Data 621 Homework 3

Layla Quinones

10/24/2021

Libraries

```
library(tidyverse)
library(ggplot2)
library(VIM)
library(GGally)
library(caret)
library(broom)
```

EDA

```
# Load data
# Training
rawTrain <- read.csv("https://raw.githubusercontent.com/MsQCompSci/Data621Group4/main/HW3/crime-training

#Testing data
rawTest <- read.csv("https://raw.githubusercontent.com/MsQCompSci/Data621Group4/main/HW3/crime-evaluati
```

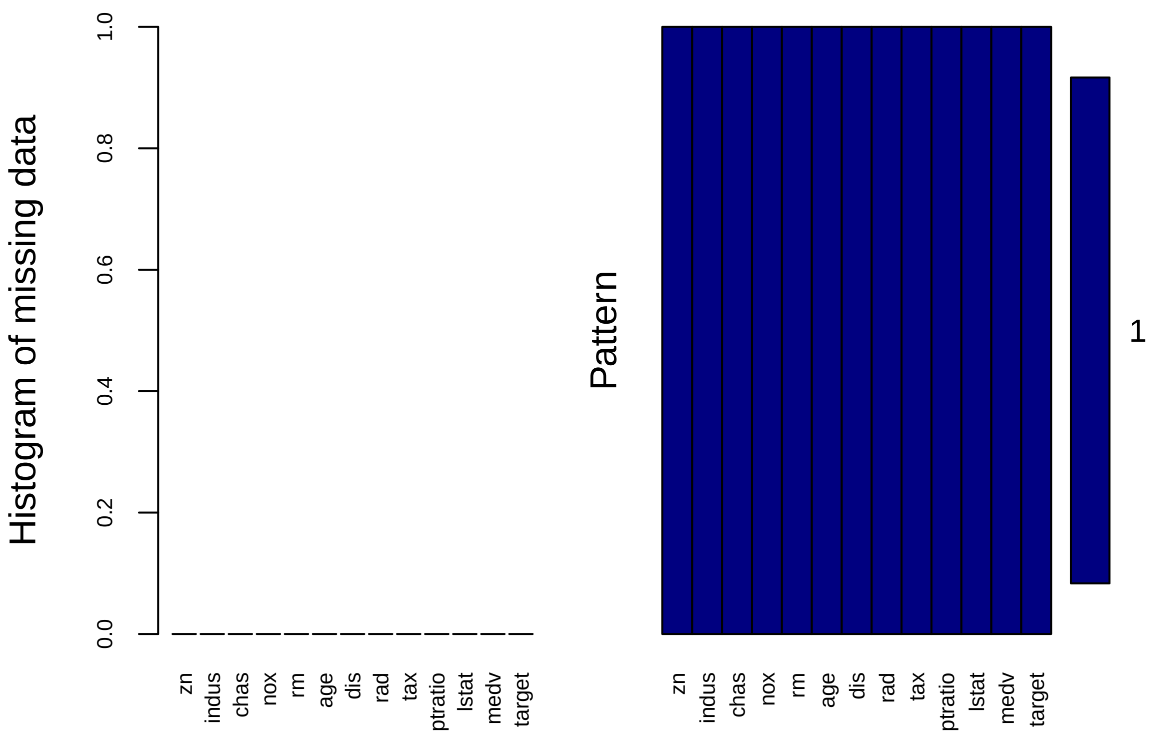
```
# check to see if we need to clean the data
# gives us a sense of what each predictor is
glimpse(rawTrain)
```

```
## Rows: 466
## Columns: 13
## $ zn      <dbl> 0, 0, 0, 30, 0, 0, 0, 0, 0, 80, 22, 0, 0, 22, 0, 0, 100, 20...
## $ indus   <dbl> 19.58, 19.58, 18.10, 4.93, 2.46, 8.56, 18.10, 18.10, 5.19, ...
## $ chas    <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ nox     <dbl> 0.605, 0.871, 0.740, 0.428, 0.488, 0.520, 0.693, 0.693, 0.5...
## $ rm      <dbl> 7.929, 5.403, 6.485, 6.393, 7.155, 6.781, 5.453, 4.519, 6.3...
## $ age     <dbl> 96.2, 100.0, 100.0, 7.8, 92.2, 71.3, 100.0, 100.0, 38.1, 19...
## $ dis     <dbl> 2.0459, 1.3216, 1.9784, 7.0355, 2.7006, 2.8561, 1.4896, 1.6...
## $ rad     <int> 5, 5, 24, 6, 3, 5, 24, 24, 5, 1, 7, 5, 24, 7, 3, 3, 5, 5, 2...
## $ tax     <int> 403, 403, 666, 300, 193, 384, 666, 666, 224, 315, 330, 398,...
## $ ptratio <dbl> 14.7, 14.7, 20.2, 16.6, 17.8, 20.9, 20.2, 20.2, 20.2, 16.4,...
## $ lstat   <dbl> 3.70, 26.82, 18.85, 5.19, 4.82, 7.67, 30.59, 36.98, 5.68, 9...
## $ medv    <dbl> 50.0, 13.4, 15.4, 23.7, 37.9, 26.5, 5.0, 7.0, 22.2, 20.9, 2...
## $ target  <int> 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1,...
```

```r
# All varaibles are numeric
# categorical variables
# chas

#dicrete
#rad, zn, tax

#all others are continuous
```

