Data 621 Homework 3: Code Appendix

Jeff Nieman, Scott Karr, James Topor, Armenoush Aslanian-Persico

Contents

Part 1. Data Exploration

```
library(bestglm)
library(alr3)
library(car)
library(pROC)
```

```
hw3 <- read.csv("https://raw.githubusercontent.com/jtopor/CUNY-MSDA-621/master/HW-3/crime-training-data

attach(hw3)

summary(hw3)
```

```
##        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
```

```r
nrow(hw3)
```

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

Correlation Matrix of Raw Data

```r
# correlation plot
round(cor(hw3), 2)
```

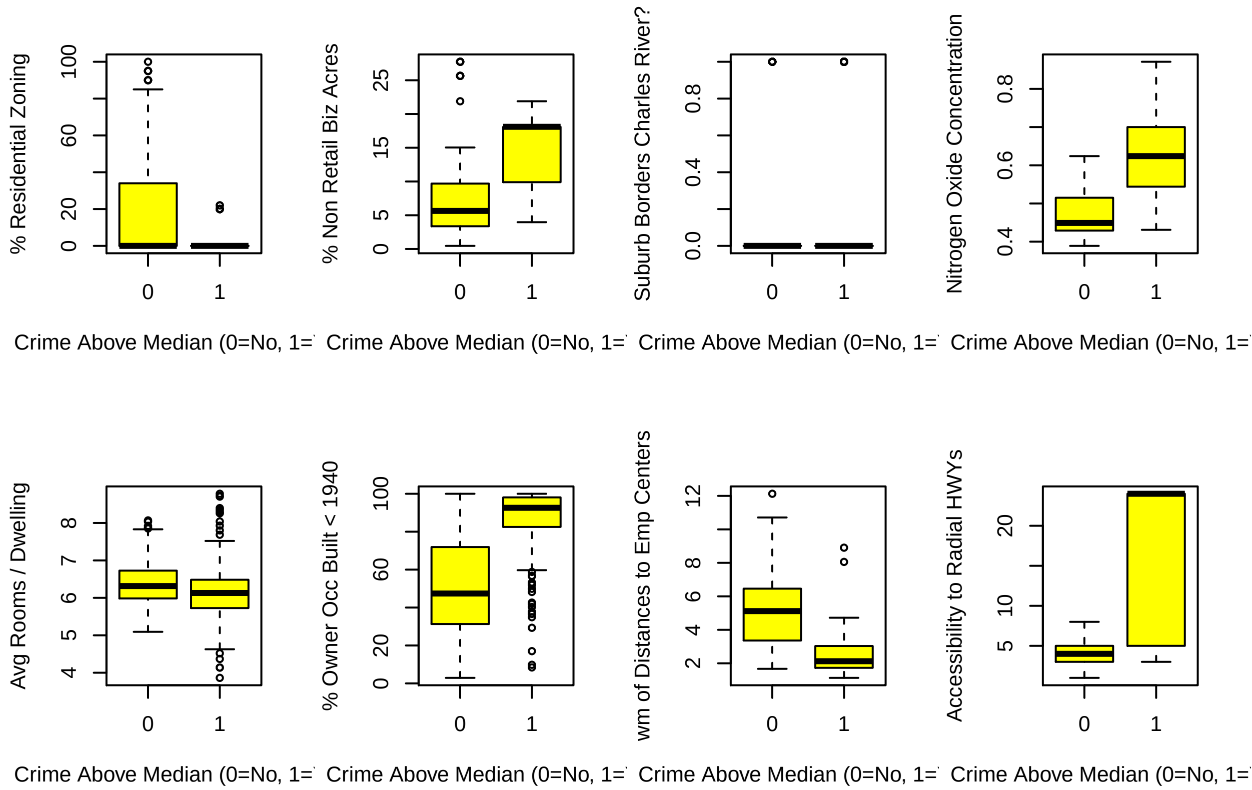
```
##            zn indus  chas   nox    rm   age   dis   rad   tax ptratio
## zn       1.00 -0.54 -0.04 -0.52  0.32 -0.57  0.66 -0.32 -0.32   -0.39
## indus   -0.54  1.00  0.06  0.76 -0.39  0.64 -0.70  0.60  0.73    0.39
## chas    -0.04  0.06  1.00  0.10  0.09  0.08 -0.10 -0.02 -0.05   -0.13
## nox     -0.52  0.76  0.10  1.00 -0.30  0.74 -0.77  0.60  0.65    0.18
## rm       0.32 -0.39  0.09 -0.30  1.00 -0.23  0.20 -0.21 -0.30   -0.36
## age     -0.57  0.64  0.08  0.74 -0.23  1.00 -0.75  0.46  0.51    0.26
## dis      0.66 -0.70 -0.10 -0.77  0.20 -0.75  1.00 -0.49 -0.53   -0.23
## rad     -0.32  0.60 -0.02  0.60 -0.21  0.46 -0.49  1.00  0.91    0.47
## tax     -0.32  0.73 -0.05  0.65 -0.30  0.51 -0.53  0.91  1.00    0.47
## ptratio -0.39  0.39 -0.13  0.18 -0.36  0.26 -0.23  0.47  0.47    1.00
## black    0.18 -0.36  0.04 -0.38  0.13 -0.27  0.29 -0.45 -0.44   -0.18
## lstat   -0.43  0.61 -0.05  0.60 -0.63  0.61 -0.51  0.50  0.56    0.38
## medv     0.38 -0.50  0.16 -0.43  0.71 -0.38  0.26 -0.40 -0.49   -0.52
## target  -0.43  0.60  0.08  0.73 -0.15  0.63 -0.62  0.63  0.61    0.25
##         black lstat  medv target
## zn       0.18 -0.43  0.38  -0.43
## indus   -0.36  0.61 -0.50   0.60
## chas     0.04 -0.05  0.16   0.08
## nox     -0.38  0.60 -0.43   0.73
## rm       0.13 -0.63  0.71  -0.15
## age     -0.27  0.61 -0.38   0.63
## dis      0.29 -0.51  0.26  -0.62
## rad     -0.45  0.50 -0.40   0.63
## tax     -0.44  0.56 -0.49   0.61
## ptratio -0.18  0.38 -0.52   0.25
```

```
## black    1.00 -0.35  0.33  -0.35
## lstat   -0.35  1.00 -0.74   0.47
## medv     0.33 -0.74  1.00  -0.27
## target  -0.35  0.47 -0.27   1.00
```

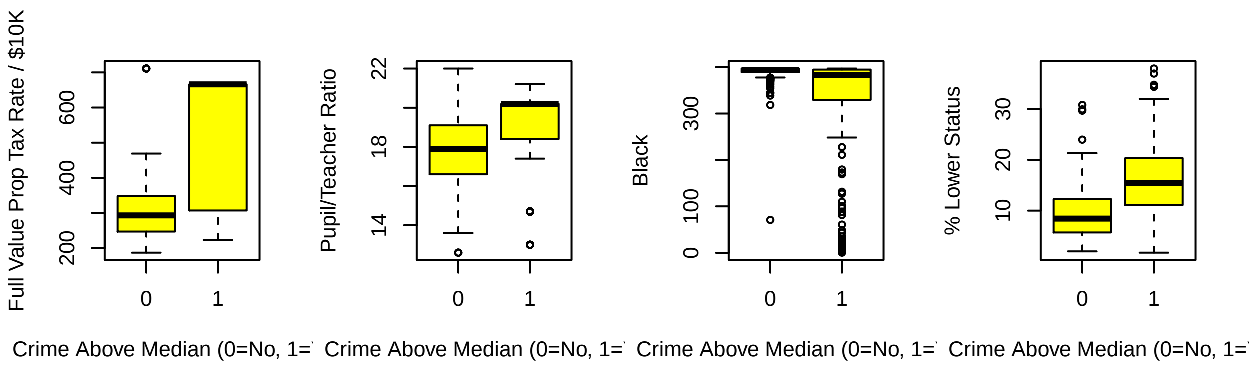
The initial correlation matrix shows some evidence of potential correlation between various variables, with the .91 covariance indicated for the 'rad' and 'tax' variables being of particular note. However, additional data exploration must be conducted before we can conclude that these initial correlations are offering a valid explanation of the training data.

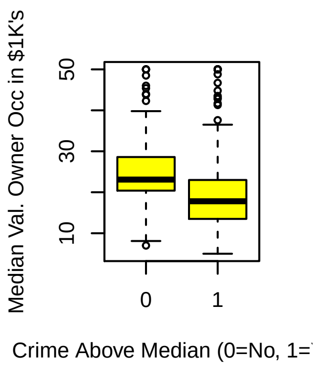
Boxplots of each independent variable relative to the binary response variable are one way in which we can begin to gain insight into the predictive aspects of the training data:

```
# box plots of each predictor variable relative to the response

# See Figure 8.8 on page 286
par(mfrow=c(2,4))

boxplot(zn~target, ylab="% Residential Zoning",
        xlab="Crime Above Median (0=No, 1=Yes)", col = "yellow")

boxplot(indus~target, ylab="% Non Retail Biz Acres",
        xlab="Crime Above Median (0=No, 1=Yes)", col = "yellow")

boxplot(chas~target, ylab="Suburb Borders Charles River?",
        xlab="Crime Above Median (0=No, 1=Yes)", col = "yellow")

boxplot(nox~target, ylab="Nitrogen Oxide Concentration",
        xlab="Crime Above Median (0=No, 1=Yes)", col = "yellow")

boxplot(rm~target, ylab="Avg Rooms / Dwelling",
        xlab="Crime Above Median (0=No, 1=Yes)", col = "yellow")

boxplot(age~target, ylab="% Owner Occ Built < 1940",
        xlab="Crime Above Median (0=No, 1=Yes)", col = "yellow")

boxplot(dis~target, ylab="wm of Distances to Emp Centers",
        xlab="Crime Above Median (0=No, 1=Yes)", col = "yellow")

boxplot(rad~target, ylab="Accessibility to Radial HWYs",
        xlab="Crime Above Median (0=No, 1=Yes)", col = "yellow")
```




```
# ---------------------
par(mfrow=c(2,4))

boxplot(tax~target, ylab="Full Value Prop Tax Rate / $10K",
        xlab="Crime Above Median (0=No, 1=Yes)", col = "yellow")

boxplot(ptratio~target, ylab="Pupil/Teacher Ratio",
        xlab="Crime Above Median (0=No, 1=Yes)", col = "yellow")

boxplot(black~target, ylab="Black",
        xlab="Crime Above Median (0=No, 1=Yes)", col = "yellow")

boxplot(lstat~target, ylab="% Lower Status",
        xlab="Crime Above Median (0=No, 1=Yes)", col = "yellow")

boxplot(medv~target, ylab="Median Val. Owner Occ in $1K's",
        xlab="Crime Above Median (0=No, 1=Yes)", col = "yellow")

# ------------------------------------------------------------------------------
# comments on individual variables

# - zn
# check count of zn variable = 0 => 72% of records have zn = 0
# maybe change to a binary variable? e.g., has zoning for large lots & doesn't?
# nrow(subset(hw3, hw3$zn == 0)) / nrow(hw3)
```



