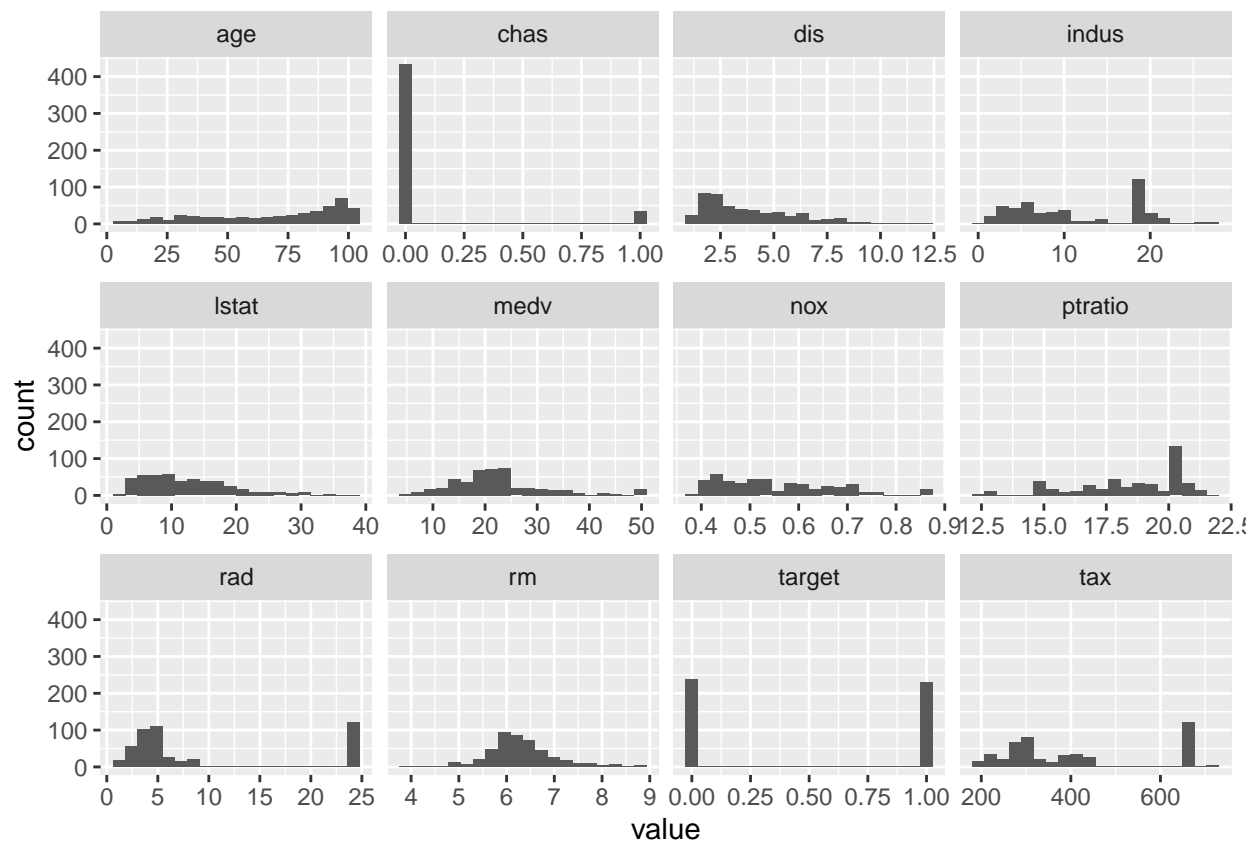


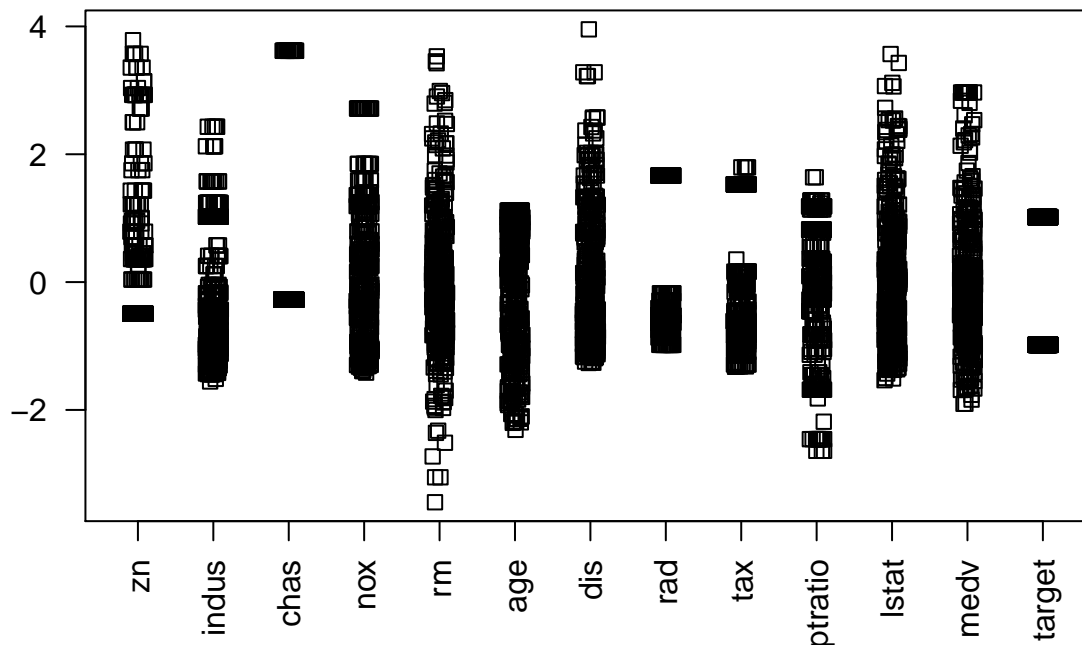
##	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target
## 1	0	19.58	0	0.605	7.929	96.2	2.0459	5	403	14.7	3.70	50.0	1
## 2	0	19.58	1	0.871	5.403	100.0	1.3216	5	403	14.7	26.82	13.4	1
## 3	0	18.10	0	0.740	6.485	100.0	1.9784	24	666	20.2	18.85	15.4	1
## 4	30	4.93	0	0.428	6.393	7.8	7.0355	6	300	16.6	5.19	23.7	0
## 5	0	2.46	0	0.488	7.155	92.2	2.7006	3	193	17.8	4.82	37.9	0
## 6	0	8.56	0	0.520	6.781	71.3	2.8561	5	384	20.9	7.67	26.5	0

Missing and Invalid Data

No missing data was found in the dataset.

With missing data assessed, we can look into the data in more detail. To visualize this we plot histograms for each data. Several predictors like dist, chas, rad, zn and tax are not normally distributed and noticeable outliers.





DATA PREPARATION

Fix missing values

No data was found missing.

Mathematical transformations.

Box Cox The Box Cox transformation tries to transform non-normal data into a normal distribution. This transformation attempts to estimate the λ for Y. With the exception of tax, all predictors have either no transformation estimate or were given a fudge value of 0.

```
## $zn
## Box-Cox Transformation
##
## 466 data points used to estimate Lambda
##
## Input data summary:
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.00   0.00   0.00  11.58  16.25  100.00
##
## Lambda could not be estimated; no transformation is applied
##
##
## $indus
## Box-Cox Transformation
```

```

##
## 466 data points used to estimate Lambda
##
## Input data summary:
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##    0.460   5.145   9.690  11.105  18.100  27.740
##
## Largest/Smallest: 60.3
## Sample Skewness: 0.289
##
## Estimated Lambda: 0.4
##
##
## $chas
## Box-Cox Transformation
##
## 466 data points used to estimate Lambda
##
## Input data summary:
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
## 0.00000 0.00000 0.00000 0.07082 0.00000 1.00000
##
## Lambda could not be estimated; no transformation is applied
##
##
## $nox
## Box-Cox Transformation
##
## 466 data points used to estimate Lambda
##
## Input data summary:
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
## 0.3890 0.4480 0.5380 0.5543 0.6240 0.8710
##
## Largest/Smallest: 2.24
## Sample Skewness: 0.746
##
## Estimated Lambda: -0.9
##
##
## $rm
## Box-Cox Transformation
##
## 466 data points used to estimate Lambda
##
## Input data summary:
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
## 3.863 5.887 6.210 6.291 6.630 8.780
##
## Largest/Smallest: 2.27
## Sample Skewness: 0.479
##
## Estimated Lambda: 0.2
##

```

```

##
## $age
## Box-Cox Transformation
##
## 466 data points used to estimate Lambda
##
## Input data summary:
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      2.90  43.88   77.15   68.37   94.10  100.00
##
## Largest/Smallest: 34.5
## Sample Skewness: -0.578
##
## Estimated Lambda: 1.3
##
##
## $dis
## Box-Cox Transformation
##
## 466 data points used to estimate Lambda
##
## Input data summary:
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      1.130   2.101   3.191   3.796   5.215  12.127
##
## Largest/Smallest: 10.7
## Sample Skewness: 0.999
##
## Estimated Lambda: -0.1
## With fudge factor, Lambda = 0 will be used for transformations
##
##
## $rad
## Box-Cox Transformation
##
## 466 data points used to estimate Lambda
##
## Input data summary:
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      1.00   4.00   5.00   9.53   24.00   24.00
##
## Largest/Smallest: 24
## Sample Skewness: 1.01
##
## Estimated Lambda: -0.2
## With fudge factor, Lambda = 0 will be used for transformations
##
##
## $tax
## Box-Cox Transformation
##
## 466 data points used to estimate Lambda
##
## Input data summary:

```

```

##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##    187.0   281.0   334.5   409.5   666.0   711.0
##
## Largest/Smallest: 3.8
## Sample Skewness: 0.659
##
## Estimated Lambda: -0.5
##
##
## $ptratio
## Box-Cox Transformation
##
## 466 data points used to estimate Lambda
##
## Input data summary:
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##     12.6    16.9    18.9    18.4    20.2    22.0
##
## Largest/Smallest: 1.75
## Sample Skewness: -0.754
##
## Estimated Lambda: 2
##
##
## $lstat
## Box-Cox Transformation
##
## 466 data points used to estimate Lambda
##
## Input data summary:
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##     1.730    7.043   11.350   12.631   16.930   37.970
##
## Largest/Smallest: 21.9
## Sample Skewness: 0.906
##
## Estimated Lambda: 0.2
##
##
## $medv
## Box-Cox Transformation
##
## 466 data points used to estimate Lambda
##
## Input data summary:
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      5.00   17.02   21.20   22.59   25.00   50.00
##
## Largest/Smallest: 10
## Sample Skewness: 1.08
##
## Estimated Lambda: 0.2
##
##

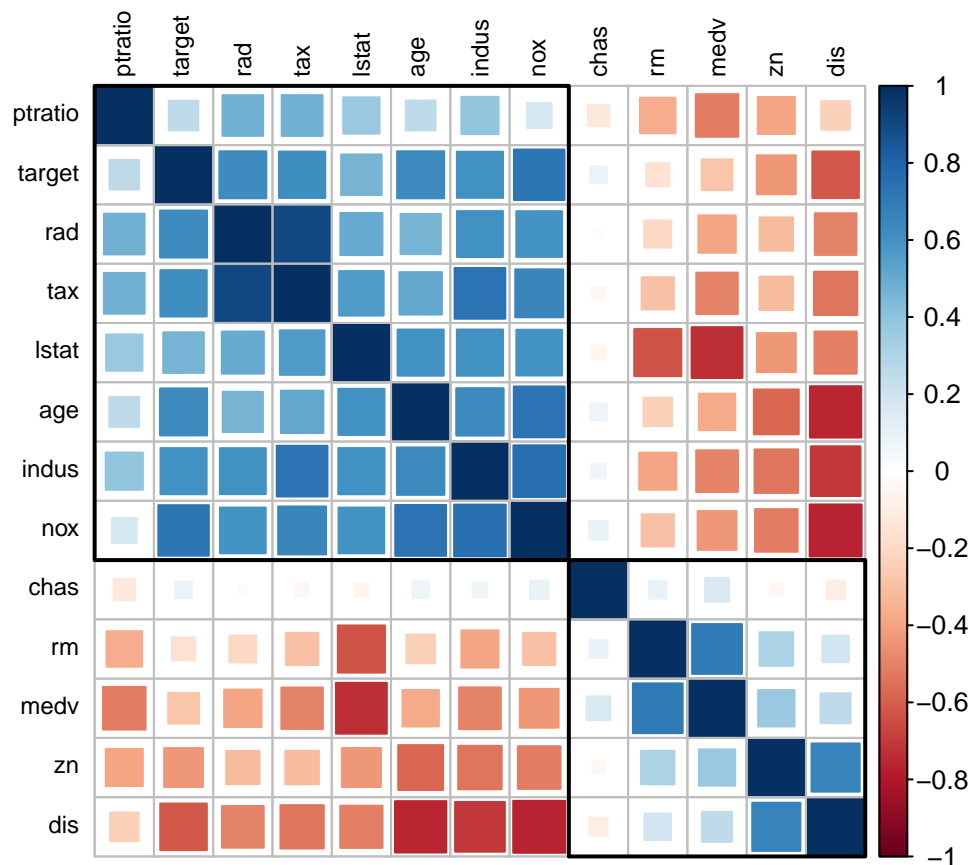
```

```
## $target
## Box-Cox Transformation
##
## 466 data points used to estimate Lambda
##
## Input data summary:
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.0000  0.0000  0.0000  0.4914  1.0000  1.0000
##
## Lambda could not be estimated; no transformation is applied
```

Variable Creation / Removal

To determine how we can combine variables to create new one we start by looking at a correlation plot. The plot and cor function lists nox, age, rad, tax and indus as the strongest positively correlated predictors, while rad and distance are the strongest negatively correlated predictors.

```
##      indus      chas      nox      rm      age      dis
## [1,] 0.6048507 0.08004187 0.7261062 -0.1525533 0.6301062 -0.6186731
##      rad      tax  ptratio  lstat  medv target
## [1,] 0.6281049 0.6111133 0.2508489 0.469127 -0.2705507      1
```



BUILD MODELS

```
### General regression
```

We start by building a model with all the predictors in the dataset.

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

The Summary of this model shows several predictor are not relevant. We build a second model without these predictors.

```
##
## Call:
## glm(formula = target ~ nox + age + dis + rad + tax + ptratio +
##      medv, family = binomial(link = "logit"), data = crimeTrain)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.01059  -0.19744  -0.01371   0.00402   3.06424
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -36.824228   5.858405  -6.286 3.26e-10 ***
## nox         42.338378   6.639207   6.377 1.81e-10 ***
## age          0.031882   0.010693   2.982 0.002867 **
## dis          0.429555   0.171849   2.500 0.012433 *
## rad          0.701767   0.139426   5.033 4.82e-07 ***
## tax         -0.008237   0.002534  -3.250 0.001153 **
```

```
## ptratio      0.376575    0.108912    3.458 0.000545 ***
## medv         0.093653    0.033556    2.791 0.005255 **
## ---
## 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: 203.45  on 458  degrees of freedom
## AIC: 219.45
##
## Number of Fisher Scoring iterations: 9
## [1] 1
## [1] 1
```

The new model has a slightly higher AIC which would tell us the first model is slightly less complex. For the 2 data sets p-value = $1 - \text{pchisq}(\text{deviance}, \text{degrees of freedom})$ are 1. The Null hypothesis is still supported.

AIC Step Method

Another way of selecting which predictors to use in the model is by calculating the AIC of the model. This metric is similar to the adjusted R-square of a model in that it penalizes models with more predictors over simpler model with few predictors. We use StepAIC function in R to find the lowest AIC with different predictors.

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



```

## - age      1    199.11 221.11
## - ptratio  1    203.53 225.53
## - dis      1    203.85 225.85
## - medv     1    205.35 227.35
## - rad      1    233.81 255.81
## - nox      1    265.14 287.14
##
## Step: AIC=216.24
## target ~ zn + indus + nox + age + dis + rad + tax + ptratio +
##         lstat + medv
##
##           Df Deviance    AIC
## - indus    1    195.51 215.51
## <none>      194.24 216.24
## - lstat    1    196.33 216.33
## - zn       1    200.59 220.59
## - tax      1    200.75 220.75
## - age      1    201.00 221.00
## - ptratio  1    203.94 223.94
## - dis      1    204.83 224.83
## - medv     1    207.12 227.12
## - rad      1    241.41 261.41
## - nox      1    265.19 285.19
##
## Step: AIC=215.51
## target ~ zn + nox + age + dis + rad + tax + ptratio + lstat +
##         medv
##
##           Df Deviance    AIC
## - lstat    1    197.32 215.32
## <none>      195.51 215.51
## - zn       1    202.05 220.05
## - age      1    202.23 220.23
## - ptratio  1    205.01 223.01
## - dis      1    205.96 223.96
## - tax      1    206.60 224.60
## - medv     1    208.13 226.13
## - rad      1    249.55 267.55
## - nox      1    270.59 288.59
##
## Step: AIC=215.32
## target ~ zn + nox + age + dis + rad + tax + ptratio + medv
##
##           Df Deviance    AIC
## <none>      197.32 215.32
## - zn       1    203.45 219.45
## - ptratio  1    206.27 222.27
## - age      1    207.13 223.13
## - tax      1    207.62 223.62
## - dis      1    207.64 223.64
## - medv     1    208.65 224.65
## - rad      1    250.98 266.98
## - nox      1    273.18 289.18

```

```
##
## Call:
## glm(formula = target ~ zn + nox + age + dis + rad + tax + ptratio +
##      medv, family = binomial(link = "logit"), data = crimeTrain)
##
## 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 ***
## zn          -0.068648   0.032019  -2.144  0.03203 *
## nox          42.807768   6.678692   6.410 1.46e-10 ***
## age           0.032950   0.010951   3.009  0.00262 **
## dis           0.654896   0.214050   3.060  0.00222 **
## 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 **
## medv          0.110472   0.035445   3.117  0.00183 **
## ---
## 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
```

This reduces the predictors used in the model to these: zn nox age dis rad tax ptratio medv

It Removes these predictors: indus chas rm#

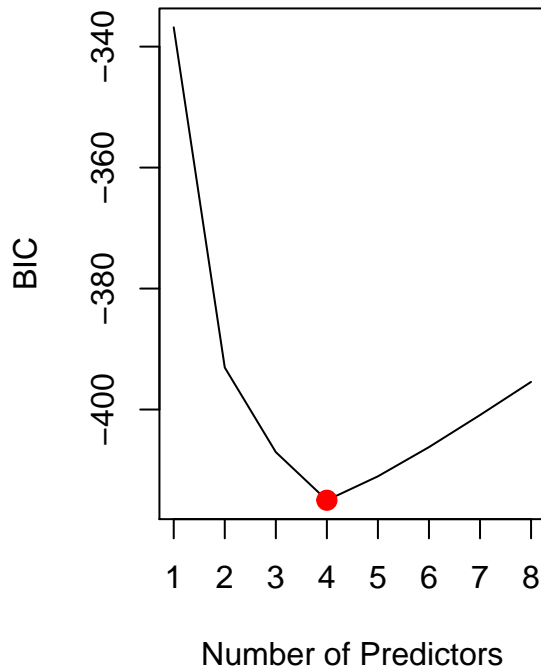
The AIC improves marginally from 218.05 (our original general model) to 215.32, but we also benefit by having a simpler model less prone to overfitting.

Also, the predictors in the model now are all significant (under 0.05 pr level) and all but one under .01 or very significant. Which is much improved over the prior model

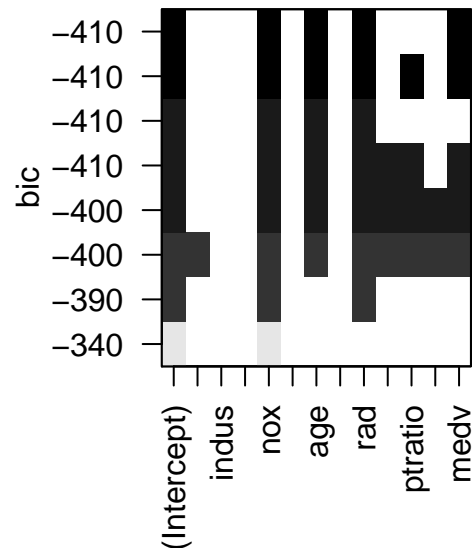
###BIC Method

To determine the number of predictors and which predictors to be used we will use the Bayesian Information Criterion (BIC).

Subset Selection Using BIC



Predictors vs. BIC



The plot on the right shows that the number of predictors with the lowest BIC are **nox** , **age**, **rad**, and **medv**. We will use those predictors to build the next model

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-17.63	2.168	-8.131	4.246e-16
nox	23.62	3.936	6.003	1.942e-09
age	0.01824	0.009172	1.989	0.04673
rad	0.4528	0.1093	4.144	3.413e-05
medv	0.04481	0.02319	1.932	0.05338

(Dispersion parameter for binomial family taken to be 1)

Null deviance:	645.9 on 465 degrees of freedom
Residual deviance:	232.8 on 461 degrees of freedom

###Forward Selection Method using some BoxCox transformed independent variables:

```
m4 <- step(glm(target~1, data=crimeTrain), direction = "forward", scope = ~zn + I(log(indus)) + I(sqrt(
```

```
## Start: AIC=680.3
```

```
## target ~ 1
```

```
##
```

```
##          Df Deviance    AIC
```

```
## + I(nox^-1)    1  50.349 291.51
```

```

## + I(age^2)      1  66.713 422.64
## + I(dis^-0.5)   1  66.801 423.26
## + rad           1  70.518 448.50
## + I(log(indus)) 1  74.068 471.38
## + I(tax^-1)     1  74.547 474.39
## + lstat         1  90.834 566.47
## + zn           1  94.762 586.20
## + I(ptratio^2)  1 107.479 644.88
## + medv          1 107.941 646.88
## + I(log(rm))    1 112.912 667.86
## + I(sqrt(chas)) 1 115.720 679.31
## <none>          116.466 680.30
##
## Step:  AIC=291.51
## target ~ I(nox^-1)
##
##           Df Deviance   AIC
## + rad      1  45.272 243.97
## + I(tax^-1) 1  46.956 260.99
## + I(age^2)  1  49.650 286.99
## + I(ptratio^2) 1 49.778 288.19
## + I(log(rm)) 1  49.876 289.10
## + medv      1  49.907 289.40
## + I(log(indus)) 1 50.043 290.66
## <none>      50.349 291.51
## + zn        1  50.147 291.63
## + I(sqrt(chas)) 1 50.305 293.10
## + lstat     1  50.336 293.38
## + I(dis^-0.5) 1  50.345 293.46
##
## Step:  AIC=243.97
## target ~ I(nox^-1) + rad
##
##           Df Deviance   AIC
## + medv      1  44.061 233.33
## + I(age^2)   1  44.442 237.35
## + I(log(rm)) 1  44.674 239.77
## + I(tax^-1)  1  45.017 243.34
## <none>      45.272 243.97
## + I(sqrt(chas)) 1 45.113 244.33
## + lstat     1  45.149 244.71
## + I(dis^-0.5) 1  45.180 245.03
## + zn        1  45.223 245.47
## + I(ptratio^2) 1 45.240 245.64
## + I(log(indus)) 1 45.267 245.92
##
## Step:  AIC=233.33
## target ~ I(nox^-1) + rad + medv
##
##           Df Deviance   AIC
## + I(age^2)   1  43.027 224.27
## + I(tax^-1)  1  43.368 227.96
## + I(dis^-0.5) 1  43.827 232.86
## + lstat     1  43.834 232.93