No Missing Values

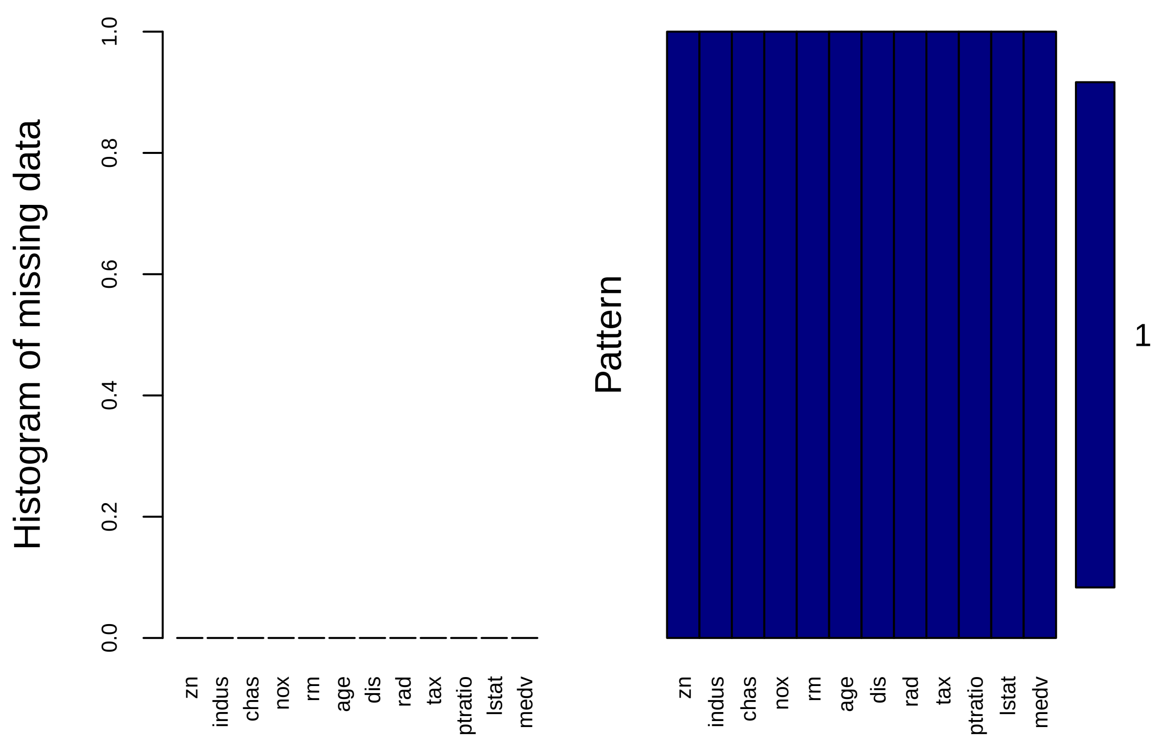
```r
#plot missing values using VIM package
aggr(rawTrain , col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(rawTrain), cex.axis=
```




```
##
##  Variables sorted by number of missings:
##  Variable Count
##        zn     0
##     indus     0
##      chas     0
##       nox     0
##        rm     0
##       age     0
```

```
##       dis     0
##       rad     0
##       tax     0
##   ptratio     0
##     lstat     0
##      medv     0
##    target     0
```

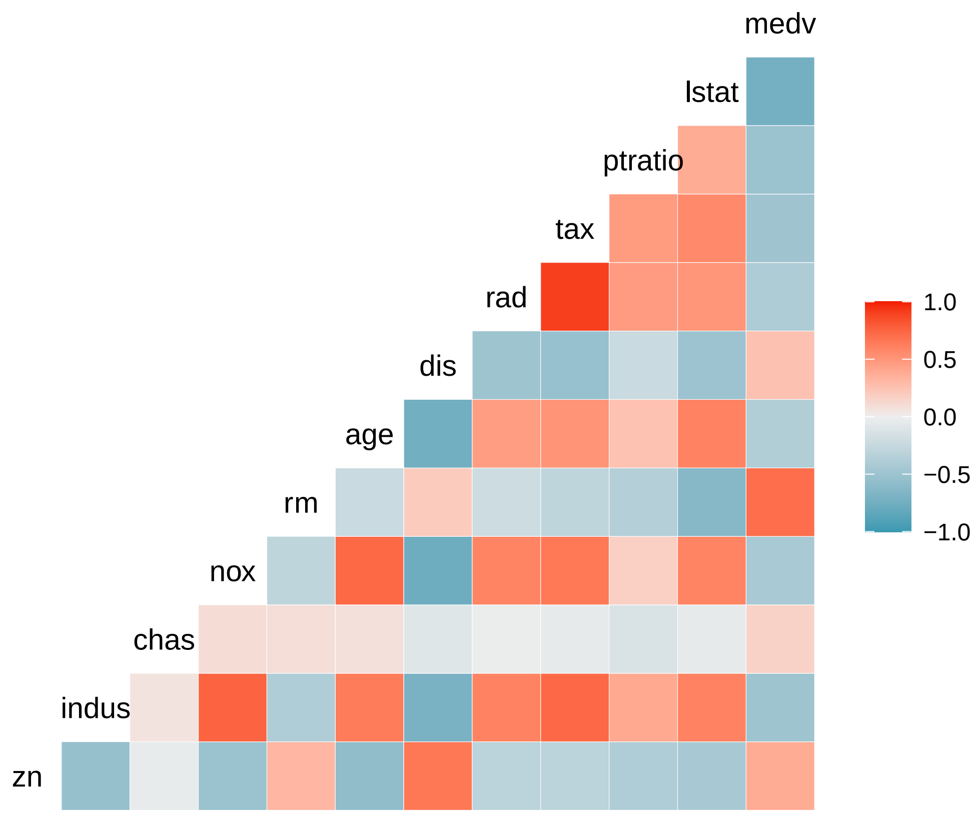
```r
#plot missing values using VIM package
aggr(rawTest , col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(rawTrain), cex.axis=.7
```



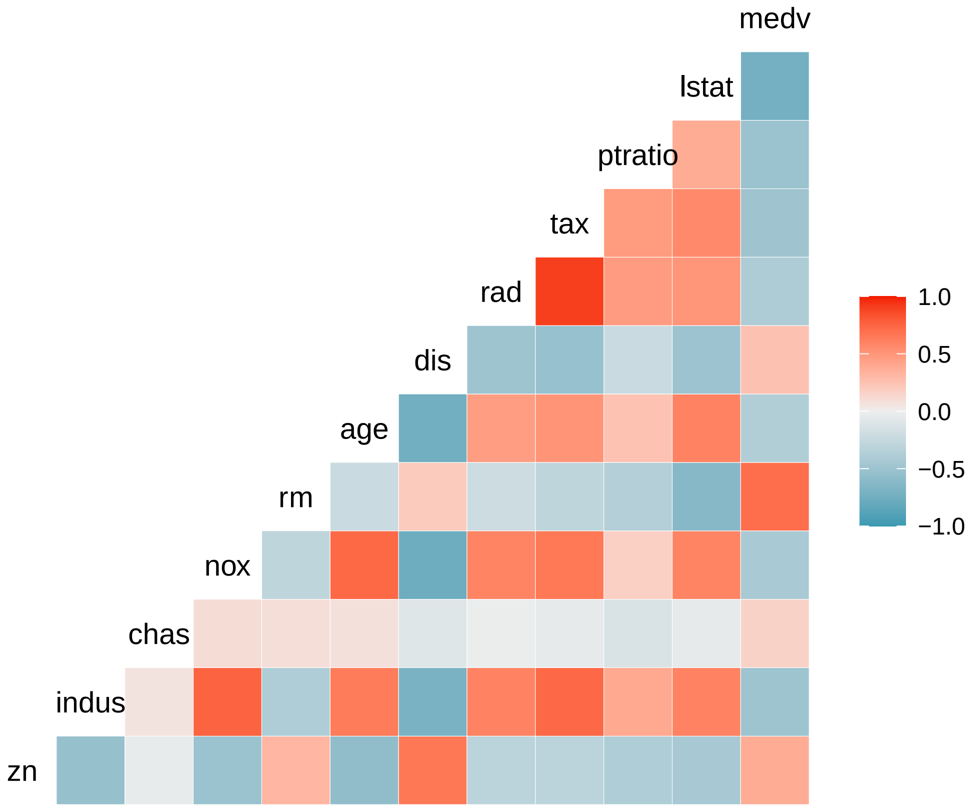

```
##
##  Variables sorted by number of missings:
##  Variable Count
##        zn     0
##     indus     0
##      chas     0
##       nox     0
##        rm     0
##       age     0
##       dis     0
##       rad     0
##       tax     0
##   ptratio     0
##     lstat     0
##      medv     0
```

