HW 3

Team 2

April 10, 2019

Contents

Overview 1
- Objective 2
- Dependencies 2

Data Exploration 2
- Summary Statistics 2
- Histogram 3
- Correlation 6

Data Preparation 7
- Transformations for Multicollinearity 7
- Log Transformations 8
- New Variables 10

Build Models 12
- MODEL 1 12
- MODEL 2 17
- MODEL 3 20
- MODEL 4 23
- MODEL 5 24

Select Models 26

Prediction 28

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

1. `zn`: proportion of residential land zoned for large lots (over 25000 square feet) (predictor variable)
2. `indus`: proportion of non-retail business acres per suburb (predictor variable)
3. `chas`: a dummy var. for whether the suburb borders the Charles River (1) or not (0) (predictor variable)
4. `nox`: nitrogen oxides concentration (parts per 10 million) (predictor variable)
5. `rm`: average number of rooms per dwelling (predictor variable)
6. `age`: proportion of owner-occupied units built prior to 1940 (predictor variable)
7. `dis`: weighted mean of distances to five Boston employment centers (predictor variable)
8. `rad`: index of accessibility to radial highways (predictor variable)
9. `tax`: full-value property-tax rate per $10,000 (predictor variable)
10. `ptratio`: pupil-teacher ratio by town (predictor variable)
11. `black`: 1000(Bk - 0.63)2 where Bk is the proportion of blacks by town (predictor variable)
12. `lstat`: lower status of the population (percent) (predictor variable)

13. `medv`: median value of owner-occupied homes in $1000s (predictor variable)
14. `target`: whether the crime rate is above the median crime rate (1) or not (0) (response variable)

Objective

Your objective is to build a binary logistic regression model on the training data set to predict whether the neighborhood will be at risk for high crime levels. You will provide classifications and probabilities for the evaluation data set using your binary logistic regression model. You can only use the variables given to you (or variables that you derive from the variables provided).

Dependencies

Replication of our work requires the following packages in Rstudio:

```r
#install.packages('corrplot')
#install.packages('randomForest')
#install.packages('olsrr')

require(ggplot2)
require(dplyr)
require(tidyr)
require(corrplot)
require(randomForest)
require(olsrr)
```

Data Exploration

First, we read the data as a csv then performed some simple statistical calculations so that we could explore the data. Below we can see a sample of the data output as it was read from the csv.

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

We can explore how many `NAs` are in each column to see if we need to impute any of the variables:

zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target
466	466	466	466	466	466	466	466	466	466	466	466	466

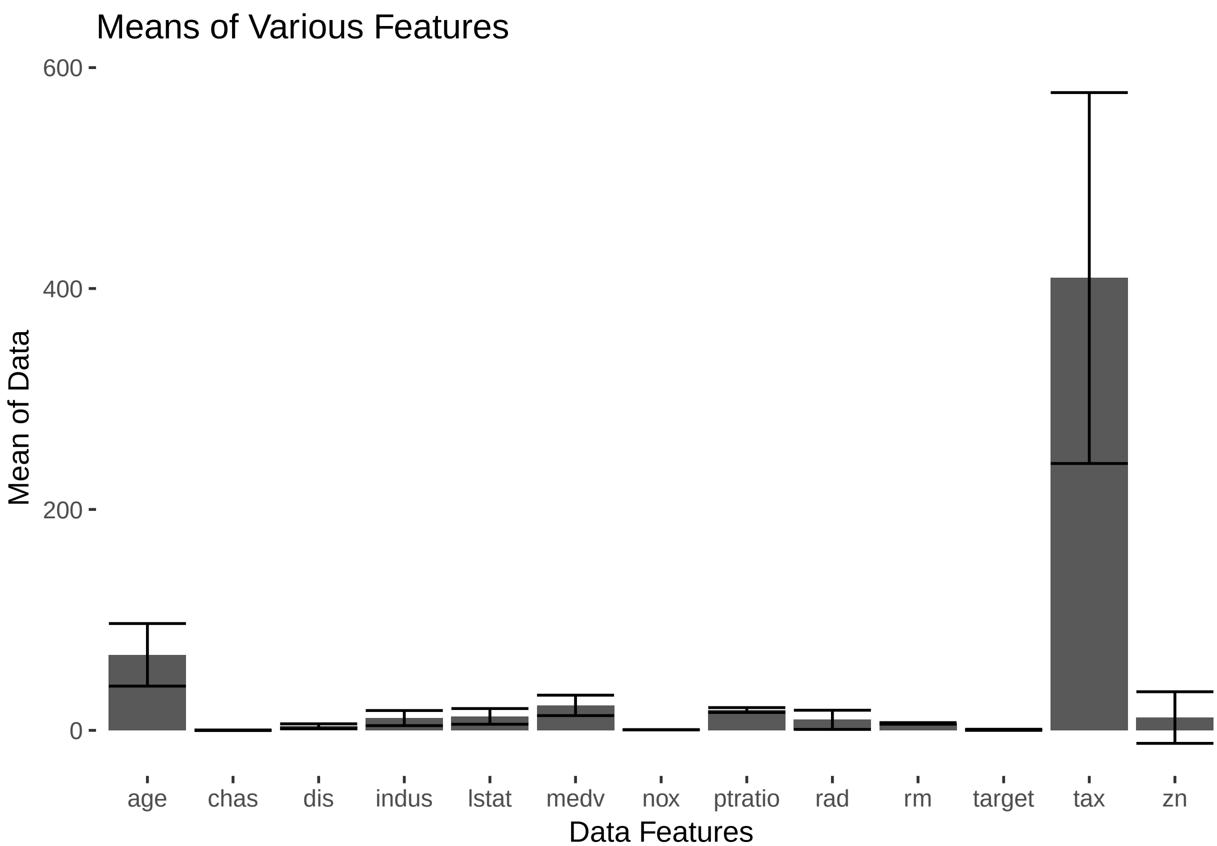
As we can see, each data vector has the same number of entries, 466. Thus, imputation will not be necessary.

Summary Statistics

We then calculated the mean and standard deviation for each data vector:

	means	sds
zn	11.5772532	23.3646511
indus	11.1050215	6.8458549
chas	0.0708155	0.2567920
nox	0.5543105	0.1166667
rm	6.2906738	0.7048513
age	68.3675966	28.3213784
dis	3.7956929	2.1069496
rad	9.5300429	8.6859272
tax	409.5021459	167.9000887
ptratio	18.3984979	2.1968447
lstat	12.6314592	7.1018907
medv	22.5892704	9.2396814
target	0.4914163	0.5004636

Below is a bar chart that illutrates the average and standard deviation for each of our data vectors. As we can see, the `tax` vector is a totally different magnitude than the rest. Models involving this vector will benefit from normalization or scaling.

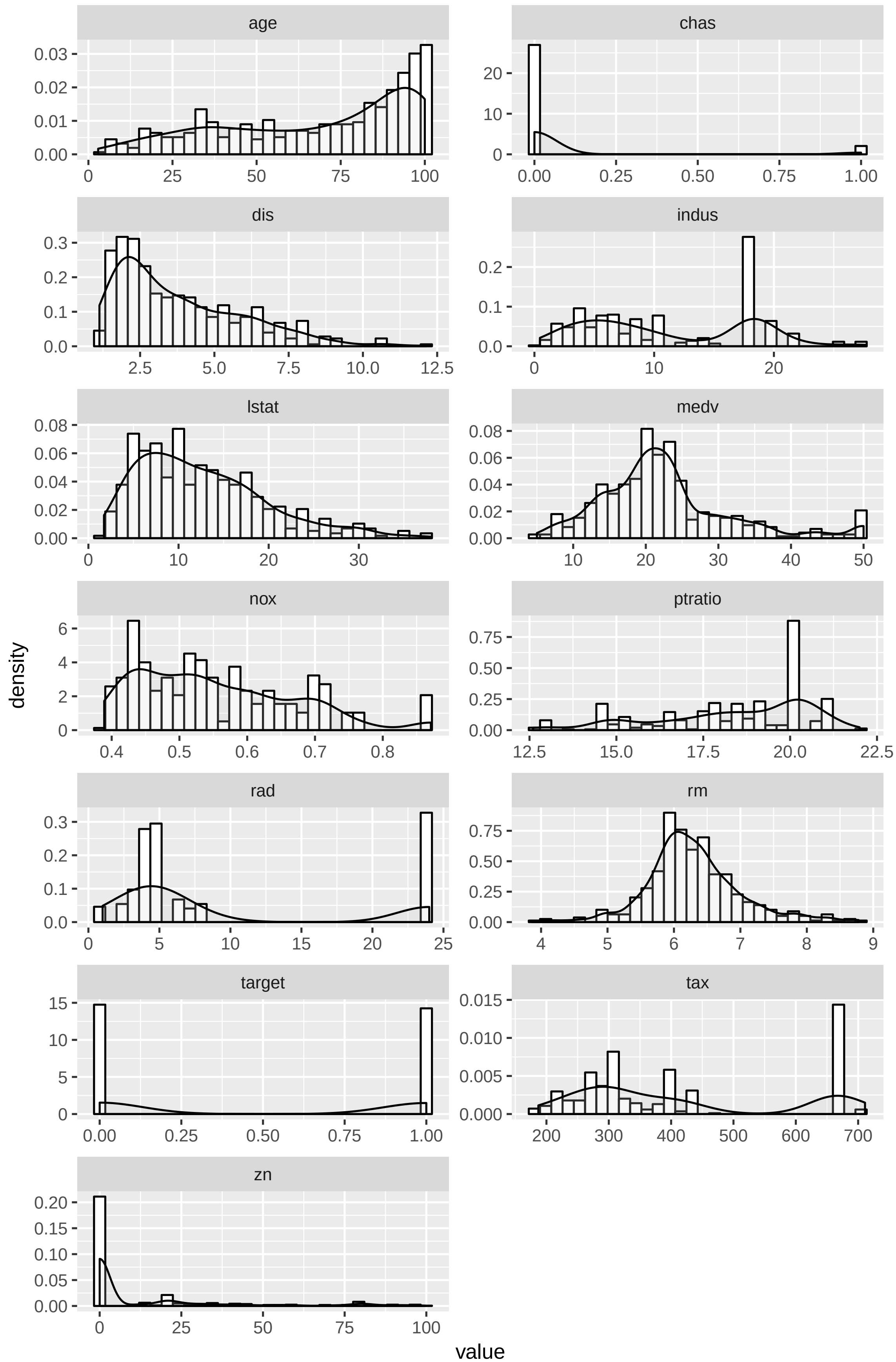


Histogram

The following histograms help visualize the spread and skewness of the raw data.

