DATA621_Homework3

Jenny

November 4, 2018

install.packages("leaps") install.packages("lattice")

```
library(caret)
```

```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(ggplot2)
library(reshape2)
library(leaps)
```

Introduction

In this homework, I 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).

My 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. I will provide classifications and probabilities for the evaluation data set using my binary logistic regression model.

Read dataset

```
train_data <- read.csv("https://raw.githubusercontent.com/JennierJ/DATA621/master/Homework3/crime-trai
eval_data <- read.csv("https://raw.githubusercontent.com/JennierJ/DATA621/master/Homework3/crime-evalua

head(train_data)
```

```
##   zn indus chas   nox    rm   age    dis rad tax ptratio  black lstat medv
## 1  0 19.58    0 0.605 7.929  96.2 2.0459   5 403    14.7 369.30  3.70 50.0
## 2  0 19.58    1 0.871 5.403 100.0 1.3216   5 403    14.7 396.90 26.82 13.4
## 3  0 18.10    0 0.740 6.485 100.0 1.9784  24 666    20.2 386.73 18.85 15.4
## 4 30  4.93    0 0.428 6.393   7.8 7.0355   6 300    16.6 374.71  5.19 23.7
## 5  0  2.46    0 0.488 7.155  92.2 2.7006   3 193    17.8 394.12  4.82 37.9
## 6  0  8.56    0 0.520 6.781  71.3 2.8561   5 384    20.9 395.58  7.67 26.5
##   target
## 1      1
## 2      1
## 3      1
## 4      0
## 5      0
## 6      0
```

1. Data Exploration & Data Visulization

```
dim(train_data)
```

```
## [1] 466  14
```

```
summary(train_data)
```

```
##        zn             indus             chas              nox
##  Min.   :  0.00   Min.   : 0.460   Min.   :0.00000   Min.   :0.3890
##  1st Qu.:  0.00   1st Qu.: 5.145   1st Qu.:0.00000   1st Qu.:0.4480
##  Median :  0.00   Median : 9.690   Median :0.00000   Median :0.5380
##  Mean   : 11.58   Mean   :11.105   Mean   :0.07082   Mean   :0.5543
##  3rd Qu.: 16.25   3rd Qu.:18.100   3rd Qu.:0.00000   3rd Qu.:0.6240
##  Max.   :100.00   Max.   :27.740   Max.   :1.00000   Max.   :0.8710
##        rm             age              dis               rad
##  Min.   :3.863   Min.   :  2.90   Min.   : 1.130   Min.   : 1.00
##  1st Qu.:5.887   1st Qu.: 43.88   1st Qu.: 2.101   1st Qu.: 4.00
##  Median :6.210   Median : 77.15   Median : 3.191   Median : 5.00
##  Mean   :6.291   Mean   : 68.37   Mean   : 3.796   Mean   : 9.53
##  3rd Qu.:6.630   3rd Qu.: 94.10   3rd Qu.: 5.215   3rd Qu.:24.00
##  Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.00
##       tax           ptratio         black            lstat
##  Min.   :187.0   Min.   :12.6   Min.   :  0.32   Min.   : 1.730
##  1st Qu.:281.0   1st Qu.:16.9   1st Qu.:375.61   1st Qu.: 7.043
##  Median :334.5   Median :18.9   Median :391.34   Median :11.350
##  Mean   :409.5   Mean   :18.4   Mean   :357.12   Mean   :12.631
##  3rd Qu.:666.0   3rd Qu.:20.2   3rd Qu.:396.24   3rd Qu.:16.930
##  Max.   :711.0   Max.   :22.0   Max.   :396.90   Max.   :37.970
##       medv           target
##  Min.   : 5.00   Min.   :0.0000
##  1st Qu.:17.02   1st Qu.:0.0000
##  Median :21.20   Median :0.0000
##  Mean   :22.59   Mean   :0.4914
##  3rd Qu.:25.00   3rd Qu.:1.0000
##  Max.   :50.00   Max.   :1.0000
```

The training dataset has 466 rows with 13 variables, with no NA values.

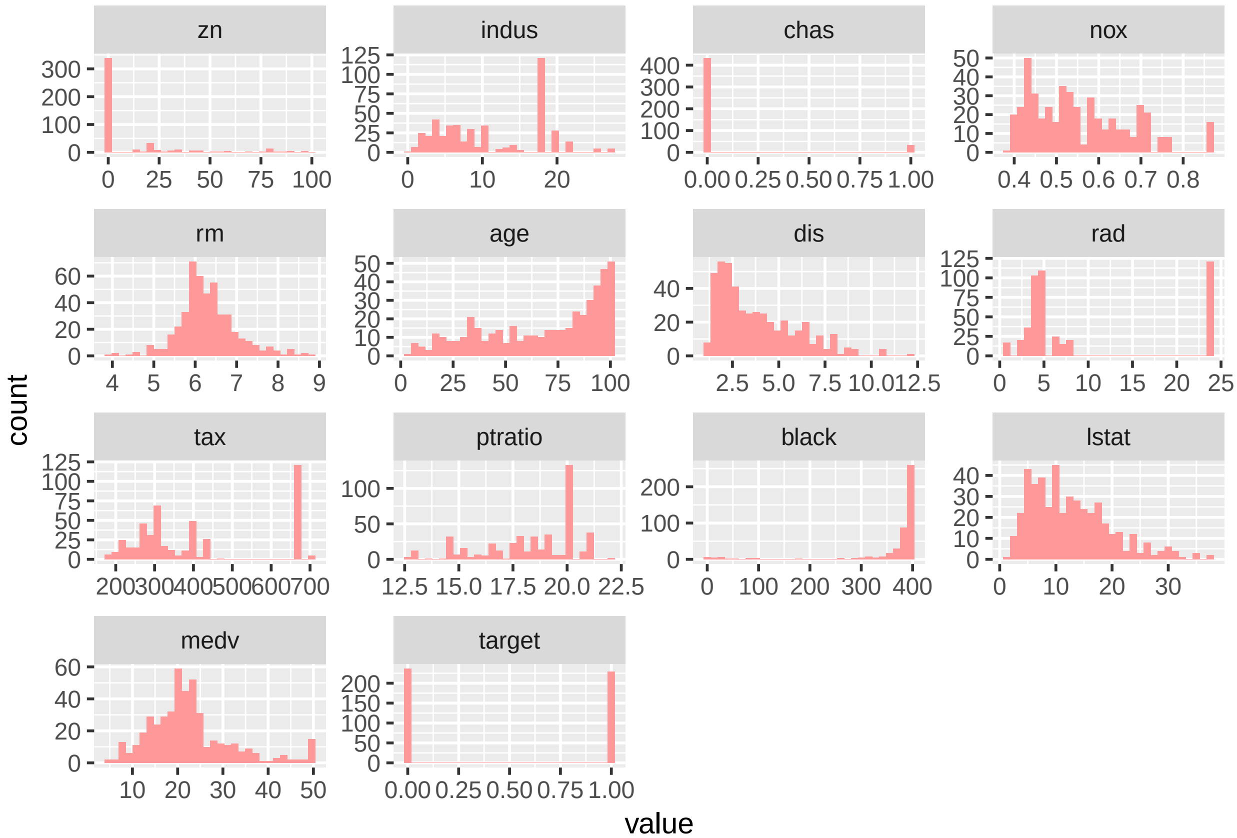
1.1. Histograms

```
v <- melt(train_data)
```

```
## No id variables; using all as measure variables
```

```
ggplot(v, aes(value)) + geom_histogram(fill = "#FF9999") + facet_wrap(~variable,
                                                                      scales = "free")
```

