DATA 621: BUSINESS ANALYTICS AND DATA MINING

HOMEWORK#3: LOGISTIC REGRESSION

03.28.2018

Yun Mai
CUNY SPS
MS in DATA SCIENCE

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).

Goals

1. Build a binary logistic regression model on the training data set to predict whether the neighborhood will be at risk for high crime levels.
2. Provide classifications and probabilities for the evaluation data set using your binary logistic regression model.

Specification

Only the variables given (or, variables that you derive from the variables provided) could be used in to modeling.

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

• zn: proportion of residential land zoned for large lots (over 25000 square feet) (predictor variable)

• indus: proportion of non-retail business acres per suburb (predictor variable)

• chas: a dummy var. for whether the suburb borders the Charles River (1) or not (0) (predictor variable)

• nox: nitrogen oxides concentration (parts per 10 million) (predictor variable)

• rm: average number of rooms per dwelling (predictor variable)

• age: proportion of owner-occupied units built prior to 1940 (predictor variable)

• dis: weighted mean of distances to five Boston employment centers (predictor variable)

• rad: index of accessibility to radial highways (predictor variable)

• tax: full-value property-tax rate per $10,000 (predictor variable)

• ptratio: pupil-teacher ratio by town (predictor variable)

• lstat: lower status of the population (percent) (predictor variable)

• medv: median value of owner-occupied homes in $1000s (predictor variable)

• target: whether the crime rate is above the median crime rate (1) or not (0) (response variable)

• A write-up submitted in PDF format. Your write-up should have four sections. Each one is described below. You may assume you are addressing me as a fellow data scientist, so do not need to shy away from technical details.

• Assigned prediction (probabilities, classifications) for the evaluation data set. Use 0.5 threshold. Include your R statistical programming code in an Appendix.

1. DATA EXPLORATION

Summary of the train data set

In table below, we can see the sample size, the range of the value , the minimum, the maximum, the mean, the median, the standard deviation of each variables, the missing data, the range of the value of each variable. The missing data here are actually 0s which are the real values for binary data set. There is no data missing as number of NA is 0 for each variable.

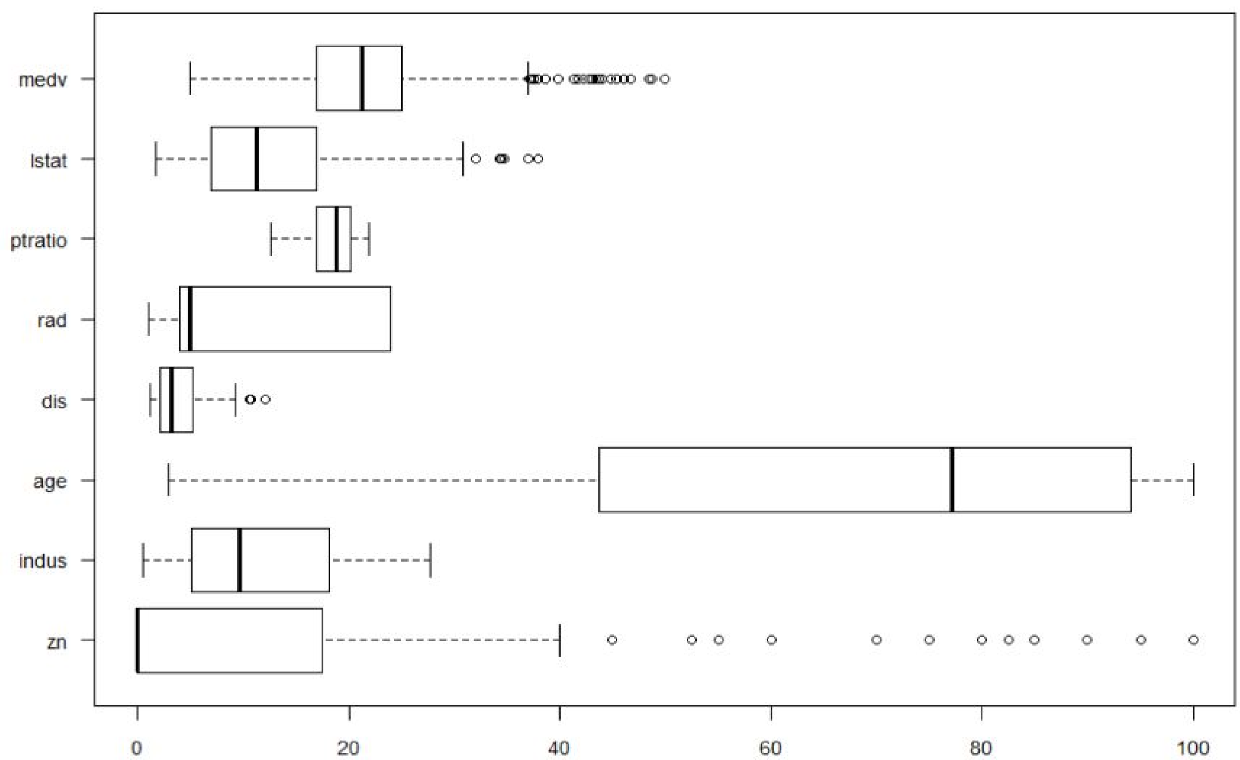
	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target
nbr.val	466.0	466.00	466.00	466.00	466.00	466.00	466.00	466.00	466.00	466.00	466.00	466.00	466.00
nbr.null	339.0	0.00	433.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	237.00
nbr.na	0.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
min	0.0	0.46	0.00	0.39	3.86	2.90	1.13	1.00	187.00	12.60	1.73	5.00	0.00
max	100.0	27.74	1.00	0.87	8.78	100.00	12.13	24.00	711.00	22.00	37.97	50.00	1.00
range	100.0	27.28	1.00	0.48	4.92	97.10	11.00	23.00	524.00	9.40	36.24	45.00	1.00
sum	5395.0	5174.94	33.00	258.31	2931.45	31859.30	1768.79	4441.00	190828.00	8573.70	5886.26	10526.60	229.00
median	0.0	9.69	0.00	0.54	6.21	77.15	3.19	5.00	334.50	18.90	11.35	21.20	0.00
mean	11.6	11.11	0.07	0.55	6.29	68.37	3.80	9.53	409.50	18.40	12.63	22.59	0.49
SE.mean	1.1	0.32	0.01	0.01	0.03	1.31	0.10	0.40	7.78	0.10	0.33	0.43	0.02
CI.mean.0.95	2.1	0.62	0.02	0.01	0.06	2.58	0.19	0.79	15.28	0.20	0.65	0.84	0.05
var	545.9	46.87	0.07	0.01	0.50	802.10	4.44	75.45	28190.44	4.83	50.44	85.37	0.25
std.dev	23.4	6.85	0.26	0.12	0.70	28.32	2.11	8.69	167.90	2.20	7.10	9.24	0.50
coef.var	2.0	0.62	3.63	0.21	0.11	0.41	0.56	0.91	0.41	0.12	0.56	0.41	1.02

In the table below we can see the first and third quartile of each variable.

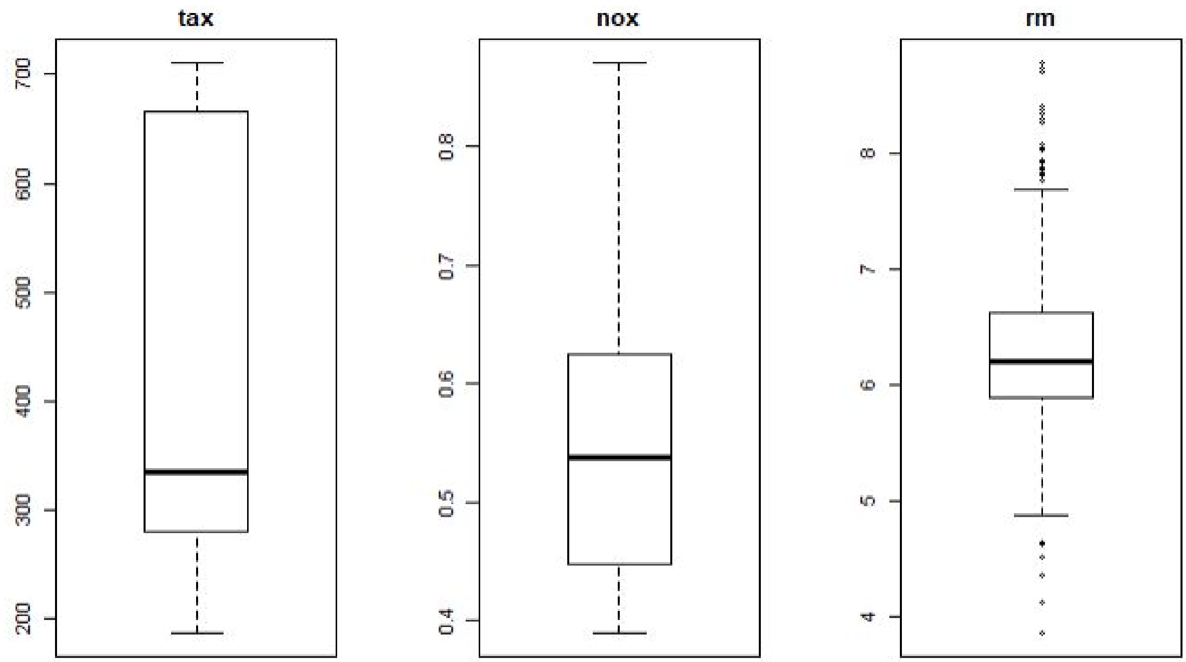
	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target
Min.	0	0.5	0.00	0.39	3.9	3	1.1	1.0	187	12.6	2	5	0.00
1st Qu.	0	5.1	0.00	0.45	5.9	44	2.1	4.0	281	16.9	7	17	0.00
Median	0	9.7	0.00	0.54	6.2	77	3.2	5.0	334	18.9	11	21	0.00
Mean	12	11.1	0.07	0.55	6.3	68	3.8	9.5	410	18.4	13	23	0.49
3rd Qu.	16	18.1	0.00	0.62	6.6	94	5.2	24.0	666	20.2	17	25	1.00
Max	100	27.7	1.00	0.87	8.8	100	12.1	24.0	711	22.0	38	50	1.00

To get an idea whether there are outliers, plot the data in boxplot. In this plot, the binary variables target and chat were not shown. To get a better view, the variable tax whose values are much larger than other variables and nox whose values are much smaller than other variables are not shown in the plot.

Outliers

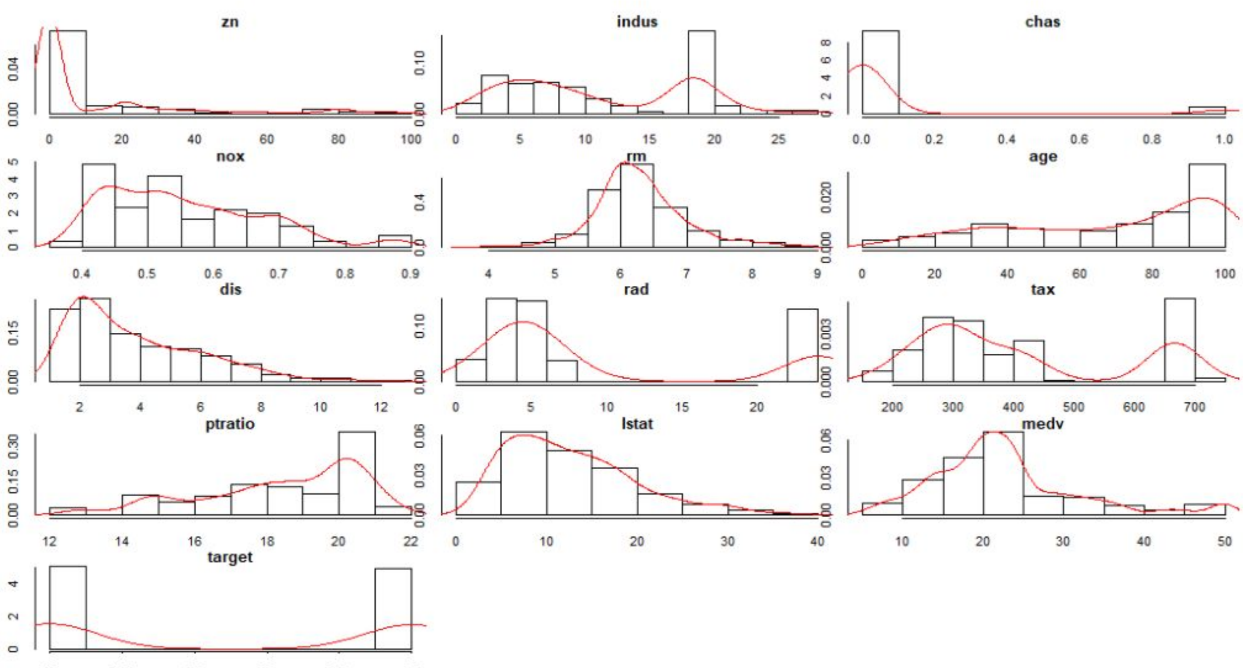


The boxplots of Tax, nox and rm are shown as follows.