```

```

## + I(log(indus)) 1 43.856 233.17
## <none> 44.061 233.33
## + I(ptratio^2) 1 43.956 234.23
## + I(sqrt(chas)) 1 44.030 235.01
## + I(log(rm)) 1 44.052 235.24
## + zn 1 44.060 235.33
##
## Step: AIC=224.27
## target ~ I(nox^-1) + rad + medv + I(age^2)
##
## Df Deviance AIC
## + I(dis^-0.5) 1 42.184 217.05
## + I(tax^-1) 1 42.397 219.40
## <none> 43.027 224.27
## + I(log(indus)) 1 42.888 224.77
## + I(ptratio^2) 1 42.975 225.71
## + I(sqrt(chas)) 1 43.004 226.02
## + lstat 1 43.006 226.05
## + zn 1 43.013 226.12
## + I(log(rm)) 1 43.024 226.24
##
## Step: AIC=217.05
## target ~ I(nox^-1) + rad + medv + I(age^2) + I(dis^-0.5)
##
## Df Deviance AIC
## + I(tax^-1) 1 41.399 210.29
## + I(log(indus)) 1 41.866 215.53
## <none> 42.184 217.05
## + lstat 1 42.036 217.41
## + I(ptratio^2) 1 42.124 218.39
## + I(log(rm)) 1 42.150 218.67
## + I(sqrt(chas)) 1 42.169 218.89
## + zn 1 42.173 218.93
##
## Step: AIC=210.29
## target ~ I(nox^-1) + rad + medv + I(age^2) + I(dis^-0.5) + I(tax^-1)
##
## Df Deviance AIC
## + lstat 1 41.180 209.83
## <none> 41.399 210.29
## + I(log(indus)) 1 41.232 210.42
## + I(ptratio^2) 1 41.318 211.38
## + I(log(rm)) 1 41.360 211.86
## + I(sqrt(chas)) 1 41.374 212.02
## + zn 1 41.396 212.27
##
## Step: AIC=209.83
## target ~ I(nox^-1) + rad + medv + I(age^2) + I(dis^-0.5) + I(tax^-1) +
## lstat
##
## Df Deviance AIC
## <none> 41.180 209.83
## + I(log(indus)) 1 41.012 209.92
## + I(ptratio^2) 1 41.062 210.49

```

```
## + I(sqrt(chas)) 1 41.159 211.59
## + zn 1 41.174 211.76
## + I(log(rm)) 1 41.178 211.80
```

```
summary(m4)
```

```
##
## Call:
## glm(formula = target ~ I(nox^-1) + rad + medv + I(age^2) + I(dis^-0.5) +
##      I(tax^-1) + lstat, data = crimeTrain)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.70627  -0.18647  -0.02143   0.13160   0.98687
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.099e+00  2.682e-01   7.826 3.52e-14 ***
## I(nox^-1)    -8.407e-01  8.932e-02  -9.413 < 2e-16 ***
## rad          1.258e-02  2.698e-03   4.661 4.13e-06 ***
## medv         1.195e-02  2.464e-03   4.850 1.69e-06 ***
## I(age^2)     2.846e-05  7.544e-06   3.772 0.000183 ***
## I(dis^-0.5) -7.689e-01  2.123e-01  -3.621 0.000326 ***
## I(tax^-1)    -7.196e+01  2.332e+01  -3.086 0.002155 **
## lstat        5.686e-03  3.647e-03   1.559 0.119675
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.08991273)
##
##      Null deviance: 116.47  on 465  degrees of freedom
## Residual deviance:  41.18  on 458  degrees of freedom
## AIC: 209.83
##
## Number of Fisher Scoring iterations: 2
```

SELECT MODELS

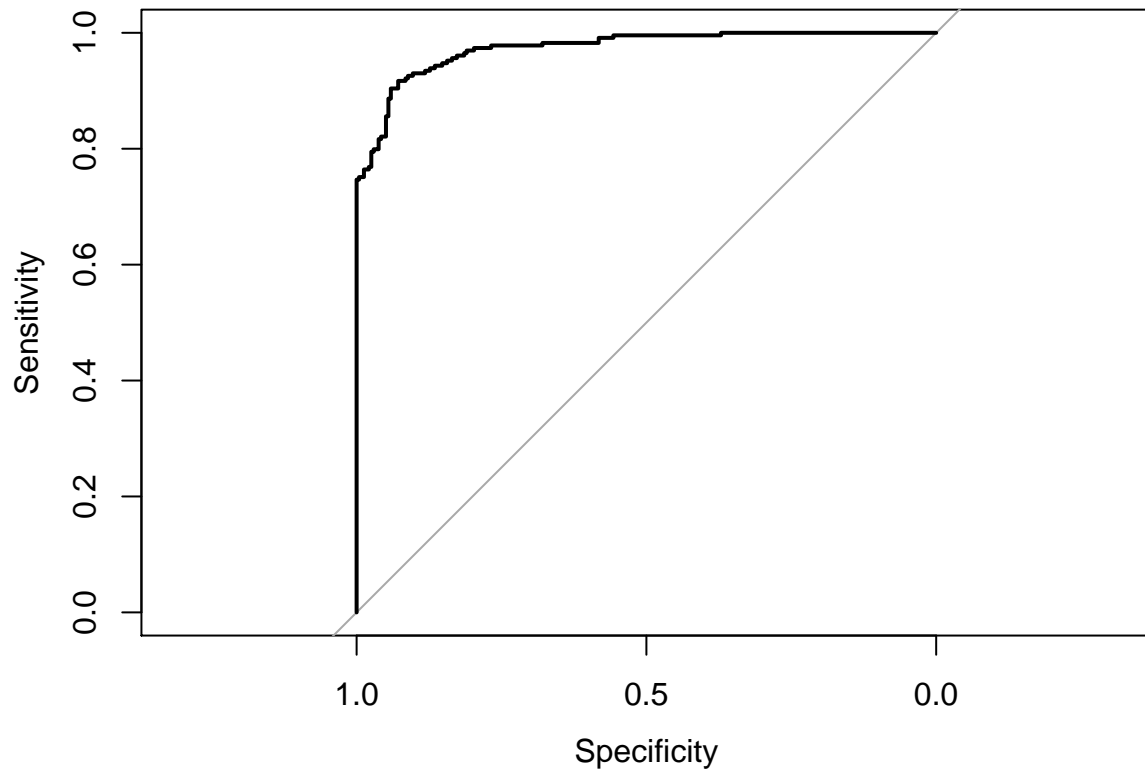
Compare Model Statistics

```
###Model 1 - General Model
```

```
####Complete general model
```

ROC Curve

The ROC Curve helps measure true positives and true negative. A high AUC or area under the curve tells us the model is predicting well.

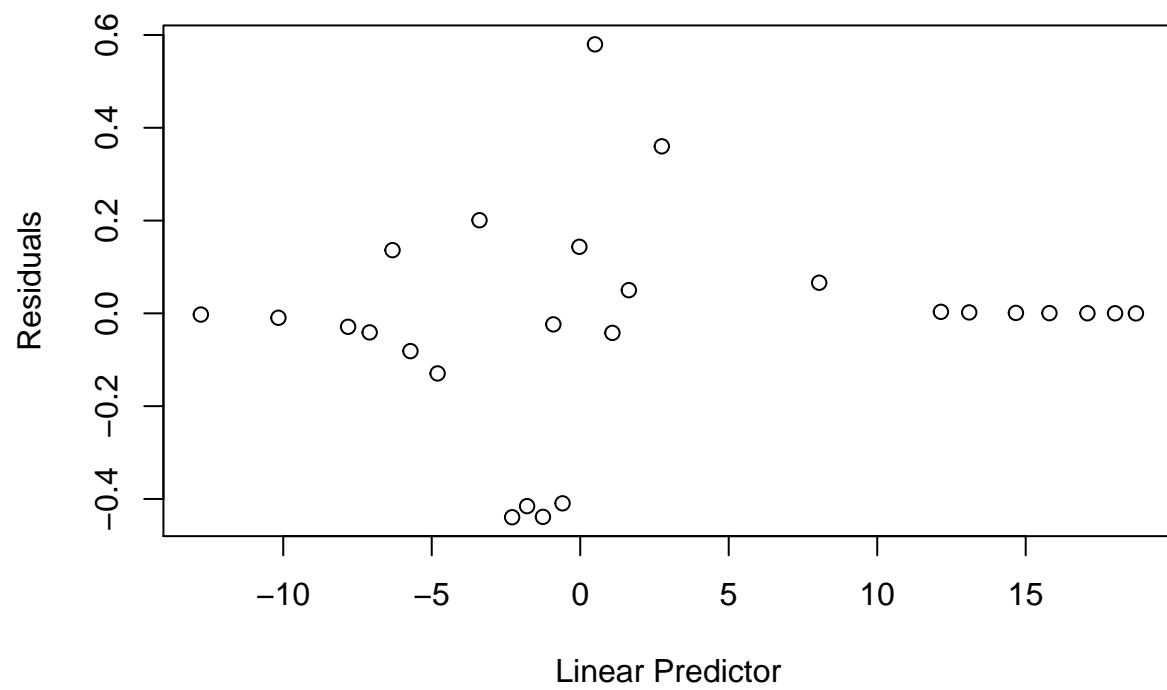


The AUC value of 0.97, tells us this model predicted values are accurate.

