hw3

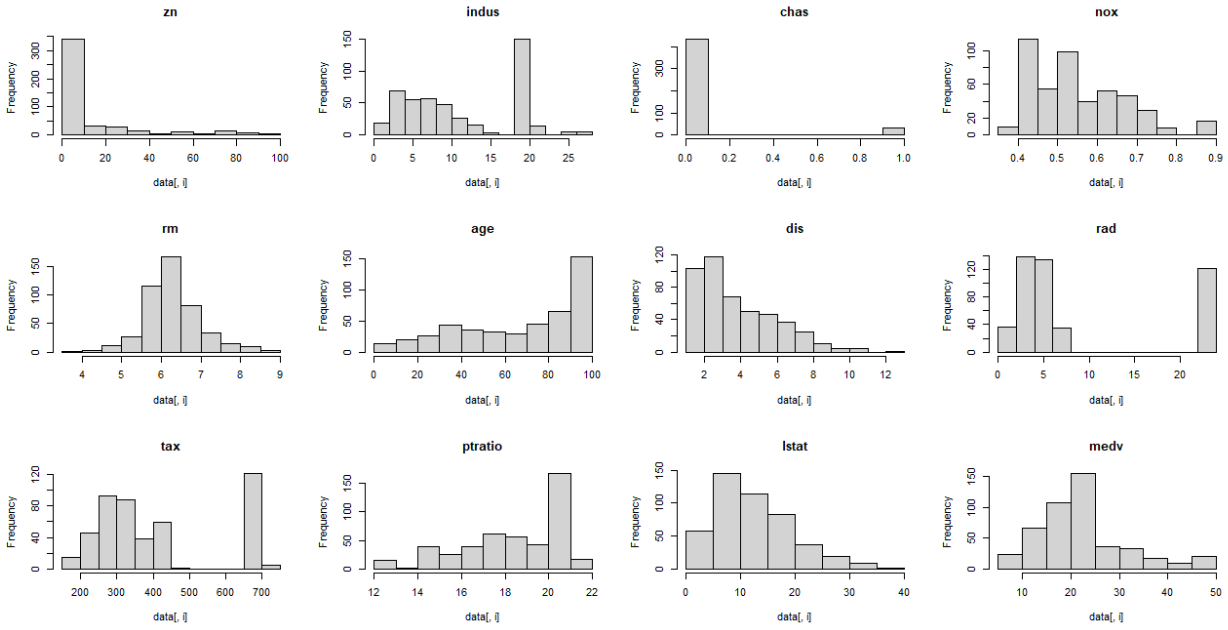
David Moste, Sadia Perveen, Vanita Thompson