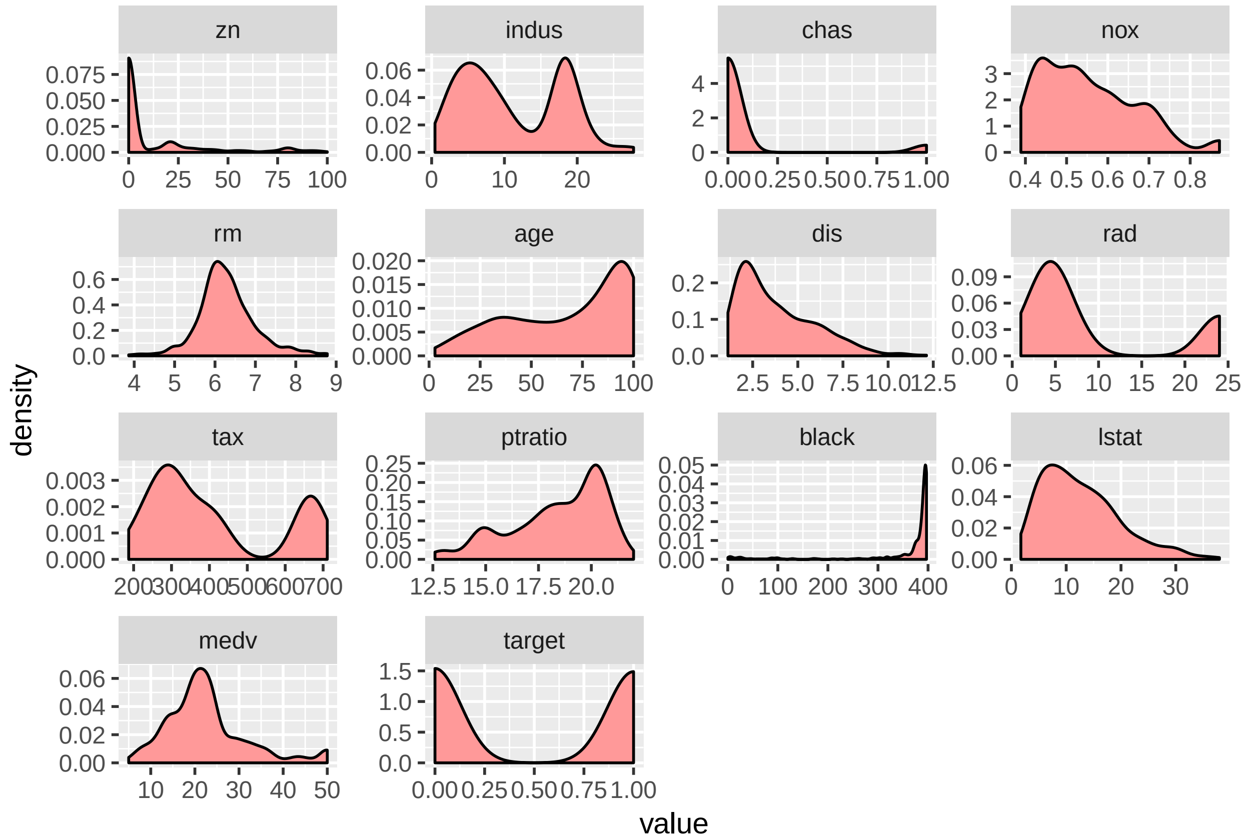
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



From the histograms above, it looks like that the variable zn, dis and age are skewed.