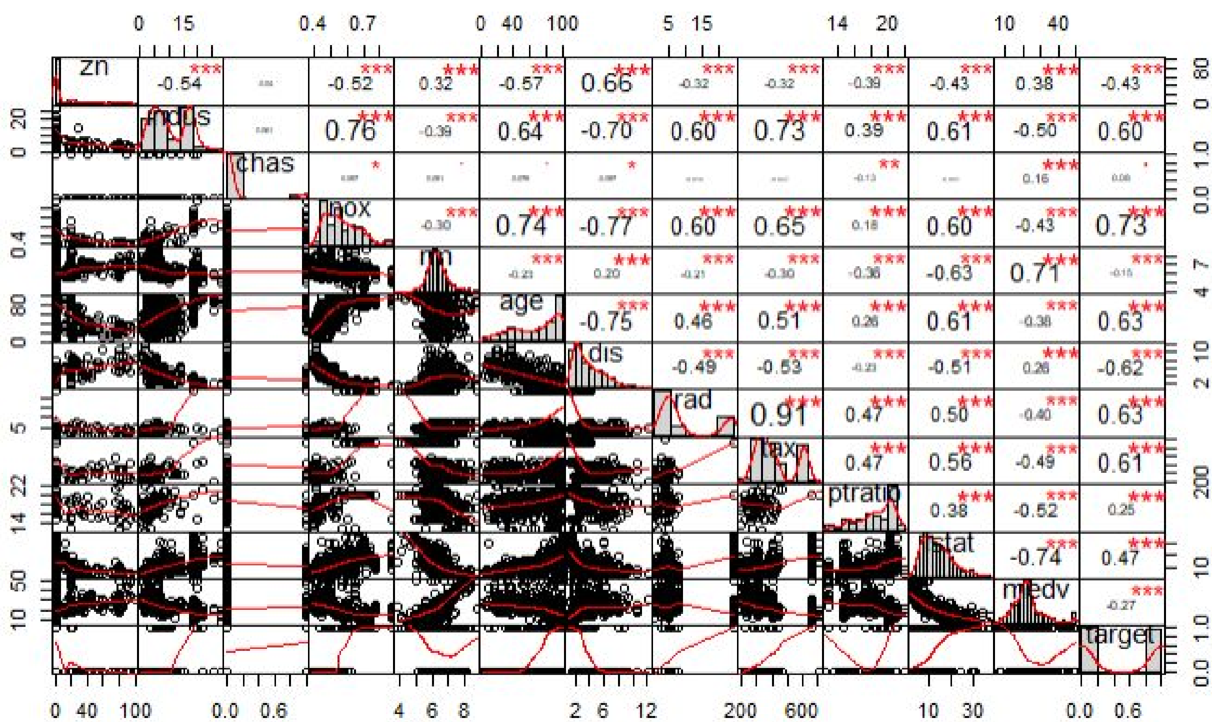
From the boxplots, we can see there are outliers in mdev, lstat, ds, zn and rm. I would tend to think these values are not real outliers and it just these variables have high variability. I will do transformation for these variable before fitting the model.

The Distribution of Variables



From the histogram, only rm follows normal distribution.

Collinearities



From the correlation matrix above shows collinearities exists among the predictors if above 0.75 is considered a strong correlation. For example, indus and nox are strongly correlated. Then the variance inflation factor (VIF) for the model were calculated. When VIF > 4, the variable will be removed from the model.

```
         GVIF
zn       2.3
indus    4.1
chas     1.1
nox      5.1
rm       2.4
age      3.2
dis      4.2
rad      7.1
tax      9.2
ptratio  2.0
lstat    3.7
medv     3.7
target   2.6
[1] Drop tax.
```

→

```
         GVIF
zn       2.2
indus    3.2
chas     1.1
nox      5.1
rm       2.4
age      3.2
dis      4.2
rad      2.5
ptratio  2.0
lstat    3.7
medv     3.7
target   2.6
[1] 5.1
[1] Drop nox.
```

→

```
         GVIF
zn       2.2
indus    2.9
chas     1.1
rm       2.4
age      3.1
dis      3.8
rad      2.4
ptratio  1.7
lstat    3.6
medv     3.4
target   2.3
[1] 3.8
```

→

```
         GVIF
zn       2.2
indus    2.9
chas     1.1
rm       2.4
age      3.1
dis      3.8
rad      2.4
ptratio  1.7
lstat    3.6
medv     3.4
target   2.3
```

So tax and nox will be removed from the predictors variable list to avoid the collinearities. The predictors now include "zn", "indus", "chas" , "rm" , "age", "dis", "rad", "ptratio", "lstat" "medv", and "target" .

Using the same cutoff, the findCorrelation function suggests to remove two more variables, indus and age, comparing to the VIF test. I will keep these two variables for now because these two variables could be important for building the model. I tend to think that the area where there are more non-retail business acres and more old houses could have more crimes.

2. DATA PREPARATION

2.1 Take care of the collinearities

2.1.1 Center the variables

Perform VIF test again after centering the variables tax and nox . There is still high collinearities indicating to delete these two variables.

```
Variance inflation factors

[1] Drop tax_ct.

Variance inflation factors

[1] 4.5
[1] Drop nox_ct.

Variance inflation factors

[1] 3.8
$VIFS

$XVars
 [1] "zn"      "indus"   "chas"    "rm"      "age"     "dis"     "rad"     "ptratio"
 [9] "lstat"   "medv"

$X
```

2.1.2 Whether transformation will help in reduce the collinearities?

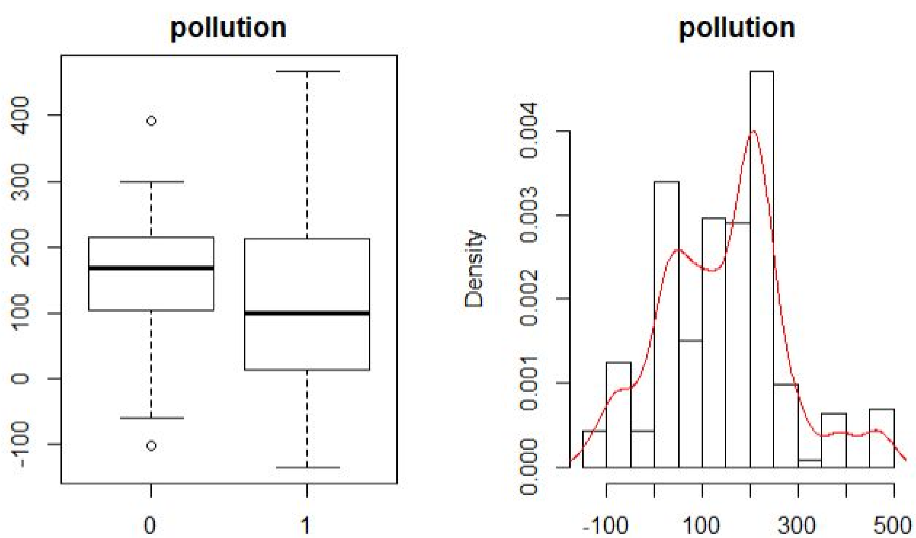
Log transformation of the variables tax and nox does not change the collinerities.

```
Variance inflation factors

[1] Drop l.tax.

Variance inflation factors

[1] 5.5
[1] Drop l.nox.

Variance inflation factors

[1] 3.8
$VIFS

$XVars
 [1] "zn"      "indus"   "chas"    "rm"      "age"     "dis"     "rad"     "ptratio"
 [9] "lstat"   "medv"

$X
```

2.1.3 Make a new variable for the operational purposes?

From the correlation matrix, we can see that tax is highly related to indus and rad and that nox is highly related to indus, age, and dis (cor above 0.74). It make sense that nitrogen oxide levels is higher in the industry area and the neighborhood with the larger amount of old houses using heating system that will generate more waste. It is reasonable that full-value property-tax rate is higher in industry area and the places near radical highways.

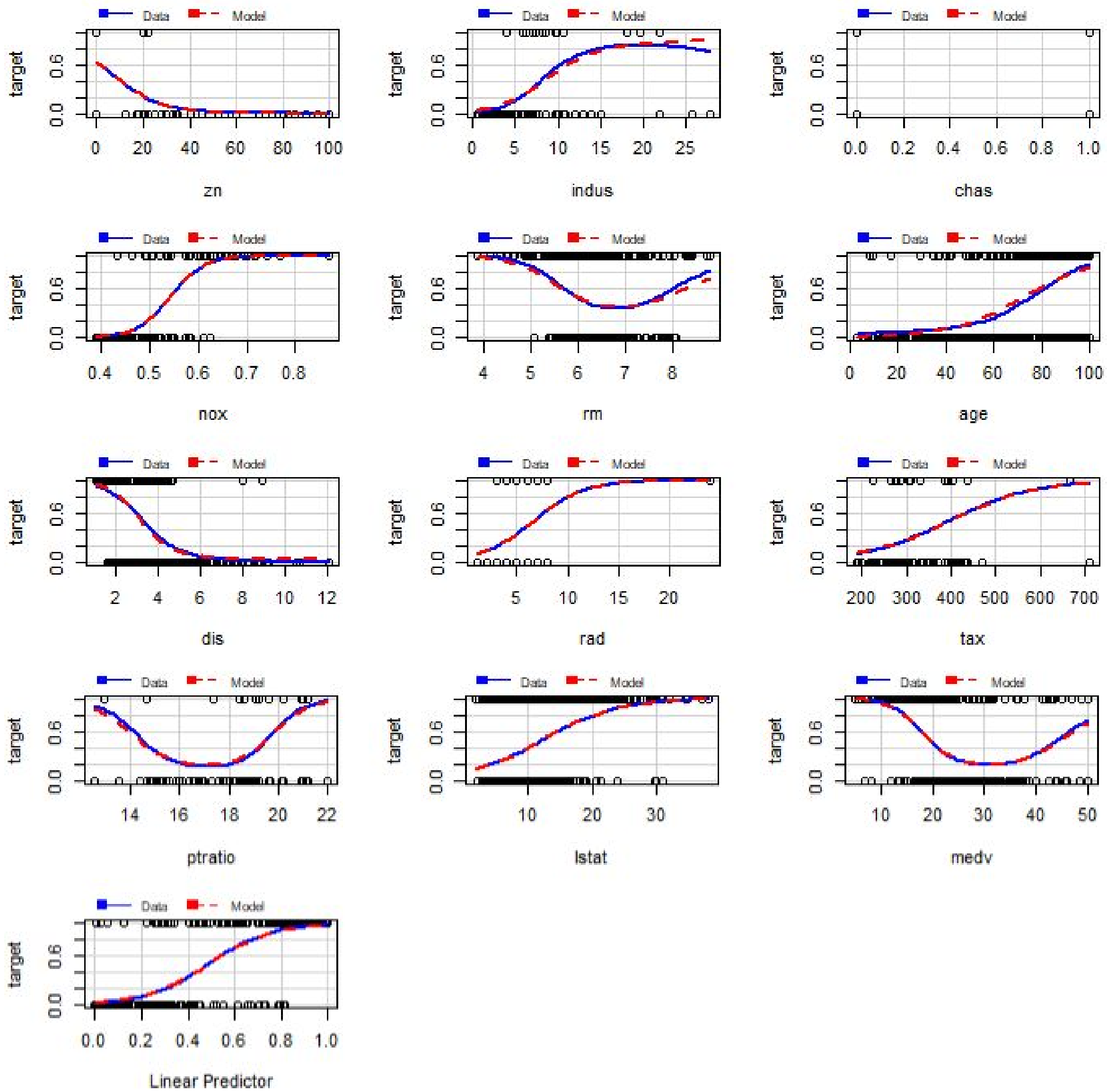
Keeping tax and nox make cause the model unstable since they show high correlations with some other variables. It is possible that they carry the redundant information. But they could also have information useful for building the model. So I will combine these two variables. Since these two variables seems affecting the crime rate in opposite directions, I will do the subtraction. A new variable 'pollution' (nox*1000 - tax) and it does not contribute to the collinearity . The distribution of the new variable is as follows.



2.2 Log or quadratic transformation for the predictor variables

2.2.1 Marginal model plots

Use the marginal model plots to check whether there is a need to add extra predictor terms. The model and the data agree to each other quite well for each predictor. There is no need to transform any predictor variable by looking at the plots.



But The boxplots for the predictor variables show some predictors are skewed. Usually, the right skewed variables (zn, dis and lstat) will need log transformation and the left skewed variables (indus, age, rad and ptratio) quadratic transformation.

Before doing the transformation, use residual plots to check which predictors need transformation.

2.2.2 Residual plots

Use the residuals to check whether there is a need to add extra predictor terms.