Correlation

```
#correlation matrix for predictors
ggcorr(rawTrain%>% select(zn:medv))
```




```
#Idetify highly correlated variables
ggcorr(rawTrain%>% select(zn:medv))
```




```r
#Lets look at some highly correlated variables and drop them
findCorrelation(cor(rawTrain%>% select(zn:medv)),
                cutoff = 0.75,
                verbose = TRUE,
                names = TRUE)
```

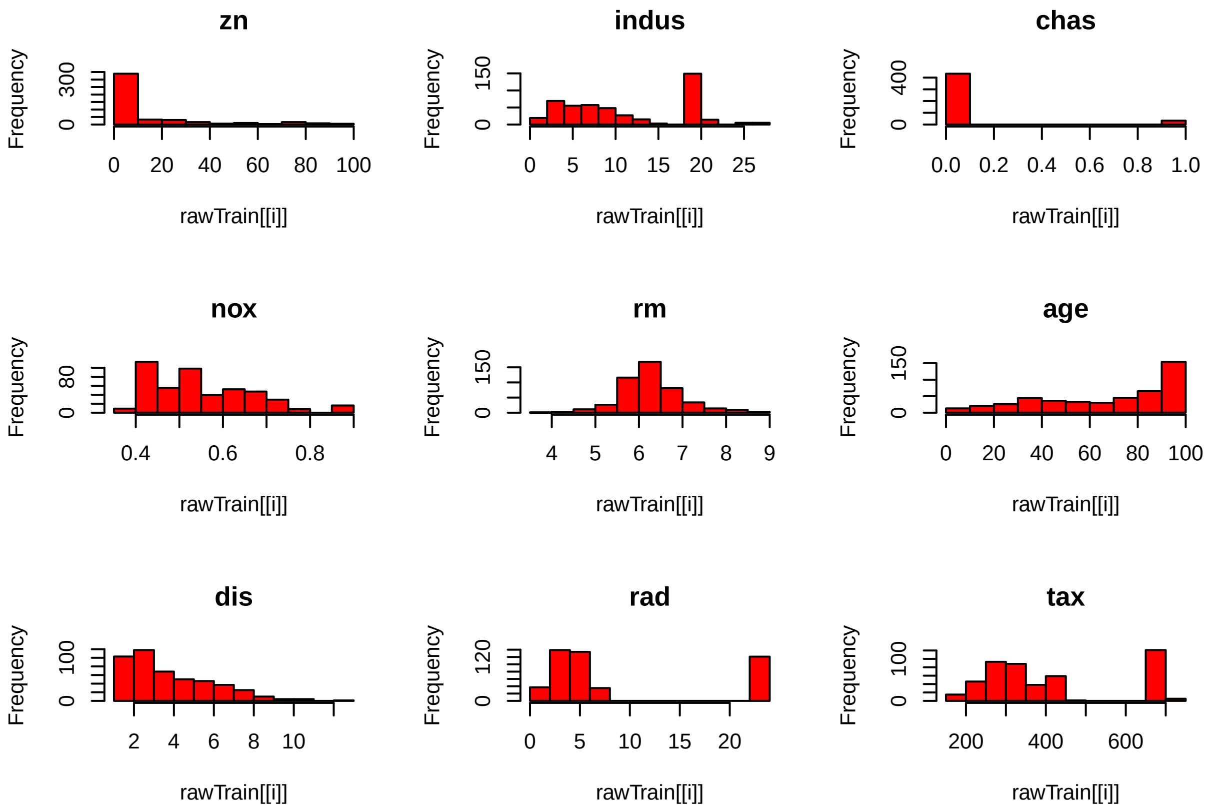
```
## Compare row 2  and column  4 with corr  0.76
##   Means:  0.539 vs 0.416 so flagging column 2
## Compare row 4  and column  7 with corr  0.769
##   Means:  0.487 vs 0.395 so flagging column 4
## Compare row 9  and column  8 with corr  0.906
##   Means:  0.46 vs 0.377 so flagging column 9
## Compare row 6  and column  7 with corr  0.751
##   Means:  0.417 vs 0.357 so flagging column 6
## All correlations <= 0.75

## [1] "indus" "nox"   "tax"   "age"
```

```r
# There are 4 highly correlated variables
# I will drop the highest one which is tax which seems to be the most highly correlated
#tax and rad are 0.9 correlated lets look at their relationship to the predictor to see which one to dr
```