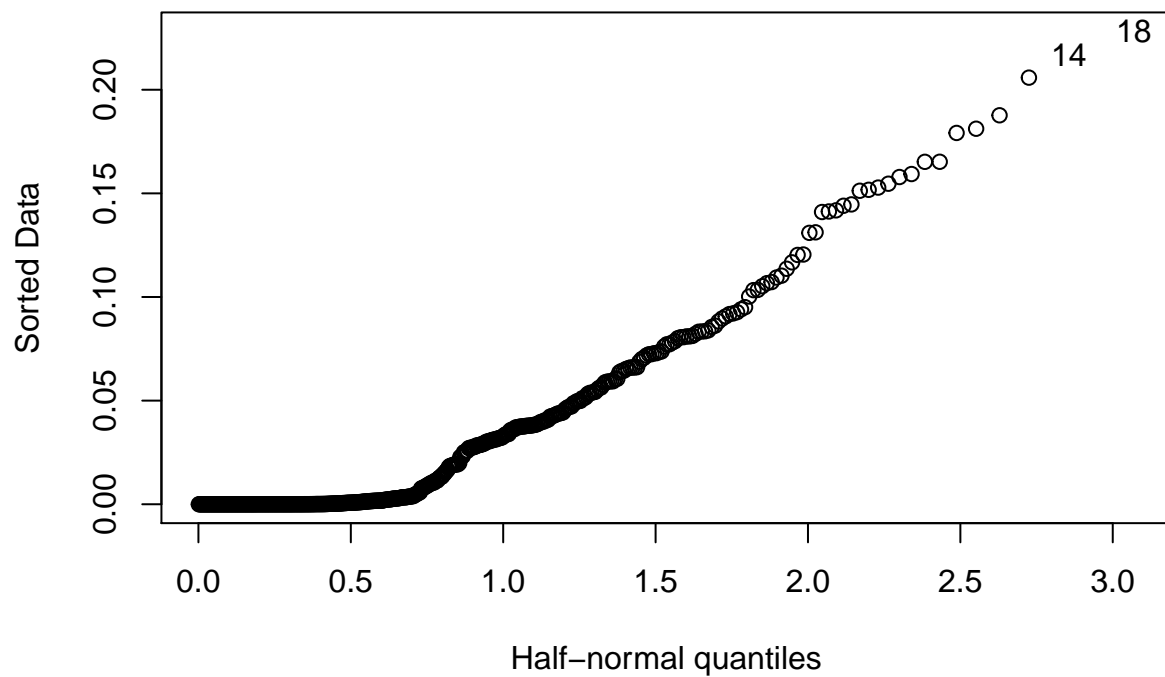
Confusion Matrix

```
##
## targetthat  0  1
##           0 220 22
##           1  17 207
```

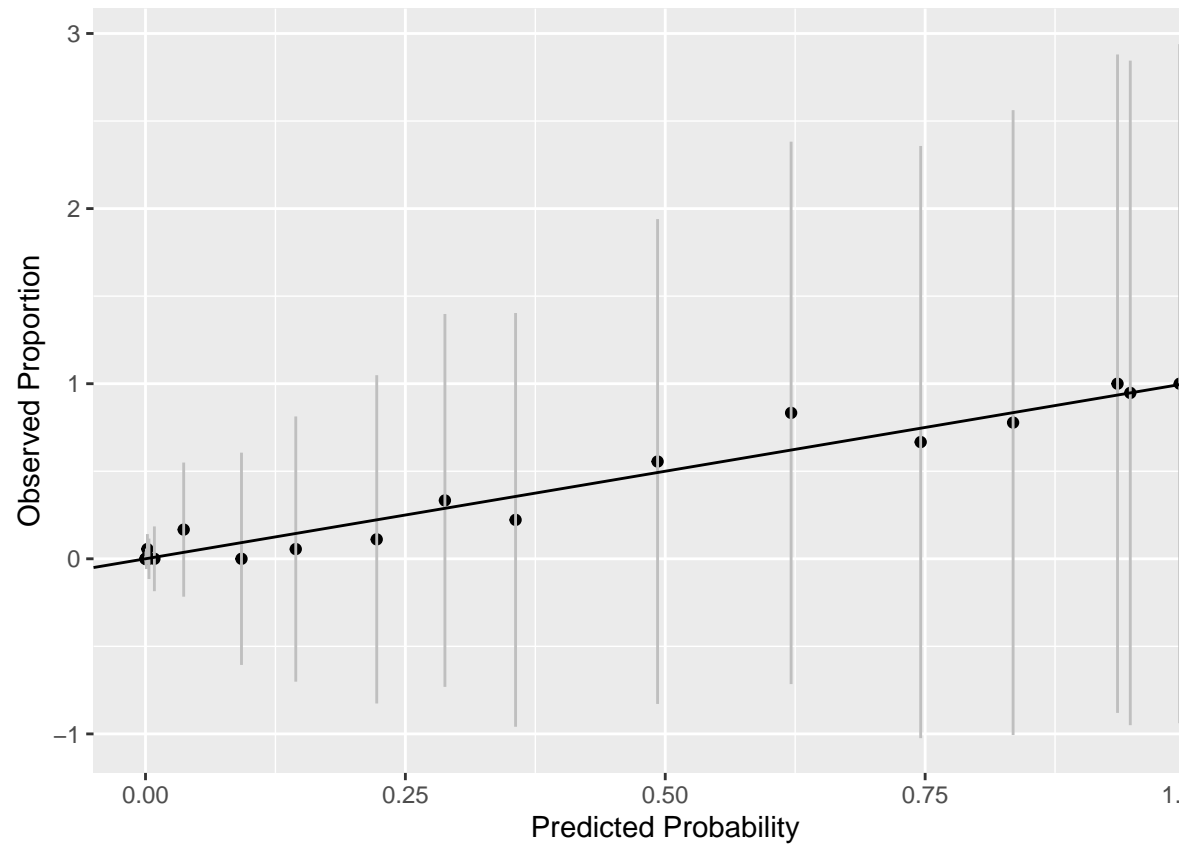
Create a binned diagnostic plot of residuals vs prediction There are definite patterns here, which bear investigating.



Plot leverages.



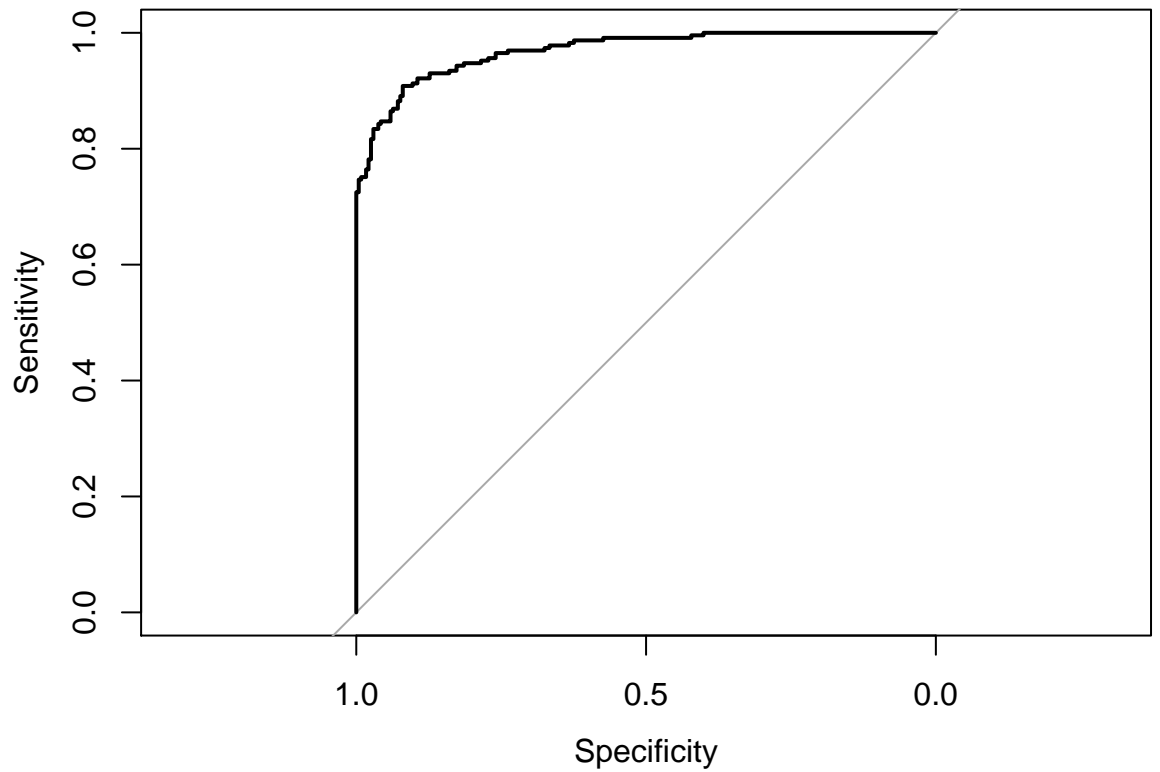
We don't see any strong outliers with the leverage plot. The points identified (14,18) are essentially in the plot of the line formed, so they are not likely pulling our model in any direction.



Plot Goodness of fit

We see that our predictors fall close to the line. (Note to group, need do adjust the min max line)

####Reduced general model



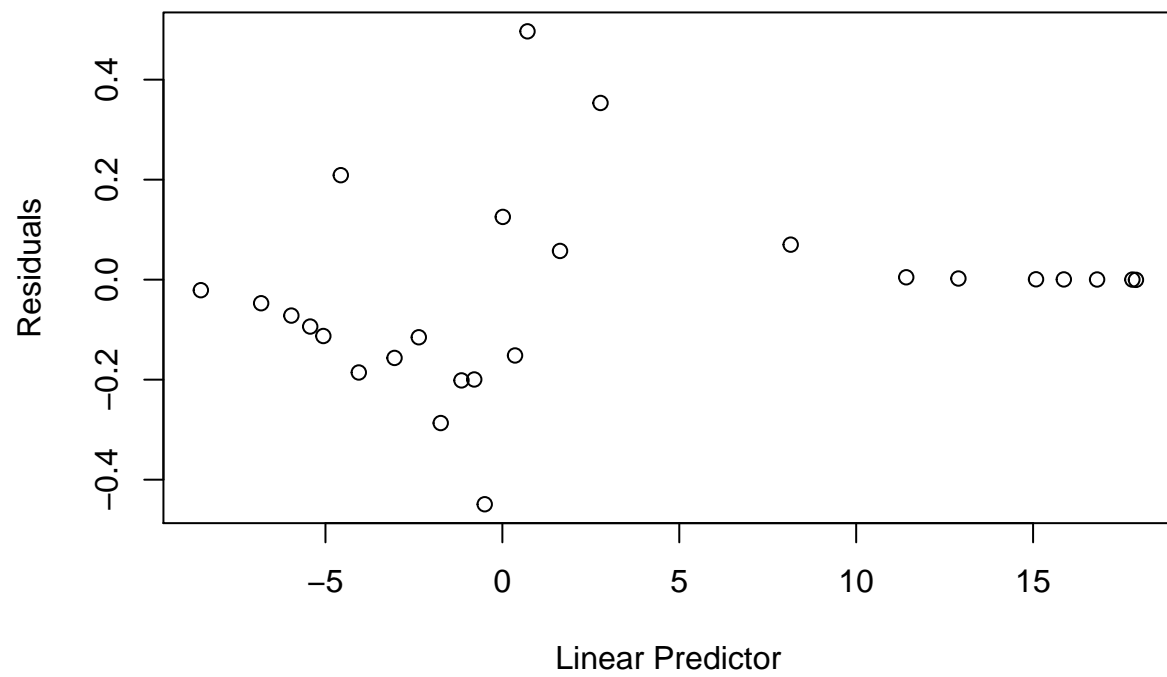
ROC Curve

This model also show a high AUC value of 0.97. This tells us predicted values are accurate, although slightly lower.