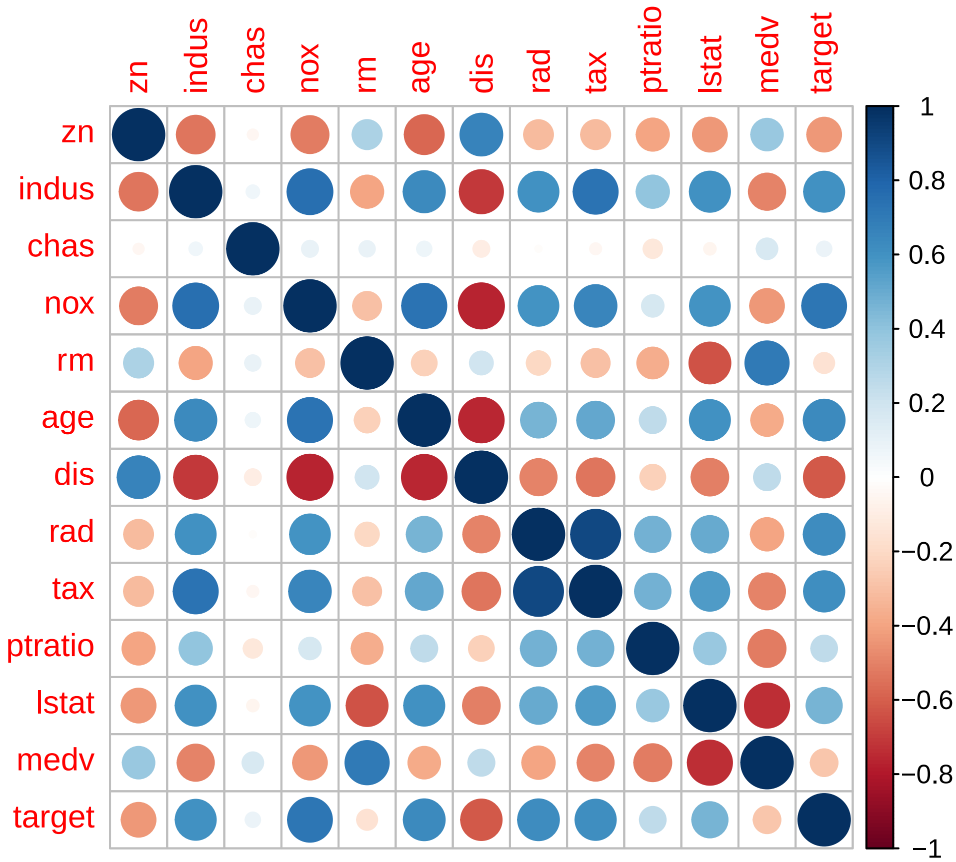
```
ggplot(data = gather(training), mapping = aes(x = value)) +
  geom_histogram(aes(y=..density..), colour="black", fill="white")+
  geom_density(alpha=.2, fill="lightgrey")+
  facet_wrap(~key, ncol = 2, scales = 'free')
```

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


age
chas
dis
indus
lstat
medv
nox
ptratio
rad
rm
target
tax
zn
density
value

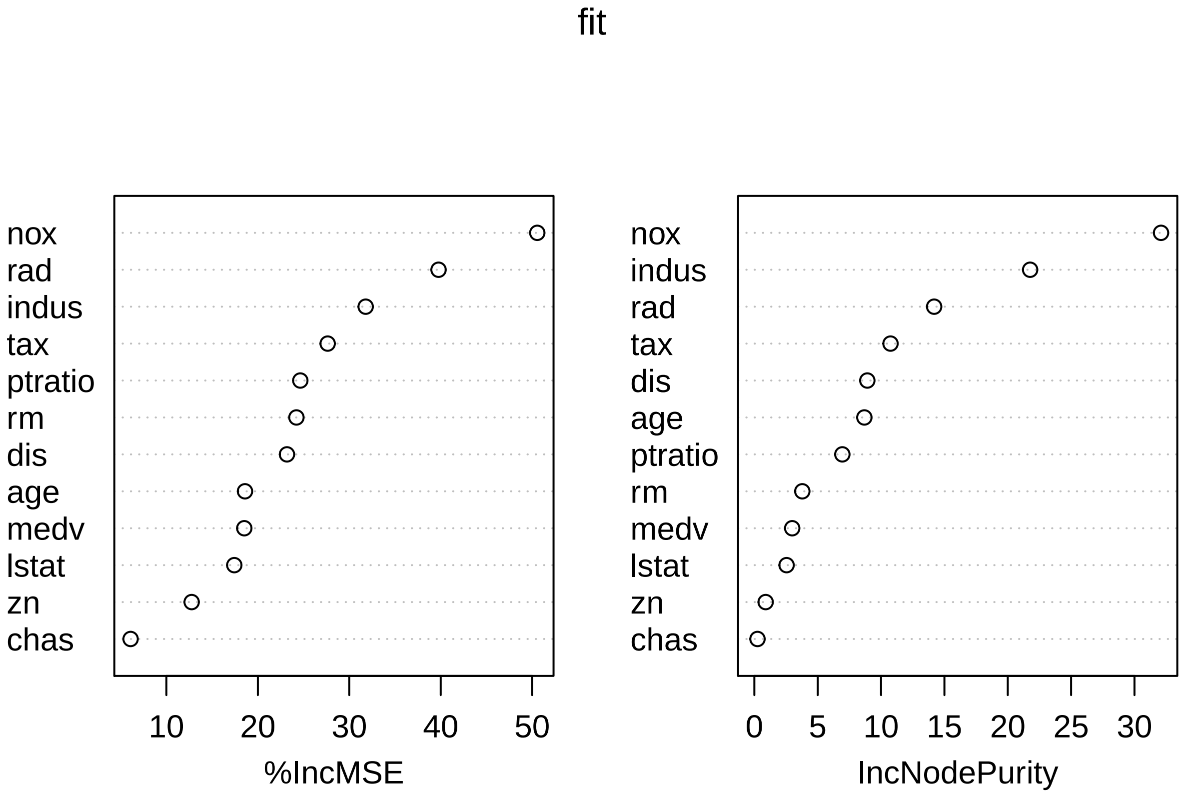
Correlation

We can see our correlation matrix below. A dark blue circle represents a strong positive relationship and a dark red circle represents a strong negative relationship between two variables. We can see that `indus`, `nox`, `target`, and `dis` have the most colinearity. Likewise, these vectors are the best predictors for the target value. Note that this plot only includes rows tha have data in each column.



Finally, we can use the `randomforest` package to verify our assumptions from the correlation plot.

```
## Warning in randomForest.default(training2, target, importance = TRUE, ntree
## = 1000): The response has five or fewer unique values. Are you sure you
## want to do regression?
```



We verified our assumptions above using 1000 random forests. The `nox`, `rad`, `indus`, and `tax` have the most effect. While `dis`is strongly colinear, it has less effect on the target. This is likely due to it encoding information stored redundantly in another vector.

Data Preparation

In the following section, we will prepare and transform our variables for our model:

Transformations for Multicollinearity

We saw some correlation between our predictor variables in our exploritory correlation plots. We can test this correlation using variance inflation factors (VIF) to ensure our model is not affected by multicollinearity.