4/15/2021

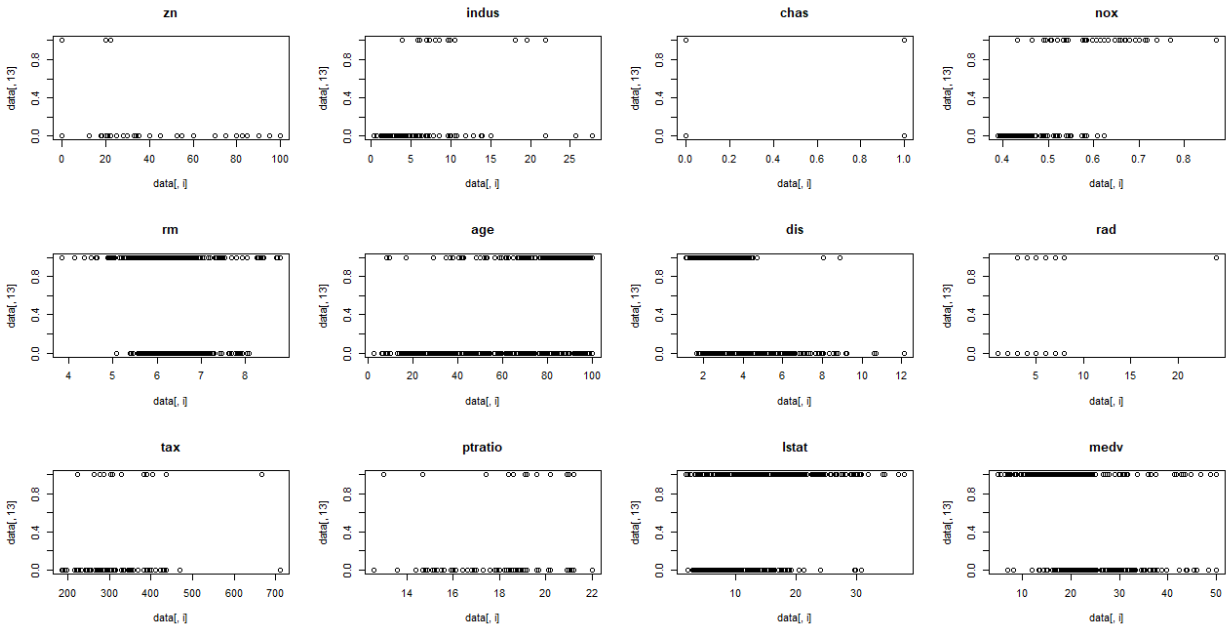
Data Exploration

After grabbing the data, we first checked out a summary of the data to see the predictor variables provided along with their summary statistics. This also allowed us to check if there was any missing data, which there was not.

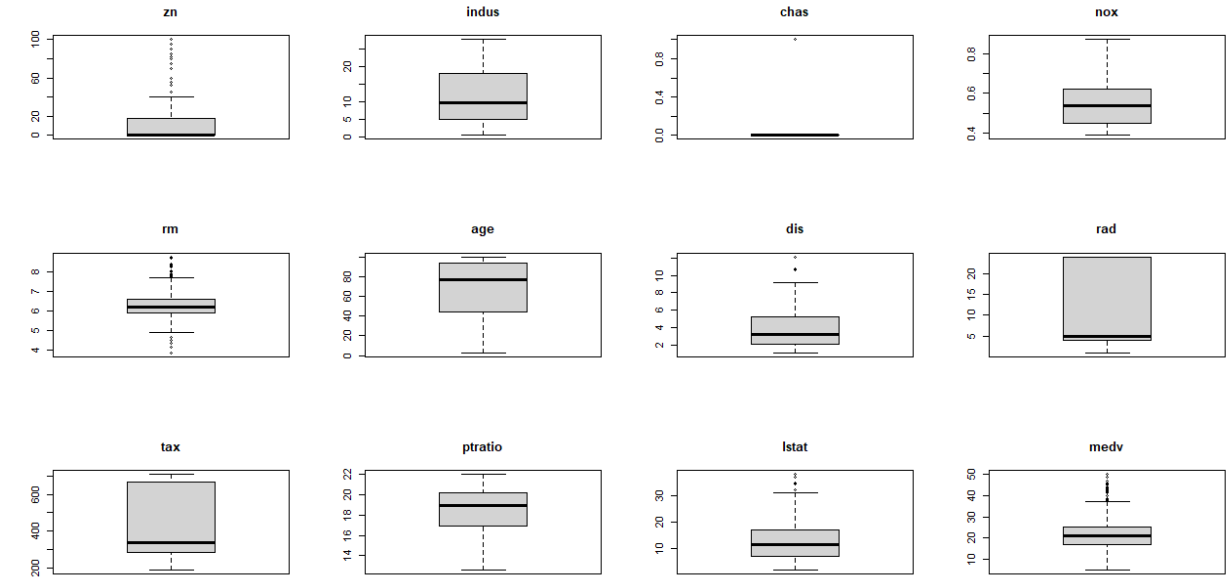
We then created three plots: a histogram, a scatter plot, and a boxplot.



The purpose of the histogram was to get a sense of the normality of each variable. Upon looking at the histogram, it was easy to see that zn, dis, lstat, nox, and age were skewed and would need to be transformed.