The original set of predictor variables:

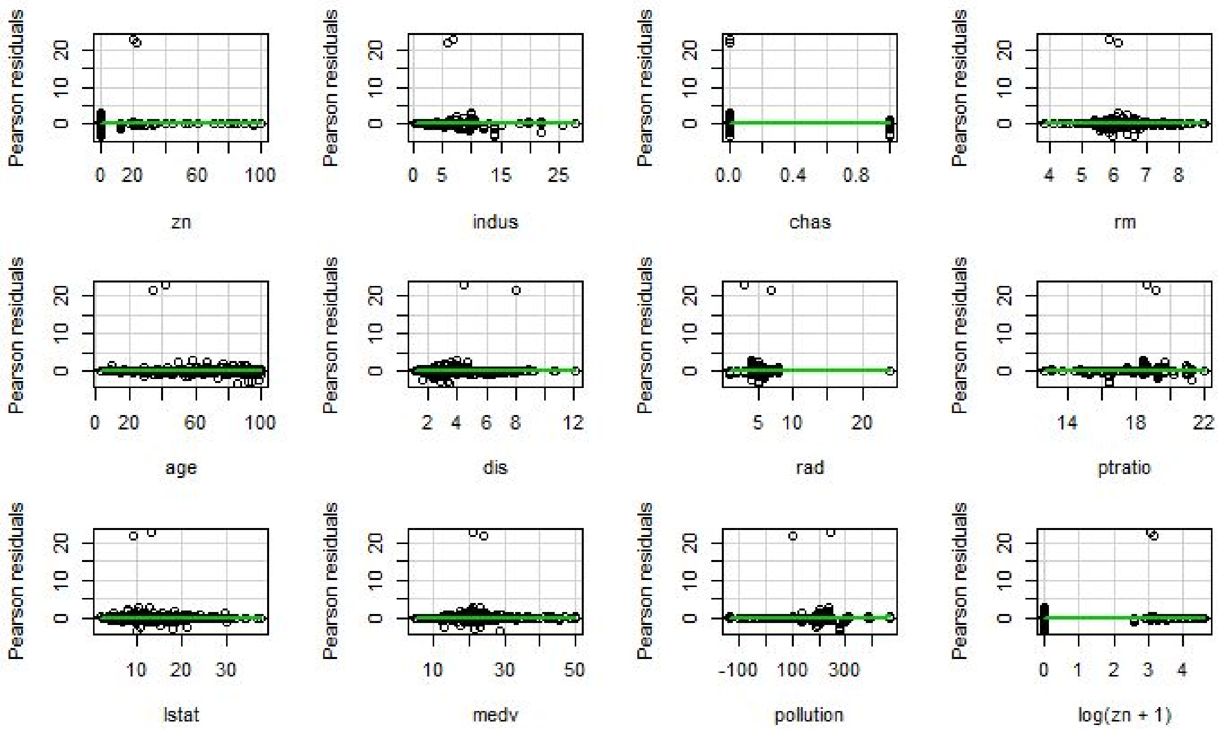
	Test stat	Pr(>|t|)
zn	0.001	0.975
indus	8.532	0.003
chas	0.000	1.000
nox	1.234	0.267
rm	11.103	0.001
age	6.819	0.009
dis	5.896	0.015
rad	0.009	0.926
tax	20.896	0.000
ptratio	8.596	0.003
lstat	2.404	0.121
medv	1.564	0.211

The set of predictor variables with combined variable pollution but without variables tax and nox:

	Test stat	Pr(>|t|)
zn	5.6	0.018
indus	21.6	0.000
chas	0.0	1.000
rm	16.1	0.000
age	9.0	0.003
dis	1.3	0.259
rad	1.5	0.218
ptratio	39.0	0.000
lstat	1.9	0.165
medv	7.8	0.005
pollution	8.9	0.003

The residual plots results are different for the original predictor list and the new predictors set with combined variable pollution. I will do the transformation based on the latter results. Usually, the right skewed variables like zn will need log transformed and the left skewed variables like indus, age , ptratio will be quadratic transformed. I will quadratic transformation for those not skewed variables rm, medv and pollution. Because there are a lot of 0 in zn, I shift zn 1 unit to the right.

After transformation, the residual plots and the statistical results show as follows. There is no significant relationship between the residuals with each predictor.



Then generate new columns for the transformed variables for convenience purpose. The following table represent the first 5 rows of the new data.

zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target	pollution	l.zn	q.indus	q.rm	q.age	q.ptratio	q.medv	q.pollution
0	19.6	0	0.60	7.9	96.2	2.0	5	403	15	3.7	50	1	202	0.0	383	63	9254	216	2500	40804
0	19.6	1	0.87	5.4	100.0	1.3	5	403	15	26.8	13	1	468	0.0	383	29	10000	216	180	219024
0	18.1	0	0.74	6.5	100.0	2.0	24	666	20	18.9	15	1	74	0.0	328	42	10000	408	237	5476
30	4.9	0	0.43	6.4	7.8	7.0	6	300	17	5.2	24	0	128	3.4	24	41	61	276	562	16384
0	2.5	0	0.49	7.2	92.2	2.7	3	193	18	4.8	38	0	295	0.0	6	51	8501	317	1436	87025

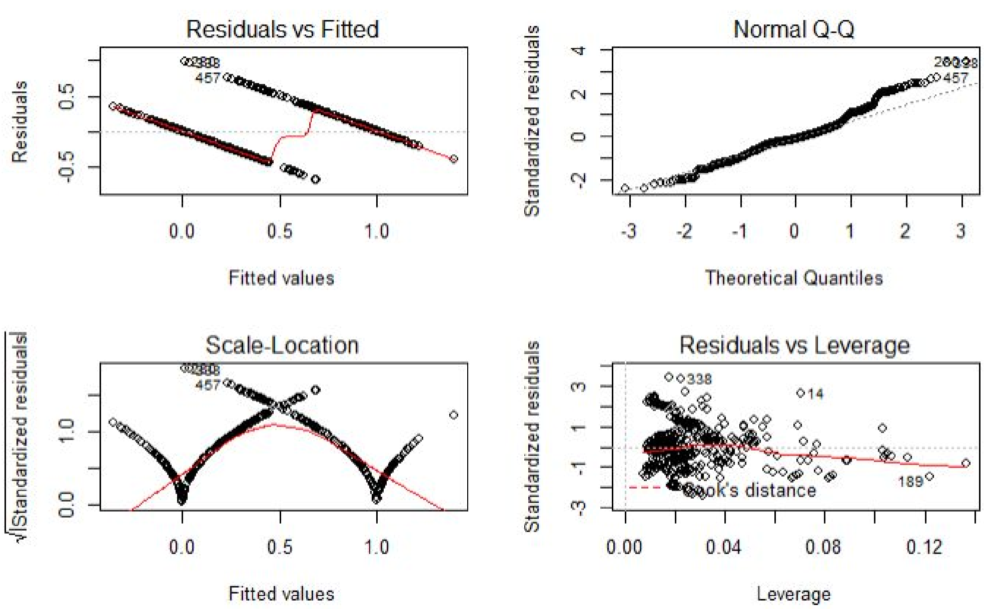
3. BUILD MODELS

3.1 Ordinary least squares regression model - backward selection

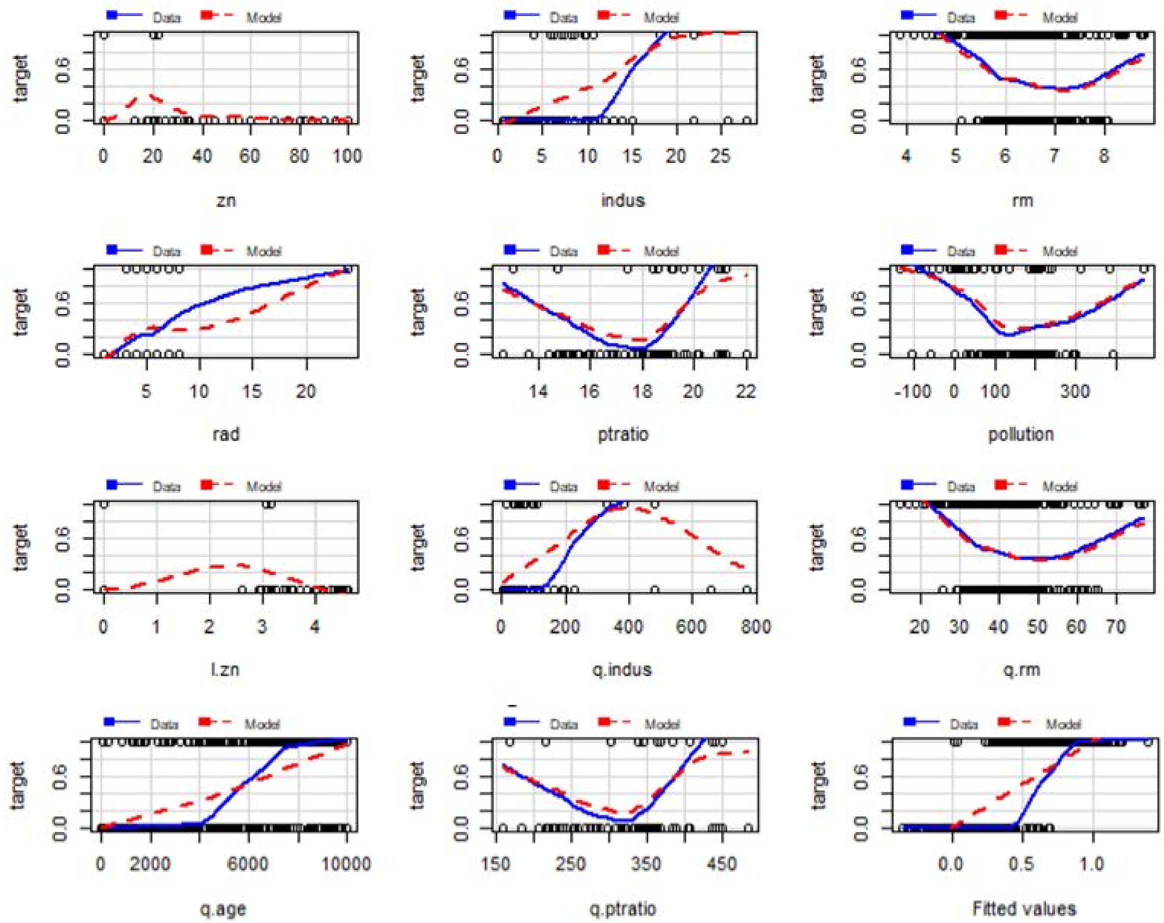
First I want to see what will linear regression model do just for fun. The final model is as follows:

```
Call:
lm(formula = target ~ zn + indus + rm + rad + ptratio + pollution +
    l.zn + q.indus + q.rm + q.age + q.ptratio, data = trsf_df)

Residuals:
   Min     1Q Median     3Q    Max 
-0.691 -0.171 -0.034  0.126  0.984 

Coefficients:
               Estimate  Std. Error t value             Pr(>|t|)    
(Intercept)  9.02143669  1.15677068    7.80     0.00000000000043 ***
zn           0.00216628  0.00143422    1.51              0.13163    
indus        0.05281919  0.01172203    4.51     0.00000841524061 ***
rm          -0.59330282  0.20107658   -2.95              0.00334 ** 
rad          0.02893280  0.00290855    9.95 < 0.0000000000000002 ***
ptratio     -0.87045503  0.11094371   -7.85     0.00000000000031 ***
pollution    0.00081159  0.00017264    4.70     0.00000343779664 ***
l.zn        -0.05817470  0.02108616   -2.76              0.00603 ** 
q.indus     -0.00168792  0.00042013   -4.02     0.00006881272781 ***
q.rm         0.04989151  0.01554596    3.21              0.00142 ** 
q.age        0.00002424  0.00000633    3.83              0.00015 ***
q.ptratio    0.02453885  0.00313309    7.83     0.00000000000034 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.29 on 454 degrees of freedom
Multiple R-squared:  0.673,	Adjusted R-squared:  0.665 
F-statistic:   85 on 11 and 454 DF,  p-value: <0.0000000000000002
```

The diagnostic plots:



Residual plots are problematic when the data are binary. Residual does not provide an assessment of the goodness-of-fit of model. Then use the marginal model plots to evaluate the goodness-of-fit.



From the bottom right-hand plot which uses these fitted values as the horizontal axis, we can see that the two lines do not agree to each other. The model is not reproducing the data in that direction. So I conclude that ordinary linear regression is not a right tool to fit the data with binary response. logistic or Poisson will be appropriate link functions

3.2 Logistic regression model - likelihood-ratio-test-based backward selection

The full model:

```
glm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredSingle term deletions

Model:
target ~ zn + indus + chas + rm + age + dis + rad + ptratio +
    lstat + medv + pollution + l.zn + q.indus + q.rm + q.age +
    q.ptratio + q.medv + q.pollution
            Df Deviance AIC   LRT             Pr(>Chi)
<none>              148 186
zn           1      148 184   0.0              0.83683
indus        1      161 197  12.8              0.00034 ***
chas         1      149 185   0.7              0.40003
rm           1      162 198  13.4              0.00025 ***
age          1      150 186   1.0              0.30736
dis          1      149 185   0.4              0.54031
rad          1      254 290 106.1 < 0.0000000000000002 ***
ptratio      1      169 205  20.3            0.0000065 ***
lstat        1      149 185   0.3              0.58069
medv         1      150 186   1.1              0.28356
pollution    1      168 204  19.6            0.0000096 ***
l.zn         1      149 185   0.8              0.38611
q.indus      1      158 194   9.2              0.00243 **
q.rm         1      161 197  12.5              0.00040 ***
q.age        1      151 187   2.5              0.11051
q.ptratio    1      171 207  22.7            0.0000019 ***
q.medv       1      150 186   1.1              0.28997
q.pollution  1      158 194  10.0              0.00158 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Then drop zn which is least significant and get the new model as follows.

```
glm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredSingle term deletions

Model:
target ~ indus + chas + rm + age + dis + rad + ptratio + lstat
+
    medv + pollution + l.zn + q.indus + q.rm + q.age +
q.ptratio +
    q.medv + q.pollution
            Df Deviance AIC   LRT              Pr(>Chi)
<none>              148 184
indus        1      161 195  12.8               0.00034 ***
chas         1      149 183   0.7               0.40966
rm           1      162 196  13.4               0.00026 ***
age          1      150 184   1.1               0.30240
dis          1      149 183   0.3               0.56235
rad          1      256 290 107.6 < 0.0000000000000002 ***
ptratio      1      172 206  23.1            0.00000151 ***
lstat        1      149 183   0.3               0.56154
medv         1      150 184   1.1               0.29261
pollution    1      168 202  19.6            0.00000979 ***
l.zn         1      152 186   3.8               0.05182 .
q.indus      1      158 192   9.2               0.00248 **
q.rm         1      161 195  12.5               0.00041 ***
q.age        1      151 185   2.6               0.10948
q.ptratio    1      174 208  25.2            0.00000051 ***
q.medv       1      150 184   1.1               0.29915
q.pollution  1      158 192  10.0               0.00160 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Then drop dis which is least significant and get the new model as follows.

```
glm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredSingle term deletions

Model:
target ~ indus + chas + rm + age + rad + ptratio + lstat + medv
+
    pollution + l.zn + q.indus + q.rm + q.age + q.ptratio +
q.medv +
    q.pollution
            Df Deviance AIC   LRT             Pr(>Chi)
<none>             149 183
indus        1      161 193  12.6              0.00038 ***
chas         1      149 181   0.6              0.43188
rm           1      162 194  13.1              0.00029 ***
age          1      150 182   1.5              0.21432
rad          1      257 289 107.9 < 0.0000000000000002 ***
ptratio      1      172 204  23.6           0.00000116 ***
lstat        1      149 181   0.4              0.52206
medv         1      150 182   0.9              0.35091
pollution    1      171 203  22.0           0.00000271 ***
l.zn         1      152 184   3.6              0.05790 .
q.indus      1      158 190   9.1              0.00250 **
q.rm         1      161 193  12.3              0.00045 ***
q.age        1      152 184   3.0              0.08244 .
q.ptratio    1      175 207  25.9           0.00000036 ***
q.medv       1      150 182   0.9              0.33461
q.pollution  1      160 192  11.5              0.00069 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Then drop lstat which is least significant and get the new model as follows.