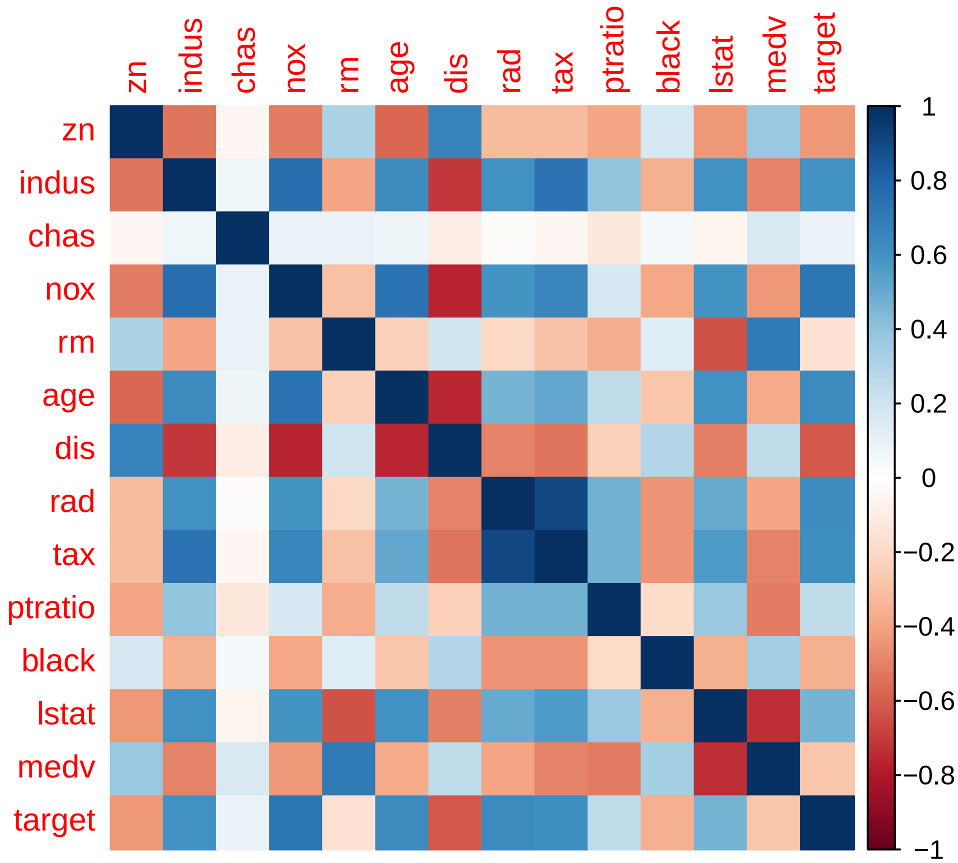
1.2 Density

```
ggplot(v, aes(value)) + geom_density(fill = "#FF9999") + facet_wrap(~variable,
                                                          scales = "free")
```



1.3 Correlation

```r
corrplot(cor(train_data), method = "color")
```



The correlation plot has shown that how variable in the dataset are related to each other.

2. Data Preparation

2.1. Data Transformation

I would like to do log transformation to medv and rad.

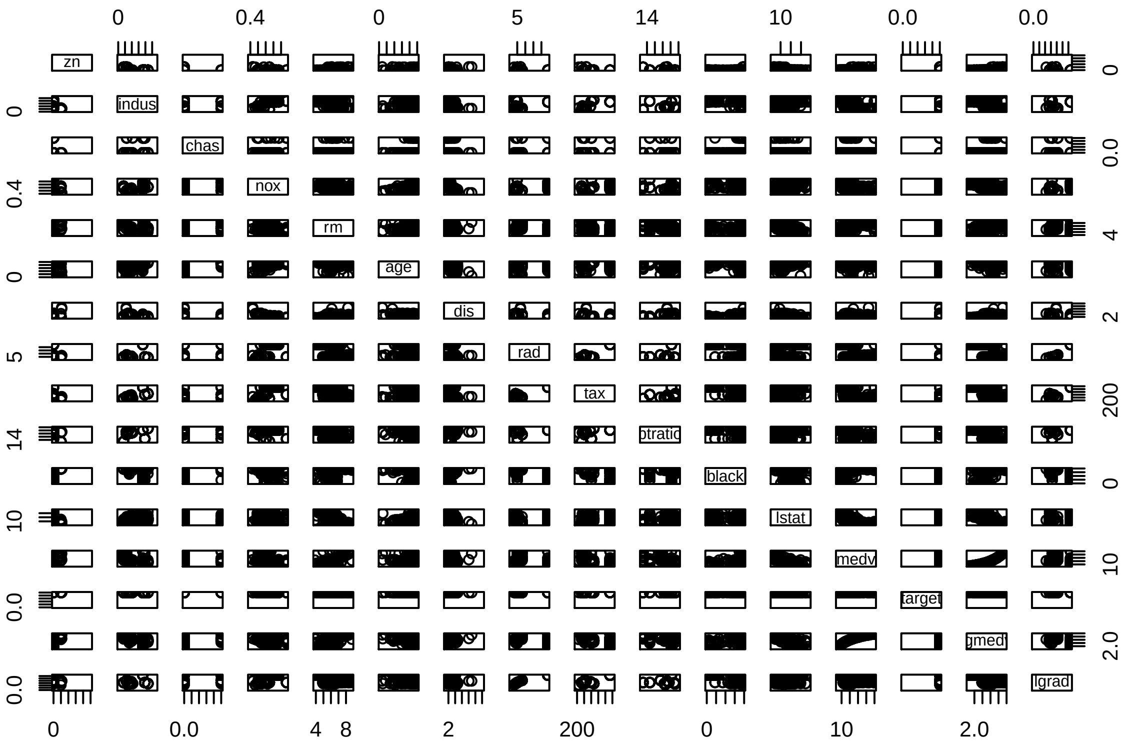
```
train_data$lgmedv <- log(train_data$medv)
train_data$lgrad <- log(train_data$rad)

head(train_data)
```

```
##   zn indus chas   nox    rm   age    dis rad tax ptratio  black lstat medv
## 1  0 19.58    0 0.605 7.929  96.2 2.0459   5 403    14.7 369.30  3.70 50.0
## 2  0 19.58    1 0.871 5.403 100.0 1.3216   5 403    14.7 396.90 26.82 13.4
## 3  0 18.10    0 0.740 6.485 100.0 1.9784  24 666    20.2 386.73 18.85 15.4
## 4 30  4.93    0 0.428 6.393   7.8 7.0355   6 300    16.6 374.71  5.19 23.7
## 5  0  2.46    0 0.488 7.155  92.2 2.7006   3 193    17.8 394.12  4.82 37.9
## 6  0  8.56    0 0.520 6.781  71.3 2.8561   5 384    20.9 395.58  7.67 26.5
##   target   lgmedv    lgrad
## 1      1 3.912023 1.609438
## 2      1 2.595255 1.609438
## 3      1 2.734368 3.178054
## 4      0 3.165475 1.791759
```

```
## 5       0 3.634951 1.098612
## 6       0 3.277145 1.609438
```

3. Build Model

```r
pairs(train_data, col = train_data$target)
```



I would like to use two methods to create models.