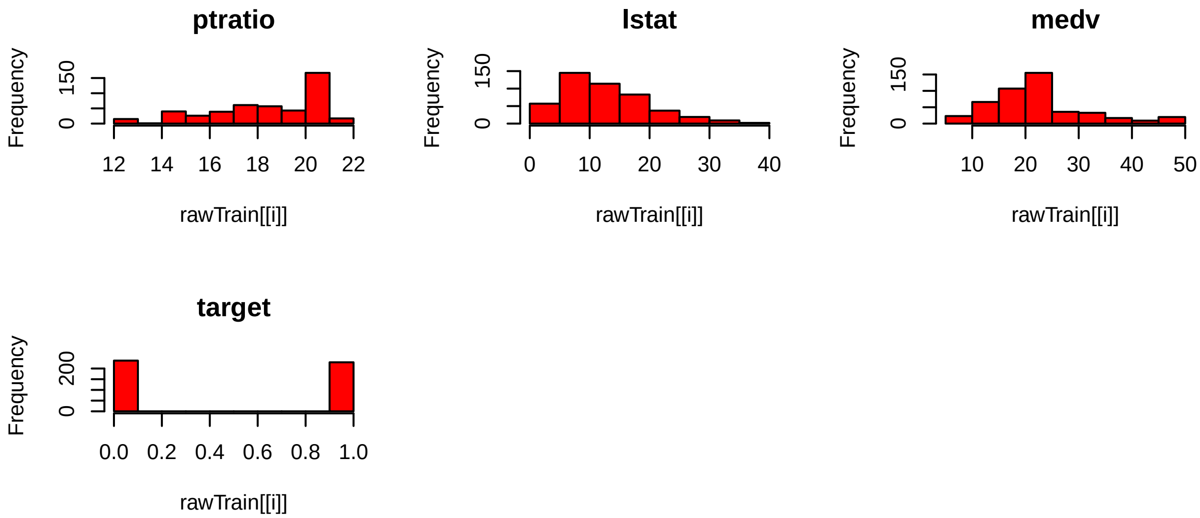
Distribution of Predictors

ADD VARIANCE AND INFLATION FACTORS TO THIS SECTION

```r
par(mfrow = c(3,3))
for(i in 1:ncol(rawTrain)) {#distribution of each variable
  hist(rawTrain[[i]], main = colnames(rawTrain[i]), col = "red")
}
```

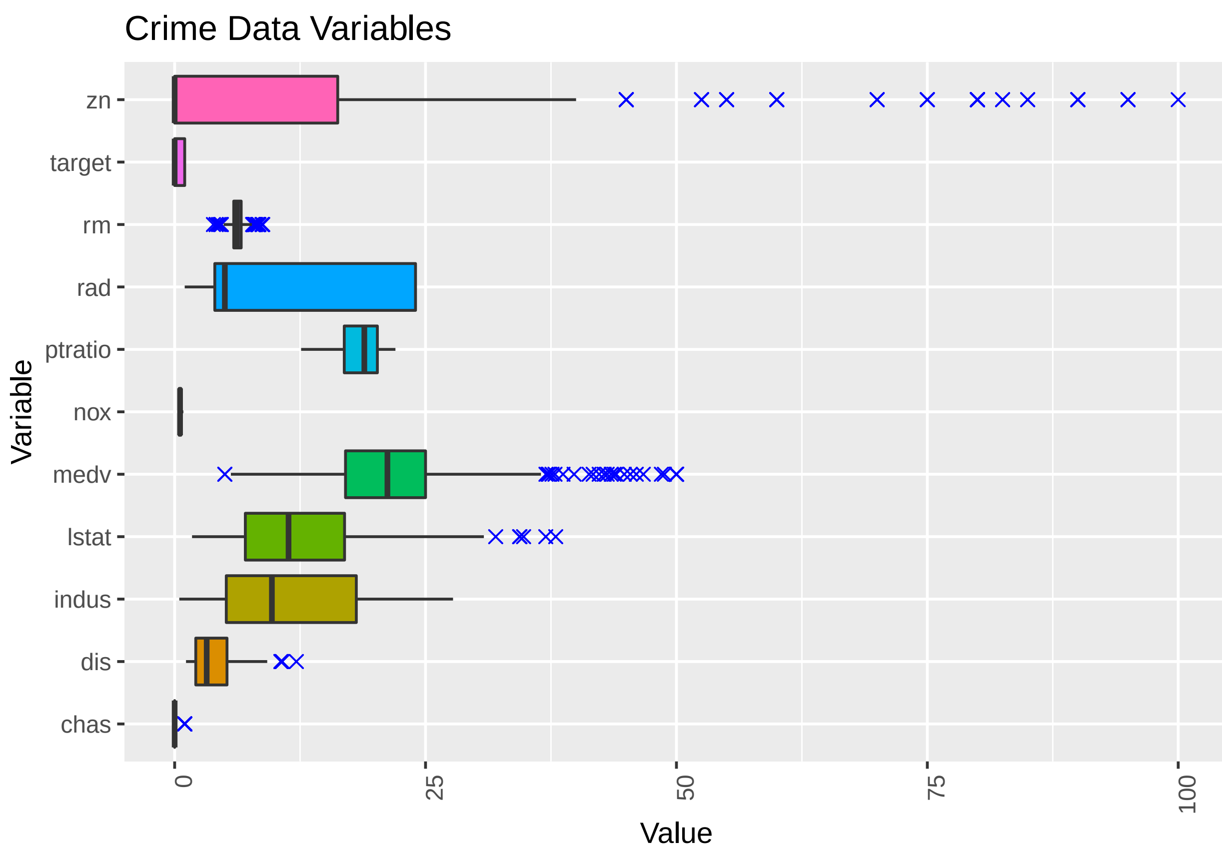



```r
#binomial data
# indus, tax and rad

#all other variables ar skewed excpet RM
```