```
glm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredSingle term deletions

Model:
target ~ indus + chas + rm + age + rad + ptratio + medv +
pollution +
    l.zn + q.indus + q.rm + q.age + q.ptratio + q.medv +
q.pollution
            Df Deviance AIC   LRT             Pr(>Chi)
<none>              149 181
indus        1      162 192  12.6              0.00038 ***
chas         1      150 180   0.8              0.37278
rm           1      162 192  12.7              0.00036 ***
age          1      150 180   1.2              0.27869
rad          1      257 287 107.8 < 0.0000000000000002 ***
ptratio      1      174 204  25.0           0.00000058 ***
medv         1      150 180   1.0              0.31030
pollution    1      171 201  22.1           0.00000262 ***
l.zn         1      153 183   4.0              0.04424 *
q.indus      1      158 188   9.2              0.00245 **
q.rm         1      161 191  12.0              0.00053 ***
q.age        1      152 182   2.6              0.10540
q.ptratio    1      177 207  27.4           0.00000017 ***
q.medv       1      150 180   1.0              0.30665
q.pollution  1      161 191  11.6              0.00066 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Then drop chas which is least significant and get the new model as follows.

```
glm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredSingle term deletions

Model:
target ~ indus + rm + age + rad + ptratio + medv + pollution + 
    l.zn + q.indus + q.rm + q.age + q.ptratio + q.medv + 
q.pollution
            Df Deviance AIC   LRT              Pr(>Chi)    
<none>              150 180                                
indus        1      162 190  11.8               0.00058 ***
rm           1      162 190  12.5               0.00040 ***
age          1      151 179   1.3               0.25699    
rad          1      259 287 109.1 < 0.0000000000000002 ***
ptratio      1      177 205  26.9           0.000000210 ***
medv         1      151 179   0.8               0.36049    
pollution    1      171 199  21.3           0.000003935 ***
l.zn         1      154 182   3.7               0.05371 .  
q.indus      1      158 186   8.4               0.00371 ** 
q.rm         1      162 190  11.8               0.00059 ***
q.age        1      153 181   2.7               0.10318    
q.ptratio    1      180 208  29.6           0.000000054 ***
q.medv       1      151 179   0.9               0.34277    
q.pollution  1      161 189  11.1               0.00087 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Then drop medv which is least significant and get the new model as follows.

```
glm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredSingle term deletions

Model:
target ~ indus + rm + age + rad + ptratio + pollution + l.zn + 
    q.indus + q.rm + q.age + q.ptratio + q.medv + q.pollution
            Df Deviance AIC   LRT              Pr(>Chi)    
<none>              151 179                                
indus        1      162 188  11.3               0.00077 ***
rm           1      163 189  12.4               0.00043 ***
age          1      152 178   1.4               0.24404    
rad          1      259 285 108.4 < 0.0000000000000002 ***
ptratio      1      177 203  26.1           0.000000316 ***
pollution    1      172 198  21.0           0.000004518 ***
l.zn         1      155 181   4.0               0.04551 *  
q.indus      1      159 185   8.0               0.00465 ** 
q.rm         1      162 188  11.5               0.00071 ***
q.age        1      153 179   2.5               0.11168    
q.ptratio    1      180 206  28.7           0.000000084 ***
q.medv       1      151 177   0.1               0.78067    
q.pollution  1      162 188  10.6               0.00113 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Then drop q.medv which is least significant and get the new model as follows.

```
glm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredSingle term deletions

Model:
target ~ indus + rm + age + rad + ptratio + pollution + l.zn +
    q.indus + q.rm + q.age + q.ptratio + q.pollution
            Df Deviance AIC   LRT             Pr(>Chi)
<none>             151 177
indus        1     164 188  12.8              0.00035 ***
rm           1     169 193  17.7          0.000026161 ***
age          1     152 176   1.4              0.23213
rad          1     259 283 108.4 < 0.0000000000000002 ***
ptratio      1     178 202  27.0          0.000000208 ***
pollution    1     172 196  21.0          0.000004576 ***
l.zn         1     155 179   4.0              0.04514 *
q.indus      1     160 184   9.3              0.00224 **
q.rm         1     169 193  18.4          0.000018245 ***
q.age        1     154 178   2.8              0.09431 .
q.ptratio    1     180 204  29.2          0.000000067 ***
q.pollution  1     162 186  10.5              0.00117 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Then drop age which is least significant and get the new model as follows.

```
glm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredSingle term deletions

Model:
target ~ indus + rm + rad + ptratio + pollution + l.zn +
q.indus +
    q.rm + q.age + q.ptratio + q.pollution
            Df Deviance AIC   LRT             Pr(>Chi)
<none>             152 176
indus        1     166 188  14.1              0.00017 ***
rm           1     174 196  21.1          0.000004279 ***
rad          1     259 281 107.0 < 0.0000000000000002 ***
ptratio      1     182 204  29.2          0.000000065 ***
pollution    1     173 195  20.5          0.000006029 ***
l.zn         1     156 178   3.7              0.05309 .
q.indus      1     162 184  10.0              0.00159 **
q.rm         1     174 196  21.8          0.000003003 ***
q.age        1     159 181   6.2              0.01251 *
q.ptratio    1     184 206  31.6          0.000000019 ***
q.pollution  1     163 185  10.2              0.00138 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

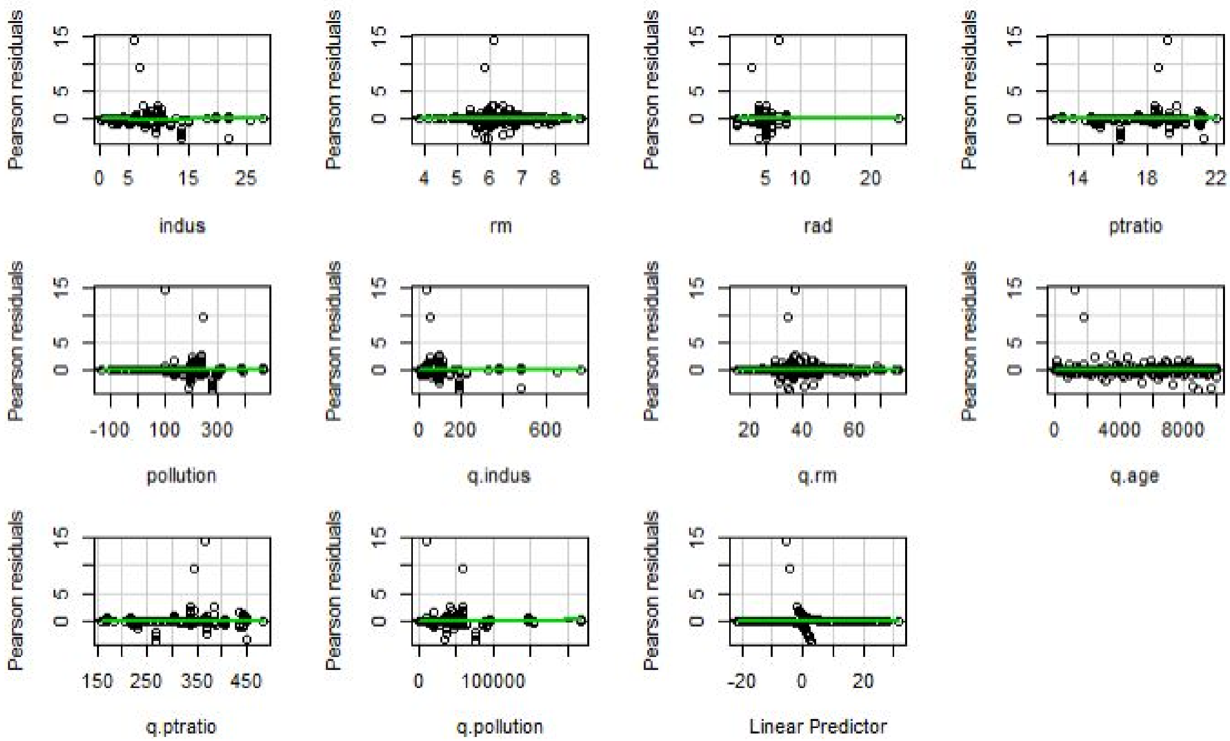
Then drop l.zn which is least significant and get the new model as follows.

```
glm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredSingle term deletions

Model:
target ~ indus + rm + rad + ptratio + pollution + q.indus +
q.rm +
    q.age + q.ptratio + q.pollution
            Df Deviance AIC   LRT             Pr(>Chi)
<none>             156 178
indus        1      182 202  26.1          0.00000033 ***
rm           1      177 197  21.0          0.00000465 ***
rad          1      266 286 109.9 < 0.0000000000000002 ***
ptratio      1      182 202  25.6          0.00000041 ***
pollution    1      179 199  22.7          0.00000191 ***
q.indus      1      174 194  18.3          0.00001873 ***
q.rm         1      178 198  21.8          0.00000307 ***
q.age        1      164 184   8.0              0.0046 **
q.ptratio    1      184 204  28.1          0.00000011 ***
q.pollution  1      167 187  10.8              0.0010 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The final model is as follows:

```
glm.fit: fitted probabilities numerically 0 or 1 occurred
Call:
glm(formula = target ~ indus + rm + rad + ptratio + pollution +
    q.indus + q.rm + q.age + q.ptratio + q.pollution, family =
binomial(link = "logit"),
    data = trsf_df)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.302  -0.086   0.000   0.009   3.270

Coefficients:
               Estimate  Std. Error z value      Pr(>|z|)
(Intercept) 117.8775462  25.3672650    4.65 0.00000337063 ***
indus         1.0471630   0.2514366    4.16 0.00003117358 ***
rm          -25.7538098   6.4066182   -4.02 0.00005822882 ***
rad           1.3317066   0.2115315    6.30 0.00000000031 ***
ptratio      -7.3623416   1.7022766   -4.32 0.00001525337 ***
pollution     0.0729795   0.0162017    4.50 0.00000665520 ***
q.indus      -0.0313415   0.0087167   -3.60       0.00032 ***
q.rm          2.0087972   0.4919438    4.08 0.00004438396 ***
q.age         0.0002367   0.0000869    2.72       0.00646 **
q.ptratio     0.2159273   0.0478716    4.51 0.00000646582 ***
q.pollution  -0.0001032   0.0000282   -3.66       0.00025 ***
---
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: 156.11  on 455  degrees of freedom
AIC: 178.1

Number of Fisher Scoring iterations: 10
```

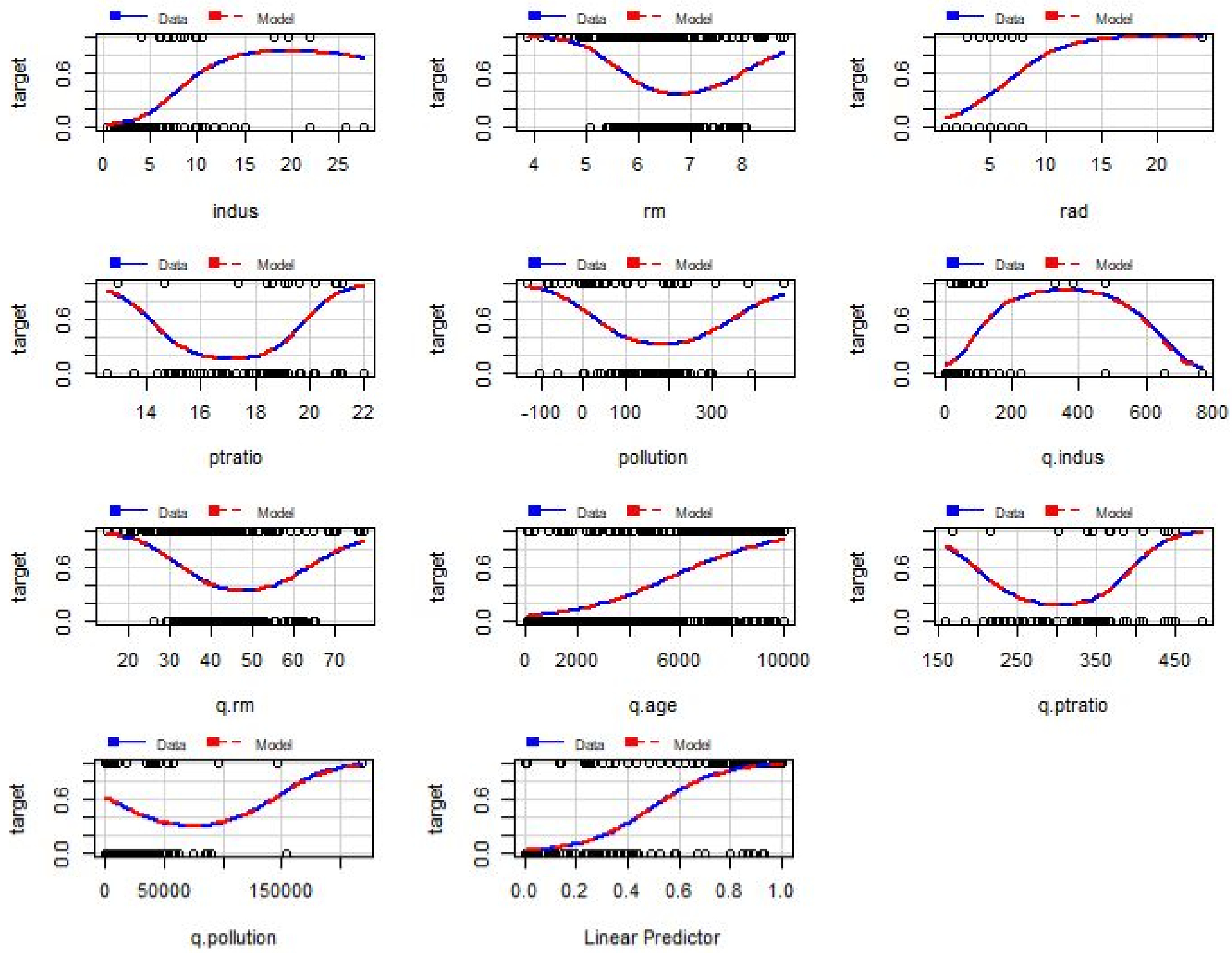
residualPlots() function performs lack-of-fit test to see if a variable has relationship with residuals. Pearson residuals are plotted against predictors one by one. From the plots we can see the relationship between Pearson residuals and each variable is linear (the green line).



The statistical results of the residual plots are shown as below:

```
glm.fit: fitted probabilities numerically 0 or 1 occurred
Test stat Pr(>|t|)
indus          0.000    1.000
rm             0.000    1.000
rad            0.026    0.871
ptratio        0.000    1.000
pollution      0.000    1.000
q.indus        0.834    0.361
q.rm           0.291    0.590
q.age          0.183    0.669
q.ptratio      2.912    0.088
q.pollution    0.680    0.410
```

The marginal model plots are shown as follows.

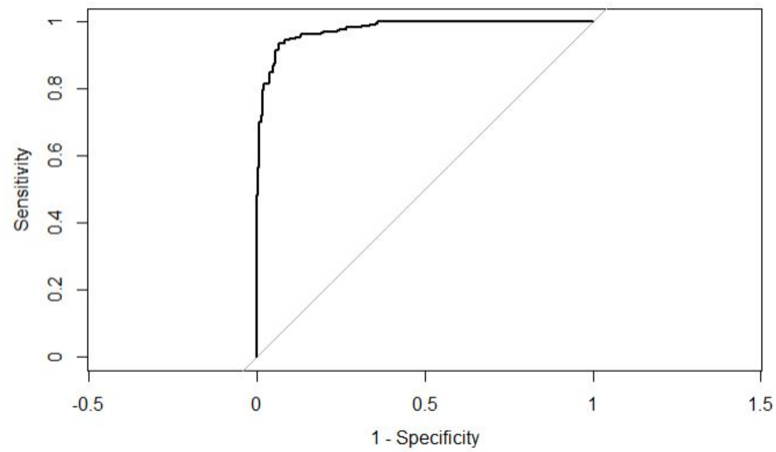


The model and the data agrees to each other quite well. But I got warning message "glm.fit: fitted probabilities numerically 0 or 1 occurred" while doing manually likelihood-ratio-test-based backward selection. I still get the model but the coefficient estimates will be inflated. To avoid this problem, I will try penalized regression by applying glmnet package later.

The confusion matrix is as follows:

```
          Reference
Prediction   0   1
         0 222  17
         1  15 212
```

The ROC curve:



3.3 Logistic regression model - Automated likelihood-ratio-test-based backward selection

Build the model:

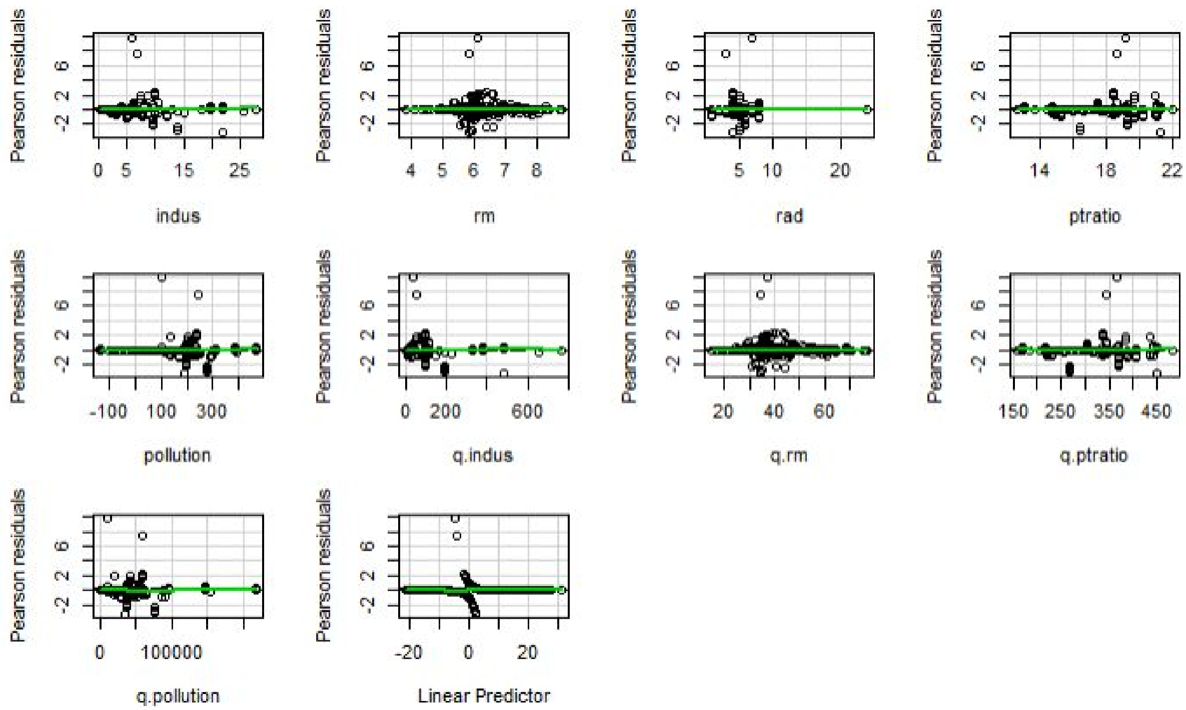
```
 Deleted Chi-Sq d.f. P      Residual d.f. P     AIC
 zn      0.05   1    0.821  0.05    1    0.82  -1.9
 dis     0.33   1    0.564  0.38    2    0.82  -3.6
 lstat   0.40   1    0.525  0.79    3    0.85  -5.2
 chas    0.74   1    0.390  1.53    4    0.82  -6.5
 medv    0.78   1    0.377  2.31    5    0.80  -7.7
 q.medv  0.04   1    0.844  2.35    6    0.89  -9.6
 age     1.25   1    0.263  3.60    7    0.82 -10.4
 l.zn    3.00   1    0.083  6.60    8    0.58  -9.4
 q.age   6.55   1    0.010 13.15    9    0.16  -4.8

Approximate Estimates after Deleting Factors

               Coef   S.E. Wald Z             P
Intercept     1.944 0.7707  2.522 0.01165804829
indus         6.646 1.7045  3.899 0.00009659160
rm          -15.340 4.8623 -3.155 0.00160559163
rad           9.962 1.7939  5.553 0.00000002806
ptratio     -17.293 3.6346 -4.758 0.00000195710
pollution     7.993 2.0209  3.955 0.00007652818
q.indus      -4.706 1.4752 -3.190 0.00142117377
q.rm         15.565 4.8973  3.178 0.00148162063
q.ptratio    17.638 3.5809  4.925 0.00000084161
q.pollution  -4.276 1.3144 -3.253 0.00114253531

Factors in Final Model

[1] indus       rm          rad         ptratio     pollution   q.indus     q.rm
[8] q.ptratio   q.pollution
```

The final model is not the same as the one manually selected. The variable q.age has been removed in this model.

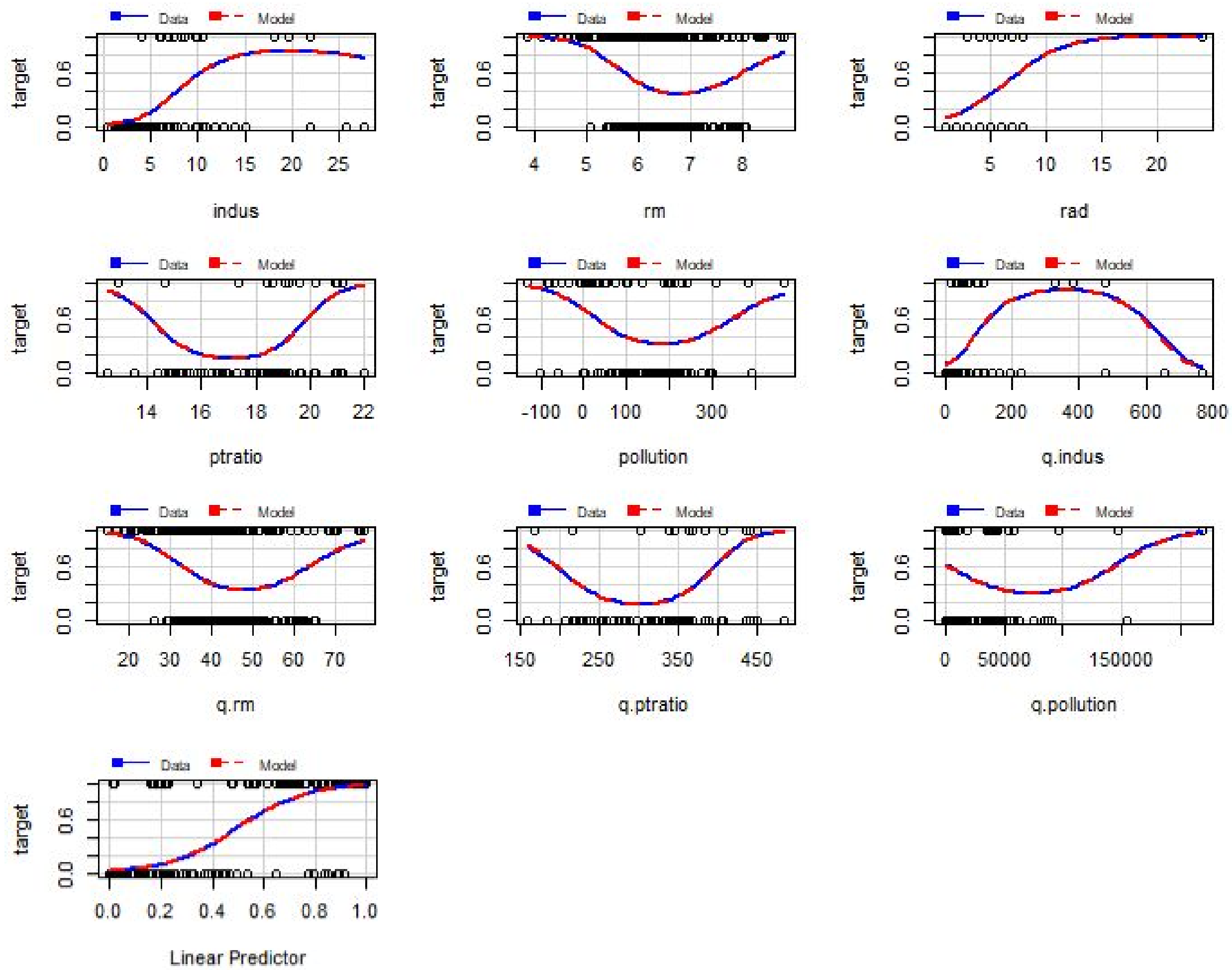
```
Call:
glm(formula = target ~ indus + rm + rad + ptratio + pollution +
    q.indus + q.rm + q.ptratio + q.pollution, family = binomial(link = "logit"),
    data = trsf_df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1923  -0.1115  -0.0001   0.0054   3.0266

Coefficients:
               Estimate  Std. Error z value      Pr(>|z|)
(Intercept) 127.4822907  25.5978270    4.98 0.00000063519 ***
indus         1.1619440   0.2597213    4.47 0.00000768375 ***
rm          -23.5160986   6.3005184   -3.73       0.00019 ***
rad           1.3499651   0.2095711    6.44 0.00000000012 ***
ptratio      -9.1736992   1.6953565   -5.41 0.00000006265 ***
pollution     0.0723076   0.0167655    4.31 0.00001611455 ***
q.indus      -0.0336299   0.0090742   -3.71       0.00021 ***
q.rm          1.8364492   0.4854077    3.78       0.00015 ***
q.ptratio     0.2657231   0.0476575    5.58 0.00000002466 ***
q.pollution  -0.0001036   0.0000297   -3.49       0.00048 ***
---
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: 164.13  on 456  degrees of freedom
AIC: 184.1

Number of Fisher Scoring iterations: 10
```

Lack-of-fit test were performed again to check the relationship between the residuals and variables. From the plots Pearson residuals vs. predictors(the green lines) and the statistical results, we can see the there is no significant relationship between Pearson residuals and each variable.