The purpose of this scatter plot was to get a sense of any relationship between each variable and the target.



The purpose of the boxplot was to see the data in another light and to get a sense of where there were outliers. It was easy to see at this point that dis and zn contained a bunch of outliers at the top of the range.

Data Preparation

To start data preparation, we performed a few transformations. In order to correct some skewing, we performed a log transformation on the zn, dis, lstat, and nox predictors and a cube root transformation on the age predictor.

Build Models

The first model we built was simply every variable in the data. The AIC for this model was 196.65.

The second model we built was done through backward elimination. The AIC for this model was 187.97.

The final model we built was based on hand-picked variables. The AIC for this model was 290.96.

Select Models

Based on the AIC values, we chose to pursue the second model. When we ran our model on the training data, we recorded the following values:

Metric	Value
Accuracy	0.9313
Classification Error	0.9311
Precision	0.9218
Sensitivity	0.9451
Specificity	0.9170
F1	0.9333
AUC	0.9797

”	0	1
0	224	19
1	13	210

Appendix

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------ tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts --------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.4

## Loading required package: lattice
```

```
##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.0.4

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
# Read in data and view summary statistics
data <- read.csv("https://raw.githubusercontent.com/dmoste/DATA621/master/hw3/crime-training-data_modifi
head(data)
```

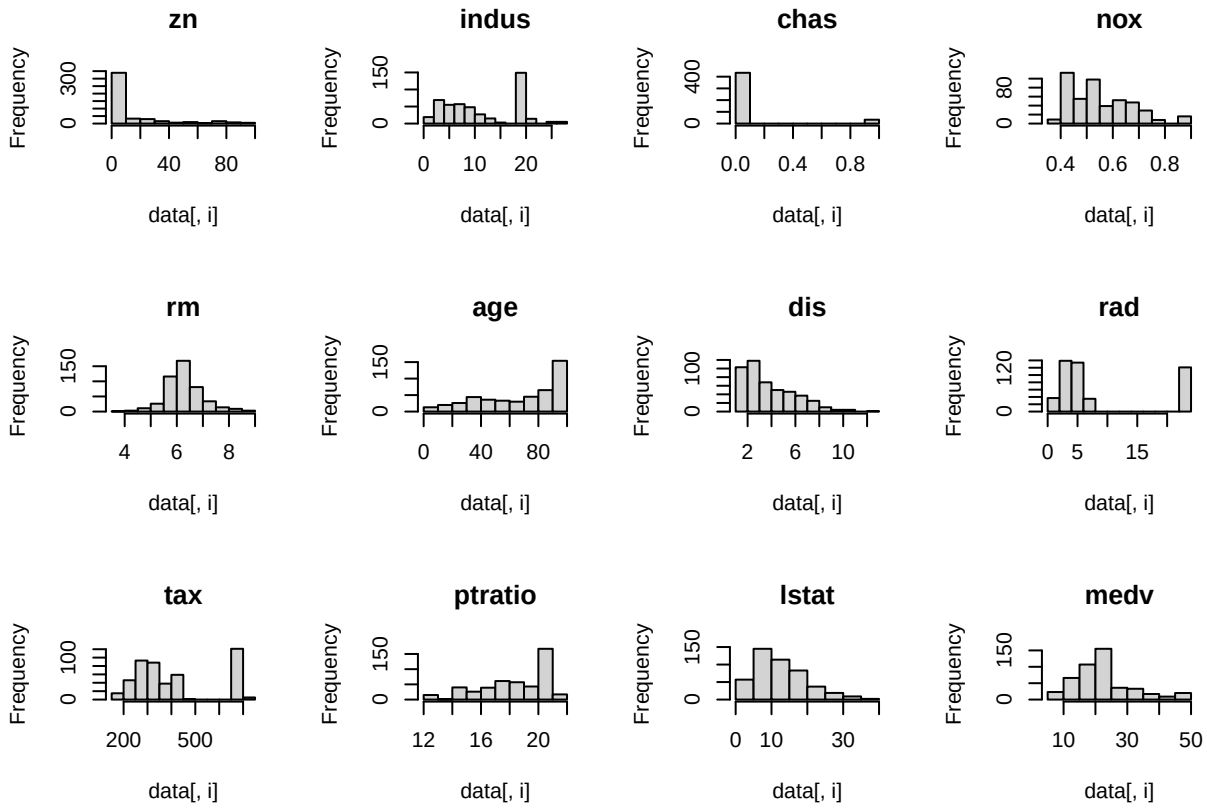
```
##   zn indus chas   nox    rm   age    dis rad tax ptratio lstat medv target
## 1  0 19.58    0 0.605 7.929  96.2 2.0459   5 403    14.7  3.70 50.0      1
## 2  0 19.58    1 0.871 5.403 100.0 1.3216   5 403    14.7 26.82 13.4      1
## 3  0 18.10    0 0.740 6.485 100.0 1.9784  24 666    20.2 18.85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

```
summary(data)
```