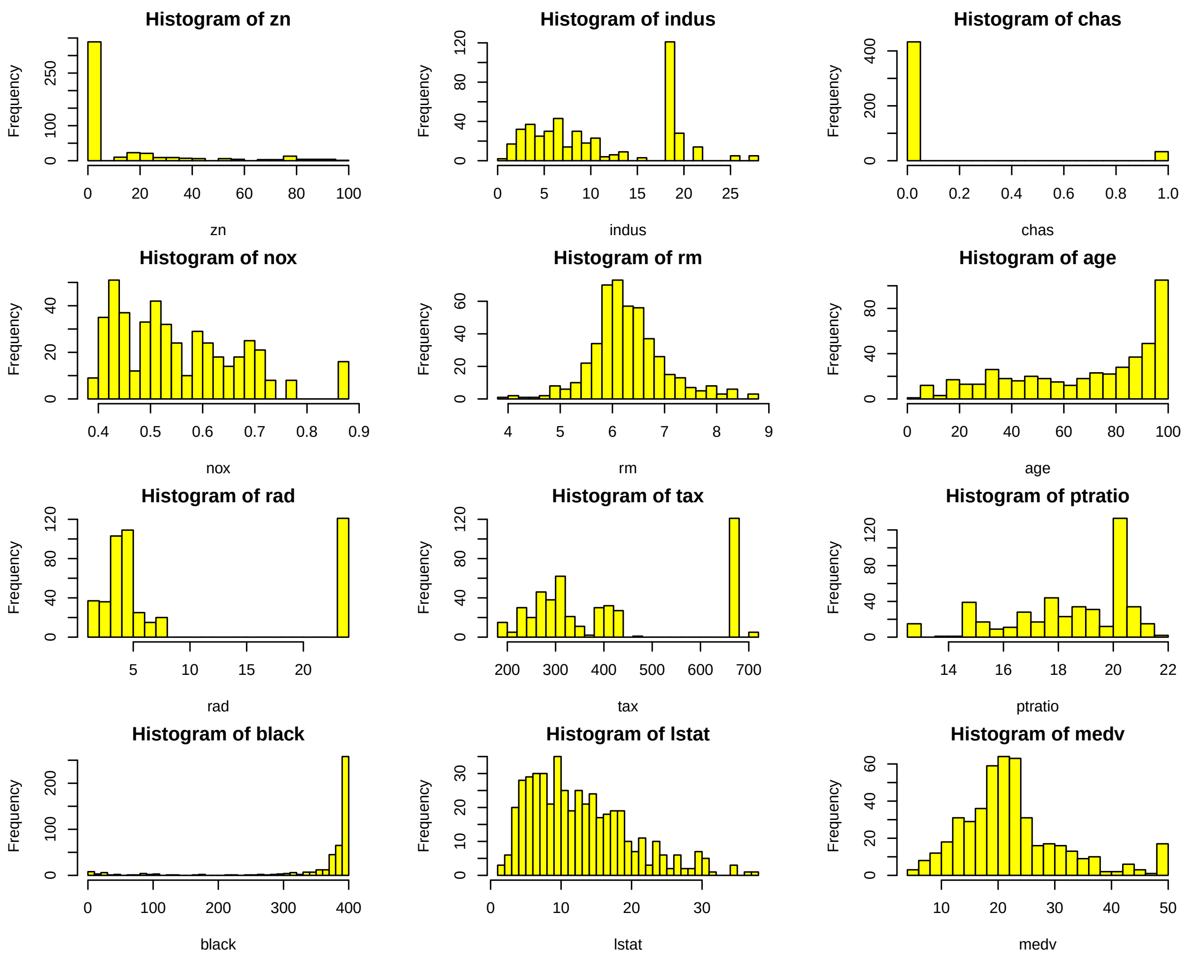


The boxplots show evidence of skew for several of the predictor variables: the 'zon', 'age', 'rad', 'tax', 'ptratio', and 'black' variables each display asymetrical distributions relative to one or both values of the response variable. Such skew may be the result of the presence of outliers or can simply be reflecting the inherent nature of the variable. For example, we would expect the 'zon' variable to be skewed due simply to what it characterizes, namely the proportion of residential land zoned for large lots. Obviously, most typical neighborhoods are unlikely to have been zoned to allow for such large lots so we should expect most instances of the variable to be either equal to zero or to be relatively small numbers.

While boxplots are useful for helping to identify potential skew, histograms allow us to more thoroughly examine whether the distribution of a variable is being dominated by a particular set of data values. Histograms for each of the twelve potential predictor variables are provided below.

```
########################################################################
# Make small histograms for each variable

par(mfrow = c(4,3), oma = c(1, 1, 0, 0), mar=c(2, 2, 0, 1) + 2)


hist(zn, breaks = 30, col = 'yellow')

# - indus
# non-symmetric distribution for TARGET == 1.
# skewed distribution for TARGET == 0

hist(indus, breaks = 30, col = 'yellow')

# - chas
```

```r
# remove from model - not correlated with TARGET
hist(chas, breaks = 30, col = 'yellow')

# - nox
# TARGET == 0 is slightly skewed while TARGET == 1 isn't
hist(nox, breaks = 30, col = 'yellow')

# - rm (rooms)'
# both are reasonably symmetric
hist(rm, breaks = 30, col = 'yellow')

# - age
# TARGET == 1 is skewed
hist(age, breaks = 30, col = 'yellow')


# - rad is skewed for TARGET == 1
hist(rad, breaks = 30, col = 'yellow')

# - tax is skewed for TARGET == 1
hist(tax, breaks = 30, col = 'yellow')

# - ptratio is skewed for TARGET == 1
hist(ptratio, breaks = 30, col = 'yellow')

# - black is skewed
hist(black, breaks = 30, col = 'yellow')

# transform black back to a proportion
# bk <- (sqrt(hw3$black) + 19.92235) / 31.62278

# now transform back to validate
# bk2 <- 1000 * (bk - .63)^2


# lstat
hist(lstat, breaks = 30, col = 'yellow')


# medv
hist(medv, breaks = 30, col = 'yellow')
```



The histograms show that the ‘zon’, ‘chas’, ‘black, ’indus’, ‘rad’, ‘tax’, and ‘ptratio’ variables each have an unusually large number of identical values. Of these, ‘zon’ and ‘chas’ can be explained by their nature: we wouldn’t expect ‘zon’ to have a value greater than zero in most instances and ‘chas’ is a binary categorical variable that can only assume values of either ‘0’ or ‘1’. For the ‘black’ variable it may be the case that many of the neighborhoods represented in the data set have very similar proportions of black residents.

For the ‘indus’, ‘rad’, ‘tax’, and ‘ptratio’ variables, analysis reveals that 121 rows of the training data contain recurring values for each of these variables. The recurring values are summarized below.

Variable	Value
indus	18.1
rad	24
tax	666
ptratio	20.2

In fact, further analysis reveals that for the ‘indus’, ‘rad’, and ‘tax’ variables, the values recorded in those 121 rows are distinct relative to the rest of the training data: no other records within the training data set contain those specific recurring values for the indicated variables.

Variable that might be dropped: CHAS, TAX, (NOX / DIS), (RM / MEDV), (AGE / INDUS)

Part 2 - Data Preparation

```r
# read eval data set
hw3.e <- read.csv("https://raw.githubusercontent.com/jtopor/CUNY-MSDA-621/master/HW-3/crime-evaluation-d

# add dummy variable 'target' to eval data
hw3.e$target <- 0

hw3.t <- hw3
hw3.et <- hw3.e



############## zn transformation ##############################
# 127 zn values > 0
#sum(hw3$zn > 0)

# Transform zn to a binary variable: > 0 = 1 in TRAINING data set
hw3.t$zn[which(hw3$zn > 0)] <- 1
hw3.t$zn <- factor(hw3.t$zn)
# summary(hw3.t$zn)

# ------- eval data set
# 7 zn values > 0 in eval data
# sum(hw3.e$zn > 0)

# Transform zn to a binary variable: > 0 = 1 in EVAL data set
hw3.et$zn[which(hw3.e$zn > 0)] <- 1
hw3.et$zn <- factor(hw3.et$zn)
# summary(hw3.et$zn)

############## age transformation ##############################
# 219 age values > 80
# sum(hw3$age > 80)

# Transform age to a binary variable: > 80 = 1
hw3.t$age[which(hw3$age > 80)] <- 1
hw3.t$age[which(hw3$age <= 80)] <- 0
hw3.t$age <- factor(hw3.t$age)
# summary(hw3.t$age)


# ---------------------------------------------------------
# 21 age values > 80
# sum(hw3.e$age > 80)

# Transform age to a binary variable: > 80 = 1
hw3.et$age[which(hw3.e$age > 80)] <- 1
hw3.et$age[which(hw3.e$age <= 80)] <- 0
hw3.et$age <- factor(hw3.et$age)
# summary(hw3.et$age)
```

```
############# black transformation #############################
# Transform black to proportional number to make the coefficient interpretable

# training data
hw3.t$black <- round( (sqrt(hw3$black) + 19.92235) / 31.62278, 4)

# eval data
hw3.et$black <- round( (sqrt(hw3.e$black) + 19.92235) / 31.62278, 4)
```

```
# now write prepped data sets to csv files

# set the path relative to your own local environment
# write.csv(hw3.t, file = "C:/SQLData/621-HW3-Clean-Data.csv", row.names = FALSE)

# write.csv(hw3.et, file = "C:/SQLData/621-HW3-Clean-EvalData-.csv", row.names = FALSE)
```