```
glm.fit: fitted probabilities numerically 0 or 1 occurred
Test stat Pr(>|t|)
indus           0.000     1.00
rm              0.000     1.00
rad             0.017     0.90
ptratio         0.000     1.00
pollution       0.000     1.00
q.indus         1.968     0.16
q.rm            0.991     0.32
q.ptratio       0.740     0.39
q.pollution     1.265     0.26
```



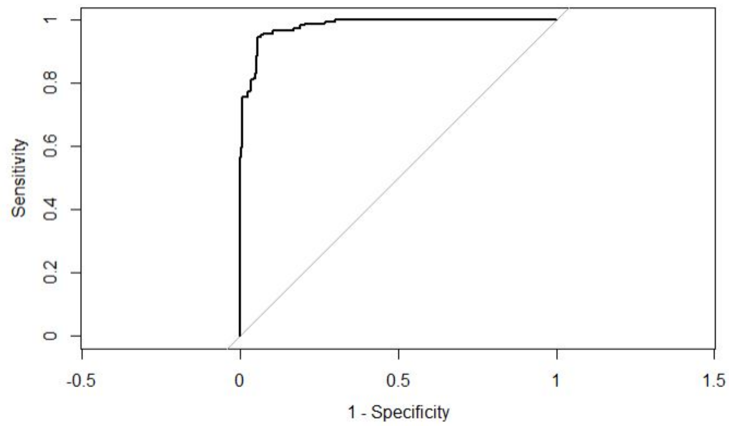
The marginal model plots show that the model and the data agree each other very well.



The confusion matrix of this model is:

```
          Reference
Prediction   0   1
         0 226  15
         1  11 214
```

The ROC plot:

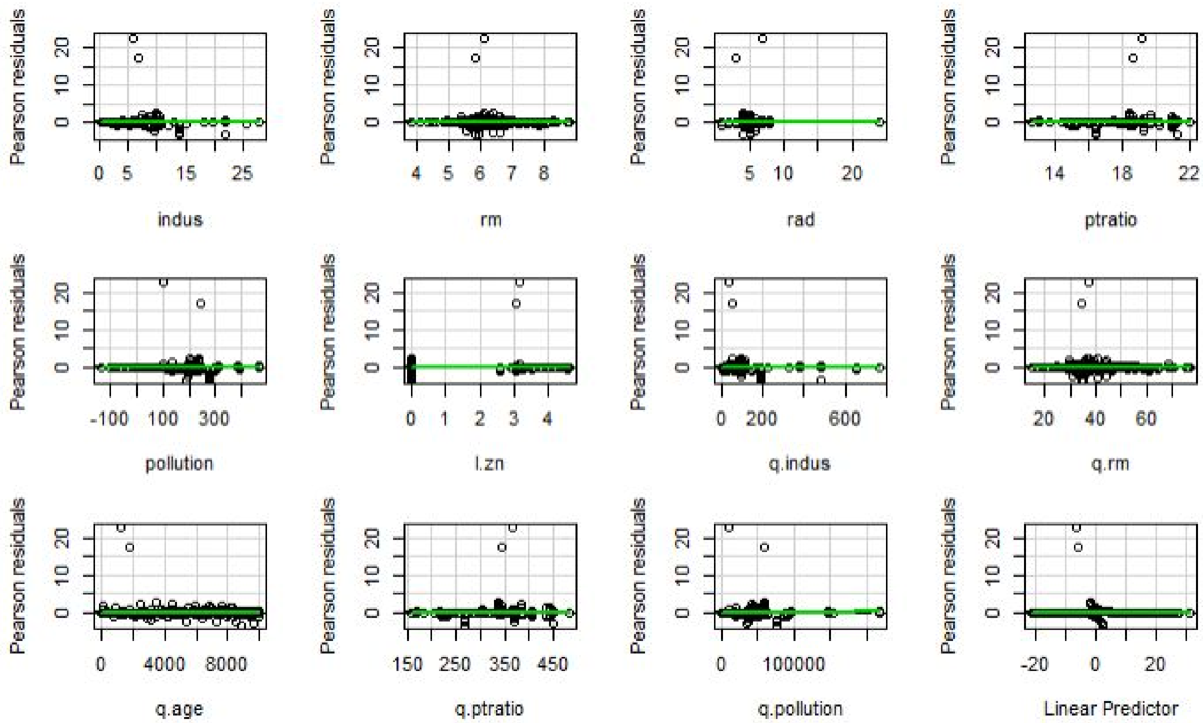


3.4 Logistic regression model - AIC-based automated enumeration approach

The summary of the final model is as follows:

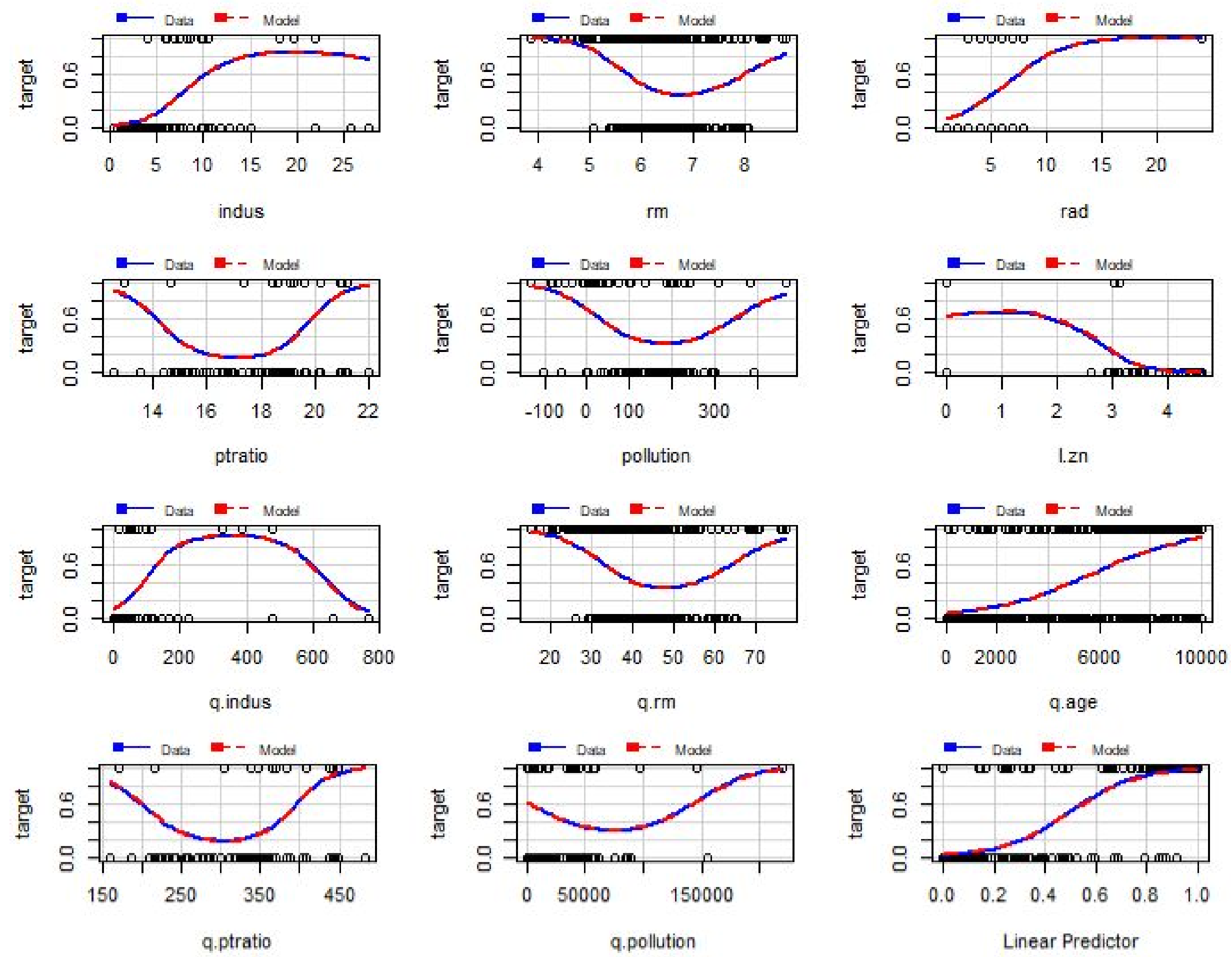
```
Logistic Regression Model

 lrm(formula = target ~ zn + indus + chas + rm + age + dis + rad +
     ptratio + lstat + medv + pollution + l.zn + q.indus + q.rm +
     q.age + q.ptratio + q.medv + q.pollution, data =
data.frame(scale(trsf_df)),
     maxit = 50)

                           Model Likelihood      Discrimination     Rank
Discrim.
                              Ratio Test            Indexes
Indexes
 Obs              466      LR chi2      497.43    R2          0.875    C
0.985
   -0.981922216472607237   d.f.             18    g          12.513    Dxy
0.970
  1.01622517600003229      Pr(> chi2) <0.0001    gr     271927.333    gamma
0.970
 max |deriv| 0.0001                              gp          0.485    tau-a
0.486
                                                 Brier       0.045

             Coef     S.E.   Wald Z Pr(>|Z|)
 Intercept     1.9457 0.8578  2.27  0.0233
 zn            0.3774 1.6662  0.23  0.8208
 indus         6.0803 1.9168  3.17  0.0015
 chas         -0.1840 0.2199 -0.84  0.4028
 rm          -22.9208 7.2788 -3.15  0.0016
 age          -1.6983 1.6328 -1.04  0.2983
 dis           0.3013 0.4909  0.61  0.5394
 rad          11.5551 1.9711  5.86  <0.0001
 ptratio     -18.9110 4.6639 -4.05  <0.0001
 lstat        -0.2612 0.4740 -0.55  0.5816
 medv          2.3525 2.2039  1.07  0.2858
 pollution     9.2676 2.1540  4.30  <0.0001
 l.zn         -1.2444 1.1057 -1.13  0.2604
 q.indus      -4.5019 1.6281 -2.77  0.0057
 q.rm         23.2447 7.5861  3.06  0.0022
 q.age         2.6745 1.6644  1.61  0.1081
 q.ptratio    19.0574 4.4876  4.25  <0.0001
 q.medv       -2.5101 2.3512 -1.07  0.2857
 q.pollution  -4.9204 1.3873 -3.55  0.0004
```

The lack of fitness test results:




```
glm.fit: fitted probabilities numerically 0 or 1 occurred
Test stat Pr(>|t|)
indus           0.000     1.000
rm              0.000     1.000
rad             0.099     0.754
ptratio         0.000     1.000
pollution       0.000     1.000
l.zn            0.033     0.856
q.indus         0.811     0.368
q.rm            0.177     0.674
q.age           0.366     0.545
q.ptratio       3.591     0.058
q.pollution     0.807     0.369
```

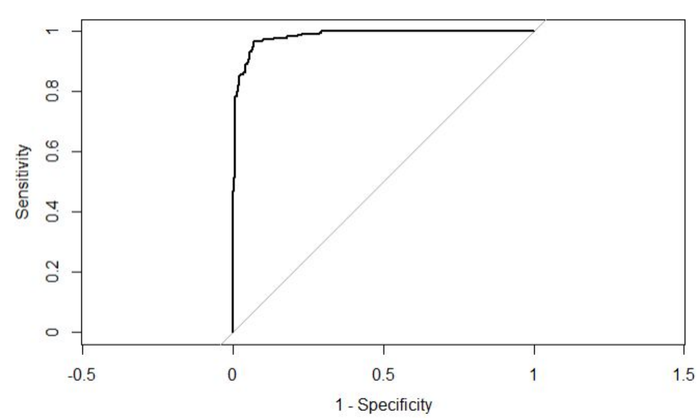
There is no statistically significant relationship between Pearson residuals and the predictor variables. The marginal model plots in below show the agreement between the model and the data.



The confusion matrix:

```
          Reference
Prediction   0   1
         0 226  16
         1  11 213
```

The ROC curve:

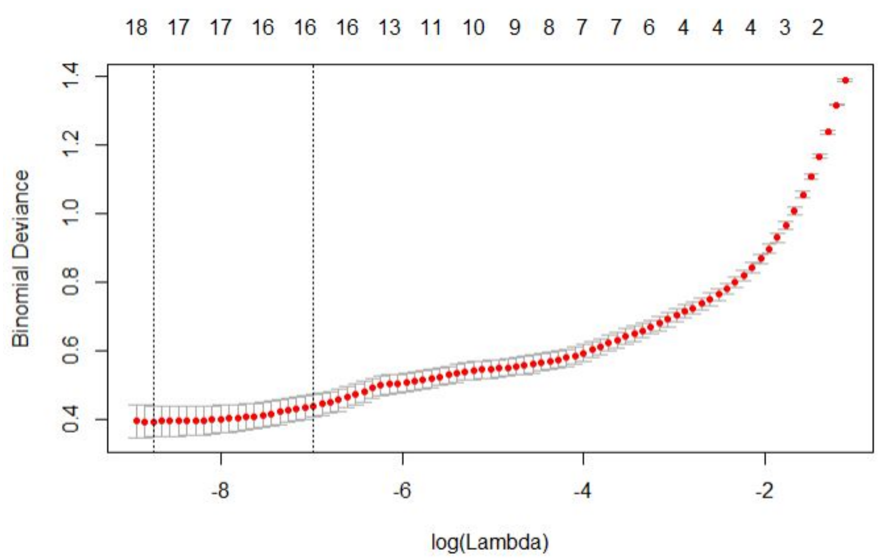


3.5 Logistic regression model - LASSO regression model

I got warning message "glm.fit: fitted probabilities numerically 0 or 1 occurred" while doing manually likelihood-ratio-test-based backward selection. I still get the model but the coefficient estimates will be inflated. To avoid this problem, I try penalized regression by applying glmnet package. Glmnet package fits a generalized linear model via penalized maximum likelihood. The object of the regression is to a model with the smallest number of coefficients that also gives a good accuracy. The hyperparameter lambda (lambda.1se) gives the simplest model but also lies within one standard error of the optimal value of lambda. This value of lambda is what will be used in the the future computation. Here, cv.glmnet function will do k-fold cross-validation to automatically find a value for the value of lambda.

The summary of the model:

```
           Length Class  Mode
lambda     85     -none- numeric
cvm        85     -none- numeric
cvsd       85     -none- numeric
cvup       85     -none- numeric
cvlo       85     -none- numeric
nzero      85     -none- numeric
name        1     -none- character
glmnet.fit 13     lognet list
lambda.min  1     -none- numeric
lambda.1se  1     -none- numeric
```



The plot shows that the log of the optimal value of lambda (i.e. the one that minimises the root mean square error) is approximately -8. Extract the lambda value from the model then lambda. min = 0.00016. lambda.1se = 0.00093.

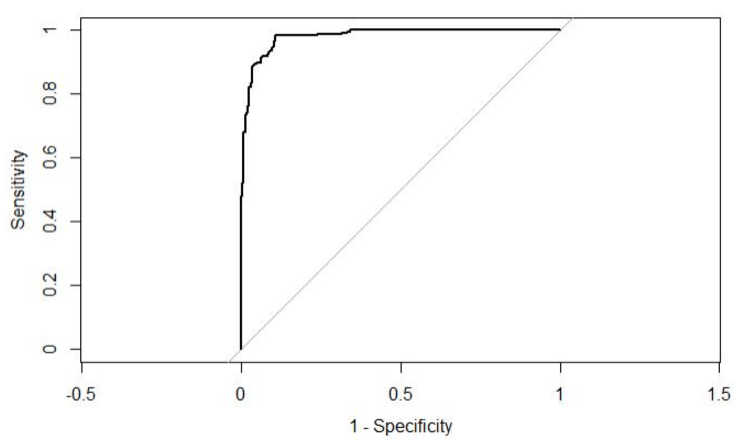
The coefficients:

```
19 x 1 sparse Matrix of class "dgCMatrix"
                       1
(Intercept)   9.651307
zn           -0.018054
indus         0.596014
chas         -0.357413
rm           -0.606821
age          -0.038543
dis           .
rad           0.906759
ptratio      -2.774919
lstat         0.007951
medv          .
pollution     0.035909
l.zn         -0.098923
q.indus      -0.018271
q.rm          .
q.age         0.000582
q.ptratio     0.085225
q.medv        0.002232
q.pollution  -0.000039
```

The confusion matrix:

```
          Reference
Prediction   0   1
         0 222  22
         1  15 207
```

The ROC curve:



3.6 Logistic regression model - AIC-based backward selection

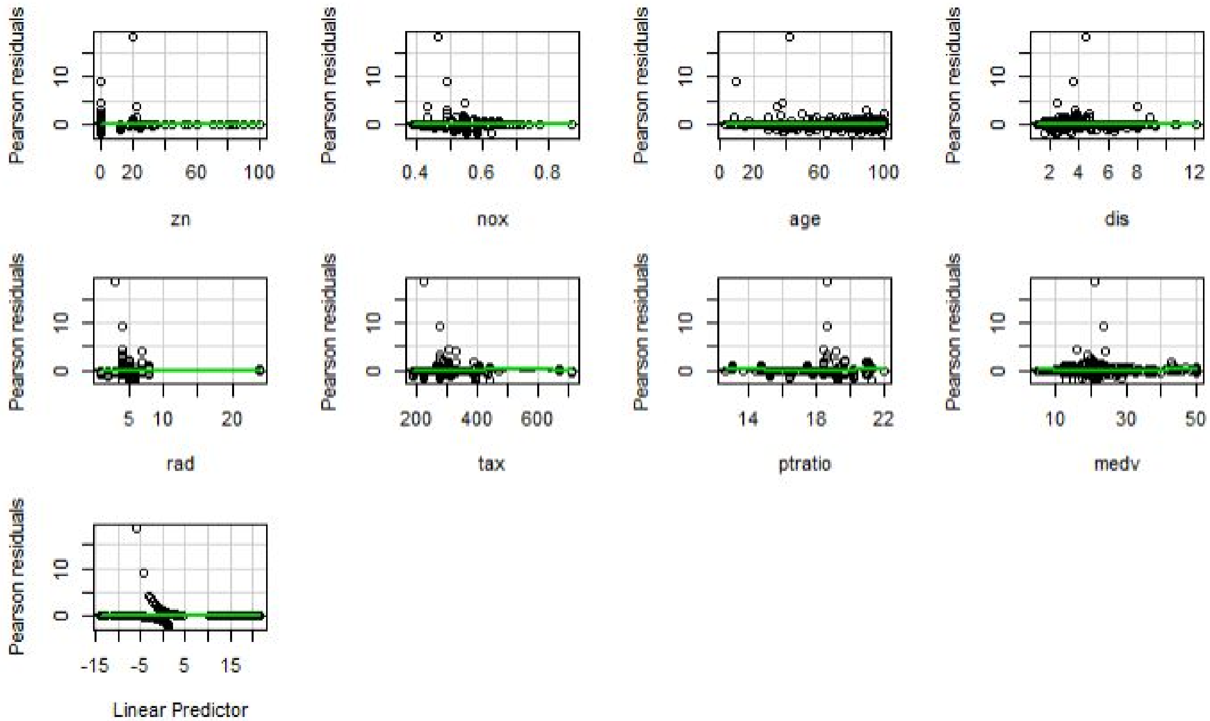
Because the memory is not big enough to do bestglm computation for more than 12 variables, I used to untransformed variables including nox and tax as initiate dataset to search for the best predictor variable set with bestglm function. I tried AIC-based method first and then try BIC-based method in the next section (3.7).

The initial model:

```
Call:
glm(formula = y ~ ., family = family, data = Xi, weights = weights)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.830  -0.175  -0.002   0.003   3.419

Coefficients:
             Estimate Std. Error z value      Pr(>|z|)
(Intercept) -37.41592    6.03501   -6.20 0.00000000057 ***
zn           -0.06865    0.03202   -2.14        0.0320 *
nox          42.80777    6.67869    6.41 0.00000000015 ***
age           0.03295    0.01095    3.01        0.0026 **
dis           0.65490    0.21405    3.06        0.0022 **
rad           0.72511    0.14979    4.84 0.00000129256 ***
tax          -0.00776    0.00265   -2.92        0.0035 **
ptratio       0.32363    0.11139    2.91        0.0037 **
medv          0.11047    0.03545    3.12        0.0018 **
---
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: 215.3

Number of Fisher Scoring iterations: 9
```

The lack of fitness test results:

```
        Test stat Pr(>|t|)
zn           0.07    0.792
nox          0.61    0.433
age          5.92    0.015
dis          7.54    0.006
rad          0.30    0.585
tax         21.10    0.000
ptratio      9.48    0.002
medv         2.47    0.116
```



There is statistically significant relationship between Pearson residuals and the predictor variables age, dis, ptratio and tax, suggesting adding the quadratic terms in previous models for age, dis, ptratio is reasonable.

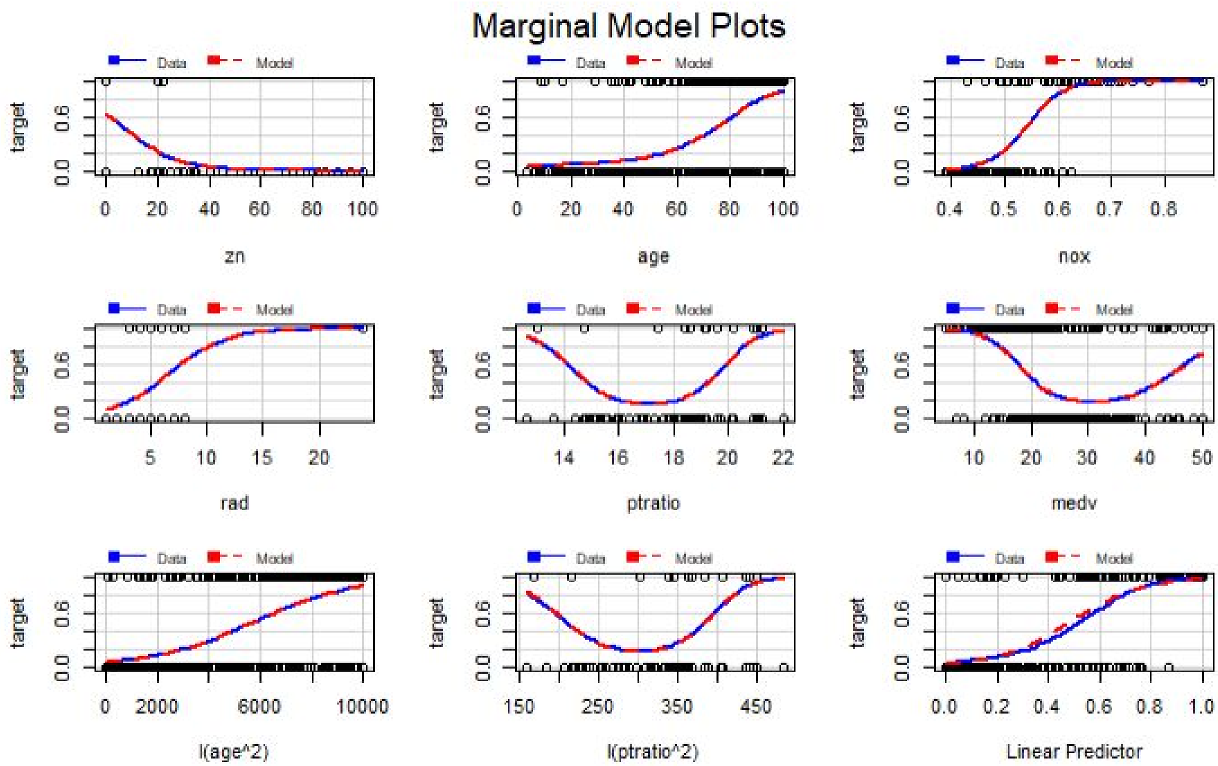
After testing, removing dis and quadratic dis will make the residual vs predictor variable reasonable. So the final model is:

```
Call:
glm(formula = target ~ zn + age + nox + rad + ptratio + medv +
    I(age^2) + I(ptratio^2), family = binomial(link = "logit"),
    data = crime_train)

Deviance Residuals:
   Min       1Q   Median       3Q      Max
-1.997   -0.325   -0.005    0.003    3.320

Coefficients:
              Estimate Std. Error z value  Pr(>|z|)
(Intercept)  41.173734  17.523797    2.35   0.01879 *
zn           -0.084562   0.032672   -2.59   0.00965 **
age          -0.049510   0.036004   -1.38   0.16909
nox          13.459534   4.647311    2.90   0.00378 **
rad           0.649376   0.136123    4.77 0.0000018 ***
ptratio      -6.156163   1.798073   -3.42   0.00062 ***
medv          0.076511   0.031808    2.41   0.01616 *
I(age^2)      0.000512   0.000293    1.75   0.08096 .
I(ptratio^2)  0.175988   0.049360    3.57   0.00036 ***
---
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: 202.94  on 457  degrees of freedom
AIC: 220.9

Number of Fisher Scoring iterations: 8
```

The residual plots statistical results:

```
             Test stat Pr(>|t|)
zn               0.095     0.76
age              0.000     1.00
nox              0.000     1.00
rad              1.680     0.20
ptratio          0.000     1.00
medv             0.108     0.74
I(age^2)         0.258     0.61
I(ptratio^2)     0.401     0.53
```

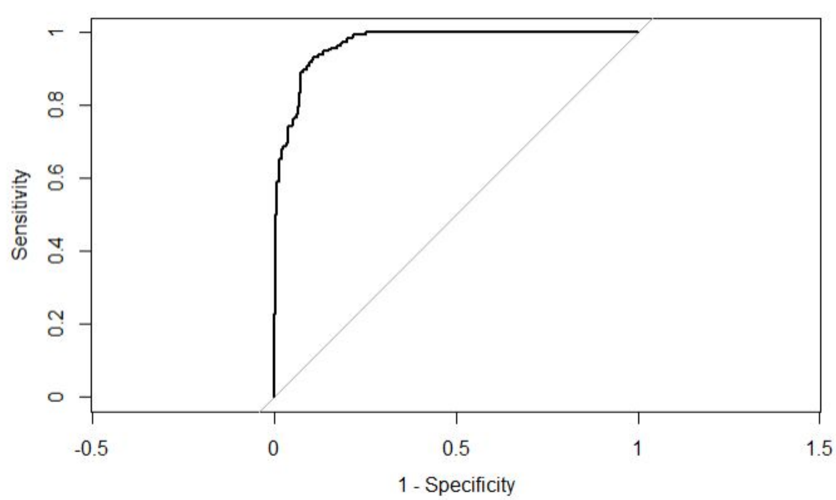
The marginal model plots in below show the agreement between the model and the data.



Confusion matrix

```
          Reference
Prediction   0   1
         0 218  22
         1  19 207
```

ROC curve



3.7 Logistic regression model - BIC-based bestglm

The initial model:

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

Call:
glm(formula = y ~ ., family = family, data = Xi, weights = weights)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8972  -0.2780  -0.0400   0.0056   2.5595

Coefficients:
             Estimate Std. Error z value             Pr(>|z|)
(Intercept) -19.86742    2.36832   -8.39 < 0.0000000000000002 ***
nox          35.63352    4.52368    7.88   0.0000000000000034 ***
rad           0.63764    0.11944    5.34   0.0000000937677682 ***
tax          -0.00815    0.00233   -3.49              0.00048 ***
---
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: 224.47  on 462  degrees of freedom
AIC: 232.5

Number of Fisher Scoring iterations: 8
```

The lack of fitness test results for the initial model:

```
    Test stat Pr(>|t|)
nox     0.112     0.74
rad     0.015     0.90
tax    21.389     0.00
```

There is statistically significant relationship between Pearson residuals and the predictor variables tax. But adding the quadratic terms for tax there is still significant relationship between the residual and tax and quadratic tax. So I remove tax and there is no significant relationship between the residual and predictor variables anymore. But the model become too simple with only two predictor variables.