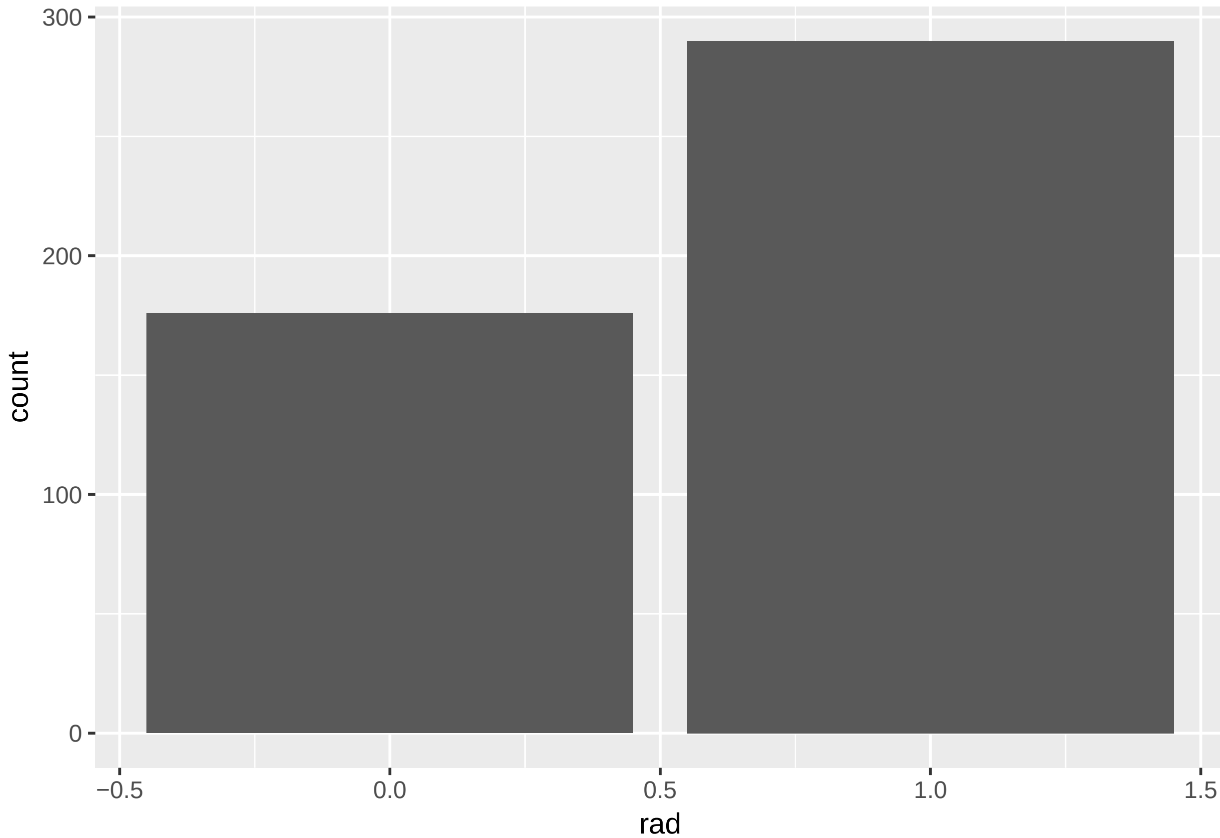
Variables	Tolerance	VIF	Standard_Error
rad	0.1474632	6.781354	2.604103
tax	0.1084925	9.217228	3.035989

This test shows us that the `rad` and `tax` variables have high multicollinearity above 5. Both variables should not be used together, without transformation in our model. The above table shows that as the standard error for both exceeds 2 times the amount then if these variables were not related.

Rad is an index variable that represents accessibility to radial highways. We choose to bifucate this data using the median value, 5.

Variables	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv
Tolerance	0.4137525	0.2483377	0.9221830	0.2210512	0.4298120	0.3213630	0.2354194	0.5836388	0.2531428	0.4765876	0.2799084	0.2775283
VIF	2.416904	4.026775	1.084383	4.523839	2.326598	3.111737	4.247738	1.713388	3.950339	2.098250	3.572597	3.603236
Standard_Error	1.554639	2.006683	1.041337	2.126932	1.525319	1.764012	2.061004	1.308965	1.987546	1.448534	1.890131	1.898219

```r
ggplot(training2, aes(x=rad))+geom_bar()
```



Through this change, the `tax` and `rad` variables are no longer affected by multicollinearity.

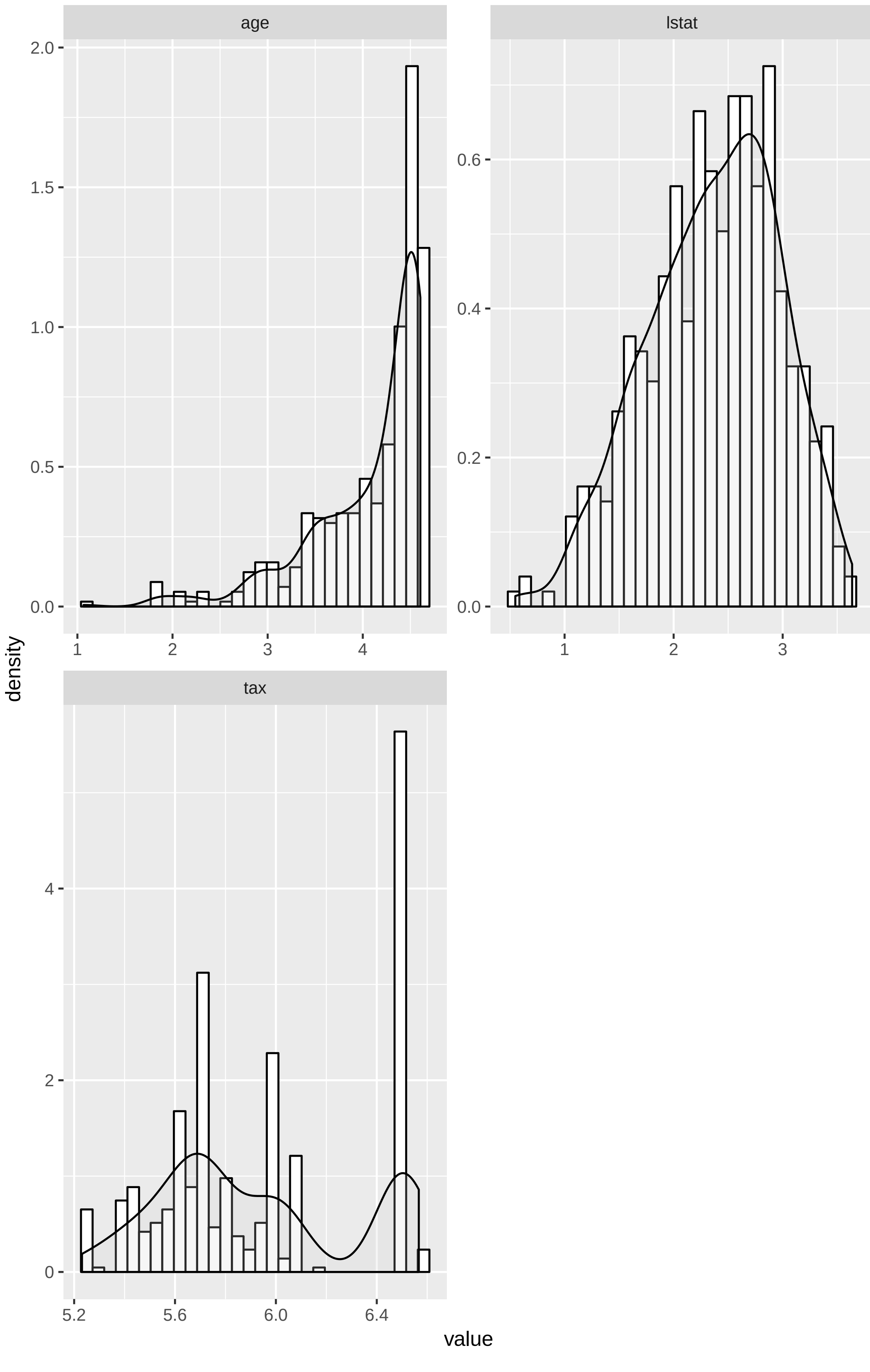
Log Transformations

While logistic modeling does not require normalized data, we choose to apply log transformations to adjust the scales for `age`, `lstat`, and `tax` so that the variables better fit our models.

```r
training2 <- training2 %>%
   mutate_at(.vars = vars(age, lstat, tax), .funs = log)

training2 %>% select(age, lstat,tax) %>% gather() %>% ggplot(mapping = aes(x = value)) +
  geom_histogram(aes(y=..density..), colour="black", fill="white")+
  geom_density(alpha=.2, fill="lightgrey")+
  facet_wrap(~key, ncol = 2, scales = 'free')
```

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


age
lstat
tax
density
value
2.0
1.5
1.0
0.5
0.0
0.6
0.4
0.2
0.0
1
2
3
4
1
2
3
4
2
0
5.2
5.6
6.0
6.4

This transformation helps center the `age` and normalize the `lstat` and `tax` variables.