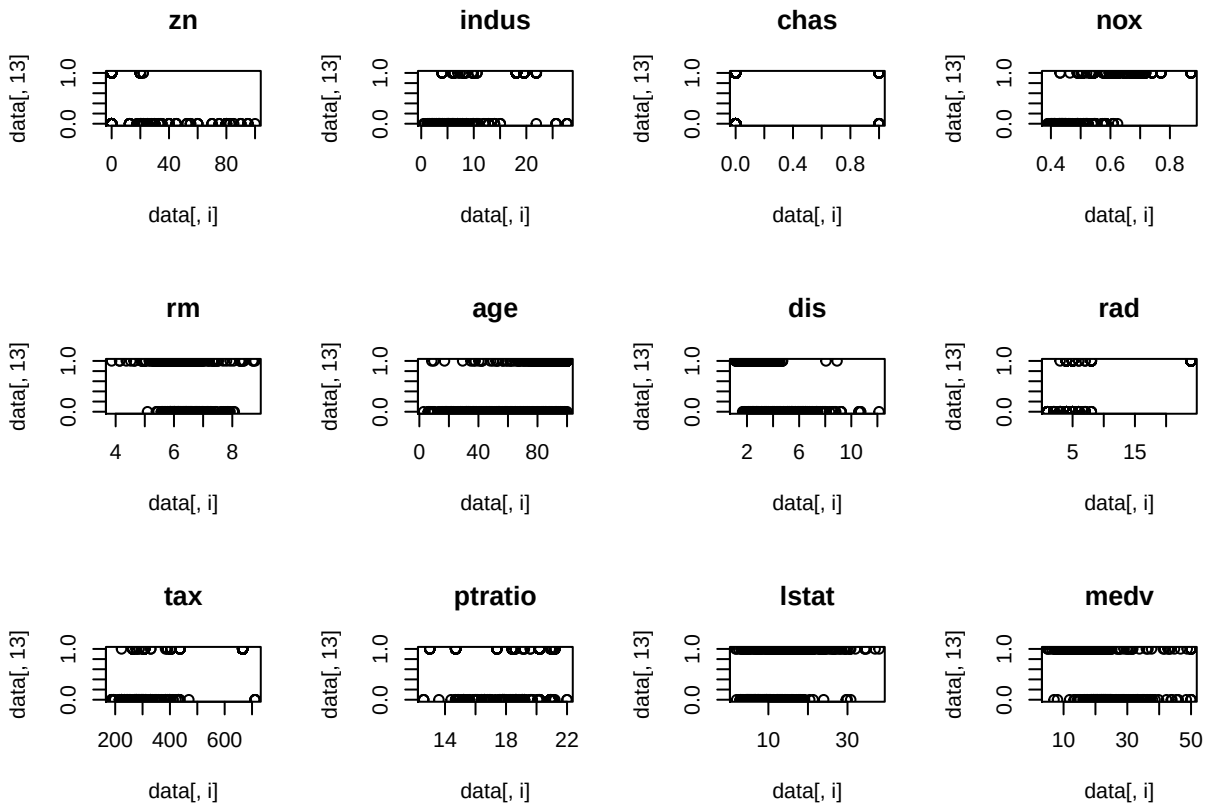
```
##        zn              indus             chas              nox
##  Min.   :  0.00   Min.   : 0.460   Min.   :0.00000   Min.   :0.3890
##  1st Qu.:  0.00   1st Qu.: 5.145   1st Qu.:0.00000   1st Qu.:0.4480
##  Median :  0.00   Median : 9.690   Median :0.00000   Median :0.5380
##  Mean   : 11.58   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         lstat             medv
##  Min.   :187.0   Min.   :12.6   Min.   : 1.730   Min.   : 5.00
##  1st Qu.:281.0   1st Qu.:16.9   1st Qu.: 7.043   1st Qu.:17.02
```