Part 3 - Build Models

```
# Load R functions for model statistics

accuracy <- function(actual, predicted){

  # Equation to be modeled: (TP + TN) / (TP + FP + TN + FN)

  # derive confusion matrix cell values
  c.mat <- data.frame(table(actual, predicted))

  # extract all four confusion matrix values from the data frame
  TN <- as.numeric(as.character(c.mat[1,3]))
  FN <- as.numeric(as.character(c.mat[2,3]))
  FP <- as.numeric(as.character(c.mat[3,3]))
  TP <- as.numeric(as.character(c.mat[4,3]))

  # now calculate the required metric
  return( (TP + TN) / (TP + FP + TN + FN) )
}
```

```
classif.err.rate <- function(actual, predicted) {

  # Equation to be modeled: (FP + FN) / (TP + FP + TN + FN)

  # derive confusion matrix cell values
  c.mat <- data.frame(table(actual, predicted))

  # extract all four confusion matrix values from the data frame
  TN <- as.numeric(as.character(c.mat[1,3]))
  FN <- as.numeric(as.character(c.mat[2,3]))
  FP <- as.numeric(as.character(c.mat[3,3]))
  TP <- as.numeric(as.character(c.mat[4,3]))
```

```
  # now calculate the required metric
  return( (FP + FN) / (TP + FP + TN + FN) )
}
```

```
precision <- function(actual, predicted) {

  # Precision : the proportion of positive cases that were correctly identified.

  # Equation to be modeled: TP / (TP + FP)

  # derive confusion matrix cell values
  c.mat <- data.frame(table(actual, predicted))

  # extract all four confusion matrix values from the data frame
  TN <- as.numeric(as.character(c.mat[1,3]))
  FN <- as.numeric(as.character(c.mat[2,3]))
  FP <- as.numeric(as.character(c.mat[3,3]))
  TP <- as.numeric(as.character(c.mat[4,3]))

  # now calculate the required metric
  return( TP / (TP + FP) )
}
```

```
sensitivity <- function(actual, predicted) {

  # Equation to be modeled: TP / (TP + FN)

  # derive confusion matrix cell values
  c.mat <- data.frame(table(actual, predicted))

  # extract all four confusion matrix values from the data frame
  TN <- as.numeric(as.character(c.mat[1,3]))
  FN <- as.numeric(as.character(c.mat[2,3]))
  FP <- as.numeric(as.character(c.mat[3,3]))
  TP <- as.numeric(as.character(c.mat[4,3]))

  # now calculate the required metric
  return( TP / (TP + FN) )
}
```

```
specificity <- function(actual, predicted) {

  # Equation to be modeled: TN / (TN + FP)

  # derive confusion matrix cell values
  c.mat <- data.frame(table(actual, predicted))

  # extract all four confusion matrix values from the data frame
  TN <- as.numeric(as.character(c.mat[1,3]))
  FN <- as.numeric(as.character(c.mat[2,3]))
  FP <- as.numeric(as.character(c.mat[3,3]))
  TP <- as.numeric(as.character(c.mat[4,3]))
```

```r
  # now calculate the required metric
  return( TN / (TN + FP) )
}
```

```r
F1.Score <- function(actual, predicted) {

  # Equation to be modeled: ( 2 * precision * sensitivity) / (precision + sensitivity)

  # now calculate the required metric
  return( ( 2 * precision(actual, predicted) * sensitivity(actual, predicted))
          / (precision(actual, predicted) + sensitivity(actual, predicted)) )
}
```

Load Training Data

```r
hw3.t <- read.csv("https://raw.githubusercontent.com/spsstudent15/2016-02-621-W2/master/HW-3/621-HW3-Cl

str(hw3.t)
```

```
## 'data.frame':    466 obs. of  14 variables:
##  $ zn     : int  0 0 0 1 0 0 0 0 0 1 ...
##  $ indus  : num  19.58 19.58 18.1 4.93 2.46 ...
##  $ chas   : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.605 0.871 0.74 0.428 0.488 0.52 0.693 0.693 0.515 0.392 ...
##  $ rm     : num  7.93 5.4 6.49 6.39 7.16 ...
##  $ age    : int  1 1 1 0 1 0 1 1 0 0 ...
##  $ dis    : num  2.05 1.32 1.98 7.04 2.7 ...
##  $ rad    : int  5 5 24 6 3 5 24 24 5 1 ...
##  $ tax    : int  403 403 666 300 193 384 666 666 224 315 ...
##  $ ptratio: num  14.7 14.7 20.2 16.6 17.8 20.9 20.2 20.2 20.2 16.4 ...
##  $ black  : num  1.24 1.26 1.25 1.24 1.26 ...
##  $ lstat  : num  3.7 26.82 18.85 5.19 4.82 ...
##  $ medv   : num  50 13.4 15.4 23.7 37.9 26.5 5 7 22.2 20.9 ...
##  $ target : int  1 1 1 0 0 0 1 1 0 0 ...
```

```r
hw3.t$zn <- factor(hw3.t$zn)

hw3.t$age <- factor(hw3.t$age)
```

Model 1: Use the bestglm Function to Build a Model

Use all variables + AIC

```r
# Use __bestglm__ function to find model with lowest AIC using PREPPED data set (prepped as above)
# build a model using all potential predictors

X <- hw3.t[, 1:13]
y <- hw3.t[, 14]

xy <- cbind(as.data.frame(X), y)
```

```r
# method = backward search: yields same result as exhaustive
best.bm <- bestglm(xy, family = binomial(link = "logit"), IC = "AIC", method = "backward")

# show best models - best has lowest AIC (see "Criterion" column)
best.bm$BestModels

# show results for BEST overall model
summary(best.bm$BestModel)
# vif(m1)

# now rebuild by hand so that mmps function can work with it
m1 <- glm(data = hw3.t, target ~ zn + indus + nox + age + dis + rad + tax + ptratio + black + lstat + me
summary(m1)
```

Check for outliers: This MUST be done by hand - the identify function requires that you click on points that are of interest to you so that it can label them. Does not seem possible to use this in a writeup.

```r
#Figure 8.13 on page 291
par(mfrow=c(1,1))
hvalues <- influence(m1)$hat
stanresDeviance <- residuals(m1)/sqrt(1-hvalues)

plot(hvalues,stanresDeviance,ylab="Standardized Deviance Residuals",
     xlab="Leverage Values",ylim=c(-3,3),xlim=c(-0.05,0.7))

# NOTE: the '12' indicated here is found by adding 1 to the number of predictor variables
# used in the final model
abline(v=2* 12 / nrow(hw3.t),lty=2)

hw3.names <- as.character(seq(1:nrow(hw3.t)))

# need to click on potential outliers using the mouse and then click "finish" in the plot window
identify(hvalues, stanresDeviance, labels = hw3.names, cex=0.75)
```

Results say remove rows 14, 18, 159

```r
# remove rows 14, 18, 159 and refit
hw3.re <- hw3.t[-c(14, 18, 159),]

# build a model using all potential predictors

X <- hw3.re[, 1:13]
y <- hw3.re[, 14]

xy <- cbind(as.data.frame(X), y)

# method = backward search: yields same result as exhaustive
best.bm <- bestglm(xy, family = binomial(link = "logit"), IC = "AIC", method = "backward")

# show best models - best has lowest AIC (see "Criterion" column)
best.bm$BestModels
```

```r
# show results for BEST overall model
summary(best.bm$BestModel)
# vif(m1)

# now rebuild by hand so that mmps function can work with it
m.re <- glm(data = hw3.re, target ~ zn + indus + chas + nox + age + dis + rad + tax + ptratio + black +
summary(m.re)

# ------------------------
# marginal model plots
mmps(m.re,layout=c(4,3),key=TRUE)
```

Now check for outliers again

```r
#Figure 8.13 on page 291
par(mfrow=c(1,1))
hvalues <- influence(m.re)$hat
stanresDeviance <- residuals(m.re)/sqrt(1-hvalues)

plot(hvalues,stanresDeviance,ylab="Standardized Deviance Residuals",
     xlab="Leverage Values",ylim=c(-3,3),xlim=c(-0.05,0.7))

# NOTE: the '12' indicated here is found by adding 1 to the number of predictor variables
# used in the final model
abline(v=2* 12 / nrow(hw3.re),lty=2)

hw3.names <- as.character(seq(1:nrow(hw3.re)))

# need to click on potential outliers using the mouse and then click "finish" in the plot window
identify(hvalues, stanresDeviance, labels = hw3.names, cex=0.75)
```

Results say remove 152, 83, 215

```r
# remove rows 14, 18, 159 and refit
hw3.re <- hw3.re[-c(83, 152, 215),]

# build a model using all potential predictors

X <- hw3.re[, 1:13]
y <- hw3.re[, 14]

xy <- cbind(as.data.frame(X), y)

# method = backward search: yields same result as exhaustive
best.bm <- bestglm(xy, family = binomial(link = "logit"), IC = "AIC", method = "backward")

# show best models - best has lowest AIC (see "Criterion" column)
best.bm$BestModels

# show results for BEST overall model
summary(best.bm$BestModel)
# vif(m1)
```

```r
# now rebuild by hand so that mmps function can work with it
m.re <- glm(data = hw3.re, target ~ zn + indus + chas + nox + age + dis + rad + tax + ptratio + black +
summary(m.re)

# -------------------------
# marginal model plots
mmps(m.re,layout=c(4,3),key=TRUE)
```