3.1 Logistic Regression Backward Selection

```r
glm.fit = glm(train_data$target ~ train_data$zn + train_data$indus + train_data$chas + train_data$nox
              + train_data$rm + train_data$age + train_data$dis + train_data$lgrad + train_data$tax +
                train_data$ptratio + train_data$black + train_data$lstat + train_data$lgmedv, data = tra
              family = binomial)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = train_data$target ~ train_data$zn + train_data$indus +
##     train_data$chas + train_data$nox + train_data$rm + train_data$age +
##     train_data$dis + train_data$lgrad + train_data$tax + train_data$ptratio +
##     train_data$black + train_data$lstat + train_data$lgmedv,
##     family = binomial, data = train_data)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5718  -0.1243  -0.0008   0.0671   3.4238
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -48.528140   9.439591  -5.141 2.73e-07 ***
## train_data$zn       -0.055872   0.032045  -1.744  0.08124 .
## train_data$indus    -0.039320   0.050844  -0.773  0.43932
## train_data$chas      0.830527   0.764727   1.086  0.27746
## train_data$nox      46.955469   7.912562   5.934 2.95e-09 ***
## train_data$rm        0.131597   0.623713   0.211  0.83290
## train_data$age       0.028696   0.013133   2.185  0.02888 *
## train_data$dis       0.648672   0.225915   2.871  0.00409 **
## train_data$lgrad     3.391947   0.771016   4.399 1.09e-05 ***
## train_data$tax      -0.007772   0.003508  -2.215  0.02674 *
## train_data$ptratio   0.441686   0.134305   3.289  0.00101 **
## train_data$black    -0.013830   0.007319  -1.890  0.05880 .
## train_data$lstat     0.071499   0.055665   1.284  0.19898
## train_data$lgmedv    3.823376   1.760816   2.171  0.02990 *
## ---
## 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: 189.17  on 452  degrees of freedom
## AIC: 217.17
##
## Number of Fisher Scoring iterations: 8
```

In the summary of model, I would like to remove the variable with a P value higher than 0.05, - variable rm

```r
glm.fit1 = glm(train_data$target ~ train_data$zn + train_data$indus + train_data$chas + train_data$nox
              +  train_data$age + train_data$dis + train_data$lgrad + train_data$tax +
                train_data$ptratio + train_data$black + train_data$lstat + train_data$lgmedv, data = tra
              family = binomial)
summary(glm.fit1)
```

```
##
## Call:
## glm(formula = train_data$target ~ train_data$zn + train_data$indus +
##     train_data$chas + train_data$nox + train_data$age + train_data$dis +
##     train_data$lgrad + train_data$tax + train_data$ptratio +
##     train_data$black + train_data$lstat + train_data$lgmedv,
##     family = binomial, data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5751  -0.1227  -0.0008   0.0678   3.4264
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)        -48.916647   9.311444  -5.253 1.49e-07 ***
## train_data$zn       -0.054929   0.031674  -1.734 0.082879 .
## train_data$indus    -0.040254   0.050722  -0.794 0.427415
## train_data$chas      0.822779   0.763423   1.078 0.281145
## train_data$nox      47.222154   7.839626   6.024 1.71e-09 ***
## train_data$age       0.030069   0.011446   2.627 0.008614 **
## train_data$dis       0.658434   0.221763   2.969 0.002987 **
## train_data$lgrad     3.407807   0.769793   4.427 9.56e-06 ***
## train_data$tax      -0.007741   0.003517  -2.201 0.027725 *
## train_data$ptratio   0.451348   0.126940   3.556 0.000377 ***
## train_data$black    -0.013966   0.007298  -1.914 0.055656 .
## train_data$lstat     0.068638   0.054105   1.269 0.204578
## train_data$lgmedv    4.085496   1.253737   3.259 0.001119 **
## ---
## 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: 189.21  on 453  degrees of freedom
## AIC: 215.21
##
## Number of Fisher Scoring iterations: 8
```

- variable indus

```
glm.fit2 = glm(train_data$target ~ train_data$zn + train_data$chas + train_data$nox
               +  train_data$age + train_data$dis + train_data$lgrad + train_data$tax +
                 train_data$ptratio + train_data$black + train_data$lstat + train_data$lgmedv, data = t
               family = binomial)
summary(glm.fit2)
```

```
##
## Call:
## glm(formula = train_data$target ~ train_data$zn + train_data$chas +
##     train_data$nox + train_data$age + train_data$dis + train_data$lgrad +
##     train_data$tax + train_data$ptratio + train_data$black +
##     train_data$lstat + train_data$lgmedv, family = binomial,
##     data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4950  -0.1260  -0.0007   0.0642   3.3997
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -47.459673   9.074063  -5.230 1.69e-07 ***
## train_data$zn       -0.054303   0.030963  -1.754  0.07947 .
## train_data$chas      0.676716   0.742654   0.911  0.36218
## train_data$nox      44.662991   6.973847   6.404 1.51e-10 ***
## train_data$age       0.029758   0.011328   2.627  0.00862 **
## train_data$dis       0.639033   0.219027   2.918  0.00353 **
## train_data$lgrad     3.660451   0.726530   5.038 4.70e-07 ***
## train_data$tax      -0.009000   0.003116  -2.888  0.00388 **
```

```
## train_data$ptratio   0.440650   0.126604   3.481  0.00050 ***
## train_data$black    -0.013482   0.007108  -1.897  0.05785 .
## train_data$lstat     0.065282   0.053675   1.216  0.22389
## train_data$lgmedv    3.982421   1.246465   3.195  0.00140 **
## ---
## 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: 189.86  on 454  degrees of freedom
## AIC: 213.86
##
## Number of Fisher Scoring iterations: 8
```