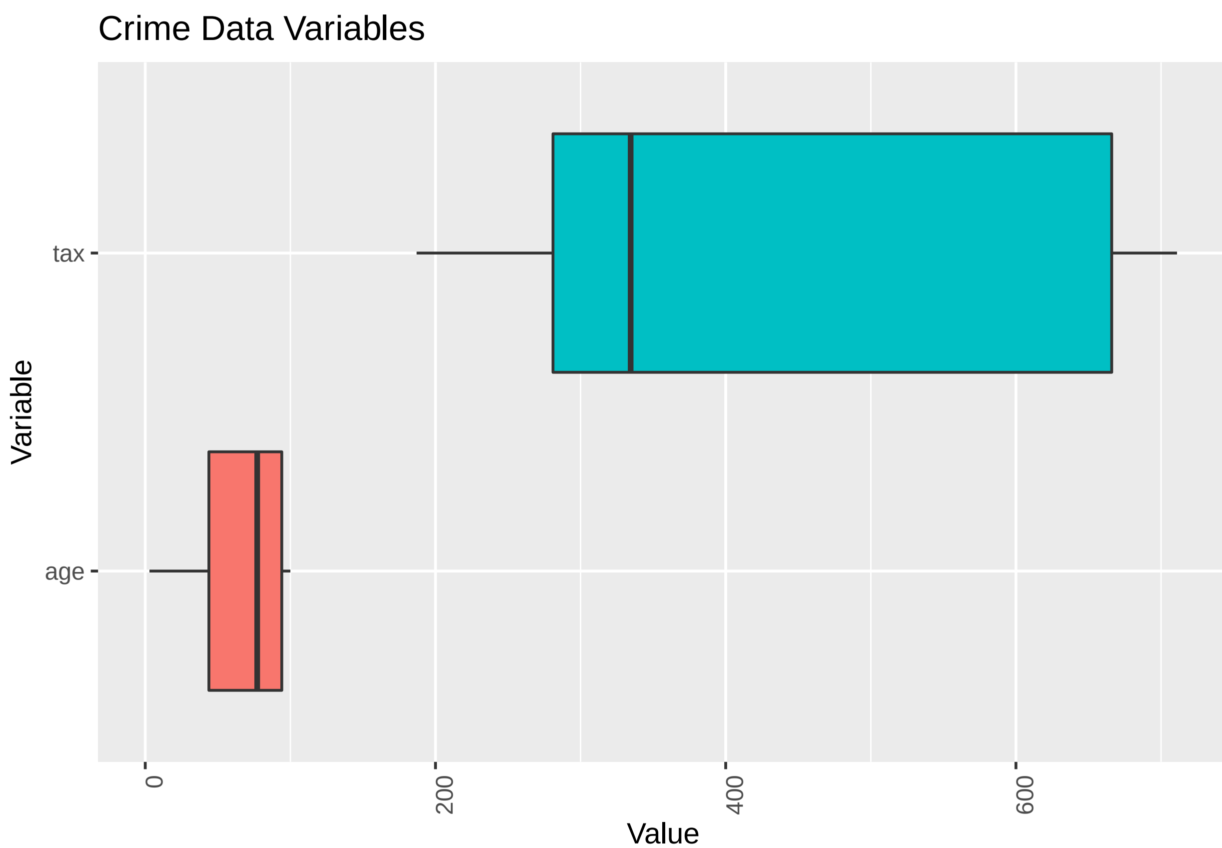
Box Plots

```
#make long
#tax and age has a much different scale so we are seperating it here
longData <- rawTrain %>%
  select(-tax, -age) %>%
  gather(key = Variable, value = Value)

# generate boxplot to identify outliers
ggplot(longData, aes(Variable, Value, fill = Variable)) +
  geom_boxplot(outlier.colour="blue",
               outlier.shape=4,
               outlier.size=2,
               show.legend=FALSE) +
   theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
   coord_flip()+
  labs(title="Crime Data Variables", y="Value")
```




```
#we can see that zn, medv and lstat has MANY outliers
```

```
#make long
#tax and age has a much different scale so we are seperating it here
longData <- rawTrain %>%
  select(tax, age) %>%
  gather(key = Variable, value = Value)

# generate boxplot to identify outliers
ggplot(longData, aes(Variable, Value, fill = Variable)) +
  geom_boxplot(outlier.colour="blue",
               outlier.shape=4,
               outlier.size=2,
               show.legend=FALSE) +
   theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
   coord_flip()+
  labs(title="Crime Data Variables", y="Value")
```




```r
# no outliers for tax and age
```

```r
#Train/Test Split
dt = sort(sample(nrow(rawTrain), nrow(rawTrain)*.8))
train<-rawTrain[dt,]
test<-rawTrain[-dt,]
```

Model Building

```r
#remove Tax due to high correlation with other variables
modelOne <- glm(target ~ zn + indus + chas + nox + rm + age + dis + rad + ptratio + lstat + medv , data

modelOne
```

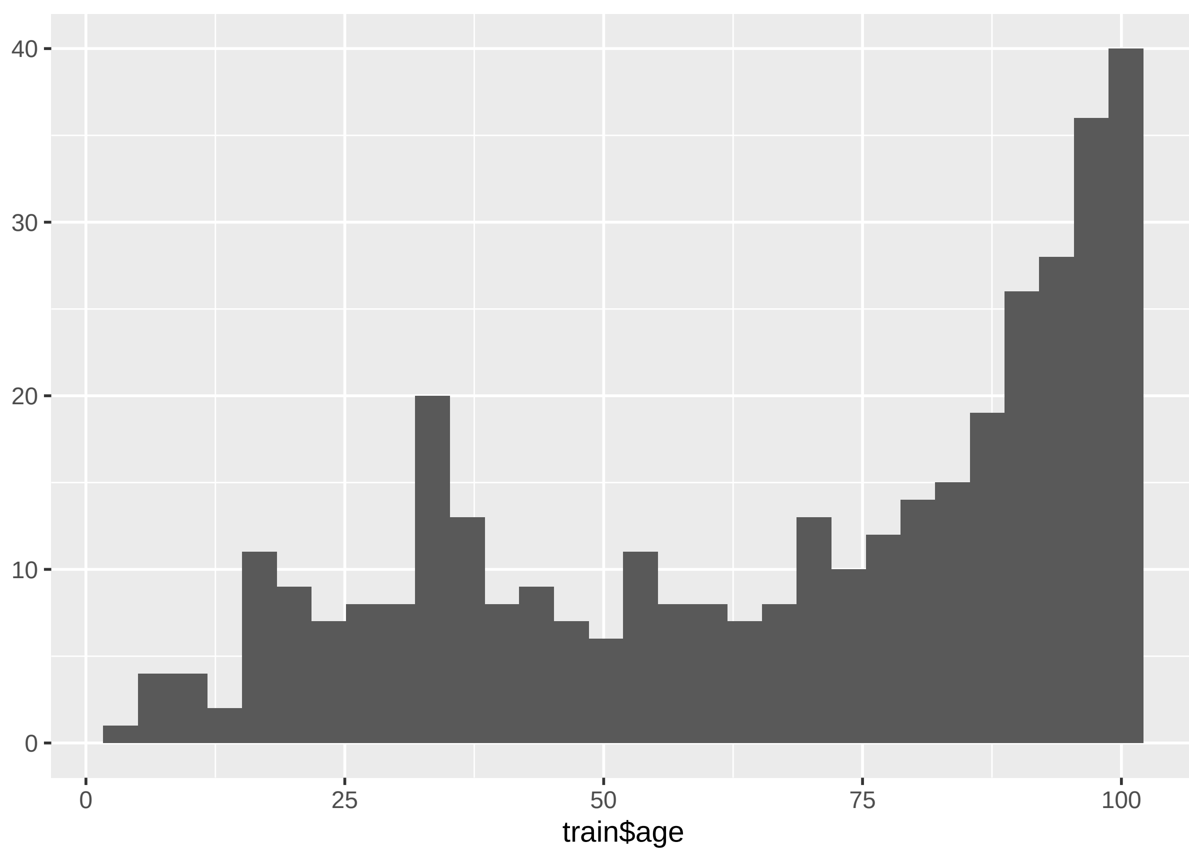
```
##
## Call:  glm(formula = target ~ zn + indus + chas + nox + rm + age + dis +
##     rad + ptratio + lstat + medv, family = "binomial", data = train)
##
## Coefficients:
## (Intercept)           zn        indus         chas          nox           rm
##   -36.68816     -0.06465     -0.04780      1.18303     40.32164     -0.56808
##         age          dis          rad      ptratio        lstat         medv
```

```
##     0.02676      0.71335      0.45953      0.36870      0.06568      0.19556
##
## Degrees of Freedom: 371 Total (i.e. Null);  360 Residual
## Null Deviance:       515
## Residual Deviance: 170.1     AIC: 194.1
```