```
##  Median :334.5   Median :18.9   Median :11.350   Median :21.20
##  Mean   :409.5   Mean   :18.4   Mean   :12.631   Mean   :22.59
##  3rd Qu.:666.0   3rd Qu.:20.2   3rd Qu.:16.930   3rd Qu.:25.00
##  Max.   :711.0   Max.   :22.0   Max.   :37.970   Max.   :50.00
##      target
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.4914
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

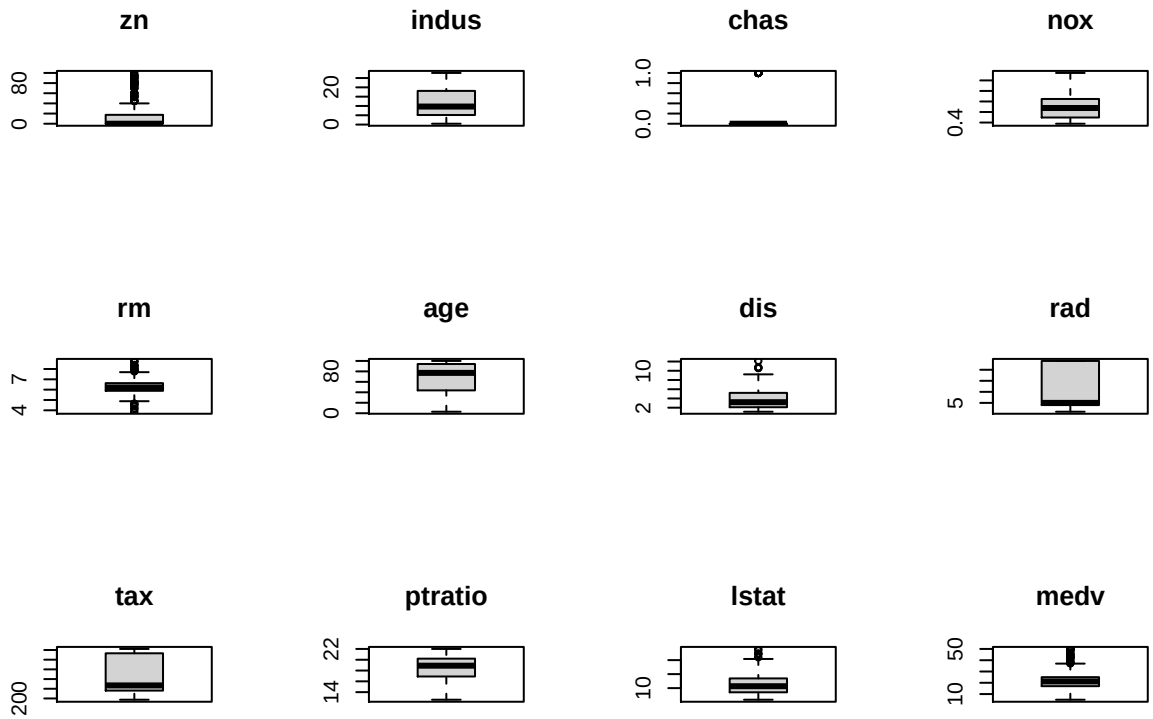
```r
par(mfrow = c(3,4))
# View histogram for all data
for(i in 1:12) {
  hist(data[,i], main = names(data)[i])
}
```