- variable chas

```
glm.fit3 = glm(train_data$target ~ train_data$zn + train_data$nox
               +  train_data$age + train_data$dis + train_data$lgrad + train_data$tax +
                 train_data$ptratio + train_data$black + train_data$lstat + train_data$lgmedv, data = tr
               family = binomial)
summary(glm.fit3)
```

```
##
## Call:
## glm(formula = train_data$target ~ train_data$zn + train_data$nox +
##     train_data$age + train_data$dis + train_data$lgrad + train_data$tax +
##     train_data$ptratio + train_data$black + train_data$lstat +
##     train_data$lgmedv, family = binomial, data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4745  -0.1274  -0.0005   0.0646   3.4076
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -47.124386   9.046356  -5.209 1.90e-07 ***
## train_data$zn       -0.059665   0.031116  -1.917 0.055177 .
## train_data$nox      43.877043   6.882902   6.375 1.83e-10 ***
## train_data$age       0.030488   0.011270   2.705 0.006825 **
## train_data$dis       0.626269   0.217594   2.878 0.004000 **
## train_data$lgrad     3.787325   0.729067   5.195 2.05e-07 ***
## train_data$tax      -0.009436   0.003149  -2.997 0.002729 **
## train_data$ptratio   0.421373   0.123778   3.404 0.000663 ***
## train_data$black    -0.013292   0.007070  -1.880 0.060099 .
## train_data$lstat     0.074672   0.052362   1.426 0.153847
## train_data$lgmedv    4.066484   1.243289   3.271 0.001073 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 190.71  on 455  degrees of freedom
```

```
## AIC: 212.71
##
## Number of Fisher Scoring iterations: 8
```

- variable zn

```
glm.fit4 = glm(train_data$target ~ train_data$nox
               +  train_data$age + train_data$dis + train_data$lgrad + train_data$tax +
                 train_data$ptratio + train_data$black + train_data$lstat + train_data$lgmedv, data = t
               family = binomial)
summary(glm.fit4)
```

```
##
## Call:
## glm(formula = train_data$target ~ train_data$nox + train_data$age +
##     train_data$dis + train_data$lgrad + train_data$tax + train_data$ptratio +
##     train_data$black + train_data$lstat + train_data$lgmedv,
##     family = binomial, data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.60483  -0.18125  -0.00248   0.05699   3.12228
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -45.638033   8.857960  -5.152 2.57e-07 ***
## train_data$nox      43.755818   6.911380   6.331 2.44e-10 ***
## train_data$age       0.029892   0.011115   2.689 0.007158 **
## train_data$dis       0.428557   0.178563   2.400 0.016394 *
## train_data$lgrad     3.735967   0.687829   5.432 5.59e-08 ***
## train_data$tax      -0.009725   0.002947  -3.300 0.000967 ***
## train_data$ptratio   0.475082   0.121170   3.921 8.83e-05 ***
## train_data$black    -0.013822   0.007300  -1.893 0.058302 .
## train_data$lstat     0.065994   0.053095   1.243 0.213884
## train_data$lgmedv    3.625056   1.216385   2.980 0.002881 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 195.50  on 456  degrees of freedom
## AIC: 215.5
##
## Number of Fisher Scoring iterations: 8
```

- variable black

```
glm.fit5 = glm(train_data$target ~ train_data$nox
               +  train_data$age + train_data$dis + train_data$lgrad + train_data$tax +
                 train_data$ptratio + train_data$lstat + train_data$lgmedv, data = train_data,
               family = binomial)
summary(glm.fit5)
```

```
##
## Call:
## glm(formula = train_data$target ~ train_data$nox + train_data$age +
##     train_data$dis + train_data$lgrad + train_data$tax + train_data$ptratio +
##     train_data$lstat + train_data$lgmedv, family = binomial,
##     data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.07722  -0.18052  -0.00237   0.07599   3.11297
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -48.422975   8.481067  -5.710 1.13e-08 ***
## train_data$nox      43.923656   6.895490   6.370 1.89e-10 ***
## train_data$age       0.029837   0.011201   2.664 0.007727 **
## train_data$dis       0.394632   0.177807   2.219 0.026457 *
## train_data$lgrad     3.782159   0.664560   5.691 1.26e-08 ***
## train_data$tax      -0.009417   0.002844  -3.311 0.000930 ***
## train_data$ptratio   0.440743   0.116615   3.779 0.000157 ***
## train_data$lstat     0.053130   0.053254   0.998 0.318441
## train_data$lgmedv    3.035936   1.161650   2.613 0.008963 **
## ---
## 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: 201.51  on 457  degrees of freedom
## AIC: 219.51
##
## Number of Fisher Scoring iterations: 8
```

- variable lstat

```
glm.fit6 = glm(target ~ nox
               +  age + dis + lgrad + tax +
                 ptratio + lgmedv, data = train_data,
               family = binomial)
summary(glm.fit6)
```

```
##
## Call:
## glm(formula = target ~ nox + age + dis + lgrad + tax + ptratio +
##     lgmedv, family = binomial, data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.02369  -0.18216  -0.00268   0.09140   3.13756
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -45.173968   7.682267  -5.880  4.1e-09 ***
## nox          43.568175   6.822642   6.386  1.7e-10 ***
```

```
## age          0.032602   0.010887   2.995 0.002748 **
## dis          0.383970   0.177580   2.162 0.030600 *
## lgrad        3.739570   0.654334   5.715  1.1e-08 ***
## tax         -0.009386   0.002837  -3.309 0.000936 ***
## ptratio      0.421167   0.114560   3.676 0.000237 ***
## lgmedv       2.335450   0.918850   2.542 0.011031 *
## ---
## 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: 202.51  on 458  degrees of freedom
## AIC: 218.51
##
## Number of Fisher Scoring iterations: 8
```

Now that the model glm.fit 6 are having variables left with P value lower than 0.05.