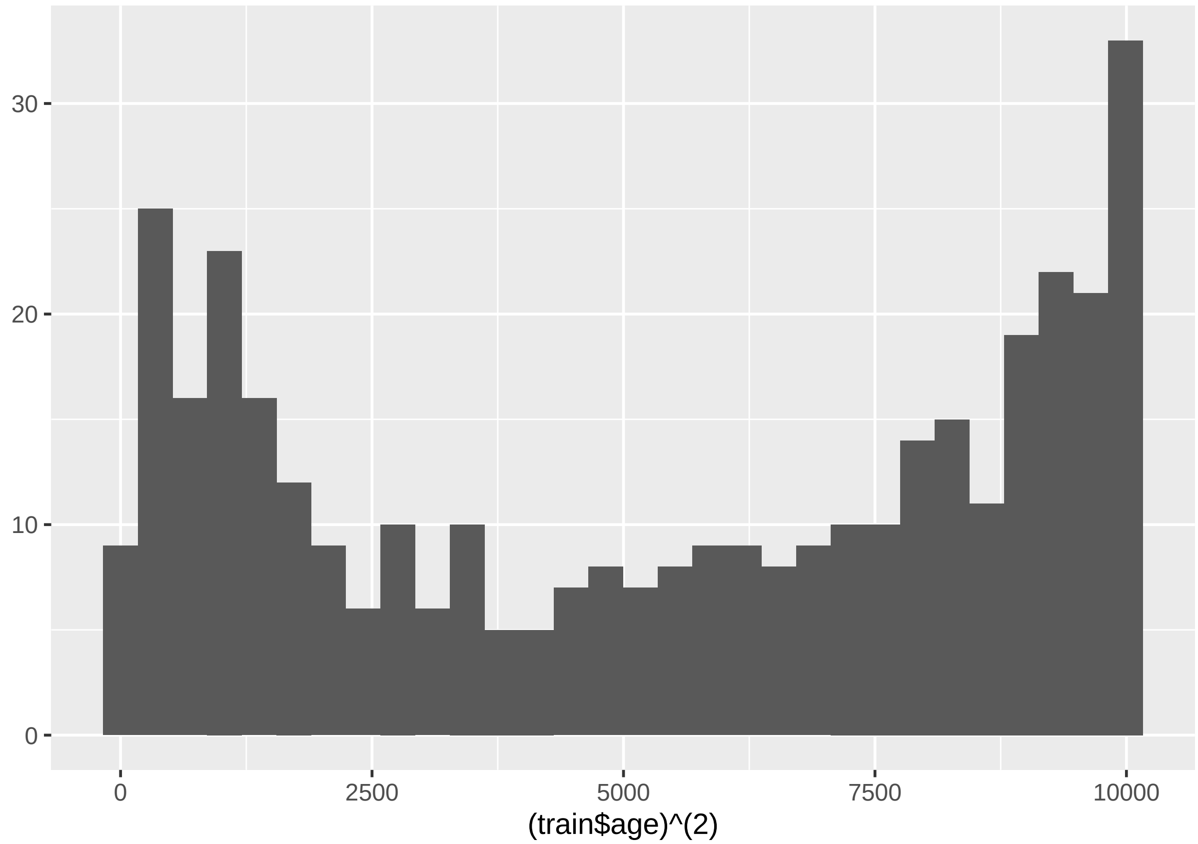
```r
# squared transformation to age and lstat

#age before squared
summary(train$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.90   40.95   76.60   67.61   93.83  100.00
```

```r
#age before squared
qplot(train$age)
```



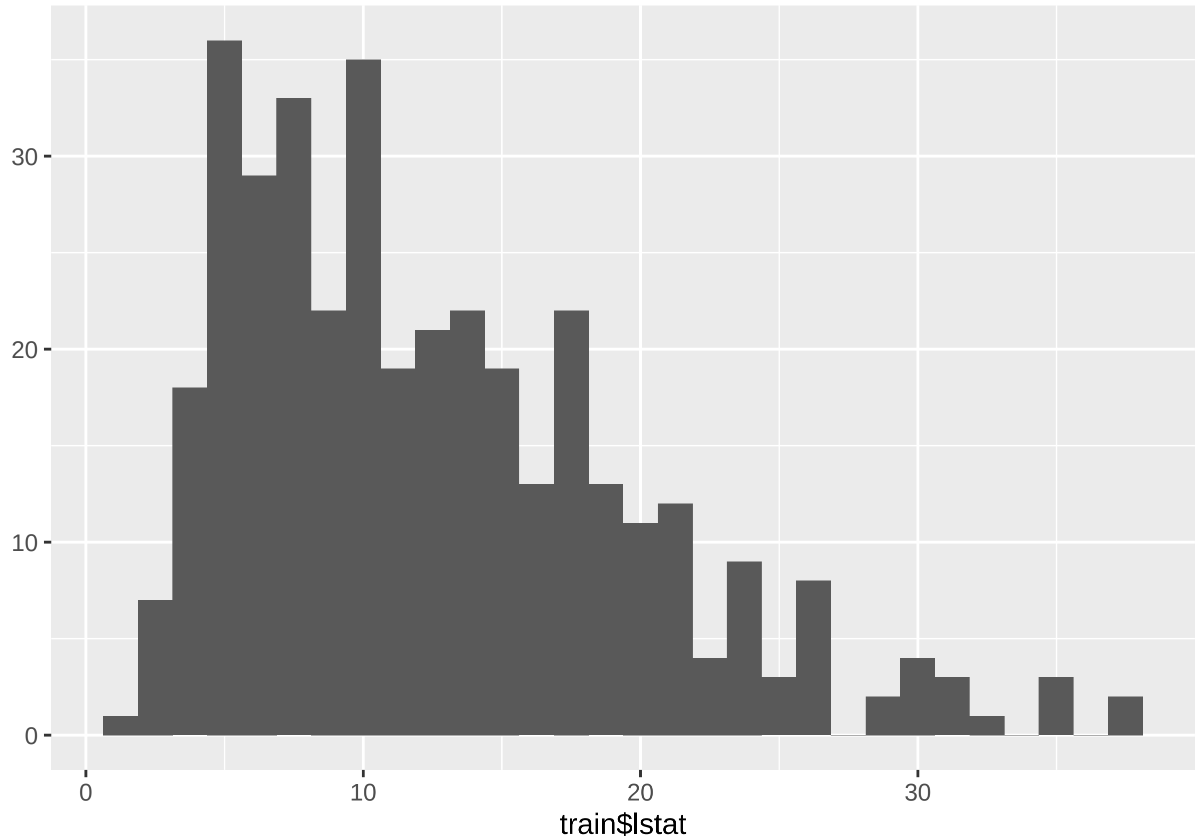

```r
#age after squared
qplot((train$age)^(2))
```