```r
# View scatter plot for all data
for(i in 1:12) {
  plot(x = data[,i], y = data[,13], main = names(data)[i])
}
```




```
# View boxplot for all data
for(i in 1:12) {
  boxplot(x = data[,i], y = data[,13], main = names(data)[i])
}
```




```
# Transformation function
my_transform <- function(data){
  data$zn_log <- log(data$zn + 1)
  data$dis_log <- log(data$dis + 1)
  data$lstat_log <- log(data$lstat + 1)
  data$nox_log <- log(data$nox + 1)
  data$age_cr <- (data$age)**(1/3)

  return(data)
}

# Transform data
t_data <- my_transform(data)

# Create first model using all parameters
fit1 <- glm(target ~ ., data = t_data, family = binomial)
```

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

```
summary(fit1)
```

```
##
## Call:
## glm(formula = target ~ ., family = binomial, data = t_data)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9773  -0.1884  -0.0004   0.0001   3.3470
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.960e+01  3.505e+01   1.130 0.258549
## zn          -6.215e-02  1.087e-01  -0.572 0.567495
## indus       -1.707e-02  6.955e-02  -0.245 0.806105
## chas         4.380e-01  7.847e-01   0.558 0.576739
## nox          8.336e+02  3.262e+02   2.555 0.010607 *
## rm          -1.539e+00  8.858e-01  -1.738 0.082270 .
## age          1.836e-01  4.989e-02   3.680 0.000233 ***
## dis         -4.746e+00  1.337e+00  -3.549 0.000387 ***
## rad          9.315e-01  2.068e-01   4.504 6.67e-06 ***
## tax         -7.082e-03  3.472e-03  -2.040 0.041370 *
## ptratio      5.929e-01  1.681e-01   3.526 0.000421 ***
## lstat        1.574e-01  1.448e-01   1.087 0.277054
## medv         2.324e-01  8.543e-02   2.721 0.006514 **
## zn_log       2.540e-01  9.138e-01   0.278 0.781057
## dis_log      2.872e+01  6.875e+00   4.178 2.94e-05 ***
## lstat_log   -2.491e+00  2.091e+00  -1.192 0.233456
## nox_log     -1.183e+03  4.953e+02  -2.389 0.016903 *
## age_cr      -5.557e+00  1.780e+00  -3.122 0.001795 **
## ---
## 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: 160.65  on 448  degrees of freedom
## AIC: 196.65
##
## Number of Fisher Scoring iterations: 10
```

```
# Create second model using backward elimination
fit2 <- glm(target ~ . - indus - zn_log - chas - zn - lstat - lstat_log,
            data = t_data, family = binomial)
```

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

```
summary(fit2)
```

```
##
## Call:
## glm(formula = target ~ . - indus - zn_log - chas - zn - lstat -
##     lstat_log, family = binomial, data = t_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0521  -0.2217  -0.0011   0.0001   3.2508
##
## Coefficients:
```

```
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.964e+01  2.962e+01   0.663 0.507183
## nox          6.992e+02  2.858e+02   2.446 0.014433 *
## rm          -1.656e+00  7.606e-01  -2.177 0.029461 *
## age          2.100e-01  4.476e-02   4.693 2.70e-06 ***
## dis         -4.721e+00  1.144e+00  -4.126 3.69e-05 ***
## rad          9.378e-01  1.785e-01   5.254 1.48e-07 ***
## tax         -7.014e-03  3.122e-03  -2.247 0.024664 *
## ptratio      6.130e-01  1.477e-01   4.149 3.34e-05 ***
## medv         2.735e-01  8.031e-02   3.405 0.000662 ***
## dis_log      2.892e+01  6.184e+00   4.676 2.92e-06 ***
## nox_log     -9.759e+02  4.329e+02  -2.254 0.024174 *
## age_cr      -6.585e+00  1.609e+00  -4.093 4.26e-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: 163.97  on 454  degrees of freedom
## AIC: 187.97
##
## Number of Fisher Scoring iterations: 9
```