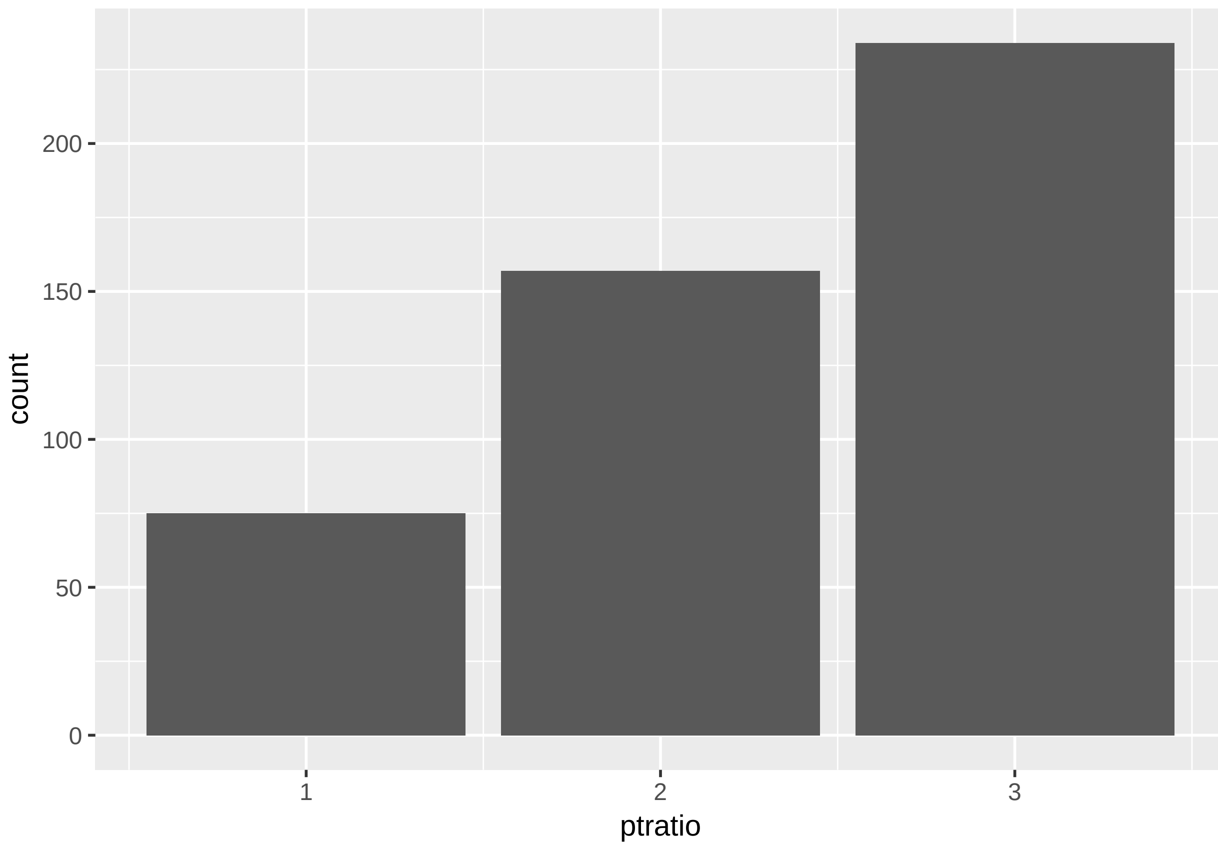
New Variables

We additionally chose to create several variables from our initial dataset.

`ptratio`

We first changed `ptratio`, a pupil-teacher ratio measurement, into a categorial variable. In the new variable, 0 represents small, 1 represents medium, and 3 represents large ratios.

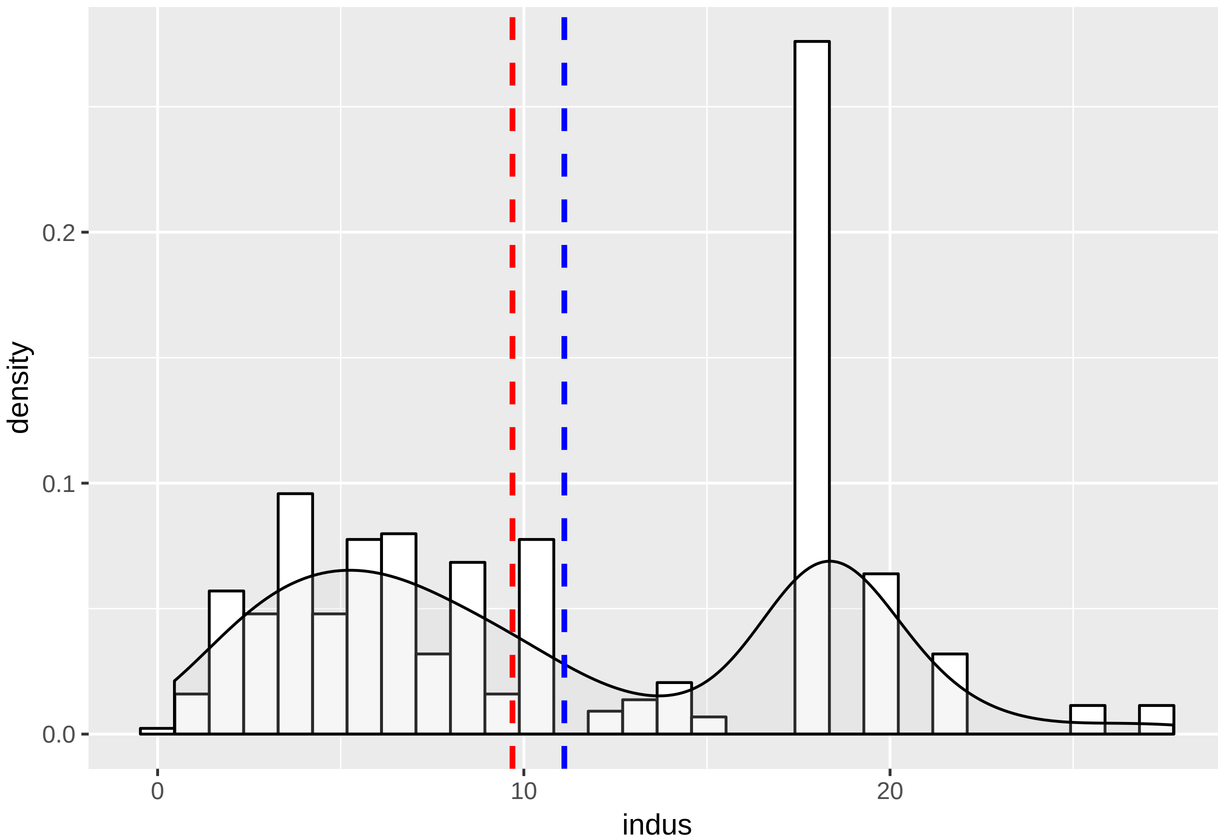
Our new variable for `ptratio` now looks like this:



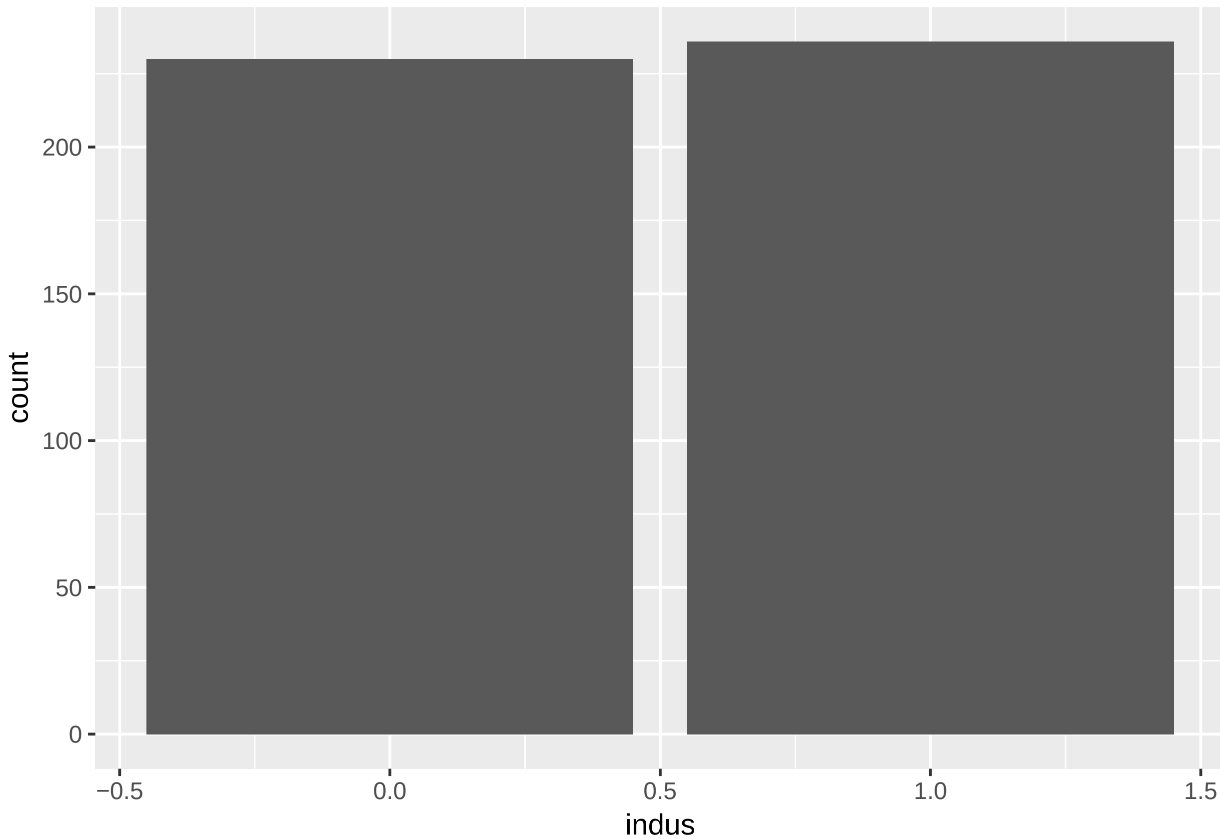
`indus`

This variable represents the proportion of non-retail business acres per suburb. The plots below show the `indus` data is bimodal, skewed right, and centered around 10. The red line shows the median, whereas the blue line depicts the mean value for this variable.

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



We choose to bifuncate this variable using its median value.