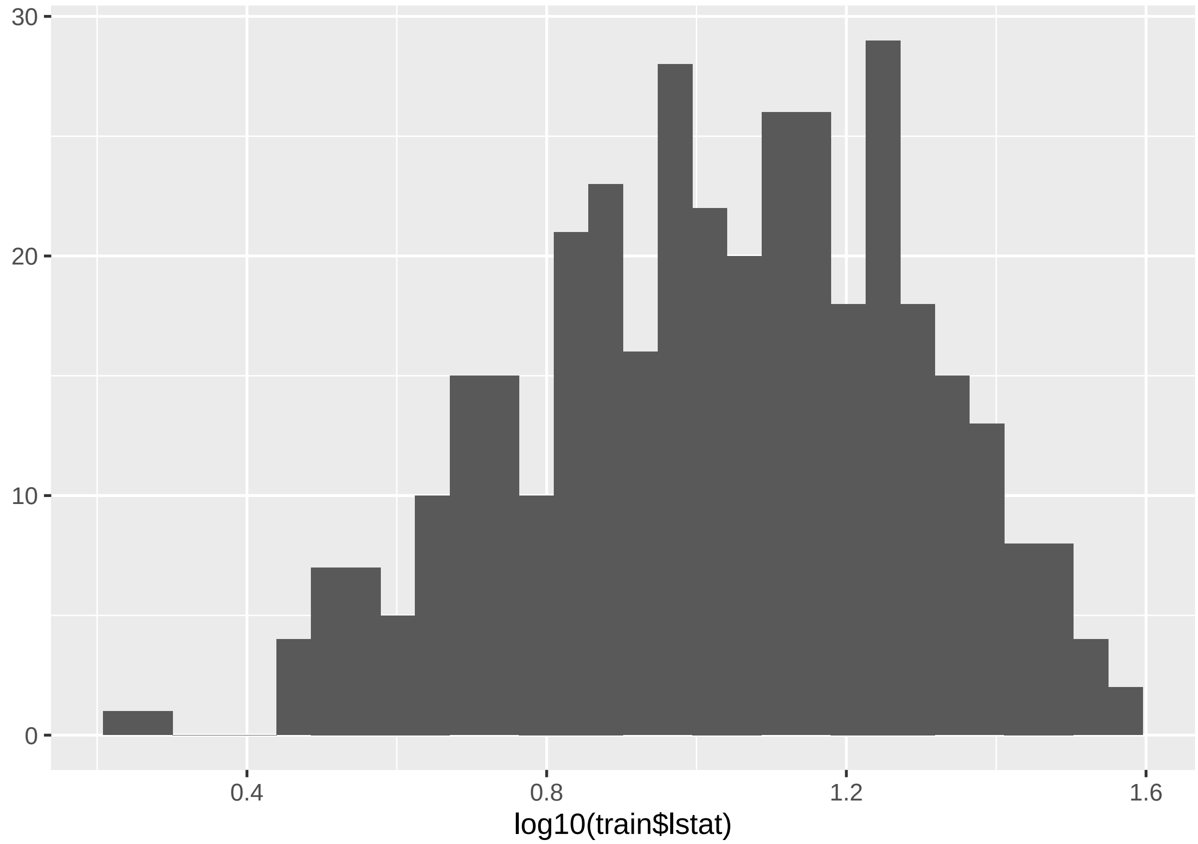

```r
#lstat before log
summary(train$lstat)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.730   6.928  11.110  12.586  17.093  37.970
```

```r
#lstat before log
qplot(train$lstat)
```




```r
#lstat afterlog
qplot(log10(train$lstat))
```




```
#remove Tax squared age and log lstat
modelTwo <- glm(target ~ zn + indus + chas + nox + rm + age^2 + dis + rad + ptratio + log10(lstat) + med

modelTwo
```

```
##
## Call:  glm(formula = target ~ zn + indus + chas + nox + rm + age^2 +
##     dis + rad + ptratio + log10(lstat) + medv, family = "binomial",
##     data = train)
##
## Coefficients:
##  (Intercept)            zn         indus          chas           nox
##    -36.02179      -0.05964      -0.04500       1.27324      40.86438
##           rm           age           dis           rad       ptratio
##     -0.81350       0.03202       0.73546       0.46057       0.38027
## log10(lstat)          medv
##      0.53610       0.19918
##
## Degrees of Freedom: 371 Total (i.e. Null);  360 Residual
## Null Deviance:       515
## Residual Deviance: 171.3     AIC: 195.3
```

```
#This one has a litter lower AIC
```

```r
#remove Tax squared age and log lstat - log dis and zn +1
modelThree <- glm(target ~ log10(zn + 1) + indus + chas + nox + rm + age^2 + log10(dis) + rad + ptratio

modelThree
```

```
##
## Call:  glm(formula = target ~ log10(zn + 1) + indus + chas + nox + rm +
##     age^2 + log10(dis) + rad + ptratio + log10(lstat) + medv,
##     family = "binomial", data = train)
##
## Coefficients:
##   (Intercept)  log10(zn + 1)          indus           chas            nox
##    -44.047678      -0.972791      -0.007393       1.079595      46.414029
##            rm            age     log10(dis)            rad        ptratio
##     -0.888001       0.039243       9.603166       0.497630       0.418534
##  log10(lstat)           medv
##      0.929415       0.241107
##
## Degrees of Freedom: 371 Total (i.e. Null);  360 Residual
## Null Deviance:       515
## Residual Deviance: 164.5     AIC: 188.5
```

```r
#AIC is lower again
```

```r
#add lstat*age
modelFour <- glm(target ~ log10(zn+ 1) + indus  + nox + rm  + log10(dis) + rad + ptratio + medv + lstat

modelFour
```

```
##
## Call:  glm(formula = target ~ log10(zn + 1) + indus + nox + rm + log10(dis) +
##     rad + ptratio + medv + lstat * age + age^2 + log10(lstat) +
##     chas, family = "binomial", data = rawTrain)
##
## Coefficients:
##   (Intercept)  log10(zn + 1)          indus            nox             rm
##    -47.867344      -0.971631      -0.079895      57.940008      -0.955576
##    log10(dis)            rad        ptratio           medv          lstat
##     10.727914       0.600422       0.455900       0.211453       0.533072
##           age   log10(lstat)           chas      lstat:age
##      0.082900      -7.758868       1.178626      -0.003236
##
## Degrees of Freedom: 465 Total (i.e. Null);  452 Residual
## Null Deviance:       645.9
## Residual Deviance: 181.2     AIC: 209.2
```

```r
#Here I decided to take lstat and age and multiply them because age is highly correlated and lstat is s
```