STOP

Now run metrics

```r
# Coefficient Interpretation

# Logit model average marginal effects - use it to generate interpretable versions of coefficients
LogitScalar <- mean(dlogis(predict(m.re, type = "link")))
LogitScalar * coef(m.re)

# Logit model predicted probabilities - yields likelihood that each eval item is '+'
#
predprob.crime<- round(predict(m.re, type="response"), 2)
summary(predprob.crime)

# Percent correctly predicted values
# NOTE: Need to create variable 'Y' for this to work - set it to response variable
Y <- hw3.re[,14]

pred.crime <- round(fitted(m.re))

table(true = Y, pred = pred.crime)


# t.r <- data.frame(table(true = Y, pred = pred.crime))
# t.r

# now use functions built in HW 2 to get required statistics
accuracy(Y, pred.crime)
classif.err.rate(Y, pred.crime)
precision(Y, pred.crime)
sensitivity(Y, pred.crime)
specificity(Y, pred.crime)
F1.Score(Y, pred.crime)

# get AUC
rocCurve <- roc(response= Y, predictor= pred.crime)
auc(rocCurve)
```

Summary Table:

Metric	Value
Number of Predictors	11
AIC	189.46
Accuracy	0.9239

Metric	Value
Classification Error Rate	0.0761
Precision	0.9357
Sensitivity	0.9067
Specificity	0.9404
F1 Score	0.9211
AUC	0.9235

Bestglm using BIC

```
# build a model using all potential predictors

X <- hw3.t[, 1:13]
y <- hw3.t[, 14]

xy <- cbind(as.data.frame(X), y)

# method = backward search: yields same result as exhaustive
best.bm <- bestglm(xy, family = binomial(link = "logit"), IC = "BIC", method = "backward")

# show best models - best has lowest AIC (see "Criterion" column)
best.bm$BestModels

# show results for BEST overall model
summary(best.bm$BestModel)
# vif(m1)

# now rebuild by hand so that mmps function can work with it
m.bic <- glm(data = hw3.t, target ~ nox + age + rad + tax, family = binomial(link = "logit"))
summary(m.bic)


# ------------------------
# marginal model plots
mmps(m.bic,layout=c(4,3),key=TRUE)

# Logit model average marginal effects - use it to generate interpretable versions of coefficients
LogitScalar <- mean(dlogis(predict(m.bic, type = "link")))
LogitScalar * coef(m.bic)
# Logit model predicted probabilities - yields likelihood that each eval item is '+'
#
predprob.crime<- round(predict(m.bic, type="response"), 2)
summary(predprob.crime)

# Percent correctly predicted values
# NOTE: Need to create variable 'Y' for this to work - set it to response variable
Y <- hw3.t[,14]

pred.crime <- round(fitted(m.bic))

table(true = Y, pred = pred.crime)
```

```r
# t.r <- data.frame(table(true = Y, pred = pred.crime))
# t.r

# now use functions built in HW 2 to get required statistics
accuracy(Y, pred.crime)
classif.err.rate(Y, pred.crime)
precision(Y, pred.crime)
sensitivity(Y, pred.crime)
specificity(Y, pred.crime)
F1.Score(Y, pred.crime)

# get AUC
rocCurve <- roc(response= Y, predictor= pred.crime)
auc(rocCurve)
```

Check for outliers: This MUST be done by hand - the identify function requires that you click on points that are of interest to you so that it can label them. Does not seem possible to use this in a writeup.

```r
#Figure 8.13 on page 291
par(mfrow=c(1,1))
hvalues <- influence(m.bic)$hat
stanresDeviance <- residuals(m.bic)/sqrt(1-hvalues)

plot(hvalues,stanresDeviance,ylab="Standardized Deviance Residuals",
     xlab="Leverage Values",ylim=c(-3,3),xlim=c(-0.05,0.7))

# NOTE: the '12' indicated here is found by adding 1 to the number of predictor variables
# used in the final model
abline(v=2* 5 / nrow(hw3.t),lty=2)

hw3.names <- as.character(seq(1:nrow(hw3.t)))

# need to click on potential outliers using the mouse and then click "finish" in the plot window
identify(hvalues, stanresDeviance, labels = hw3.names, cex=0.75)
```

NO OUTLIERS!!!

Summary Table:

Metric	Value
Number of Predictors	4
AIC	227.34
Accuracy	0.8777
Classification Error Rate	0.1223
Precision	0.8874
Sensitivity	0.8603
Specificity	0.8945
F1 Score	0.8736
AUC	0.8774

Model 2: Logit Model Using Backward Selection

```r
#start with CHAS and TAX eliminated
redo <- glm(data=hw3.t, target~.-chas - tax, family=binomial(link="logit"))
summary(redo)
```

```
##
## Call:
## glm(formula = target ~ . - chas - tax, family = binomial(link = "logit"),
##     data = hw3.t)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3238  -0.2257  -0.0184   0.0020   3.6954
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -29.38278    9.10920  -3.226 0.001257 **
## zn1          -2.40227    0.84062  -2.858 0.004267 **
## indus        -0.12937    0.04573  -2.829 0.004671 **
## nox          49.76413    7.87130   6.322 2.58e-10 ***
## rm           -0.54371    0.68080  -0.799 0.424508
## age1          1.36513    0.50068   2.727 0.006400 **
## dis           0.81940    0.21866   3.747 0.000179 ***
## rad           0.57301    0.14289   4.010 6.07e-05 ***
## ptratio       0.25510    0.12065   2.114 0.034479 *
## black        -7.53122    5.44049  -1.384 0.166269
## lstat         0.08170    0.05076   1.610 0.107482
## medv          0.20079    0.06490   3.094 0.001976 **
## ---
## 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: 195.45  on 454  degrees of freedom
## AIC: 219.45
##
## Number of Fisher Scoring iterations: 9
```

```r
vif(redo)
```

```
##       zn    indus      nox       rm      age      dis      rad  ptratio
## 2.458051 2.617444 4.668796 5.483782 1.959740 4.428175 1.364588 2.075804
##    black    lstat     medv
## 1.050568 2.635865 7.586927
```

```r
#remove rm
redo1 <- glm(data=hw3.t, target~.-chas - tax - rm, family=binomial(link="logit"))
summary(redo1)
```

```
##
## Call:
## glm(formula = target ~ . - chas - tax - rm, family = binomial(link = "logit"),
##     data = hw3.t)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2301  -0.2465  -0.0200   0.0026   3.6926
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -30.90623    8.96234  -3.448 0.000564 ***
## zn1          -2.44648    0.83523  -2.929 0.003399 **
## indus        -0.12989    0.04567  -2.844 0.004456 **
## nox          49.09457    7.76056   6.326 2.51e-10 ***
## age1          1.19155    0.44685   2.667 0.007664 **
## dis           0.79836    0.21499   3.713 0.000204 ***
## rad           0.53939    0.13325   4.048 5.17e-05 ***
## ptratio       0.22147    0.11092   1.997 0.045870 *
## black        -7.39839    5.49195  -1.347 0.177937
## lstat         0.09511    0.04790   1.986 0.047056 *
## medv          0.16225    0.04164   3.896 9.76e-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: 196.09  on 455  degrees of freedom
## AIC: 218.09
##
## Number of Fisher Scoring iterations: 9
```

```r
vif(redo1)
```

```
##       zn    indus      nox      age      dis      rad  ptratio    black
## 2.404000 2.620695 4.525562 1.562843 4.218065 1.199857 1.762695 1.051137
##    lstat     medv
## 2.347105 3.093550
```

```r
#remove black
redo2 <- glm(data=hw3.t, target~.-chas - tax - rm - black, family=binomial(link="logit"))
summary(redo2)
```

```
##
## Call:
## glm(formula = target ~ . - chas - tax - rm - black, family = binomial(link = "logit"),
##     data = hw3.t)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2516  -0.2413  -0.0204   0.0031   3.6903
##
```

```
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -39.63989    6.06071  -6.540 6.13e-11 ***
## zn1          -2.48325    0.82918  -2.995 0.002746 ** 
## indus        -0.12532    0.04508  -2.780 0.005439 ** 
## nox          48.80035    7.71002   6.329 2.46e-10 ***
## age1          1.22211    0.44386   2.753 0.005899 ** 
## dis           0.78972    0.21353   3.698 0.000217 ***
## rad           0.54603    0.13247   4.122 3.76e-05 ***
## ptratio       0.21038    0.10889   1.932 0.053351 .  
## lstat         0.09256    0.04786   1.934 0.053104 .  
## medv          0.15601    0.04112   3.794 0.000148 ***
## ---
## 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: 198.40  on 456  degrees of freedom
## AIC: 218.4
## 
## Number of Fisher Scoring iterations: 9
```

```r
vif(redo2)
```

```
##       zn    indus      nox      age      dis      rad  ptratio    lstat 
## 2.377473 2.575499 4.451908 1.565612 4.121229 1.199413 1.742364 2.345083 
##     medv 
## 3.062759
```

```r
#remove ptratio
redo3 <- glm(data=hw3.t, target~.-chas - tax - rm - black - ptratio, family=binomial(link="logit"))
summary(redo3)
```

```
## 
## Call:
## glm(formula = target ~ . - chas - tax - rm - black - ptratio, 
##     family = binomial(link = "logit"), data = hw3.t)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.1506  -0.2322  -0.0211   0.0050   3.7590  
## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -34.26432    5.16106  -6.639 3.16e-11 ***
## zn1          -3.06008    0.80363  -3.808  0.00014 ***
## indus        -0.11770    0.04453  -2.643  0.00822 ** 
## nox          47.19316    7.55639   6.245 4.23e-10 ***
## age1          1.20526    0.43953   2.742  0.00610 ** 
## dis           0.83789    0.21349   3.925 8.68e-05 ***
## rad           0.50366    0.12831   3.925 8.66e-05 ***
## lstat         0.08995    0.04677   1.923  0.05445 .  
```

```
## medv          0.13226    0.03864   3.423  0.00062 ***
## ---
## 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: 202.17  on 457  degrees of freedom
## AIC: 220.17
##
## Number of Fisher Scoring iterations: 9
```

```
vif(redo3)
```

```
##       zn    indus      nox      age      dis      rad    lstat     medv
## 2.116670 2.639962 4.443026 1.564002 4.117494 1.146209 2.379362 2.776842
```

```
#remove lstat
redo4 <- glm(data=hw3.t, target~.-chas - tax - rm - black - ptratio - lstat, family=binomial(link="logi
summary(redo4)
```