As a result of these transformations, our data now looks like this:

zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target
0	1	0	0.605	7.929	4.566429	2.0459	1	5.998937	1	1.308333	50.0	1
0	1	1	0.871	5.403	4.605170	1.3216	1	5.998937	1	3.289148	13.4	1
0	1	0	0.740	6.485	4.605170	1.9784	1	6.501290	3	2.936513	15.4	1
30	0	0	0.428	6.393	2.054124	7.0355	1	5.703782	2	1.646734	23.7	0
0	0	0	0.488	7.155	4.523960	2.7006	0	5.262690	2	1.572774	37.9	0
0	0	0	0.520	6.781	4.266896	2.8561	1	5.950643	3	2.037317	26.5	0

Build Models

Loading necessary packages for building models

MODEL 1

This is a basic model, we use all data without any transformations applied. Backward elimination method is used.

```
training$target = as.factor(training$target)
model_1<- step(glm(target~., data = training, family = 'binomial'), direction = "backward")
```

```
## Start:  AIC=218.05
## target ~ zn + indus + chas + nox + rm + age + dis + rad + tax +
##     ptratio + lstat + medv
##
```

```
##           Df Deviance    AIC
## - rm       1   192.71 216.71
## - lstat    1   192.77 216.77
## - chas     1   193.53 217.53
## - indus    1   193.99 217.99
## <none>         192.05 218.05
## - tax      1   196.59 220.59
## - zn       1   196.89 220.89
## - age      1   198.73 222.73
## - medv     1   199.95 223.95
## - ptratio  1   203.32 227.32
## - dis      1   203.84 227.84
## - rad      1   233.74 257.74
## - nox      1   265.05 289.05
##
## Step:  AIC=216.71
## target ~ zn + indus + chas + nox + age + dis + rad + tax + ptratio +
##     lstat + medv
##
##           Df Deviance    AIC
## - chas     1   194.24 216.24
## - lstat    1   194.32 216.32
## - indus    1   194.58 216.58
## <none>         192.71 216.71
## - tax      1   197.59 219.59
## - zn       1   198.07 220.07
## - age      1   199.11 221.11
## - ptratio  1   203.53 225.53
## - dis      1   203.85 225.85
## - medv     1   205.35 227.35
## - rad      1   233.81 255.81
## - nox      1   265.14 287.14
##
## Step:  AIC=216.24
## target ~ zn + indus + nox + age + dis + rad + tax + ptratio +
##     lstat + medv
##
##           Df Deviance    AIC
## - indus    1   195.51 215.51
## <none>         194.24 216.24
## - lstat    1   196.33 216.33
## - zn       1   200.59 220.59
## - tax      1   200.75 220.75
## - age      1   201.00 221.00
## - ptratio  1   203.94 223.94
## - dis      1   204.83 224.83
## - medv     1   207.12 227.12
## - rad      1   241.41 261.41
## - nox      1   265.19 285.19
##
## Step:  AIC=215.51
## target ~ zn + nox + age + dis + rad + tax + ptratio + lstat +
##     medv
##
```

```
##           Df Deviance    AIC
## - lstat    1   197.32 215.32
## <none>         195.51 215.51
## - zn       1   202.05 220.05
## - age      1   202.23 220.23
## - ptratio  1   205.01 223.01
## - dis      1   205.96 223.96
## - tax      1   206.60 224.60
## - medv     1   208.13 226.13
## - rad      1   249.55 267.55
## - nox      1   270.59 288.59
##
## Step:  AIC=215.32
## target ~ zn + nox + age + dis + rad + tax + ptratio + medv
##
##           Df Deviance    AIC
## <none>         197.32 215.32
## - zn       1   203.45 219.45
## - ptratio  1   206.27 222.27
## - age      1   207.13 223.13
## - tax      1   207.62 223.62
## - dis      1   207.64 223.64
## - medv     1   208.65 224.65
## - rad      1   250.98 266.98
## - nox      1   273.18 289.18
```

```r
summary(model_1)
```

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

```
## AIC: 215.32
##
## Number of Fisher Scoring iterations: 9
```

```r
vif(model_1)
```

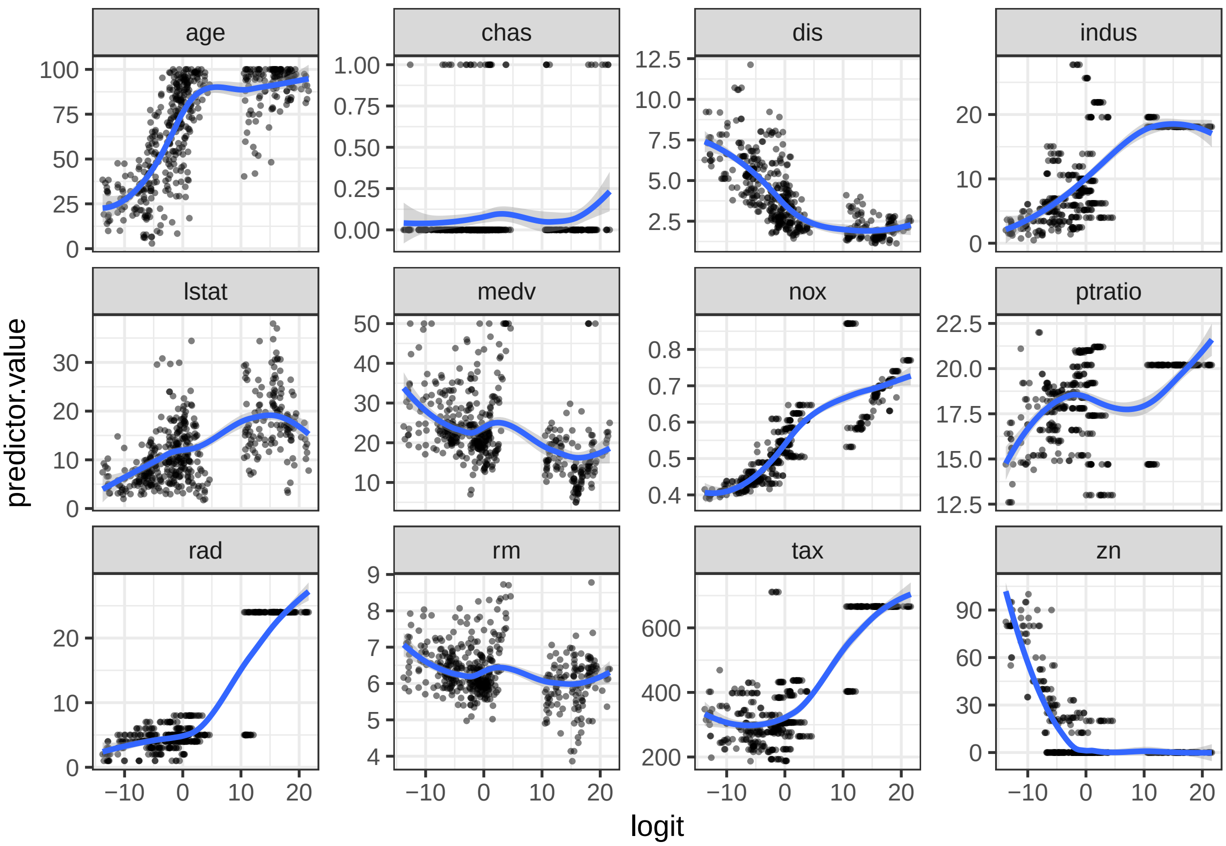
```
##       zn      nox      age      dis      rad      tax  ptratio     medv
## 1.789037 3.172660 1.701974 3.595939 1.697110 1.754274 1.865085 2.193689
```

There is no significant multicollinearity detected in model_1.

Check model_1 for the following logistic regression assumptions:

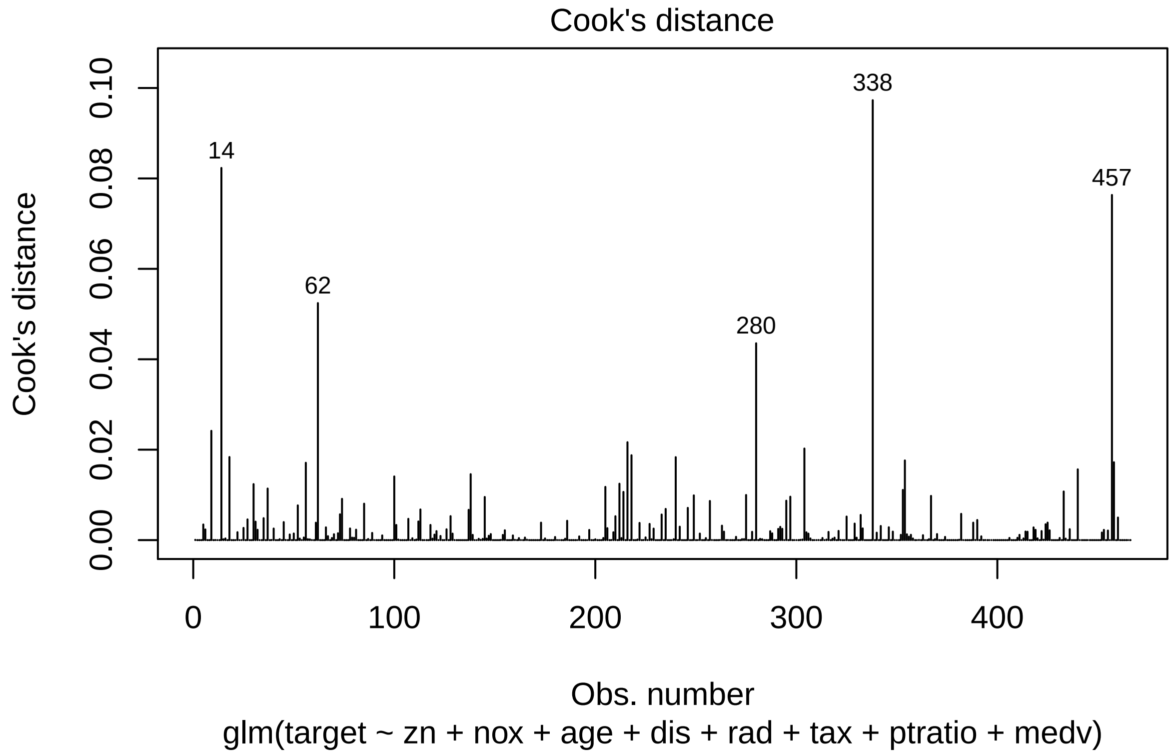
1. The outcome is a binary (True)
2. There is a linear relationship between the logit of the outcome and each predictor variables (If not, model can benefit from variables transformations)
3. There is no influential values (extreme values or outliers) in the continuous predictors.
4. There is no high intercorrelations (i.e. multicollinearity) among the predictors.