3.2 Lead Model Selection

The leaps packages is helping to generate all subset regression models.

```
regfit.full = regsubsets(target~., data = train_data, nvmax = 15)
reg.summary = summary(regfit.full)
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(target ~ ., data = train_data, nvmax = 15)
## 15 Variables  (and intercept)
##         Forced in Forced out
## zn          FALSE      FALSE
## indus       FALSE      FALSE
## chas        FALSE      FALSE
## nox         FALSE      FALSE
## rm          FALSE      FALSE
## age         FALSE      FALSE
## dis         FALSE      FALSE
## rad         FALSE      FALSE
## tax         FALSE      FALSE
## ptratio     FALSE      FALSE
## black       FALSE      FALSE
## lstat       FALSE      FALSE
## medv        FALSE      FALSE
## lgmedv      FALSE      FALSE
## lgrad       FALSE      FALSE
## 1 subsets of each size up to 15
## Selection Algorithm: exhaustive
##           zn  indus chas nox rm  age dis rad tax ptratio black lstat medv
## 1  ( 1 )  " " " "   " "  "*" " " " " " " " " " " " "     " "   " "   " "
## 2  ( 1 )  " " " "   " "  "*" " " " " " " " " " " " "     " "   " "   " "
## 3  ( 1 )  " " " "   " "  "*" " " "*" " " " " " " " "     " "   " "   " "
## 4  ( 1 )  " " " "   " "  "*" " " "*" " " " " " " " "     " "   " "   "*"
```

```
## 5  ( 1 )  " " " "   " "  "*" " " "*" " " " " " " "*"     " "   " "   "*"
## 6  ( 1 )  " " " "   " "  "*" " " "*" " " " " " " "*"     "*"   " "   "*"
## 7  ( 1 )  " " " "   " "  "*" " " "*" " " " " " " "*"     "*"   "*"   "*"
## 8  ( 1 )  " " " "   " "  "*" "*" "*" " " " " " " "*"     "*"   "*"   "*"
## 9  ( 1 )  " " " "   " "  "*" "*" "*" " " " " "*" "*"     "*"   "*"   "*"
## 10  ( 1 ) "*" " "   " "  "*" "*" "*" "*" " " " " "*"     "*"   "*"   "*"
## 11  ( 1 ) " " "*"   " "  "*" "*" "*" " " "*" "*" "*"     "*"   "*"   "*"
## 12  ( 1 ) " " "*"   " "  "*" "*" "*" "*" "*" "*" "*"     "*"   "*"   "*"
## 13  ( 1 ) "*" "*"   " "  "*" "*" "*" "*" "*" "*" "*"     "*"   "*"   "*"
## 14  ( 1 ) "*" "*"   "*"  "*" "*" "*" "*" "*" "*" "*"     "*"   "*"   "*"
## 15  ( 1 ) "*" "*"   "*"  "*" "*" "*" "*" "*" "*" "*"     "*"   "*"   "*"
##           lgmedv lgrad
## 1  ( 1 )  " "    " "
## 2  ( 1 )  " "    "*"
## 3  ( 1 )  " "    "*"
## 4  ( 1 )  " "    "*"
## 5  ( 1 )  " "    "*"
## 6  ( 1 )  " "    "*"
## 7  ( 1 )  " "    "*"
## 8  ( 1 )  " "    "*"
## 9  ( 1 )  " "    "*"
## 10  ( 1 ) " "    "*"
## 11  ( 1 ) " "    "*"
## 12  ( 1 ) " "    "*"
## 13  ( 1 ) " "    "*"
## 14  ( 1 ) " "    "*"
## 15  ( 1 ) "*"    "*"
```

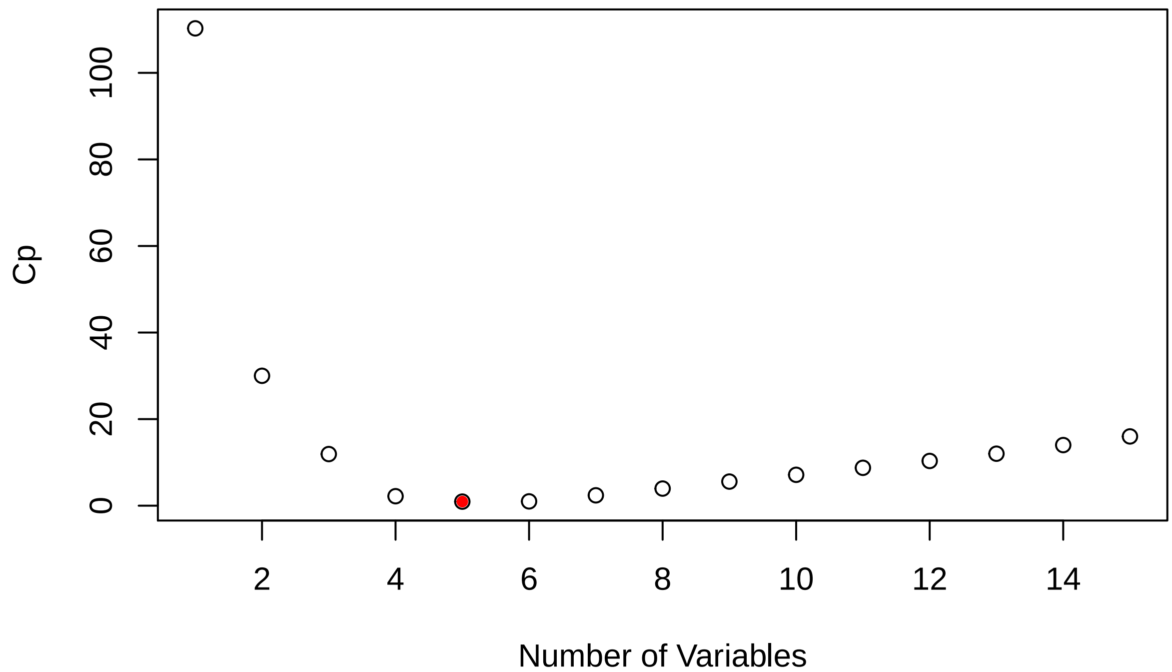
```
names(reg.summary)
```

```
## [1] "which"  "rsq"      "rss"      "adjr2"    "cp"       "bic"      "outmat"   "obj"
```

```
plot(reg.summary$cp, xlab = "Number of Variables", ylab = "Cp")
which.min(reg.summary$cp)
```

```
## [1] 5
```

```
points(5, reg.summary$cp[5], pch = 20, col = "red")
```




```r
glm.fit7 = glm(train_data$target ~ train_data$nox
               +  train_data$age + train_data$ptratio + train_data$medv + train_data$lgrad, data = trai
               family = binomial)
summary(glm.fit7)
```

```
##
## Call:
## glm(formula = train_data$target ~ train_data$nox + train_data$age +
##     train_data$ptratio + train_data$medv + train_data$lgrad,
##     family = binomial, data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.03040  -0.27870  -0.01536   0.08021   2.86196
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -27.254311   3.773829  -7.222 5.13e-13 ***
## train_data$nox      25.669907   4.098389   6.263 3.77e-10 ***
## train_data$age       0.019977   0.009248   2.160 0.030751 *
## train_data$ptratio   0.307056   0.100031   3.070 0.002143 **
## train_data$medv      0.094301   0.028525   3.306 0.000947 ***
## train_data$lgrad     2.478406   0.495238   5.004 5.60e-07 ***
## ---
## 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: 222.88  on 460  degrees of freedom
## AIC: 234.88
##
## Number of Fisher Scoring iterations: 7
```