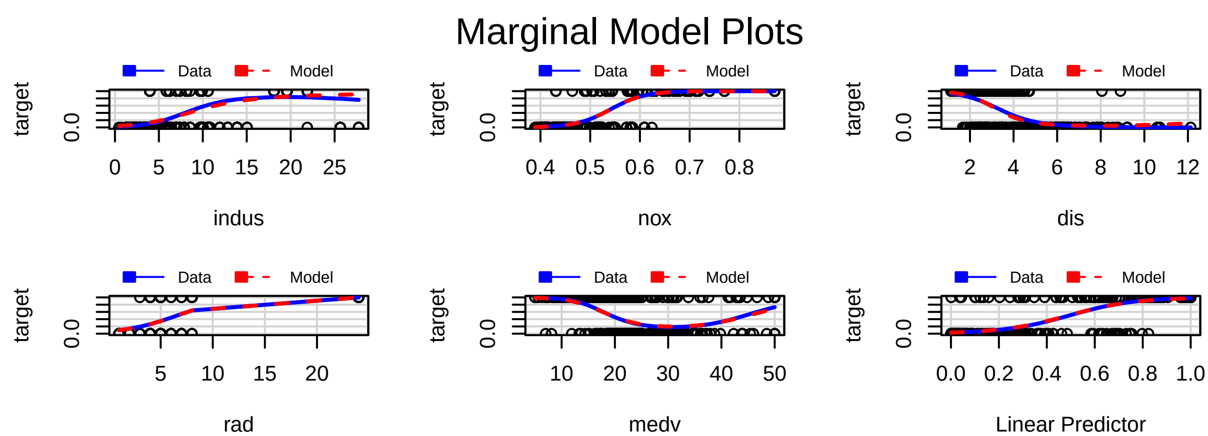
```
##
## Call:
## glm(formula = target ~ . - chas - tax - rm - black - ptratio -
##     lstat, family = binomial(link = "logit"), data = hw3.t)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8643  -0.2428  -0.0310   0.0060   3.6744
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -31.02452    4.65124  -6.670 2.56e-11 ***
## zn1          -2.63156    0.73554  -3.578 0.000347 ***
## indus        -0.10369    0.04353  -2.382 0.017219 *
## nox          45.12985    7.27072   6.207 5.40e-10 ***
## age1          1.39407    0.42773   3.259 0.001117 **
## dis           0.74725    0.20310   3.679 0.000234 ***
## rad           0.50907    0.12521   4.066 4.79e-05 ***
## medv          0.08775    0.02970   2.955 0.003128 **
## ---
## 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: 205.87  on 458  degrees of freedom
## AIC: 221.87
##
## Number of Fisher Scoring iterations: 8
```

```r
vif(redo4)
```

```
##       zn    indus      nox      age      dis      rad     medv
## 1.923310 2.522150 4.217279 1.507917 3.934578 1.163008 1.683635
```

```r
redo.fit <- round(fitted(redo4))

# ------------------------
# marginal model plots
mmps(redo4,layout=c(4,3),key=TRUE)
```

```
## Warning in mmps(redo4, layout = c(4, 3), key = TRUE): Interactions and/or
## factors skipped
```




```r
# ----------------------------
# Coefficient Interpretation

# Logit model average marginal effects - use it to generate interpretable versions of coefficients
LogitScalar.sub <- mean(dlogis(predict(redo4, type = "link")))
LogitScalar.sub * coef(redo4)
```

```
##  (Intercept)          zn1        indus          nox         age1
## -2.195496212 -0.186226051 -0.007337724  3.193680201  0.098653123
##          dis          rad         medv
##  0.052880236  0.036025387  0.006209541
```

```r
Y <- hw3.t[,14]
table(true = Y, pred = redo.fit)
```

```
##     pred
## true   0   1
##    0 218  19
##    1  22 207
```

```r
# t.r <- data.frame(table(true = Y, pred = pred.crime))
# t.r

# now use functions built in HW 2 to get required statistics
accuracy(Y, redo.fit)
```

```
## [1] 0.9120172
```

```r
classif.err.rate(Y, redo.fit)
```

```
## [1] 0.08798283
```

```r
precision(Y, redo.fit)
```

```
## [1] 0.9159292
```

```r
sensitivity(Y, redo.fit)
```

```
## [1] 0.9039301
```

```r
specificity(Y, redo.fit)
```

```
## [1] 0.9198312
```

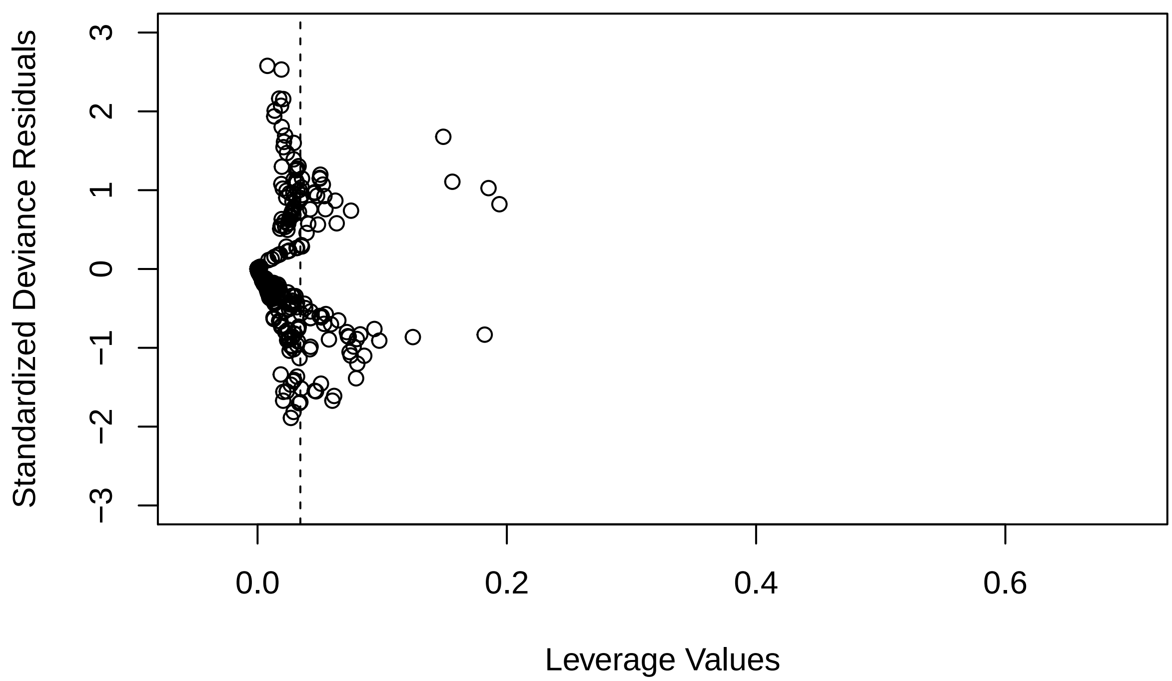
```r
F1.Score(Y, redo.fit)
```

```
## [1] 0.9098901
```

```r
#look at misses
hw3t.4 <- hw3.t
hw3t.4$predict <- fitted(redo4)
miss.4 <- subset(hw3t.4[which(hw3.t$target != redo.fit),])

#AUC
rocCurve <- roc(response= Y, predictor= redo.fit)
auc(rocCurve)
```

```
## Area under the curve: 0.9119
```

```r
#See Figure 8.13 on page 291
par(mfrow=c(1,1))
hvalues <- influence(redo4)$hat
stanresDeviance <- residuals(redo4)/sqrt(1-hvalues)

plot(hvalues,stanresDeviance,ylab="Standardized Deviance Residuals",
     xlab="Leverage Values",ylim=c(-3,3),xlim=c(-0.05,0.7))

# NOTE: the '12' indicated here is found by adding 1 to the number of predictor variables
# used in the final model
abline(v=2* 8 / nrow(hw3.t),lty=2)

hw3.names <- as.character(seq(1:nrow(hw3.t)))

# need to click on potential outliers using the mouse and then click "finish" in the plot window
identify(hvalues, stanresDeviance, labels = hw3.names, cex=0.75)
```




```
## integer(0)
```

```r
#Remove outliers #396, 18, 85, 218, 14
hw3.o <- hw3.t[-c(14,18,85,218, 396),]
redo4.1 <- glm(data=hw3.o, target~.-chas - tax - rm - black - ptratio - lstat, family=binomial(link="log
summary(redo4)
```

```
##
```

```
## Call:
## glm(formula = target ~ . - chas - tax - rm - black - ptratio -
##     lstat, family = binomial(link = "logit"), data = hw3.t)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8643  -0.2428  -0.0310   0.0060   3.6744
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -31.02452    4.65124  -6.670 2.56e-11 ***
## zn1          -2.63156    0.73554  -3.578 0.000347 ***
## indus        -0.10369    0.04353  -2.382 0.017219 *
## nox          45.12985    7.27072   6.207 5.40e-10 ***
## age1          1.39407    0.42773   3.259 0.001117 **
## dis           0.74725    0.20310   3.679 0.000234 ***
## rad           0.50907    0.12521   4.066 4.79e-05 ***
## medv          0.08775    0.02970   2.955 0.003128 **
## ---
## 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: 205.87  on 458  degrees of freedom
## AIC: 221.87
##
## Number of Fisher Scoring iterations: 8
```

```r
prediction <- round(predict(redo4.1, newdata=hw3.t, type="response"))

table(true = Y, pred = prediction)
```

```
##     pred
## true   0   1
##    0 218  19
##    1  21 208
```

```r
accuracy(Y, prediction)
```

```
## [1] 0.9141631
```

```r
classif.err.rate(Y, prediction)
```

```
## [1] 0.08583691
```

```r
precision(Y, prediction)
```

```
## [1] 0.9162996
```

```r
sensitivity(Y, prediction)
```

```
## [1] 0.9082969
```