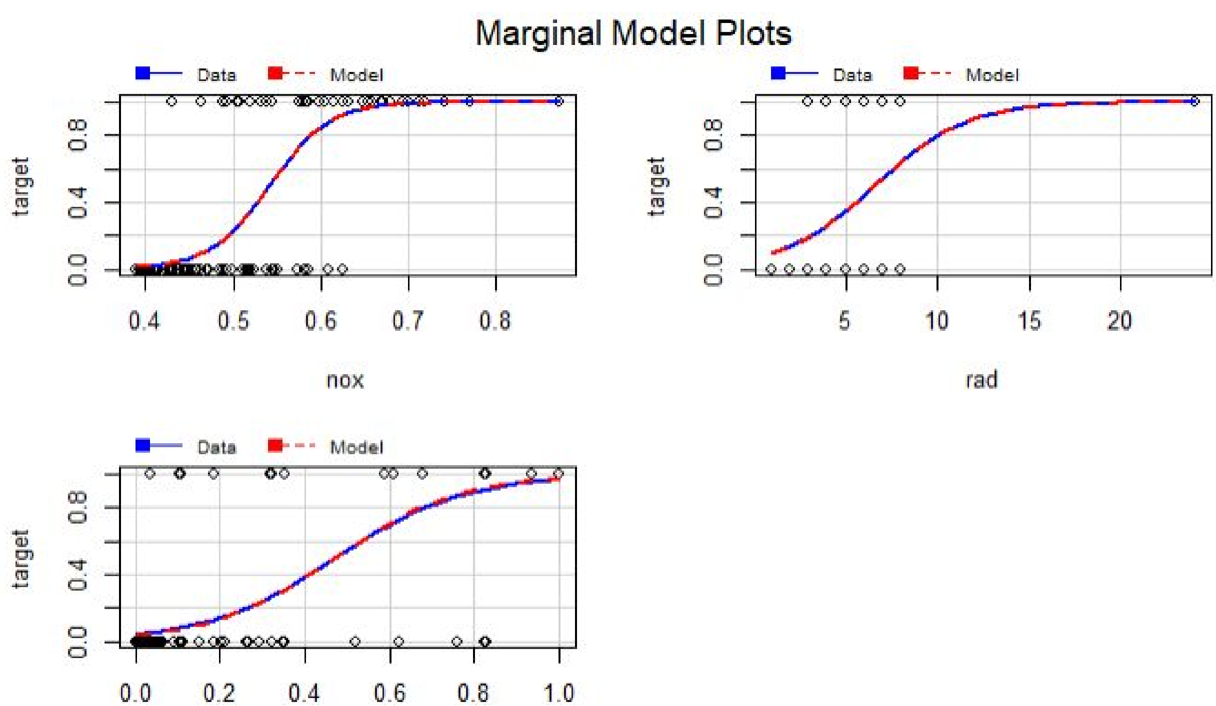
The lack of fitness test results for the final model:

```
    Test stat Pr(>|t|)
nox      1.30     0.25
rad      0.69     0.40
```

The final model is as follows:

```
Call:
glm(formula = target ~ nox + rad, family = binomial(link = "logit"),
    data = crime_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8769  -0.3447  -0.0692   0.0068   2.5803

Coefficients:
            Estimate Std. Error z value             Pr(>|z|)
(Intercept)  -17.453      1.949   -8.96 < 0.0000000000000002 ***
nox           27.196      3.232    8.42 < 0.0000000000000002 ***
rad            0.514      0.108    4.75             0.000002 ***
---
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: 239.51  on 463  degrees of freedom
AIC: 245.5

Number of Fisher Scoring iterations: 8
```

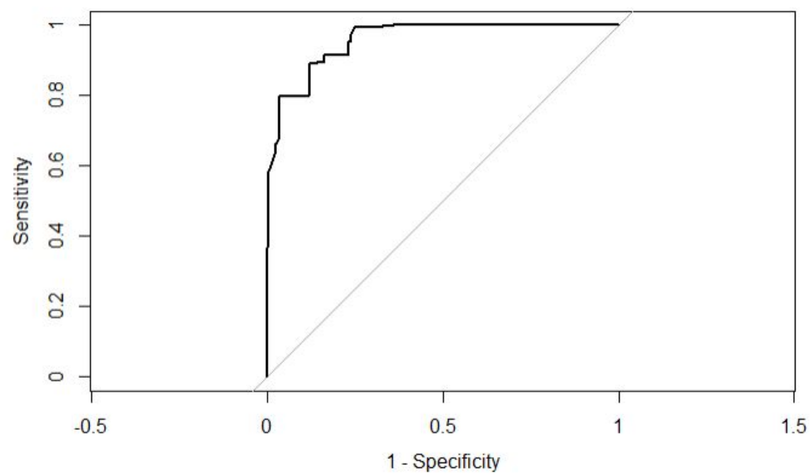
The marginal model plots in below show the agreement between the model and the data.



The confusion matrix:

```
           Reference
Prediction   0   1
         0 222  37
         1  15 192
```

The ROC curve:



3.8 Logistic regression model - train model

The initital model:

```
glm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurredglm.fit: fitted probabilities numerically 0 or 1
occurred
Call:
glm(formula = target ~ zn + rm + rad + ptratio + pollution +
    q.indus + q.rm + q.ptratio + q.pollution, family =
binomial(link = "logit"),
    data = trsf_df)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.328  -0.169   0.000   0.008   3.348

Coefficients:
               Estimate  Std. Error z value      Pr(>|z|)
(Intercept) 150.5946112  26.5012009    5.68 0.00000001327 ***
zn           -0.1321370   0.0379156   -3.49       0.00049 ***
rm          -22.7430469   5.6960208   -3.99 0.00006529885 ***
rad           1.0774629   0.1807254    5.96 0.00000000249 ***
ptratio     -10.2426381   1.7005770   -6.02 0.00000000171 ***
pollution     0.0477046   0.0136439    3.50       0.00047 ***
q.indus       0.0049233   0.0015874    3.10       0.00192 **
q.rm          1.7387473   0.4336651    4.01 0.00006086690 ***
q.ptratio     0.2887534   0.0470045    6.14 0.00000000081 ***
q.pollution  -0.0000748   0.0000242   -3.09       0.00200 **
---
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: 173.79  on 456  degrees of freedom
AIC: 193.8

Number of Fisher Scoring iterations: 9
```

The lack-of-fitness test for the initial model:

```
             Test stat Pr(>|t|)
zn               0.057    0.811
rm               0.000    1.000
rad              0.540    0.462
ptratio          0.000    1.000
pollution        0.000    1.000
q.indus          9.090    0.003
q.rm             0.593    0.441
q.ptratio        0.396    0.529
q.pollution      2.797    0.094
```

Then I remove q.indus and get the final model:

```
Call:
glm(formula = target ~ zn + rm + rad + ptratio + pollution +
    q.rm + q.ptratio + q.pollution, family = binomial(link = "logit"),
    data = trsf_df)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.799  -0.196   0.000   0.014   3.349

Coefficients:
               Estimate  Std. Error z value       Pr(>|z|)
(Intercept) 157.7617132  26.6596509    5.92 0.000000003266 ***
zn           -0.1547377   0.0401490   -3.85        0.00012 ***
rm          -20.7885742   5.4008673   -3.85        0.00012 ***
rad           0.9325898   0.1640043    5.69 0.000000012976 ***
ptratio     -11.3900700   1.7830048   -6.39 0.000000000168 ***
pollution     0.0350264   0.0125724    2.79        0.00534 **
q.rm          1.5839195   0.4107434    3.86        0.00012 ***
q.ptratio     0.3197318   0.0491423    6.51 0.000000000077 ***
q.pollution  -0.0000490   0.0000222   -2.21        0.02686 *
---
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: 184.71  on 457  degrees of freedom
AIC: 202.7

Number of Fisher Scoring iterations: 9
```

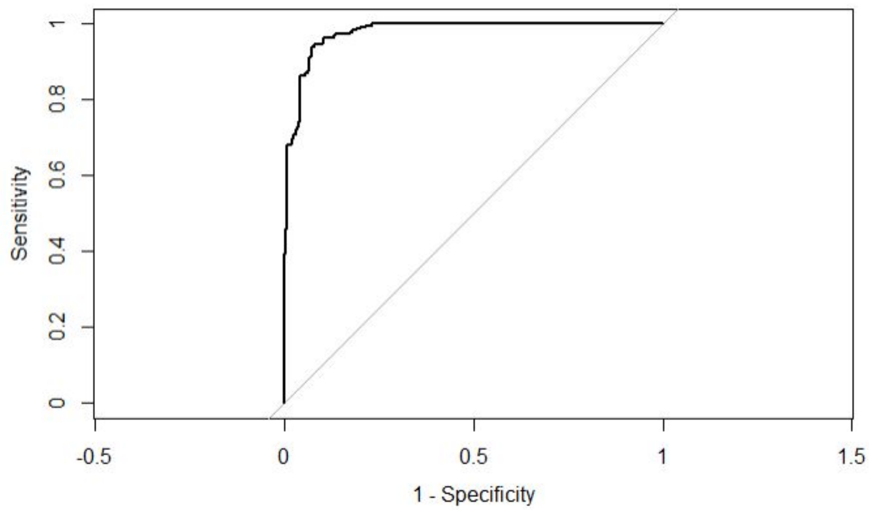
The lack-of-fitness test for the final model:

```
            Test stat Pr(>|t|)
zn               0.21    0.645
rm               0.00    1.000
rad              1.52    0.218
ptratio          0.00    1.000
pollution        0.00    1.000
q.rm             0.32    0.573
q.ptratio        0.00    0.987
q.pollution      3.38    0.066
```

The confusion matrix

```
          Reference
Prediction   0   1
         0 221  17
         1  16 212
```

The ROC curve:



4. SELECT MODELS

The confusion matrix for each model is shown as follows:

model.1 manually LRT

```
          Reference
Prediction   0   1
         0 222  17
         1  15 212
```

model.2 automatic LRT

```
          Reference
Prediction   0   1
         0 226  15
         1  11 214
```

model.3 AIC-backwards

```
          Reference
Prediction   0   1
         0 226  16
         1  11 213
```

model.4 LASSO

```
          Reference
Prediction   0   1
         0 222  22
         1  15 207
```

model.5 AIC-based bestglm

```
          Reference
Prediction   0   1
         0 217  22
         1  20 207
```

model.6 BIC-based bestglm

```
          Reference
Prediction   0   1
         0 213  37
         1  24 192
```

model.7 train model'

```
          Reference
Prediction   0   1
         0 221  17
         1  16 212
```

The summary of the

model	accuracy	error.rate	precision	sensitivity	specificity	F1	pseudo.R2	AIC	AUC	number.of.predictor
model.1 manually LRT	0.94	0.06	0.95	0.93	0.95	0.94	0.76	178	0.98	10
model.2 automatic LRT	0.94	0.06	0.95	0.93	0.95	0.94	0.75	184	0.98	9
model.3 AIC-backwards	0.94	0.06	0.95	0.93	0.95	0.94	0.76	176	0.98	11
model.4 LASSO	0.92	0.08	0.93	0.90	0.94	0.92	0.73	-439	0.98	15
model.5 AIC-based bestglm	0.91	0.09	0.91	0.90	0.92	0.91	0.69	221	0.97	8
model.6 BIC-based bestglm	0.87	0.13	0.89	0.84	0.90	0.86	0.63	246	0.98	2
model.7 train model	0.93	0.07	0.93	0.93	0.93	0.93	0.71	203	0.98	8

Because the object is prediction, the model have higher accuracy will be more favorable. The model-1 and model-2 have the highest accuracy. Based on the above summary,, Model-1 and Model-2 have the same value of error rate, precision, sensitivity, specificity, F1 score and AUC. Model-1 has lower AIC(178) than model-2 (184) . But the model-2 has less predictor variables (9) than model-1 (10) so it is a more parsimonious model. Taken together, model-2 which is built by automated likelihood-ratio-test-based backward selection is the best model among the 7 logistic models.

Then make predictions using the evaluation data set. The following table represent the first 10 rows of the evaluation data set with the predicted values.

zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	pred.prob	pred.target
0	7.1	0	0.47	7.2	61	5.0	2	242	18	4.0	35	0.00	0
0	8.1	0	0.54	6.1	84	4.5	4	307	21	10.3	18	0.67	1
0	8.1	0	0.54	6.5	94	4.4	4	307	21	12.8	18	0.64	1
0	8.1	0	0.54	6.0	82	4.0	4	307	21	27.7	13	0.71	1
0	6.0	0	0.50	5.8	42	3.9	5	279	19	8.8	21	0.11	0
25	5.1	0	0.45	5.7	66	7.2	8	284	20	13.2	19	0.63	1
25	5.1	0	0.45	6.0	93	6.8	8	284	20	14.4	16	0.52	1
0	4.5	0	0.45	6.6	56	4.4	3	247	18	6.5	27	0.00	0
0	4.5	0	0.45	6.1	57	3.8	3	247	18	8.4	22	0.00	0
0	2.9	0	0.44	6.2	70	3.5	2	276	18	11.3	21	0.00	0

The full data could be found through the following URL:
https://github.com/YunMai-SPS/DATA621_homework/blob/master/data621_assignment3/crime_test_predition.csv