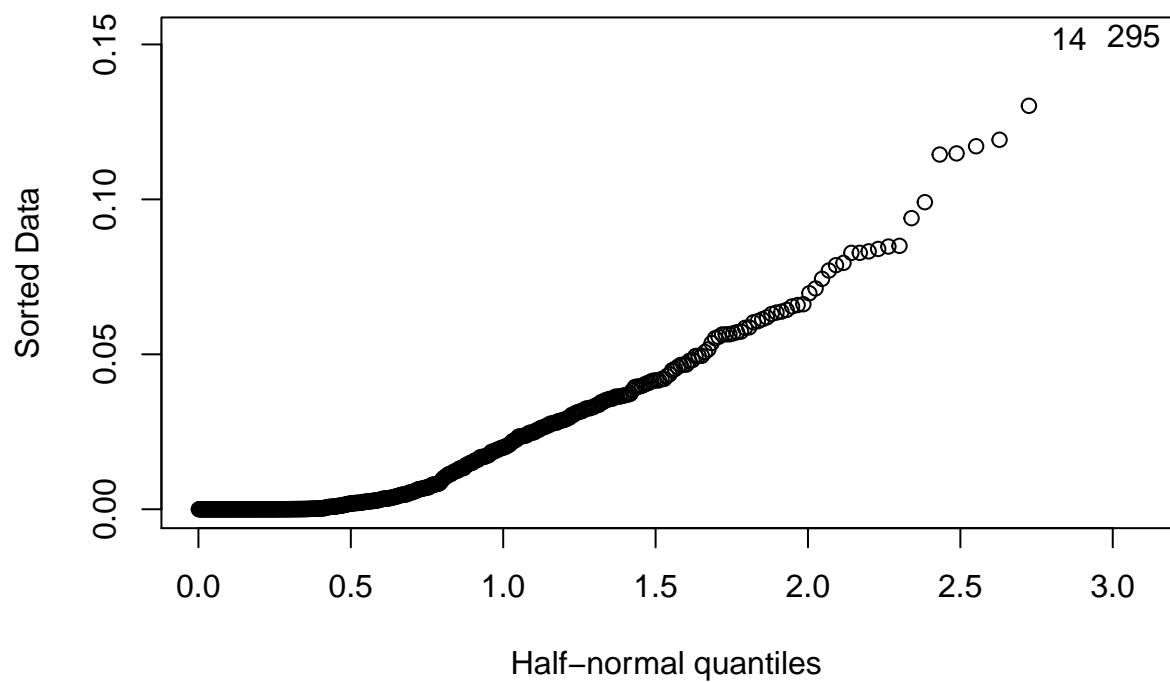
Confusion Matrix

```
##
## targetthat  0   1
##           0 218  22
##           1  19 207
```

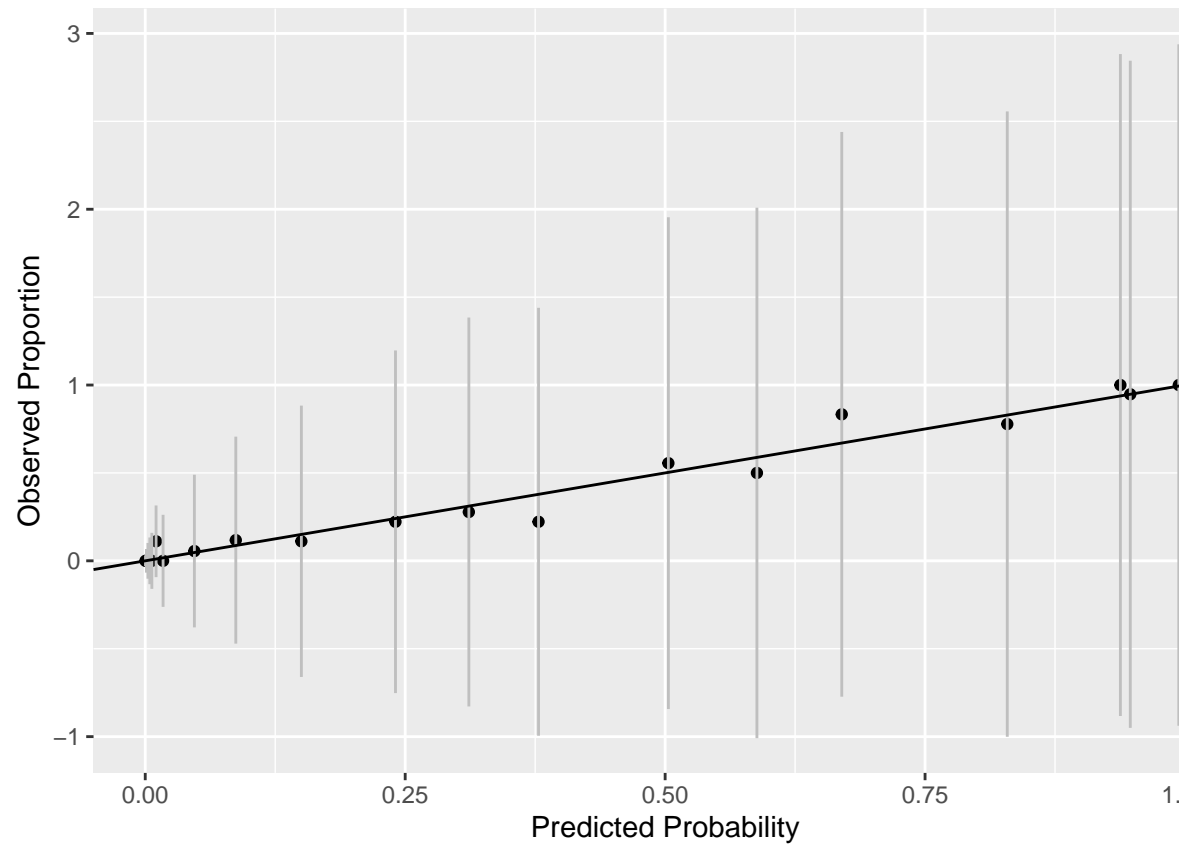
Create a binned diagnostic plot of residuals vs prediction There are definite patterns here, which bear investigating.



Plot leverages.



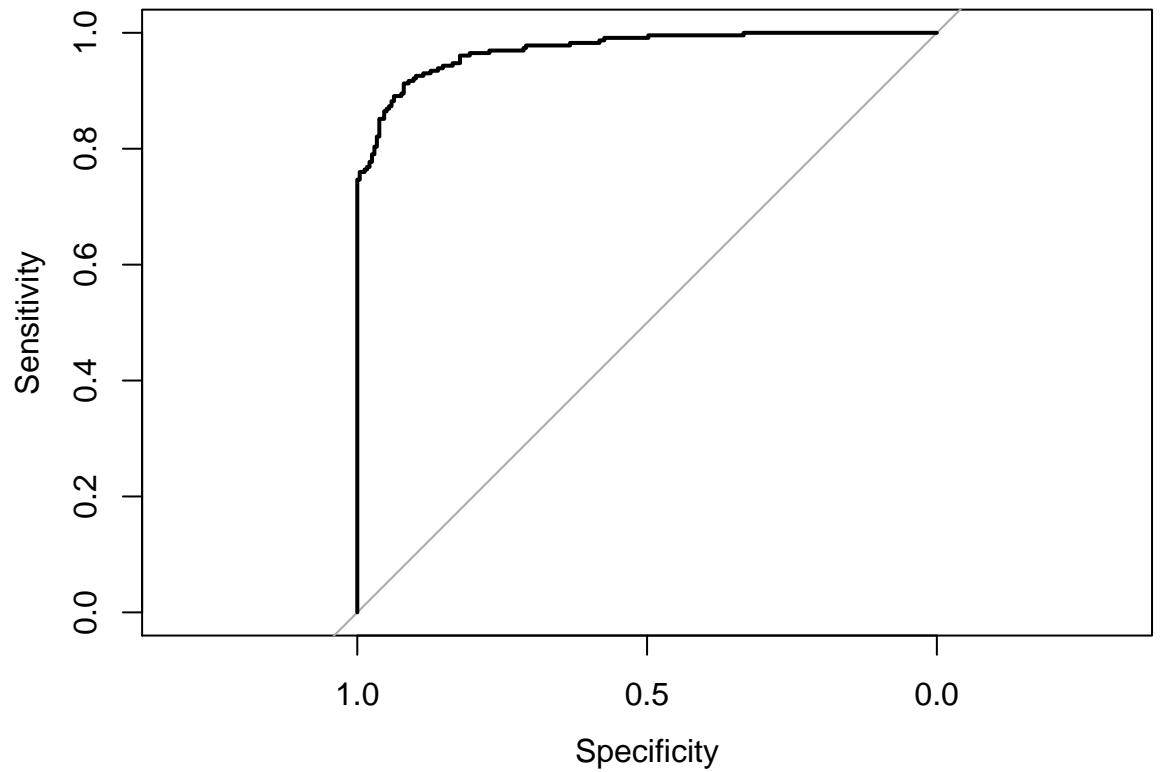
We don't see any strong outliers with the leverage plot. The points identified (14,18) are essentially in the plot of the line formed, so they are not likely pulling our model in any direction.



Plot Goodness of fit

We see that our predictors fall close to the line. (Note to group, need do adjust the min max line)

###Model 2 - AIC Model



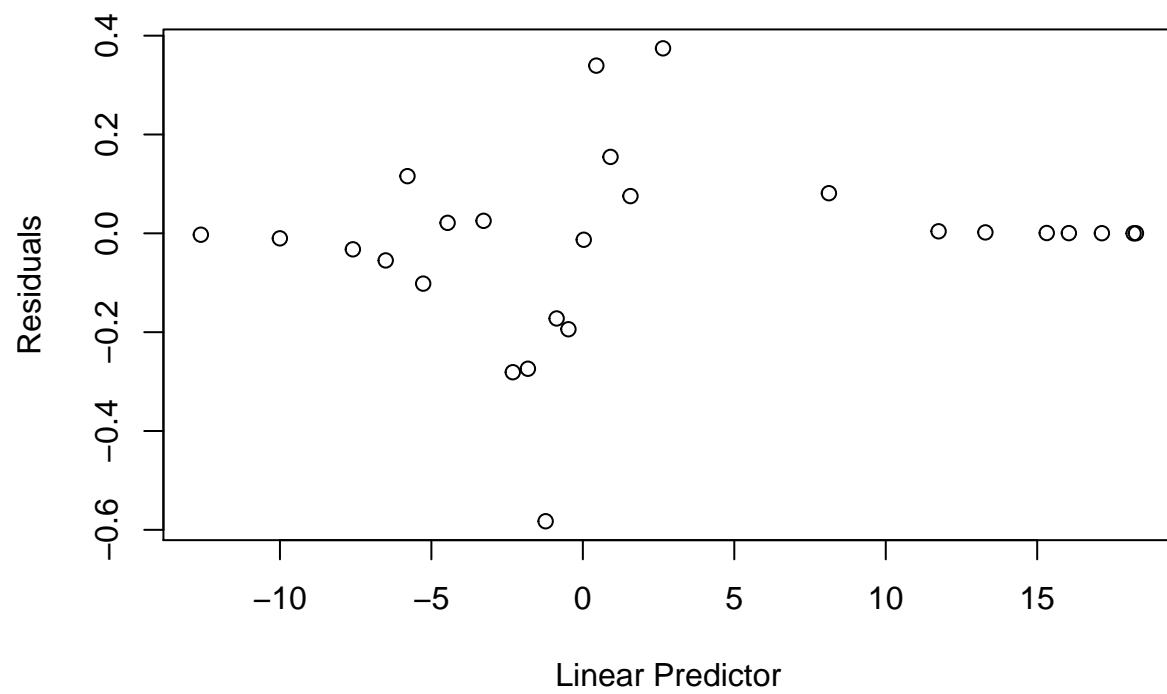
ROC Curve

The AUC value of 0.97, tells us this model predicted values are accurate.