Checking for a linear relationship between the logit of the outcome and each predictor variables

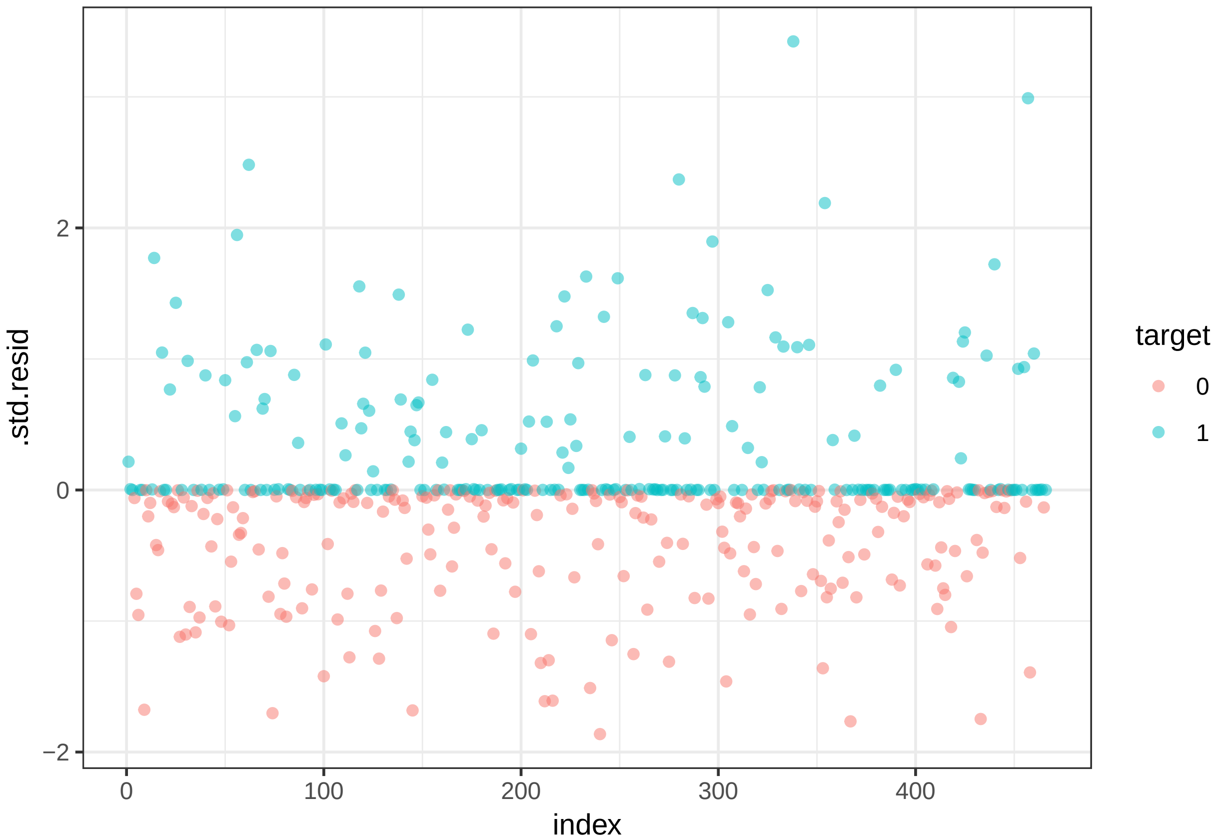


Not all the relationships are linear, model can benefit from varibles transformations.

Checking model_1 for the presence of influencial values.




```
## # A tibble: 5 x 17
##   target    zn   nox   age   dis   rad   tax ptratio  medv .fitted .se.fit
##   <fct>  <dbl> <dbl> <dbl> <dbl> <int> <int>   <dbl> <dbl>   <dbl>   <dbl>
## 1 1         22 0.431   8.4  8.91     7   330    19.1  42.8  -0.941   0.970
## 2 1          0 0.544  37.8  2.52     4   304    18.4  16.1  -2.96    0.706
## 3 1         22 0.431  34.9  8.06     7   330    19.1  24.3  -2.67    0.652
## 4 1         20 0.464  42.1  4.43     3   223    18.6  21.1  -5.84    0.936
## 5 1          0 0.489   9.8  3.59     4   277    18.6  23.7  -4.42    0.832
## # ... with 6 more variables: .resid <dbl>, .hat <dbl>, .sigma <dbl>,
## #   .cooksd <dbl>, .std.resid <dbl>, index <int>
```




```
## # A tibble: 1 x 17
##   target    zn   nox   age   dis   rad   tax ptratio  medv .fitted .se.fit
##   <fct>  <dbl> <dbl> <dbl> <dbl> <int> <int>   <dbl> <dbl>   <dbl>   <dbl>
## 1 1         20 0.464  42.1  4.43     3   223    18.6  21.1   -5.84   0.936
## # ... with 6 more variables: .resid <dbl>, .hat <dbl>, .sigma <dbl>,
## #   .cooksd <dbl>, .std.resid <dbl>, index <int>
```

Eliminating the row from training data set with influential value.

```
training_clean <-training %>%
  filter(!(nox==0.464 & age==42.1))
```

MODEL 2

Building a model based on a dateset with eliminated influential values.

```
model_2<- step(glm(target~., data = training_clean, family = 'binomial'), direction = "backward")
```

```
## Start:  AIC=204.95
## target ~ zn + indus + chas + nox + rm + age + dis + rad + tax +
##     ptratio + lstat + medv
##
##           Df Deviance    AIC
## - lstat    1   179.53 203.53
## - rm       1   179.86 203.86
## - chas     1   180.40 204.40
## <none>         178.95 204.95
## - indus    1   181.26 205.26
```

```
## - tax      1   182.93 206.93
## - zn       1   186.28 210.28
## - age      1   187.56 211.56
## - medv     1   188.43 212.43
## - ptratio  1   190.95 214.95
## - dis      1   194.36 218.36
## - rad      1   221.84 245.84
## - nox      1   258.08 282.08
##
## Step:  AIC=203.53
## target ~ zn + indus + chas + nox + rm + age + dis + rad + tax +
##     ptratio + medv
##
##           Df Deviance    AIC
## - chas     1   181.28 203.28
## - rm       1   181.38 203.38
## <none>         179.53 203.53
## - indus    1   181.76 203.76
## - tax      1   183.26 205.26
## - zn       1   186.46 208.46
## - medv     1   189.16 211.16
## - ptratio  1   192.50 214.50
## - age      1   192.97 214.97
## - dis      1   195.50 217.50
## - rad      1   222.50 244.50
## - nox      1   259.97 281.97
##
## Step:  AIC=203.28
## target ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio +
##     medv
##
##           Df Deviance    AIC
## - indus    1   182.79 202.79
## <none>         181.28 203.28
## - rm       1   183.38 203.38
## - tax      1   186.60 206.60
## - zn       1   189.44 209.44
## - medv     1   191.08 211.08
## - ptratio  1   193.09 213.09
## - age      1   195.88 215.88
## - dis      1   196.73 216.73
## - rad      1   232.03 252.03
## - nox      1   260.00 280.00
##
## Step:  AIC=202.79
## target ~ zn + nox + rm + age + dis + rad + tax + ptratio + medv
##
##           Df Deviance    AIC
## - rm       1   184.66 202.66
## <none>         182.79 202.79
## - zn       1   191.25 209.25
## - medv     1   192.17 210.17
## - tax      1   192.67 210.67
## - ptratio  1   194.12 212.12
```

```
## - age      1   196.72 214.72
## - dis      1   197.96 215.96
## - rad      1   240.79 258.79
## - nox      1   266.03 284.03
##
## Step:  AIC=202.66
## target ~ zn + nox + age + dis + rad + tax + ptratio + medv
##
##           Df Deviance    AIC
## <none>         184.66 202.66
## - zn       1   193.60 209.60
## - ptratio  1   194.15 210.15
## - tax      1   194.85 210.85
## - age      1   196.83 212.83
## - dis      1   198.25 214.25
## - medv     1   198.43 214.43
## - rad      1   240.96 256.96
## - nox      1   266.17 282.17
```

```
summary(model_2)
```

```
##
## Call:
## glm(formula = target ~ zn + nox + age + dis + rad + tax + ptratio +
##     medv, family = "binomial", data = training_clean)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8555  -0.1501  -0.0006   0.0014   3.1726
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -42.128250   6.730139  -6.260 3.86e-10 ***
## zn           -0.093902   0.036896  -2.545 0.010927 *
## nox          47.388109   7.340291   6.456 1.08e-10 ***
## age           0.038718   0.011711   3.306 0.000946 ***
## dis           0.807838   0.235713   3.427 0.000610 ***
## rad           0.806479   0.161555   4.992 5.98e-07 ***
## tax          -0.007945   0.002739  -2.900 0.003728 **
## ptratio       0.350885   0.117755   2.980 0.002884 **
## medv          0.130814   0.038521   3.396 0.000684 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 644.45  on 464  degrees of freedom
## Residual deviance: 184.66  on 456  degrees of freedom
## AIC: 202.66
##
## Number of Fisher Scoring iterations: 9
```

```
vif(model_2)
```

```
##       zn      nox      age      dis      rad      tax  ptratio     medv
```

```
## 1.960798 3.435929 1.773456 4.001177 1.770018 1.759468 1.985662 2.336037
```

There is no significant multicollinearity detected in model_2.