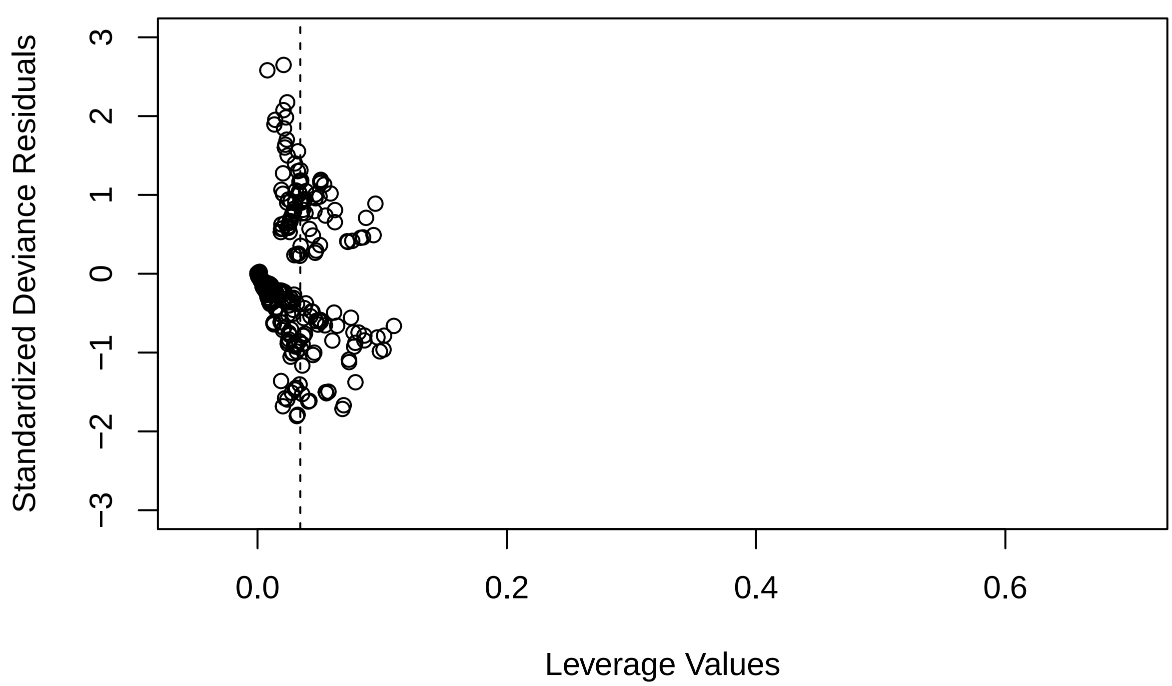
```r
specificity(Y, prediction)
```

```
## [1] 0.9198312
```

```r
F1.Score(Y, prediction)
```

```
## [1] 0.9122807
```

```r
par(mfrow=c(1,1))
hvalues <- influence(redo4.1)$hat
stanresDeviance <- residuals(redo4.1)/sqrt(1-hvalues)

plot(hvalues,stanresDeviance,ylab="Standardized Deviance Residuals",
     xlab="Leverage Values",ylim=c(-3,3),xlim=c(-0.05,0.7))

# NOTE: the '12' indicated here is found by adding 1 to the number of predictor variables
# used in the final model
abline(v=2* 8 / nrow(hw3.t),lty=2)
```



```r
hw3.names <- as.character(seq(1:nrow(hw3.t)))

#no outliers

#AUC
rocCurve <- roc(response= Y, predictor= prediction)
auc(rocCurve)
```

```
## Area under the curve: 0.9141
```

Model 3: Probit Model Using Backward Selection

```r
#probit - again starting with no TAX and CHAS
pmod <- glm(data=hw3.t, target~. - tax- chas, family=binomial(link="probit"))
```

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

```r
summary(pmod)
```

```
##
## Call:
## glm(formula = target ~ . - tax - chas, family = binomial(link = "probit"),
##     data = hw3.t)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2256  -0.2445  -0.0032   0.0000   3.9212
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -14.77129    5.00943  -2.949  0.00319 **
## zn1          -0.95557    0.42190  -2.265  0.02352 *
## indus        -0.06550    0.02488  -2.632  0.00849 **
## nox          25.58271    4.02313   6.359 2.03e-10 ***
## rm           -0.31252    0.37050  -0.844  0.39894
## age1          0.71386    0.27627   2.584  0.00977 **
## dis           0.37495    0.11551   3.246  0.00117 **
## rad           0.30129    0.07474   4.031 5.55e-05 ***
## ptratio       0.14916    0.06591   2.263  0.02363 *
## black        -4.17851    3.10279  -1.347  0.17808
## lstat         0.04098    0.02819   1.454  0.14604
## medv          0.10395    0.03389   3.067  0.00216 **
## ---
## 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: 199.77  on 454  degrees of freedom
```

```
## AIC: 223.77
##
## Number of Fisher Scoring iterations: 12
```

```r
vif(pmod)
```

```
##       zn    indus      nox       rm      age      dis      rad  ptratio
## 2.365415 2.540325 4.454924 5.483796 1.902328 4.400903 1.274948 1.954070
##    black    lstat     medv
## 1.049779 2.618295 7.023299
```

```r
#get rid of rm
pmod1 <- glm(data=hw3.t, target~. - tax- chas - rm, family=binomial(link="probit"))
```

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

```r
summary(pmod1)
```

```
##
## Call:
## glm(formula = target ~ . - tax - chas - rm, family = binomial(link = "probit"),
##     data = hw3.t)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1317  -0.2585  -0.0037   0.0000   3.9216
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -15.77148    4.87705  -3.234 0.001221 **
## zn1          -0.98509    0.41504  -2.373 0.017621 *
## indus        -0.06528    0.02481  -2.632 0.008494 **
## nox          25.22038    3.95156   6.382 1.74e-10 ***
## age1          0.60979    0.24929   2.446 0.014442 *
## dis           0.36576    0.11337   3.226 0.001255 **
## rad           0.28576    0.07089   4.031 5.56e-05 ***
## ptratio       0.13144    0.06146   2.139 0.032463 *
## black        -4.08946    3.11494  -1.313 0.189232
## lstat         0.05020    0.02622   1.914 0.055578 .
## medv          0.08268    0.02182   3.789 0.000151 ***
## ---
## 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: 200.46  on 455  degrees of freedom
## AIC: 222.46
##
## Number of Fisher Scoring iterations: 12
```

```r
vif(pmod1)
```

```
##       zn    indus      nox      age      dis      rad  ptratio    black
## 2.286815 2.542446 4.328321 1.557061 4.204075 1.165641 1.708739 1.050144
##    lstat     medv
## 2.283869 2.931991
```

```r
#get rid of black
pmod2 <- glm(data=hw3.t, target~. - tax- chas - rm - black, family=binomial(link="probit"))
```

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

```r
summary(pmod2)
```

```
##
## Call:
## glm(formula = target ~ . - tax - chas - rm - black, family = binomial(link = "probit"),
##     data = hw3.t)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1501  -0.2526  -0.0038   0.0000   3.9137
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -20.59379    3.10278  -6.637 3.20e-11 ***
## zn1          -0.99820    0.41261  -2.419 0.015554 *
## indus        -0.06285    0.02456  -2.559 0.010487 *
## nox          24.98786    3.92633   6.364 1.96e-10 ***
## age1          0.63248    0.24792   2.551 0.010738 *
## dis           0.35924    0.11291   3.182 0.001465 **
## rad           0.29002    0.07060   4.108 3.99e-05 ***
## ptratio       0.12680    0.06079   2.086 0.036980 *
## lstat         0.04866    0.02620   1.857 0.063260 .
## medv          0.07944    0.02159   3.680 0.000234 ***
## ---
## 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: 202.72  on 456  degrees of freedom
## AIC: 222.72
##
## Number of Fisher Scoring iterations: 10
```

```r
vif(pmod2)
```

```
##       zn    indus      nox      age      dis      rad  ptratio    lstat
## 2.258185 2.500816 4.257708 1.553641 4.121265 1.165747 1.691050 2.279136
##     medv
## 2.892678
```

```r
#get rid of lstat
pmod3 <- glm(data=hw3.t, target~. - tax- chas - rm - black - lstat, family=binomial(link="probit"))
```

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

```r
summary(pmod3)
```

```
##
## Call:
## glm(formula = target ~ . - tax - chas - rm - black - lstat, family = binomial(link = "probit"),
##     data = hw3.t)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8598  -0.2660  -0.0062   0.0000   3.8284
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -18.70441    2.86028  -6.539 6.18e-11 ***
## zn1          -0.86185    0.38952  -2.213  0.02692 *
## indus        -0.05485    0.02390  -2.295  0.02175 *
## nox          23.81223    3.77190   6.313 2.74e-10 ***
## age1          0.73167    0.24044   3.043  0.00234 **
## dis           0.31276    0.10758   2.907  0.00365 **
## rad           0.29011    0.06889   4.211 2.54e-05 ***
## ptratio       0.12089    0.06043   2.001  0.04544 *
## medv          0.05602    0.01706   3.283  0.00103 **
## ---
## 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: 206.46  on 457  degrees of freedom
## AIC: 224.46
##
## Number of Fisher Scoring iterations: 10
```

```r
vif(pmod3)
```

```
##       zn    indus      nox      age      dis      rad  ptratio     medv
## 2.152354 2.403601 4.112197 1.488618 3.925404 1.167530 1.703963 1.859259
```

```r
pmod.fit <- round(fitted(pmod3))

# ------------------------
# marginal model plots


# ----------------------------
# Coefficient Interpretation
```

```r
# Logit model average marginal effects - use it to generate interpretable versions of coefficients
LogitScalar.sub <- mean(dlogis(predict(pmod3, type = "link")))
LogitScalar.sub * coef(pmod3)
```

```
##  (Intercept)          zn1        indus          nox         age1
## -1.993004590 -0.091832408 -0.005844793  2.537255851  0.077961626
##          dis          rad      ptratio         medv
##  0.033325850  0.030912388  0.012881636  0.005968772
```

```r
table(true = Y, pred = pmod.fit)
```

```
##     pred
## true   0   1
##    0 218  19
##    1  21 208
```

```r
# t.r <- data.frame(table(true = Y, pred = pred.crime))
# t.r

# now use functions built in HW 2 to get required statistics
accuracy(Y, pmod.fit)
```

```
## [1] 0.9141631
```

```r
classif.err.rate(Y, pmod.fit)
```

```
## [1] 0.08583691
```

```r
precision(Y, pmod.fit)
```

```
## [1] 0.9162996
```

```r
sensitivity(Y, pmod.fit)
```

```
## [1] 0.9082969
```

```r
specificity(Y, pmod.fit)
```

```
## [1] 0.9198312
```