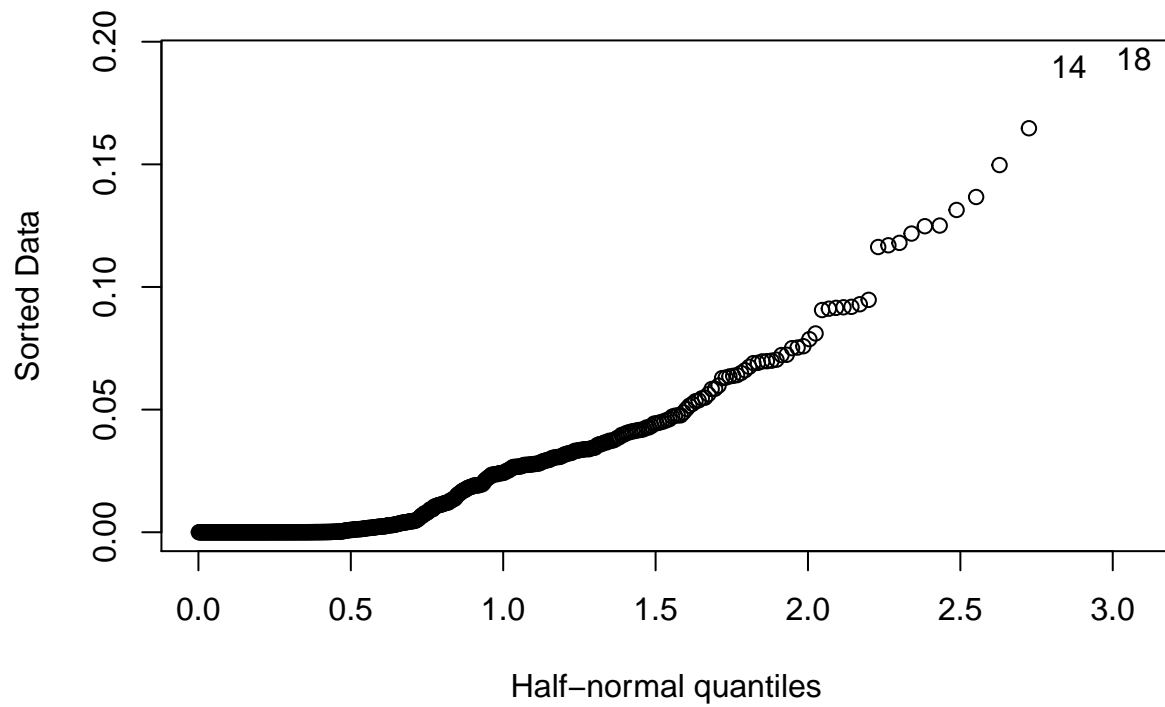
Confusion Matrix

```
##
## targethat    0    1
##           0 218  22
##           1  19 207
```

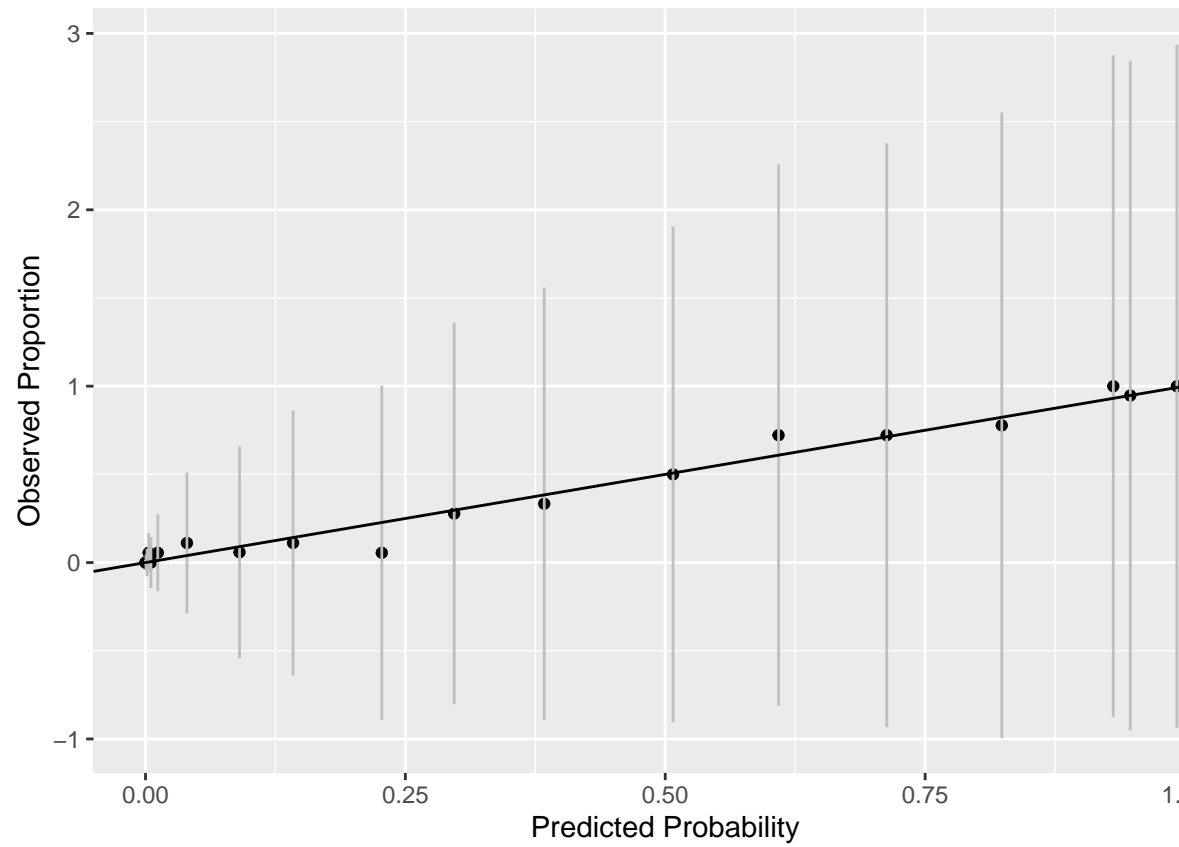
Create a binned diagnostic plot of residuals vs prediction



Plot leverages.

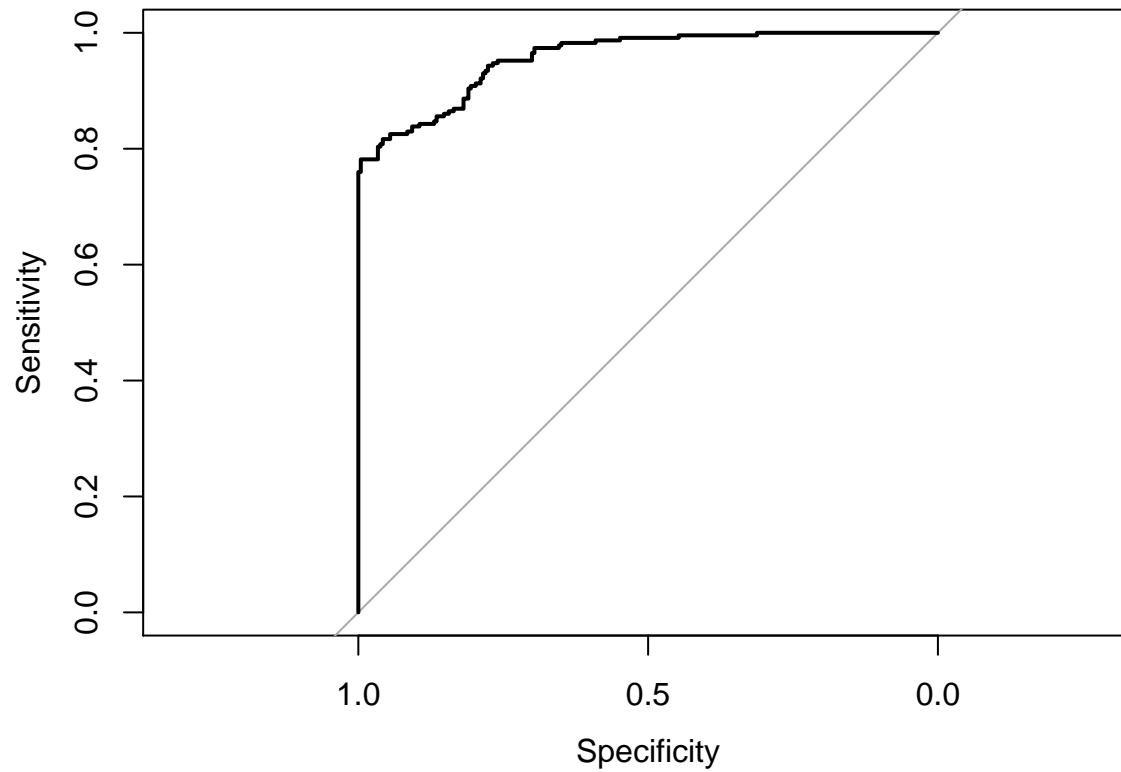


We don't see any strong outliers with the leverage plot. The points identified (14,18) are essentially in the plot of the line formed, so they are not likely pulling our model in any direction.



Plot Goodness of fit

We see that our predictors fall close to the line. (Note to group, need do adjust the min max line)