MODEL 3

This model is built based on important variables, selected using caret package function varImp()

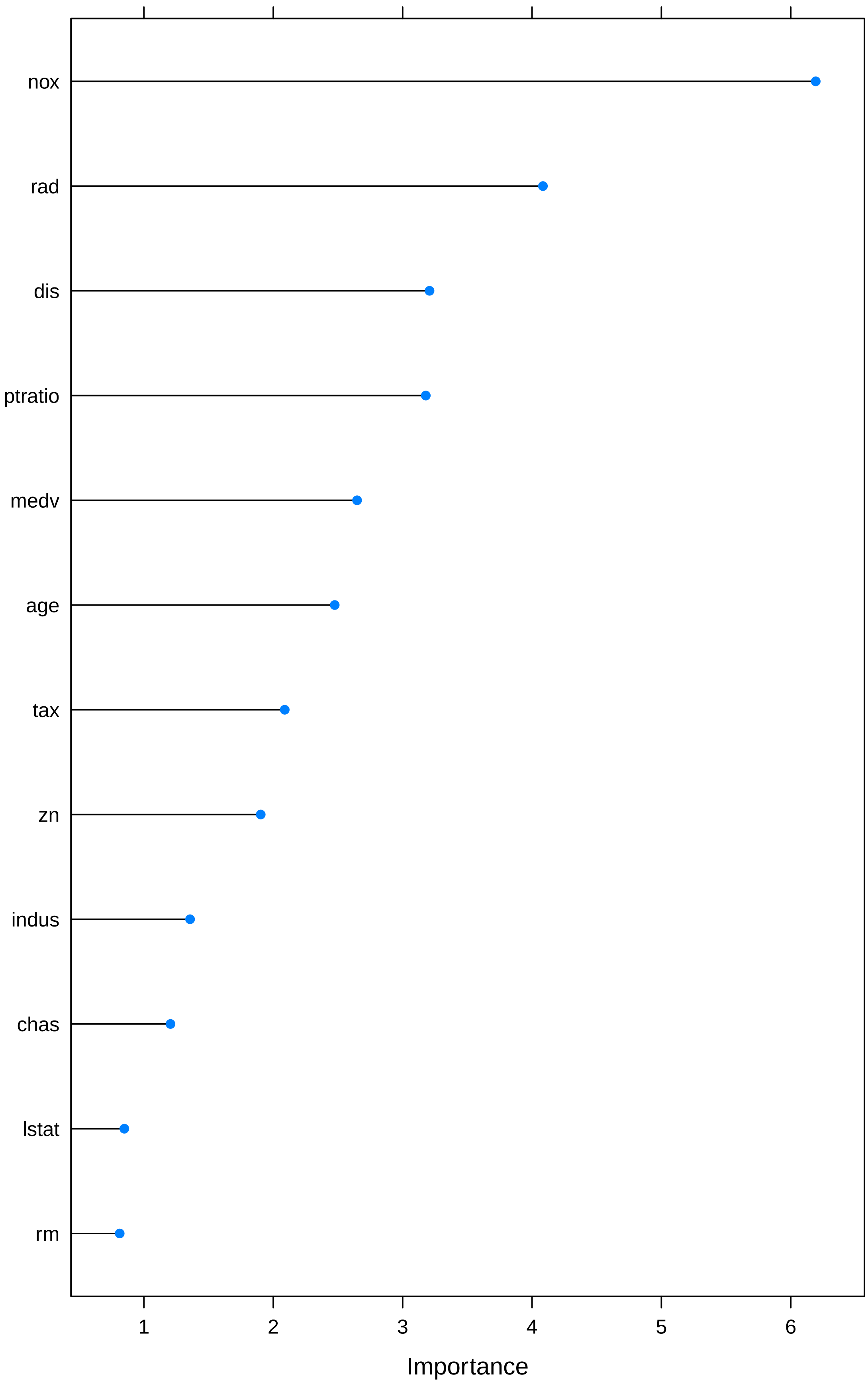
```
control <- trainControl(method="repeatedcv", number=10, repeats=3)
model <- train(target~., data=training, method="glm",  trControl=control)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
importance <- varImp(model, scale=FALSE)
print(importance)
```

```
## glm variable importance
##
##         Overall
## nox      6.1932
## rad      4.0843
## dis      3.2077
## ptratio  3.1791
## medv     2.6477
## age      2.4749
## tax      2.0887
## zn       1.9029
## indus    1.3568
## chas     1.2054
## lstat    0.8486
## rm       0.8127
```

```
plot(importance)
```



nox
rad
dis
ptratio
medv
age
tax
zn
indus
chas
lstat
rm
1
2
3
4
5
6
Importance

```r
model_3<- step(glm(target~., data = training %>% dplyr::select(-lstat, -rm), family = 'binomial'), direc
```

```
## Start:  AIC=216.32
## target ~ zn + indus + chas + nox + age + dis + rad + tax + ptratio +
##     medv
##
##           Df Deviance    AIC
## - indus    1   195.97 215.97
## <none>         194.32 216.32
## - chas     1   196.33 216.33
## - tax      1   198.83 218.83
## - zn       1   199.29 219.29
## - age      1   203.62 223.62
## - ptratio  1   204.88 224.88
## - dis      1   205.44 225.44
## - medv     1   205.98 225.98
## - rad      1   235.07 255.07
## - nox      1   267.08 287.08
##
## Step:  AIC=215.97
## target ~ zn + chas + nox + age + dis + rad + tax + ptratio +
##     medv
##
##           Df Deviance    AIC
## - chas     1   197.32 215.32
## <none>         195.97 215.97
## - zn       1   201.29 219.29
## - age      1   205.01 223.01
## - tax      1   205.20 223.20
## - ptratio  1   205.90 223.90
## - dis      1   206.82 224.82
## - medv     1   207.65 225.65
## - rad      1   244.73 262.73
## - nox      1   273.06 291.06
##
## Step:  AIC=215.32
## target ~ zn + nox + age + dis + rad + tax + ptratio + medv
##
##           Df Deviance    AIC
## <none>         197.32 215.32
## - zn       1   203.45 219.45
## - ptratio  1   206.27 222.27
## - age      1   207.13 223.13
## - tax      1   207.62 223.62
## - dis      1   207.64 223.64
## - medv     1   208.65 224.65
## - rad      1   250.98 266.98
## - nox      1   273.18 289.18
```

```r
summary(model_3)
```

```
##
## Call:
## glm(formula = target ~ zn + nox + age + dis + rad + tax + ptratio +
```

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

```r
vif(model_3)
```

```
##       zn      nox      age      dis      rad      tax  ptratio     medv
## 1.789037 3.172660 1.701974 3.595939 1.697110 1.754274 1.865085 2.193689
```

There is no significant multicollinearity detected in model_3.

MODEL 4

This model is built based on the lowest Akaike information criterion (AIC). MASS package is used.

```r
model_4 <- glm(target ~., data = training, family = binomial) %>%
  stepAIC(trace = FALSE)
summary(model_4)
```

```
##
## Call:
## glm(formula = target ~ zn + nox + age + dis + rad + tax + ptratio +
##     medv, family = binomial, data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8295  -0.1752  -0.0021   0.0032   3.4191
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -37.415922   6.035013  -6.200 5.65e-10 ***
## zn           -0.068648   0.032019  -2.144  0.03203 *
## nox          42.807768   6.678692   6.410 1.46e-10 ***
## age           0.032950   0.010951   3.009  0.00262 **
## dis           0.654896   0.214050   3.060  0.00222 **
## rad           0.725109   0.149788   4.841 1.29e-06 ***
## tax          -0.007756   0.002653  -2.924  0.00346 **
## ptratio       0.323628   0.111390   2.905  0.00367 **
## medv          0.110472   0.035445   3.117  0.00183 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 197.32  on 457  degrees of freedom
## AIC: 215.32
##
## Number of Fisher Scoring iterations: 9
```

```
vif(model_4)
```

```
##       zn      nox      age      dis      rad      tax  ptratio     medv
## 1.789037 3.172660 1.701974 3.595939 1.697110 1.754274 1.865085 2.193689
```

There is no significant multicollinearity detected in model_4.