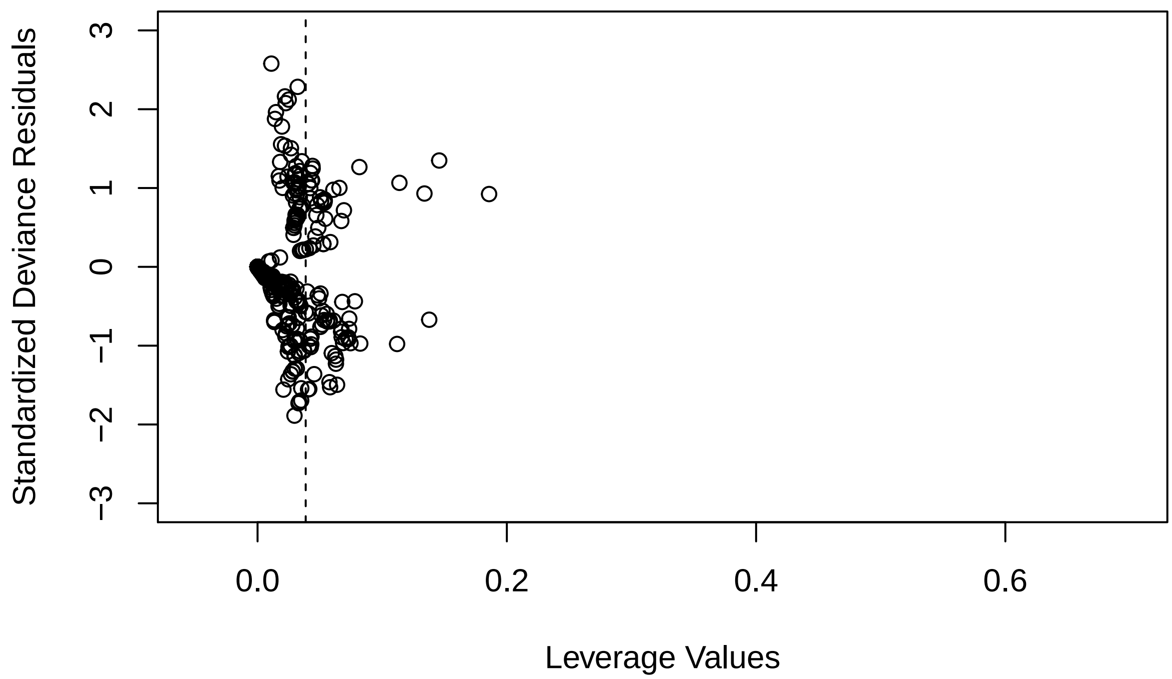
```r
F1.Score(Y, pmod.fit)
```

```
## [1] 0.9122807
```

```r
#auc
rocCurve <- roc(response= Y, predictor= pmod.fit)
auc(rocCurve)
```

```
## Area under the curve: 0.9141
```

```r
#look for outliers

#See Figure 8.13 on page 291
par(mfrow=c(1,1))
hvalues <- influence(pmod3)$hat
stanresDeviance <- residuals(pmod3)/sqrt(1-hvalues)

plot(hvalues,stanresDeviance,ylab="Standardized Deviance Residuals",
     xlab="Leverage Values",ylim=c(-3,3),xlim=c(-0.05,0.7))

# NOTE: the '12' indicated here is found by adding 1 to the number of predictor variables
# used in the final model
abline(v=2* 9 / nrow(hw3.t),lty=2)

hw3.names <- as.character(seq(1:nrow(hw3.t)))

# need to click on potential outliers using the mouse and then click "finish" in the plot window
identify(hvalues, stanresDeviance, labels = hw3.names, cex=0.75)
```




```
## integer(0)
```

```r
#Remove outliers #396, 18, 85, 14
hw3.o.p <- hw3.t[-c(14,18,85, 396),]
pmod3.1 <- glm(data=hw3.o.p, target~. - tax- chas - rm - black - lstat, family=binomial(link="probit"))
```

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

```r
summary(pmod3.1)
```

```
##
## Call:
## glm(formula = target ~ . - tax - chas - rm - black - lstat, family = binomial(link = "probit"),
##     data = hw3.o.p)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8022  -0.2647  -0.0065   0.0000   3.9000
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -17.93752    2.87908  -6.230 4.66e-10 ***
## zn1          -1.00492    0.41051  -2.448  0.01437 *
## indus        -0.05227    0.02406  -2.173  0.02981 *
## nox          22.83626    3.77716   6.046 1.49e-09 ***
## age1          0.75831    0.24499   3.095  0.00197 **
## dis           0.31715    0.11789   2.690  0.00714 **
## rad           0.28931    0.06986   4.141 3.45e-05 ***
## ptratio       0.11240    0.06143   1.830  0.06726 .
## medv          0.05001    0.01789   2.795  0.00519 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 640.25  on 461  degrees of freedom
## Residual deviance: 202.64  on 453  degrees of freedom
## AIC: 220.64
##
## Number of Fisher Scoring iterations: 10
```

```r
#remove ptratio
pmod3.2 <- glm(data=hw3.o.p, target~. - tax- chas - rm - black - lstat -ptratio, family=binomial(link="
```

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

```r
summary(pmod3.2)
```

```
##
## Call:
## glm(formula = target ~ . - tax - chas - rm - black - lstat -
##     ptratio, family = binomial(link = "probit"), data = hw3.o.p)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7329  -0.2490  -0.0123   0.0000   3.9778
##
## Coefficients:
```

```
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -15.17084    2.35792  -6.434 1.24e-10 ***
## zn1          -1.23712    0.39527  -3.130  0.00175 ** 
## indus        -0.04966    0.02380  -2.087  0.03693 *  
## nox          22.24095    3.71283   5.990 2.09e-09 ***
## age1          0.72982    0.24167   3.020  0.00253 ** 
## dis           0.33442    0.11863   2.819  0.00482 ** 
## rad           0.27352    0.06893   3.968 7.24e-05 ***
## medv          0.03579    0.01599   2.238  0.02522 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 640.25  on 461  degrees of freedom
## Residual deviance: 205.77  on 454  degrees of freedom
## AIC: 221.77
## 
## Number of Fisher Scoring iterations: 10
```

```r
prediction.p <- round(predict(pmod3.2, newdata=hw3.t, type="response"))

table(true = Y, pred = prediction.p)
```

```
##     pred
## true   0   1
##    0 218  19
##    1  22 207
```

```r
accuracy(Y, prediction.p)
```

```
## [1] 0.9120172
```

```r
classif.err.rate(Y, prediction.p)
```

```
## [1] 0.08798283
```

```r
precision(Y, prediction.p)
```

```
## [1] 0.9159292
```

```r
sensitivity(Y, prediction.p)
```

```
## [1] 0.9039301
```

```r
specificity(Y, prediction.p)
```

```
## [1] 0.9198312
```

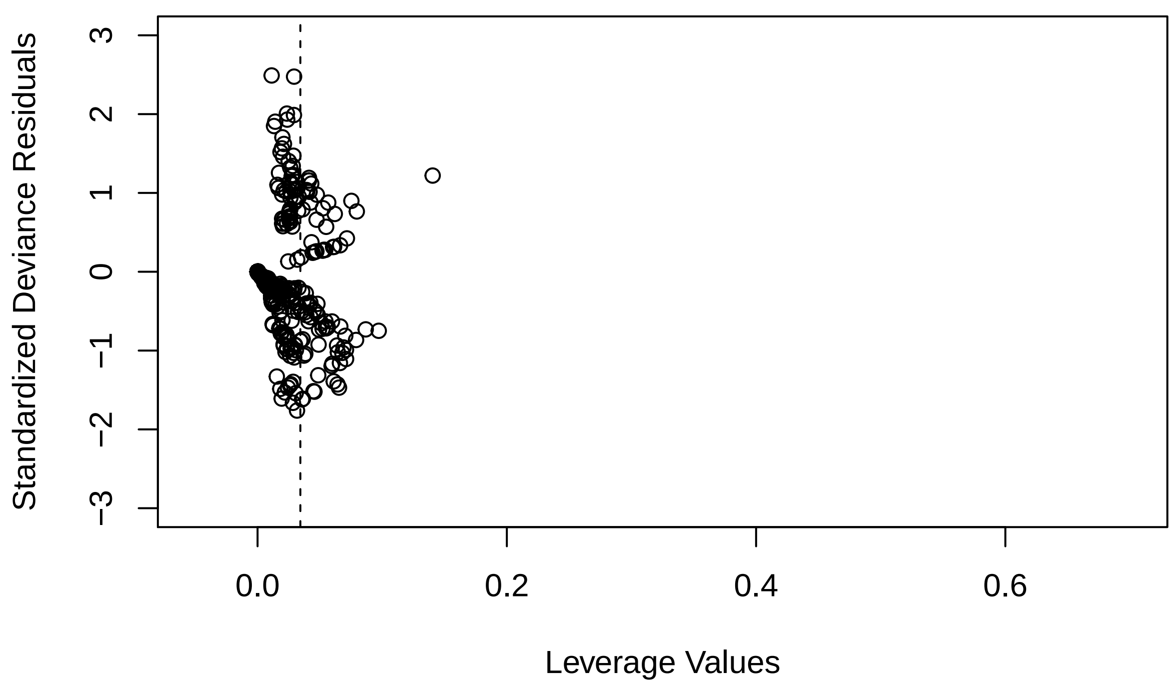
```r
F1.Score(Y, prediction.p)
```

```
## [1] 0.9098901
```

```r
#auc
rocCurve <- roc(response= Y, predictor= prediction.p)
auc(rocCurve)
```

```
## Area under the curve: 0.9119
```

```r
#check for outliers
par(mfrow=c(1,1))
hvalues <- influence(pmod3.2)$hat
stanresDeviance <- residuals(pmod3.2)/sqrt(1-hvalues)

plot(hvalues,stanresDeviance,ylab="Standardized Deviance Residuals",
     xlab="Leverage Values",ylim=c(-3,3),xlim=c(-0.05,0.7))

# NOTE: the '8' indicated here is found by adding 1 to the number of predictor variables
# used in the final model
abline(v=2* 8 / nrow(hw3.t),lty=2)
```



```r
#SCott's model updated


# BUILD MODEL
```

```r
# Use forward selection strategy to find model with lowest AIC using PREPPED data set (prepped as above
# iterate through predictors in descending order of correlation with target
# avoid highly collinear predictors with each iteration

m1 <- glm(data = hw3.t, target ~ nox, family = binomial(link = "logit"))
summary(m1)

m2 <- glm(data = hw3.t, target ~ nox + rad, family = binomial(link = "logit"))
summary(m2)

m3 <- glm(data = hw3.t, target ~ nox + rad + age, family = binomial(link = "logit"))
summary(m3)

m4 <- glm(data = hw3.t, target ~ nox + rad + age + tax, family = binomial(link = "logit"))
summary(m4)

m5 <- glm(data = hw3.t, target ~ nox + rad + age + tax + ptratio, family = binomial(link = "logit"))
summary(m5)

m <- glm(data = hw3.t, target ~ nox + rad + age + tax + ptratio + medv, family = binomial(link = "logit
summary(m)


m.fit <- round(fitted(m))

# ------------------------
# marginal model plots


# ----------------------------
# Coefficient Interpretation

# Logit model average marginal effects - use it to generate interpretable versions of coefficients
LogitScalar.sub <- mean(dlogis(predict(m.fit,type = "link")))
LogitScalar.sub * coef(m.fit)

table(true = Y, pred = m.fit)


# t.r <- data.frame(table(true = Y, pred = pred.crime))
# t.r

# now use functions built in HW 2 to get required statistics
accuracy(Y, m.fit)
classif.err.rate(Y, m.fit)
precision(Y, m.fit)
sensitivity(Y, m.fit)
specificity(Y, m.fit)
```

```r
F1.Score(Y, m.fit)

#auc
rocCurve <- roc(response= Y, predictor= m.fit)
auc(rocCurve)
```