###Model 3 - BIC Model

##

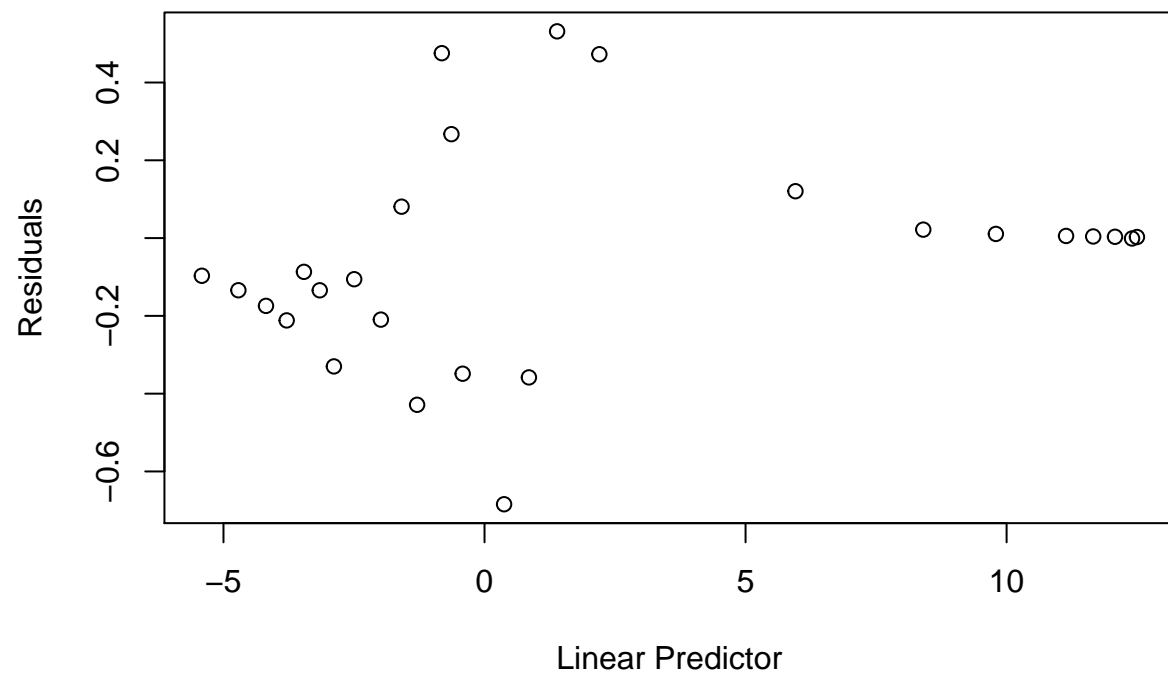
targetthat 0 1

0 214 37

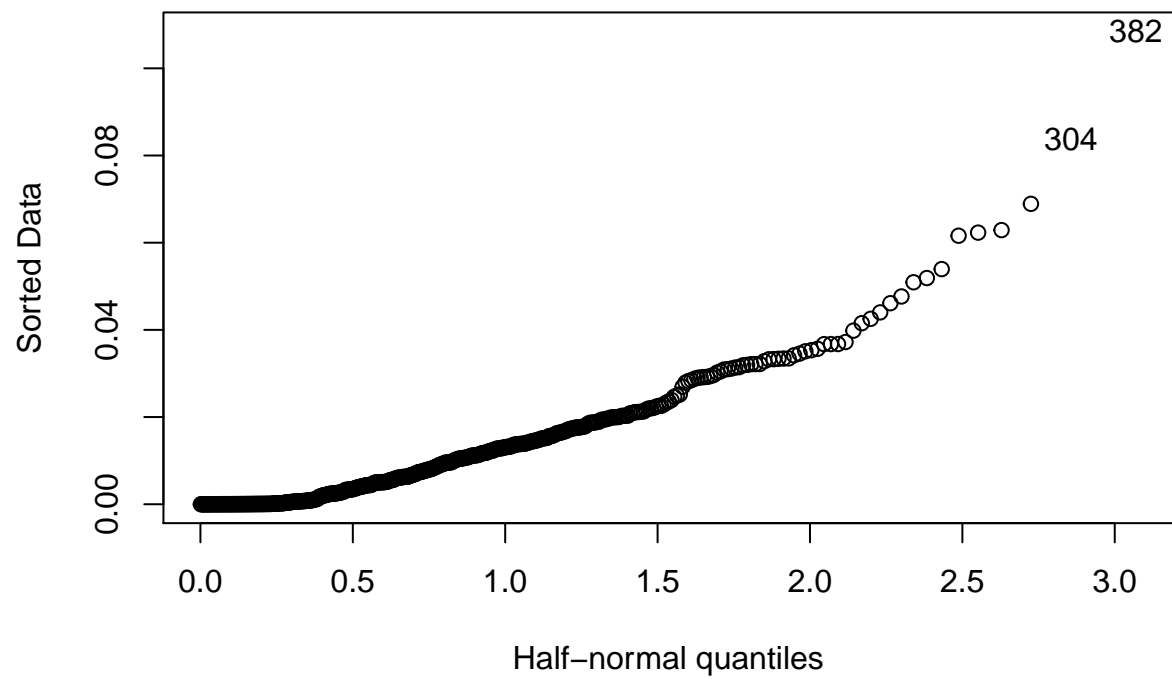
1 23 192

The AUC value of 0.96, although high for this model it has the lowest AUC score.

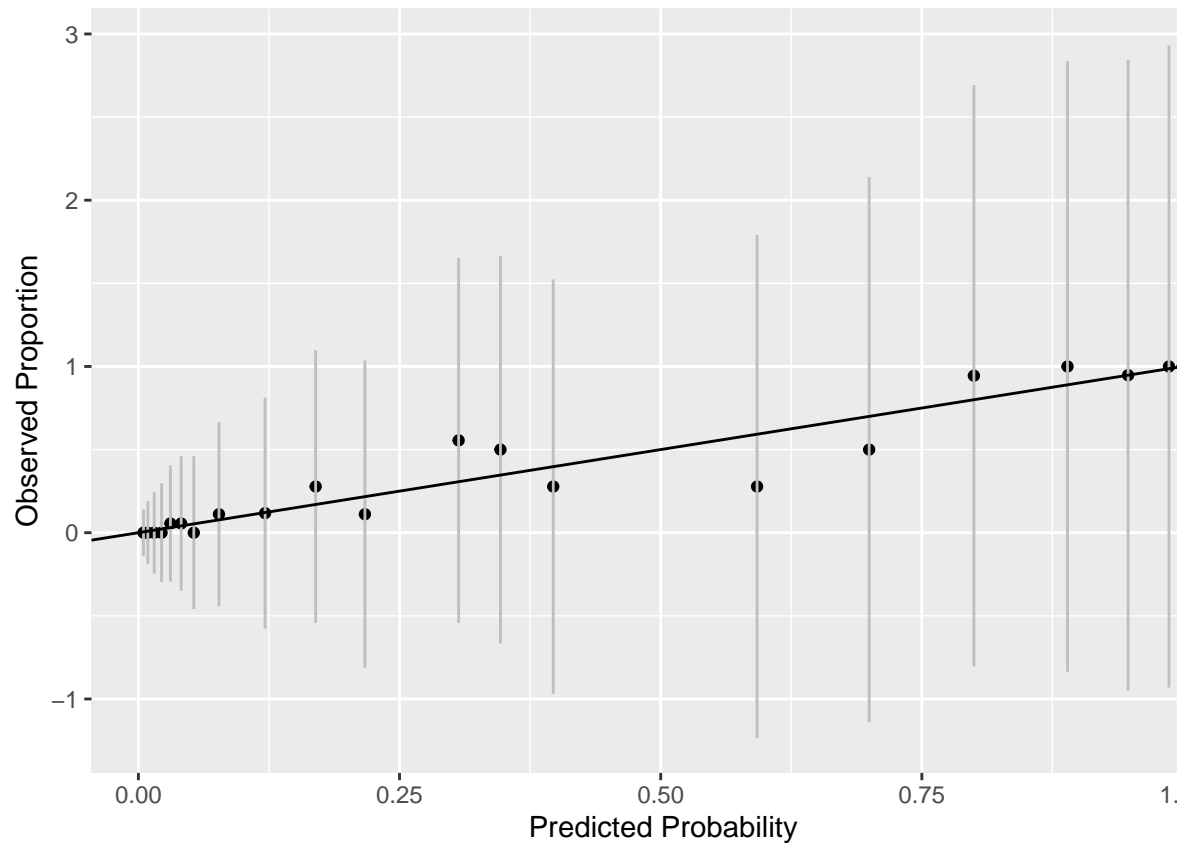
Create a binned diagnostic plot of residuals vs prediction



Plot leverages.



We don't see any strong outliers with the leverage plot. The points identified (14,18) are essentially in the plot of the line formed, so they are not likely pulling our model in any direction.



Plot Goodness of fit

We see that our predictors fall close to the line.

Pick the best regression model

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

Metric	Model 1	Model 2	Model 3	Model 4
AIC	218.0469179	215.3228528	242.7968243	209.8265226
BIC	271.9213312	252.6205235	263.5177525	247.1241933

From the above we see that Model 4, found by using the step forward selection method to do stepwise reduction of models achieves both the lowest AIC and the lowest BIC. Considering that it returns the best by both of those measures, this is the model we will use against future data (e.g., an evaluation dataset.)

Conclusion

APPENDIX

Code used in analysis

```
library(ggplot2) library(tidyr) library(MASS) library(psych) library(kableExtra) library(dplyr) li-
brary(faraway) library(gridExtra) library(reshape2) library(leaps) library(pROC) library(caret) library(naniar)
library(pander) library(pROC) crimeTrain <- read.csv("crime-training-data_modified.csv") crimeEval <-
read.csv("crime-evaluation-data_modified.csv")
```

OVERVIEW

In this homework assignment, we 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).

Objective:

The 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.

DATA EXPLORATION

Data Summary

```
crimed1 <- describe(crimeTrain, na.rm = F) crimed1$na_count <- sapply(crimeTrain, function(y) sum(length(which(is.na(y)
<- sapply(crimeTrain, function(x) round(sum(is.na(x))/nrow(crimeTrain)*100,1))

colsTrain<-ncol(crimeTrain) colsEval<-ncol(crimeEval) missingCol<-colnames(crimeTrain)[!(colnames(crimeTrain)
%in% colnames(crimeEval))]
```

The dataset consists of two data files: training and evaluation. The training dataset contains 13 columns, while the evaluation dataset contains 12. The evaluation dataset is missing column target which represent our response variable and defines whether the crime rate is above the median crime rate (1) or not (0). We will start by exploring the training data set since it will be the one used to generate the regression model.

```
text<-"a test" if(all(apply(crimeTrain,2,function(x) is.numeric(x)))==TRUE) { text<-"all data is numeric"
} else { text<-"not all data is numeric" } maxMeanMedianDiff<-round(max(abs(sapply(crimeTrain, median, na.rm = T) - sapply(crimeTrain, mean, na.rm = T))*100/(sapply(crimeTrain, max, na.rm = T)-sapply(crimeTrain, min, na.rm = T))),2)
```

First we see that all data is numeric. The dataset does contain one dummy variable to identify if the property borders the Charles River (1) or not (0).

```
nas<-as.data.frame(sapply(crimeTrain, function(x) sum(is.na(x)))) nasp<-as.data.frame(sapply(crimeTrain, function(x) round(sum(is.na(x))/nrow(crimeTrain) 100,1))) colnames(nas)<-c("name") maxna<-max(nas) maxnaname<-rownames(nas)[nas$name==maxna] percent<-round(maxna/nrow(crimeTrain)100,1)
```

An important aspect of any dataset is to determine how much, if any, data is missing. We look at all the variables to see which if any have missing data. We look at the basic descriptive statistics as well as the missing data and their percentages:

```
kable(crime1, "html", escape = F) %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed"), full_width = T) %>% column_spec(1, bold = T) %>% scroll_box(width = "100%", height = "500px") sapply(crimeTrain, function(x) round(sum(is.na(x))/nrow(crimeTrain)*100,1)) vis_miss(crimeTrain) head(crimeTrain)
```

Missing and Invalid Data

No missing data was found in the dataset.

With missing data assessed, we can look into the data in more detail. To visualize this we plot histograms for each data. Several predictors like dist, chas, rad, zn and tax are not normally distributed and noticeable outliers.

```
attach(crimeTrain[,-1]) ggplot(gather(crimeTrain[,-1]), aes(value)) + geom_histogram(bins = 20) + facet_wrap(~key, scales = "free_x") stripchart(data.frame(scale(crimeTrain)), method = "jitter", las=2, vertical=TRUE)
```

Mathematical transformations.