```
# Create third model with hand selection
fit3 <- glm(target ~ zn + rm +age + dis + rad + tax + ptratio + medv,
            data = t_data, family = binomial)
summary(fit3)
```

```
##
## Call:
## glm(formula = target ~ zn + rm + age + dis + rad + tax + ptratio +
##     medv, family = binomial, data = t_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.73904  -0.43043  -0.01829   0.01456   2.91223
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.745512   2.609299  -1.052   0.2927
## zn          -0.040881   0.020093  -2.035   0.0419 *
## rm          -0.226798   0.488674  -0.464   0.6426
## age          0.040720   0.009384   4.339 1.43e-05 ***
## dis         -0.338560   0.141597  -2.391   0.0168 *
## rad          0.535605   0.122952   4.356 1.32e-05 ***
## tax         -0.002660   0.002067  -1.287   0.1982
## ptratio     -0.019279   0.080818  -0.239   0.8115
## medv         0.034165   0.045079   0.758   0.4485
## ---
## 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: 272.96  on 457  degrees of freedom
## AIC: 290.96
##
## Number of Fisher Scoring iterations: 8
```

```r
# Make predictions from training data and add to the transformed data
likely <- predict(fit2, t_data, type = "response")
pred <- ifelse(likely > 0.5, 1, 0)
t_data <- cbind(t_data, likely, pred)

# Convert class and scored.class into factors for use with caret
t_data$target <- as.factor(t_data$target)
t_data$pred <- as.factor(t_data$pred)

# Calculate the precision with caret
caret::precision(data = t_data$pred,
                 reference = t_data$target,
                 positive = 1)
```

```
## [1] 0.9218107
```

```r
# Calculate the sensitivity with caret
caret::sensitivity(data = t_data$pred,
                   reference = t_data$target,
                   positive = 1)
```

```
## [1] 0.9170306
```

```r
# Calculate the specificity with caret
caret::specificity(data = t_data$pred,
                   reference = t_data$target,
                   negative = 0)
```

```
## [1] 0.9451477
```

```r
# Calculate the F1 with caret
caret::F_meas(data = t_data$pred,
              reference = t_data$target,
              negative = 0)
```

```
## [1] 0.9333333
```

```r
# Use pROC to obtain an roc curve
rocCurve <- pROC::roc(response = t_data$target,
                      predictor = t_data$likely)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```r
auc(rocCurve)
```

```
## Area under the curve: 0.9797
```

```r
# Produce a confusion matrix with caret
caret::confusionMatrix(data = t_data$pred,
                       reference = t_data$target,
                       mode = 'everything')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 224  19
##          1  13 210
##
##                Accuracy : 0.9313
##                  95% CI : (0.9044, 0.9526)
##     No Information Rate : 0.5086
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8626
##
##  Mcnemar's Test P-Value : 0.3768
##
##             Sensitivity : 0.9451
##             Specificity : 0.9170
##          Pos Pred Value : 0.9218
##          Neg Pred Value : 0.9417
##               Precision : 0.9218
##                  Recall : 0.9451
##                      F1 : 0.9333
##              Prevalence : 0.5086
##          Detection Rate : 0.4807
##    Detection Prevalence : 0.5215
##       Balanced Accuracy : 0.9311
##
##        'Positive' Class : 0
##
```

```r
# Read in evaluation data and transform in the same way as training data
eval <- read.csv("https://raw.githubusercontent.com/dmoste/DATA621/master/hw3/crime-evaluation-data_modi
t_eval <- my_transform(eval)

# Make predictions on the evaluation data
eval_pred <- predict(fit1, t_eval, type = "response")
eval_pred <- ifelse(eval_pred > 0.5, 1, 0)
```