The model glm.fit7 is the "best" model that with lowest CP and picked by leap pacakge

4. Model Selection

Let's see the factor of model glm.fit6

```
summary(glm.fit6)
```

```
##
## Call:
## glm(formula = target ~ nox + age + dis + lgrad + tax + ptratio +
##     lgmedv, family = binomial, data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.02369  -0.18216  -0.00268   0.09140   3.13756
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -45.173968   7.682267  -5.880  4.1e-09 ***
## nox          43.568175   6.822642   6.386  1.7e-10 ***
## age           0.032602   0.010887   2.995 0.002748 **
## dis           0.383970   0.177580   2.162 0.030600 *
## lgrad         3.739570   0.654334   5.715  1.1e-08 ***
## tax          -0.009386   0.002837  -3.309 0.000936 ***
## ptratio       0.421167   0.114560   3.676 0.000237 ***
## lgmedv        2.335450   0.918850   2.542 0.011031 *
## ---
## 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: 202.51  on 458  degrees of freedom
## AIC: 218.51
##
## Number of Fisher Scoring iterations: 8
```

```
glm.probs = predict(glm.fit6, data = train_data, type = "response")

Matrix6 <- confusionMatrix(data = factor(ifelse(glm.probs > 0.5, 1, 0)), reference = factor(train_data$t
                           positive = "1")
Matrix6
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction   0   1
##          0 219  24
##          1  18 205
##
##                Accuracy : 0.9099
##                  95% CI : (0.8801, 0.9343)
##     No Information Rate : 0.5086
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8196
##  Mcnemar's Test P-Value : 0.4404
##
##             Sensitivity : 0.8952
##             Specificity : 0.9241
##          Pos Pred Value : 0.9193
##          Neg Pred Value : 0.9012
##              Prevalence : 0.4914
##          Detection Rate : 0.4399
##    Detection Prevalence : 0.4785
##       Balanced Accuracy : 0.9096
##
##        'Positive' Class : 1
##
```

And the factor of model glm.fit7

```
summary(glm.fit7)
```

```
##
## Call:
## glm(formula = train_data$target ~ train_data$nox + train_data$age +
##     train_data$ptratio + train_data$medv + train_data$lgrad,
##     family = binomial, data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.03040  -0.27870  -0.01536   0.08021   2.86196
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -27.254311   3.773829  -7.222 5.13e-13 ***
## train_data$nox      25.669907   4.098389   6.263 3.77e-10 ***
## train_data$age       0.019977   0.009248   2.160 0.030751 *
## train_data$ptratio   0.307056   0.100031   3.070 0.002143 **
## train_data$medv      0.094301   0.028525   3.306 0.000947 ***
## train_data$lgrad     2.478406   0.495238   5.004 5.60e-07 ***
## ---
## 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: 222.88  on 460  degrees of freedom
## AIC: 234.88
##
## Number of Fisher Scoring iterations: 7
```

```
glm.probs = predict(glm.fit7, data = train_data, type = "response")

Matrix7 <- confusionMatrix(data = factor(ifelse(glm.probs > 0.5, 1, 0)), reference = factor(train_data$t
                           positive = "1")
Matrix7
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 209  36
##          1  28 193
##
##                Accuracy : 0.8627
##                  95% CI : (0.828, 0.8926)
##     No Information Rate : 0.5086
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7251
##  Mcnemar's Test P-Value : 0.3816
##
##             Sensitivity : 0.8428
##             Specificity : 0.8819
##          Pos Pred Value : 0.8733
##          Neg Pred Value : 0.8531
##              Prevalence : 0.4914
##          Detection Rate : 0.4142
##    Detection Prevalence : 0.4742
##       Balanced Accuracy : 0.8623
##
##        'Positive' Class : 1
##
```