Box Cox The Box Cox transformation tries to transform non-normal data into a normal distribution. This transformation attempts to estimate the λ for Y. With the exception of tax, all predictors have either no transformation estimate or were given a fudge value of 0.

```
crimeTrain_bct <- apply(crimeTrain, 2, BoxCoxTrans) crimeTrain_bct
```

Variable Creation / Removal

To determine how we can combine variables to create new one we start by looking at a correlation plot. The plot and cor function lists nox, age, rad, tax and indus as the strongest positively correlated predictors, while rad and distance are the strongest negatively correlated predictors. `cor(crimeTrain$target, crimeTrain[-c(1)], use="na.or.complete")`

```
corrplot::corrplot(cor(crimeTrain[,1:13]), order = "hclust", method='square', addrect = 2, tl.col = "black", tl.cex = .75, na.label = " ")
```

BUILD MODELS

General regression

We start by building a model with all the predictors in the dataset.

```
m1<-glm(target~.,data=crimeTrain,family="binomial"(link="logit")) summary(m1)
```

The Summary of this model shows several predictor are not relevant. We build a second model without these predictors.

```
m1.1<-glm(target~nox+age+dis+rad+tax+ptratio+medv,data=crimeTrain,family="binomial"(link="logit")) summary(m1.1)
```



```
1-pchisq(m1.1deviance,m1.1df.residual) 1-pchisq(m1deviance,m1df.residual)
```

The new model has a slightly higher AIC which would tell us the first model is slightly less complex. For the 2 data sets p-value = 1 - pchisq(deviance, degrees of freedom) are 1. The Null hypothesis is still supported.

AIC Step Method Another way of selecting which predictors to use in the model is by calculating the AIC of the model. This metric is similar to the adjusted R-square of a model in that it penalizes models with more predictors over simpler model with few predictors. We use Stepwise function in R to find the lowest AIC with different predictors.

```
m2 <- step(m1) summary(m2)
```

This reduces the predictors used in the model to these: zn nox age dis rad tax ptratio medv

It Removes these predictors: indus chas rm#

The AIC improves marginally from 218.05 (our original general model) to 215.32, but we also benefit by having a simpler model less prone to overfitting.

Also, the predictors in the model now are all significant (under 0.05 p level) and all but one under .01 or very significant. Which is much improved over the prior model

BIC Method

To determine the number of predictors and which predictors to be used we will use the Bayesian Information Criterion (BIC).

```
regfit.full <- regsubsets(factor(target) ~ ., data=crimeTrain) par(mfrow = c(1,2)) reg.summary <-  
summary(regfit.full) plot(reg.summary$bic,xlab = "NumberofPredictors",ylab = "BIC",type =  
"l",main = "SubsetSelectionUsingBIC")BIC_num <- which.min(reg.summary$bic) points(BIC_num,  
reg.summary$bic[BIC_num], col="red", cex=2, pch=20) plot(regfit.full, scale="bic", main="Predictors  
vs. BIC") par(mfrow = c(1,1))
```

The plot on the right shows that the number of predictors with the lowest BIC are **nox** , **age**, **rad**, and **medv**. We will use those predictors to build the next model

```
m3 <- glm(target ~ nox + age + rad + medv, family=binomial, data = crimeTrain) crimeTrainpredicted_m3 <-  
predict(m3, crimeTrain, type = 'response')crimeTraintarget_m3target <- ifelse(crimeTrainpredicted_m3>0.5,  
1, 0) pander::pander(summary(m3))
```

Forward Selection Method using some BoxCox transformed independent variables:

```
{r} m4 <- step(glm(target~1, data=crimeTrain), direction = "forward", scope = ~zn + I(log(indus)) +  
I(sqrt(chas)) + I(nox^-1) + I(log(rm)) + I(age^2) + I(dis^-.5) + rad + I(tax^-1) + I(ptratio^2) + lstat +  
medv) summary(m4)
```

SELECT MODELS Compare Model Statistics

Model 1 - General Model

Complete general model

ROC Curve

The ROC Curve helps measure true positives and true negative. A high AUC or area under the curve tells us the model is predicting well.

```
targetthat<-predict(m1,type="response") g<-roc(target~targetthat,data=crimeTrain) plot(g)
```

The AUC value of 0.96, tells us this model predicted values are accurate.

Confusion Matrix

```
targetthat[targetthat<0.5]<-0 targetthat[targetthat>=0.5]<-1 table(targetthat,crimeTrain$target)
```

Create a binned diagnostic plot of residuals vs prediction There are definite patterns here, which bear investigating.

```
crimeMut <- mutate(crimeTrain, Residuals = residuals(m1), linPred = predict(m1)) grpCrime <-
group_by(crimeMut, cut(linPred, breaks=unique(quantile(linPred, (0:25/26))))) diagCrime <- sum-
marise(grpCrime, Residuals = mean(Residuals), linPred = mean(linPred)) plot(Residuals ~ linPred, data =
diagCrime, xlab="Linear Predictor")
```

Plot leverages.

```
halfnorm(hatvalues(m1))
```

We don't see any strong outliers with the leverage plot. The points identified (14,18) are essentially in the plot of the line formed, so they are not likely pulling our model in any direction.

Plot Goodness of fit

```
linPred <- predict(m1) crimeMut <- mutate(crimeTrain, predProb = predict(m1, type = "response"))
grpCrime <- group_by(crimeMut, cut(linPred, breaks = unique(quantile(linPred, (0:25)/26))))
hlDf <- summarise(grpCrime, y= sum(target), pPred=mean(predProb), count = n()) hlDf <- mu-
tate(hlDf, se.fit=sqrt(pPred * (1-(pPred)/count))) ggplot(hlDf,aes(x=pPred,y=y/count,ymin=y/count-
2se.fit,ymax=y/count+2se.fit)) + geom_point()+geom_linerange(color=grey(0.75))+geom_abline(intercept=0,slope=1)
+ xlab("Predicted Probability") + ylab("Observed Proportion")
```

We see that our predictors fall close to the line. (Note to group, need do adjust the min max line)

Reduced general model

ROC Curve

```
targetthat<-predict(m1.1,type="response") g<-roc(target~targetthat,data=crimeTrain) plot(g)
```

This model also show a high AUC value of 0.96. This tells us predicted values are accurate, although slightly lower.

Confusion Matrix

```
targetthat[targetthat<0.5]<-0 targetthat[targetthat>=0.5]<-1 table(targetthat,crimeTrain$target)
```

Create a binned diagnostic plot of residuals vs prediction There are definite patterns here, which bear investigating.

```
crimeMut <- mutate(crimeTrain, Residuals = residuals(m1.1), linPred = predict(m1.1)) grpCrime <-
group_by(crimeMut, cut(linPred, breaks=unique(quantile(linPred, (0:25/26))))) diagCrime <- sum-
marise(grpCrime, Residuals = mean(Residuals), linPred = mean(linPred)) plot(Residuals ~ linPred, data =
diagCrime, xlab="Linear Predictor")
```

Plot leverages.

```
halfnorm(hatvalues(m1.1))
```

We don't see any strong outliers with the leverage plot. The points identified (14,18) are essentially in the plot of the line formed, so they are not likely pulling our model in any direction.

Plot Goodness of fit

```
linPred <- predict(m1.1) crimeMut <- mutate(crimeTrain, predProb = predict(m1.1, type = "response"))
grpCrime <- group_by(crimeMut, cut(linPred, breaks = unique(quantile(linPred, (0:25)/26))))
hlDf <- summarise(grpCrime, y= sum(target), pPred=mean(predProb), count = n()) hlDf <- mu-
tate(hlDf, se.fit=sqrt(pPred * (1-(pPred)/count))) ggplot(hlDf,aes(x=pPred,y=y/count,ymin=y/count-
2se.fit,ymax=y/count+2se.fit)) + geom_point()+geom_linerange(color=grey(0.75))+geom_abline(intercept=0,slope=1)
+ xlab("Predicted Probability") + ylab("Observed Proportion")
```

We see that our predictors fall close to the line. (Note to group, need do adjust the min max line)

Model 2 - AIC Model

ROC Curve

```
targetthat<-predict(m2,type="response") g<-roc(target~targetthat,data=crimeTrain) plot(g)
```

The AUC value of 0.96, tells us this model predicted values are accurate.

Confusion Matrix

```
targetthat[targetthat<0.5]<-0 targetthat[targetthat>=0.5]<-1 table(targetthat,crimeTrain$target)
```

Create a binned diagnostic plot of residuals vs prediction

```
crimeMut <- mutate(crimeTrain, Residuals = residuals(m2), linPred = predict(m2)) grpCrime <-  
group_by(crimeMut, cut(linPred, breaks=unique(quantile(linPred, (0:25/26))))) diagCrime <- sum-  
marise(grpCrime, Residuals = mean(Residuals), linPred = mean(linPred)) plot(Residuals ~ linPred, data =  
diagCrime, xlab="Linear Predictor")
```

Plot leverages.

```
halfnorm(hatvalues(m2))
```

We don't see any strong outliers with the leverage plot. The points identified (14,18) are essentially in the plot of the line formed, so they are not likely pulling our model in any direction.

Plot Goodness of fit

```
linPred <- predict(m2) crimeMut <- mutate(crimeTrain, predProb = predict(m2, type = "response"))  
grpCrime <- group_by(crimeMut, cut(linPred, breaks = unique(quantile(linPred, (0:25)/26))))  
hlDf <- summarise(grpCrime, y= sum(target), pPred=mean(predProb), count = n()) hlDf <- mu-  
tate(hlDf, se.fit=sqrt(pPred * (1-(pPred)/count))) ggplot(hlDf,aes(x=pPred,y=y/count,ymin=y/count-  
2se.fit,ymax=y/count+2se.fit)) + geom_point()+geom_linerange(color=grey(0.75))+geom_abline(intercept=0,slope=1)  
+ xlab("Predicted Probability") + ylab("Observed Proportion")
```

We see that our predictors fall close to the line. (Note to group, need do adjust the min max line)

Model 3 - BIC Model

```
targetthat<-predict(m3,type="response") g<-roc(target~targetthat,data=crimeTrain) plot(g) targetthat[targetthat<0.5]<-  
0 targetthat[targetthat>=0.5]<-1 table(targetthat,crimeTrain$target)
```

The AUC value of 0.96, although high for this model it has the lowest AUC score.

Create a binned diagnostic plot of residuals vs prediction

```
crimeMut <- mutate(crimeTrain, Residuals = residuals(m3), linPred = predict(m3)) grpCrime <-  
group_by(crimeMut, cut(linPred, breaks=unique(quantile(linPred, (0:25/26))))) diagCrime <- sum-  
marise(grpCrime, Residuals = mean(Residuals), linPred = mean(linPred)) plot(Residuals ~ linPred, data =  
diagCrime, xlab="Linear Predictor")
```

Plot leverages.

```
halfnorm(hatvalues(m3))
```

We don't see any strong outliers with the leverage plot. The points identified (14,18) are essentially in the plot of the line formed, so they are not likely pulling our model in any direction.

Plot Goodness of fit

```
linPred <- predict(m3) crimeMut <- mutate(crimeTrain, predProb = predict(m3, type = "response"))  
grpCrime <- group_by(crimeMut, cut(linPred, breaks = unique(quantile(linPred, (0:25)/26))))
```

```
h1Df <- summarise(grpCrime, y= sum(target), pPred=mean(predProb), count = n()) h1Df <- mu-
tate(h1Df, se.fit=sqrt(pPred * (1-(pPred)/count))) ggplot(h1Df,aes(x=pPred,y=y/count,ymin=y/count-
2se.fit,ymax=y/count+2se.fit)) + geom_point()+geom_linerange(color=grey(0.75))+geom_abline(intercept=0,slope=1)
+ xlab("Predicted Probability") + ylab("Observed Proportion")
```

We see that our predictors fall close to the line.

Pick the best regression model

```
m1AIC <- AIC(m1) m1BIC <- BIC(m1) m2AIC <- AIC(m2) m2BIC <- BIC(m2) m3AIC <- AIC(m3)
m3BIC <- BIC(m3) m4AIC <- AIC(m4) m4BIC <- BIC(m4) summary(m1)
```