Test Models

```r
#Make predictions
predOne = predict(modelOne,test, type = "response")
predTwo = predict(modelTwo,test, type = "response")
predThree = predict(modelThree,test, type = "response")
predFour = predict(modelFour,test, type = "response")

#measure accuracy
postResample(pred = predOne, obs = test$target)
```

```
##      RMSE  Rsquared       MAE
## 0.2249477 0.7997835 0.1203042
```

```r
#measure accuracy
postResample(pred = predTwo, obs = test$target)
```

```
##      RMSE  Rsquared       MAE
## 0.2170584 0.8144284 0.1160138
```

```r
#measure accuracy
postResample(pred = predThree, obs = test$target)
```

```
##      RMSE  Rsquared       MAE
## 0.2141277 0.8189267 0.1118686
```

```r
#measure accuracy
postResample(pred = predFour, obs = test$target)
```

```
##       RMSE   Rsquared        MAE
## 0.17975252 0.87386770 0.09000409
```

Confusion Matric and Accuracy Measurment

```r
resultsFit<- ifelse(predOne > 0.5,1,0)
resultsFit <- as.factor(resultsFit)
#confusionMatrix(test$target, resultsFit)
resultsFit
```

```
##   1   4   7  18  25  43  47  49  66  67  77  79  81  92  94 104 109 111 116 117
##   1   0   1   1   0   0   1   1   1   0   1   0   0   0   0   0   1   1   0   1
## 119 122 124 130 146 151 153 154 161 167 181 182 183 184 186 190 195 200 221 227
##   1   0   1   0   1   1   0   1   1   0   0   0   1   0   1   1   1   1   1   1
## 228 229 231 232 243 244 257 260 262 263 265 267 271 276 284 286 295 298 301 303
##   1   1   1   1   1   1   0   1   0   1   1   1   1   1   1   1   1   1   0   0
## 312 314 324 327 331 348 349 351 352 361 363 365 366 372 379 387 391 393 396 400
##   1   0   0   0   1   0   0   0   0   0   0   1   0   0   1   1   0   1   0   1
## 405 410 411 416 419 425 426 437 441 442 454 455 460 464
##   1   0   0   0   1   0   0   0   0   1   1   1   0   1
## Levels: 0 1
```

Anova Tests for each model

```r
#Looking at strength of variables
anova(modelOne, test = 'Chisq')
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: target
##
## Terms added sequentially (first to last)
##
##
##         Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                      371     515.01
## zn       1  101.237       370     413.78 < 2.2e-16 ***
## indus    1   85.715       369     328.06 < 2.2e-16 ***
## chas     1    2.602       368     325.46  0.106754
## nox      1   99.035       367     226.42 < 2.2e-16 ***
## rm       1    1.391       366     225.03  0.238154
## age      1    0.080       365     224.95  0.777458
## dis      1    5.701       364     219.25  0.016957 *
## rad      1   34.639       363     184.61 3.969e-09 ***
## ptratio  1    3.524       362     181.09  0.060475 .
## lstat    1    1.459       361     179.63  0.227119
## medv     1    9.503       360     170.13  0.002051 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Looking at strength of variables
anova(modelTwo, test = 'Chisq')
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: target
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                           371     515.01
## zn            1  101.237       370     413.78 < 2.2e-16 ***
## indus         1   85.715       369     328.06 < 2.2e-16 ***
## chas          1    2.602       368     325.46  0.106754
## nox           1   99.035       367     226.42 < 2.2e-16 ***
## rm            1    1.391       366     225.03  0.238154
## age           1    0.080       365     224.95  0.777458
## dis           1    5.701       364     219.25  0.016957 *
## rad           1   34.639       363     184.61 3.969e-09 ***
```

```
## ptratio       1    3.524       362     181.09  0.060475 .
## log10(lstat)  1    0.016       361     181.07  0.898119
## medv          1    9.727       360     171.35  0.001816 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Looking at strength of variables
anova(modelThree, test = 'Chisq')
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: target
##
## Terms added sequentially (first to last)
##
##
##                Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                             371     515.01
## log10(zn + 1)  1   93.354       370     421.66 < 2.2e-16 ***
## indus          1   88.185       369     333.47 < 2.2e-16 ***
## chas           1    2.474       368     331.00 0.1157461
## nox            1  104.010       367     226.99 < 2.2e-16 ***
## rm             1    1.301       366     225.69 0.2539696
## age            1    0.136       365     225.55 0.7122263
## log10(dis)     1    8.262       364     217.29 0.0040484 **
## rad            1   36.959       363     180.33 1.206e-09 ***
## ptratio        1    3.041       362     177.29 0.0811920 .
## log10(lstat)   1    0.000       361     177.29 0.9962682
## medv           1   12.827       360     164.46 0.0003417 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

NEXT I WANT TO TRY BOX COX TRANSFORMATIONS on things we deleted?

```r
#Looking at strength of variables (now we have all strong variables)
anova(modelFour, test = 'Chisq')
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: target
##
## Terms added sequentially (first to last)
##
##
##                Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                             465     645.88
## log10(zn + 1)  1  116.753       464     529.12 < 2.2e-16 ***
## indus          1   89.384       463     439.74 < 2.2e-16 ***
## nox            1  155.463       462     284.28 < 2.2e-16 ***
```

```
## rm             1    7.067       461     277.21  0.007852 **
## log10(dis)     1    9.104       460     268.10  0.002550 **
## rad            1   55.030       459     213.07 1.187e-13 ***
## ptratio        1    1.954       458     211.12  0.162162
## medv           1    3.969       457     207.15  0.046359 *
## lstat          1    7.083       456     200.07  0.007781 **
## age            1    9.544       455     190.53  0.002006 **
## log10(lstat)   1    1.693       454     188.83  0.193158
## chas           1    2.154       453     186.68  0.142243
## lstat:age      1    5.446       452     181.23  0.019609 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

WEE NEED QQ PLOTS AND ACCURACY

AUC or ROC curve