MODEL 5

This model is built based on the data transformation performed in "Data Preparation" part

```
model_5 <- step(glm(target ~., data = transformed.data, family = 'binomial'), direction = "backward")
```

```
## Start:  AIC=260.23
## target ~ zn + indus + chas + nox + rm + age + dis + rad + tax +
##     ptratio + lstat + medv
##
##           Df Deviance    AIC
## - rm       1   234.23 258.23
## - age      1   234.65 258.65
## - ptratio  1   234.73 258.73
## - lstat    1   234.83 258.83
## <none>         234.23 260.23
## - rad      1   238.78 262.78
## - chas     1   242.40 266.40
## - medv     1   244.24 268.24
## - dis      1   244.71 268.71
## - indus    1   246.62 270.62
## - zn       1   248.47 272.47
## - tax      1   257.10 281.10
## - nox      1   321.60 345.60
##
## Step:  AIC=258.23
## target ~ zn + indus + chas + nox + age + dis + rad + tax + ptratio +
##     lstat + medv
```

```
##
##           Df Deviance    AIC
## - age      1   234.70 256.70
## - ptratio  1   234.73 256.73
## - lstat    1   235.01 257.01
## <none>         234.23 258.23
## - rad      1   238.81 260.81
## - chas     1   242.42 264.42
## - dis      1   244.86 266.86
## - indus    1   246.77 268.77
## - zn       1   248.47 270.47
## - medv     1   249.98 271.98
## - tax      1   258.58 280.58
## - nox      1   322.16 344.16
##
## Step:  AIC=256.7
## target ~ zn + indus + chas + nox + dis + rad + tax + ptratio +
##     lstat + medv
##
##           Df Deviance    AIC
## - ptratio  1   235.02 255.02
## - lstat    1   235.79 255.79
## <none>         234.70 256.70
## - rad      1   239.59 259.59
## - chas     1   243.21 263.21
## - dis      1   244.99 264.99
## - indus    1   249.04 269.04
## - zn       1   249.67 269.67
## - medv     1   250.15 270.15
## - tax      1   259.24 279.24
## - nox      1   329.43 349.43
##
## Step:  AIC=255.02
## target ~ zn + indus + chas + nox + dis + rad + tax + lstat +
##     medv
##
##         Df Deviance    AIC
## - lstat  1   236.03 254.03
## <none>       235.02 255.02
## - rad    1   240.23 258.23
## - chas   1   243.26 261.26
## - dis    1   245.01 263.01
## - indus  1   250.30 268.30
## - medv   1   250.47 268.47
## - zn     1   251.62 269.62
## - tax    1   261.31 279.31
## - nox    1   331.22 349.22
##
## Step:  AIC=254.03
## target ~ zn + indus + chas + nox + dis + rad + tax + medv
##
##         Df Deviance    AIC
## <none>       236.03 254.03
## - rad    1   241.25 257.25
```

```
## - dis    1   245.45 261.45
## - chas   1   245.67 261.67
## - indus  1   251.11 267.11
## - zn     1   252.69 268.69
## - medv   1   256.71 272.71
## - tax    1   262.46 278.46
## - nox    1   338.18 354.18
```

```r
summary(model_5)
```

```
##
## Call:
## glm(formula = target ~ zn + indus + chas + nox + dis + rad +
##     tax + medv, family = "binomial", data = transformed.data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.89268  -0.28726  -0.00684   0.21268   3.13087
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -47.41854    6.16159  -7.696 1.41e-14 ***
## zn           -0.08984    0.03094  -2.903 0.003691 **
## indus        -1.74389    0.47409  -3.678 0.000235 ***
## chas          2.08121    0.67092   3.102 0.001922 **
## nox          44.08304    6.15874   7.158 8.20e-13 ***
## dis           0.56284    0.18574   3.030 0.002444 **
## rad          -0.88718    0.39686  -2.236 0.025383 *
## tax           3.48609    0.73586   4.737 2.16e-06 ***
## medv          0.11964    0.02851   4.197 2.71e-05 ***
## ---
## 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: 236.03  on 457  degrees of freedom
## AIC: 254.03
##
## Number of Fisher Scoring iterations: 8
```

```r
vif(model_5)
```

```
##       zn    indus     chas      nox      dis      rad      tax     medv
## 1.422571 2.042715 1.189370 3.356124 2.743851 1.431253 2.074097 1.826324
```

There is no significant multicollinearity detected in model_5.

Select Models

AIC, BIC, Loik and pseudR2 were used to select the best model.

```
##              AIC      BIC    loglik  pseudoR2
## model_1 215.3229 252.6205  -98.66143 0.6944879
## model_2 202.6583 239.9366  -92.32914 0.7134650
```

```
## model_3 215.3229 252.6205  -98.66143 0.6944879
## model_4 215.3229 252.6205  -98.66143 0.6944879
## model_5 254.0314 291.3290 -118.01568 0.6345561
```

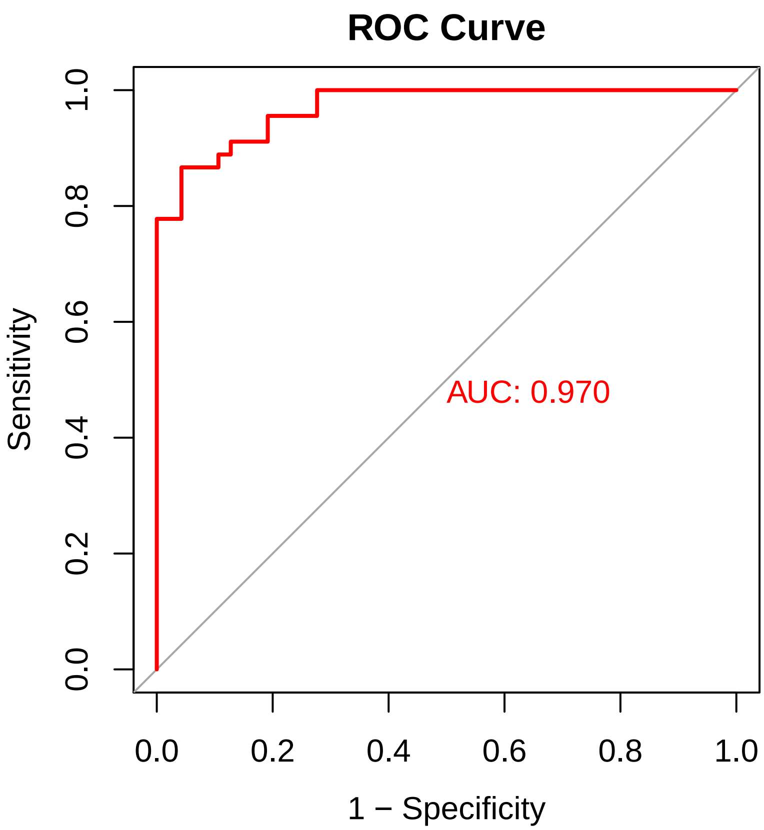
model_2 is the best model considering AIC,BIC, log likelihood and McFadden pseudoR2. model_2 has the lowest AIC, loglik and highest pseudoR2 which is indicative of a superior fit over all the other models. Although using that process might direct to choose a model that is overfitted.

We will choose model_2 as the best model for this assignment.

Splitting data set on train and test in order to assess model 2.

```
set.seed(123)
training.samples <- training$target %>%
createDataPartition(p = 0.8, list = FALSE)
train.data  <- training[training.samples, ]
test.data <- training[-training.samples, ]
```

Roc curve of model_2



Let's choose a cut off probability measure for predicting with a high or low crime rate.

```
## [1] 0.6335833
```

The value is closed to 50% let's use 50% as a cutoff.

Confusion matrix of model_2

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   0   1
```

```
##          0 41  5
##          1  6 40
##
##                Accuracy : 0.8804
##                  95% CI : (0.7961, 0.9388)
##     No Information Rate : 0.5109
##     P-Value [Acc > NIR] : 5.644e-14
##
##                   Kappa : 0.7609
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8889
##             Specificity : 0.8723
##          Pos Pred Value : 0.8696
##          Neg Pred Value : 0.8913
##              Prevalence : 0.4891
##          Detection Rate : 0.4348
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.8806
##
##        'Positive' Class : 1
##
```

```
## $accuracy
## [1] 0.8804348
##
## $error_rate
## [1] 0.1195652
##
## $precision
## [1] 0.8695652
##
## $sensitivity
## [1] 0.8888889
##
## $specificity
## [1] 0.8723404
##
## $F1
## [1] 0.8791209
```

Prediction

```
##   zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv
## 1  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8  4.03 34.7
## 2  0  8.14    0 0.538 6.096 84.5 4.4619   4 307    21.0 10.26 18.2
## 3  0  8.14    0 0.538 6.495 94.4 4.4547   4 307    21.0 12.80 18.4
## 4  0  8.14    0 0.538 5.950 82.0 3.9900   4 307    21.0 27.71 13.2
## 5  0  5.96    0 0.499 5.850 41.5 3.9342   5 279    19.2  8.77 21.0
## 6 25  5.13    0 0.453 5.741 66.2 7.2254   8 284    19.7 13.15 18.7
##   predict_prob predict_target
## 1   0.04523037              0
## 2   0.68543918              1
```

```
## 3   0.76540370                1
## 4   0.41260604                0
## 5   0.08341741                0
## 6   0.25830142                0
```