Using model glm.fit6 to predict the target with evaluation_data

```
eval_data$lgrad <- log(eval_data$rad)
eval_data$lgmedv <- log(eval_data$medv)
glm.probs.pred = predict(glm.fit6, newdata = eval_data, type = "response")
predtarget <- ifelse(glm.probs.pred > 0.5, 1, 0)
eval_data$pred <- predtarget
eval_data
```

```
##    zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
## 1   0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 2   0  8.14    0 0.538 6.096 84.5 4.4619   4 307    21.0 380.02 10.26
## 3   0  8.14    0 0.538 6.495 94.4 4.4547   4 307    21.0 387.94 12.80
## 4   0  8.14    0 0.538 5.950 82.0 3.9900   4 307    21.0 232.60 27.71
## 5   0  5.96    0 0.499 5.850 41.5 3.9342   5 279    19.2 396.90  8.77
```

```
## 6   25  5.13    0 0.453 5.741  66.2 7.2254   8 284    19.7 395.11 13.15
## 7   25  5.13    0 0.453 5.966  93.4 6.8185   8 284    19.7 378.08 14.44
## 8    0  4.49    0 0.449 6.630  56.1 4.4377   3 247    18.5 392.30  6.53
## 9    0  4.49    0 0.449 6.121  56.8 3.7476   3 247    18.5 395.15  8.44
## 10   0  2.89    0 0.445 6.163  69.6 3.4952   2 276    18.0 391.83 11.34
## 11   0 25.65    0 0.581 5.856  97.0 1.9444   2 188    19.1 370.31 25.41
## 12   0 25.65    0 0.581 5.613  95.6 1.7572   2 188    19.1 359.29 27.26
## 13   0 21.89    0 0.624 5.637  94.7 1.9799   4 437    21.2 396.90 18.34
## 14   0 19.58    0 0.605 6.101  93.0 2.2834   5 403    14.7 240.16  9.81
## 15   0 19.58    0 0.605 5.880  97.3 2.3887   5 403    14.7 348.13 12.03
## 16   0 10.59    1 0.489 5.960  92.1 3.8771   4 277    18.6 393.25 17.27
## 17   0  6.20    0 0.504 6.552  21.4 3.3751   8 307    17.4 380.34  3.76
## 18   0  6.20    0 0.507 8.247  70.4 3.6519   8 307    17.4 378.95  3.95
## 19 22  5.86    0 0.431 6.957   6.8 8.9067   7 330    19.1 386.09  3.53
## 20 90  2.97    0 0.400 7.088  20.8 7.3073   1 285    15.3 394.72  7.85
## 21 80  1.76    0 0.385 6.230  31.5 9.0892   1 241    18.2 341.60 12.93
## 22 33  2.18    0 0.472 6.616  58.1 3.3700   7 222    18.4 393.36  8.93
## 23   0  9.90    0 0.544 6.122  52.8 2.6403   4 304    18.4 396.90  5.98
## 24   0  7.38    0 0.493 6.415  40.1 4.7211   5 287    19.6 396.90  6.12
## 25   0  7.38    0 0.493 6.312  28.9 5.4159   5 287    19.6 396.90  6.15
## 26   0  5.19    0 0.515 5.895  59.6 5.6150   5 224    20.2 394.81 10.56
## 27 80  2.01    0 0.435 6.635  29.7 8.3440   4 280    17.0 390.94  5.99
## 28   0 18.10    0 0.718 3.561  87.9 1.6132  24 666    20.2 354.70  7.12
## 29   0 18.10    1 0.631 7.016  97.5 1.2024  24 666    20.2 392.05  2.96
## 30   0 18.10    0 0.584 6.348  86.1 2.0527  24 666    20.2  83.45 17.64
## 31   0 18.10    0 0.740 5.935  87.9 1.8206  24 666    20.2  68.95 34.02
## 32   0 18.10    0 0.740 5.627  93.9 1.8172  24 666    20.2 396.90 22.88
## 33   0 18.10    0 0.740 5.818  92.4 1.8662  24 666    20.2 391.45 22.11
## 34   0 18.10    0 0.740 6.219 100.0 2.0048  24 666    20.2 395.69 16.59
## 35   0 18.10    0 0.740 5.854  96.6 1.8956  24 666    20.2 240.52 23.79
## 36   0 18.10    0 0.713 6.525  86.5 2.4358  24 666    20.2  50.92 18.13
## 37   0 18.10    0 0.713 6.376  88.4 2.5671  24 666    20.2 391.43 14.65
## 38   0 18.10    0 0.655 6.209  65.4 2.9634  24 666    20.2 396.90 13.22
## 39   0  9.69    0 0.585 5.794  70.6 2.8927   6 391    19.2 396.90 14.10
## 40   0 11.93    0 0.573 6.976  91.0 2.1675   1 273    21.0 396.90  5.64
##    medv     lgrad   lgmedv pred
## 1  34.7 0.6931472 3.546740    0
## 2  18.2 1.3862944 2.901422    1
## 3  18.4 1.3862944 2.912351    1
## 4  13.2 1.3862944 2.580217    0
## 5  21.0 1.6094379 3.044522    0
## 6  18.7 2.0794415 2.928524    0
## 7  16.0 2.0794415 2.772589    1
## 8  26.6 1.0986123 3.280911    0
## 9  22.2 1.0986123 3.100092    0
## 10 21.4 0.6931472 3.063391    0
## 11 17.3 0.6931472 2.850707    0
## 12 15.7 0.6931472 2.753661    0
## 13 14.3 1.3862944 2.660260    1
## 14 25.0 1.6094379 3.218876    1
## 15 19.1 1.6094379 2.949688    1
## 16 21.7 1.3862944 3.077312    0
## 17 31.5 2.0794415 3.449988    0
## 18 48.3 2.0794415 3.877432    1
```

```
## 19 29.6 1.9459101 3.387774    0
## 20 32.2 0.0000000 3.471966    0
## 21 20.1 0.0000000 3.000720    0
## 22 28.4 1.9459101 3.346389    0
## 23 22.1 1.3862944 3.095578    0
## 24 25.0 1.6094379 3.218876    0
## 25 23.0 1.6094379 3.135494    0
## 26 18.5 1.6094379 2.917771    1
## 27 24.5 1.3862944 3.198673    0
## 28 27.5 3.1780538 3.314186    1
## 29 50.0 3.1780538 3.912023    1
## 30 14.5 3.1780538 2.674149    1
## 31  8.4 3.1780538 2.128232    1
## 32 12.8 3.1780538 2.549445    1
## 33 10.5 3.1780538 2.351375    1
## 34 18.4 3.1780538 2.912351    1
## 35 10.8 3.1780538 2.379546    1
## 36 14.1 3.1780538 2.646175    1
## 37 17.7 3.1780538 2.873565    1
## 38 21.4 3.1780538 3.063391    1
## 39 18.3 1.7917595 2.906901    1
## 40 23.9 0.0000000 3.173878    0
```

```
table(predtarget)
```

```
## predtarget
##  0  1
## 20 20
```

```
summary(glm.probs.pred)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000042 0.1394000 0.4815000 0.5164000 0.9894000 1.0000000
```