Check for outliers: This MUST be done by hand - the identify function requires that you click on points that are of interest to you so that it can label them. Does not seem possible to use this in a writeup.

```r
#Figure 8.13 on page 291
par(mfrow=c(1,1))
hvalues <- influence(m)$hat
stanresDeviance <- residuals(m)/sqrt(1-hvalues)

plot(hvalues,stanresDeviance,ylab="Standardized Deviance Residuals",
     xlab="Leverage Values",ylim=c(-3,3),xlim=c(-0.05,0.7))

# NOTE: the '7' indicated here is found by adding 1 to the number of predictor variables
# used in the final model
abline(v=2 * 7 / nrow(hw3.t),lty=2)

hw3.names <- as.character(seq(1:nrow(hw3.t)))

# need to click on potential outliers using the mouse and then click "finish" in the plot window
identify(hvalues, stanresDeviance, labels = hw3.names, cex=0.75)

![image](outliers.png)
```

There are no outliers.

Model 4: Forward Selection + AIC

```r
library(bestglm)
library(alr3)
library(car)
library(pROC)
```

Load Training Data

```
## 'data.frame':    466 obs. of  14 variables:
##  $ zn     : int  0 0 0 1 0 0 0 0 0 1 ...
##  $ indus  : num  19.58 19.58 18.1 4.93 2.46 ...
##  $ chas   : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.605 0.871 0.74 0.428 0.488 0.52 0.693 0.693 0.515 0.392 ...
##  $ rm     : num  7.93 5.4 6.49 6.39 7.16 ...
##  $ age    : int  1 1 1 0 1 0 1 1 0 0 ...
##  $ dis    : num  2.05 1.32 1.98 7.04 2.7 ...
##  $ rad    : int  5 5 24 6 3 5 24 24 5 1 ...
##  $ tax    : int  403 403 666 300 193 384 666 666 224 315 ...
```

```
##  $ ptratio: num  14.7 14.7 20.2 16.6 17.8 20.9 20.2 20.2 20.2 16.4 ...
##  $ black  : num  1.24 1.26 1.25 1.24 1.26 ...
##  $ lstat  : num  3.7 26.82 18.85 5.19 4.82 ...
##  $ medv   : num  50 13.4 15.4 23.7 37.9 26.5 5 7 22.2 20.9 ...
##  $ target : int  1 1 1 0 0 0 1 1 0 0 ...
```

BUILD MODEL

```r
# Use forward selection strategy to find model with lowest AIC using PREPPED data set (prepped as above
# iterate through predictors in descending order of correlation with target
# avoid highly collinear predictors with each iteration

m1 <- glm(data = hw3.t, target ~ nox, family = binomial(link = "logit"))
summary(m1)
```

```
##
## Call:
## glm(formula = target ~ nox, family = binomial(link = "logit"),
##     data = hw3.t)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2456  -0.3759  -0.1675   0.3707   2.5576
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -15.892      1.449  -10.97   <2e-16 ***
## nox           29.375      2.707   10.85   <2e-16 ***
## ---
## 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: 292.01  on 464  degrees of freedom
## AIC: 296.01
##
## Number of Fisher Scoring iterations: 6
```

```r
m2 <- glm(data = hw3.t, target ~ nox + rad, family = binomial(link = "logit"))
summary(m2)
```

```
##
## Call:
## glm(formula = target ~ nox + rad, family = binomial(link = "logit"),
##     data = hw3.t)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8769  -0.3447  -0.0692   0.0068   2.5803
##
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -17.4532     1.9488  -8.956  < 2e-16 ***
## nox          27.1964     3.2317   8.415  < 2e-16 ***
## rad           0.5139     0.1082   4.750 2.04e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 239.51  on 463  degrees of freedom
## AIC: 245.51
## 
## Number of Fisher Scoring iterations: 8
```

```r
m3 <- glm(data = hw3.t, target ~ nox + rad + age, family = binomial(link = "logit"))
summary(m3)
```

```
## 
## Call:
## glm(formula = target ~ nox + rad + age, family = binomial(link = "logit"), 
##     data = hw3.t)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -1.89929  -0.32307  -0.06752   0.00654   2.65597  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -16.1599     2.0306  -7.958 1.75e-15 ***
## nox          23.7736     3.5958   6.612 3.80e-11 ***
## rad           0.5439     0.1124   4.840 1.30e-06 ***
## age1          0.7733     0.3826   2.021   0.0433 *  
## ---
## 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: 235.40  on 462  degrees of freedom
## AIC: 243.4
## 
## Number of Fisher Scoring iterations: 8
```

```r
m4 <- glm(data = hw3.t, target ~ nox + rad + age + tax, family = binomial(link = "logit"))
summary(m4)
```

```
## 
## Call:
## glm(formula = target ~ nox + rad + age + tax, family = binomial(link = "logit"), 
##     data = hw3.t)
## 
## Deviance Residuals: 
```

```
##      Min        1Q    Median        3Q       Max
## -1.79812  -0.27281  -0.03378   0.00576   2.67239
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -18.574993   2.449356  -7.584 3.36e-14 ***
## nox          32.290476   4.754724   6.791 1.11e-11 ***
## rad           0.705286   0.129123   5.462 4.71e-08 ***
## age1          1.048365   0.396498   2.644 0.008192 **
## tax          -0.009264   0.002489  -3.723 0.000197 ***
## ---
## 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: 217.34  on 461  degrees of freedom
## AIC: 227.34
##
## Number of Fisher Scoring iterations: 8
```

```
m5 <- glm(data = hw3.t, target ~ nox + rad + age + tax + ptratio, family = binomial(link = "logit"))
summary(m5)
```

```
##
## Call:
## glm(formula = target ~ nox + rad + age + tax + ptratio, family = binomial(link = "logit"),
##     data = hw3.t)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.02044  -0.22600  -0.01481   0.00189   2.76906
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -23.948087   3.624270  -6.608 3.90e-11 ***
## nox          34.716322   5.130479   6.767 1.32e-11 ***
## rad           0.823704   0.144963   5.682 1.33e-08 ***
## age1          1.066780   0.400334   2.665  0.00771 **
## tax          -0.010659   0.002627  -4.058 4.95e-05 ***
## ptratio       0.211480   0.087725   2.411  0.01592 *
## ---
## 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: 211.42  on 460  degrees of freedom
## AIC: 223.42
##
## Number of Fisher Scoring iterations: 9
```

```r
m <- glm(data = hw3.t, target ~ nox + rad + age + tax + ptratio + medv, family = binomial(link = "logit
summary(m)
```

```
##
## Call:
## glm(formula = target ~ nox + rad + age + tax + ptratio + medv,
##     family = binomial(link = "logit"), data = hw3.t)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.99303  -0.21717  -0.01391   0.00267   2.85026
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -27.915631   4.216520  -6.621 3.58e-11 ***
## nox          34.939991   5.136647   6.802 1.03e-11 ***
## rad           0.778812   0.144997   5.371 7.82e-08 ***
## age1          1.190059   0.409924   2.903  0.00369 **
## tax          -0.009566   0.002633  -3.633  0.00028 ***
## ptratio       0.330006   0.106129   3.109  0.00187 **
## medv          0.062443   0.029352   2.127  0.03339 *
## ---
## 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: 206.52  on 459  degrees of freedom
## AIC: 220.52
##
## Number of Fisher Scoring iterations: 9
```

```r
# Find outliers using ~ twice the average leverage
# Avg leverage is first dotted line ~.015
# Cutoff leverage is second dotted line ~.030

# Note, the strategy in this model is forward selection and minimizing AIC
# while maintaining all predictor p-values within .05 significance levels.

# AIC minimization drove selection of outliers first, removing as many as plausible
# while staying within customary cutoff threshold
```

```r
#Figure 8.13 on page 291
par(mfrow=c(1,1))
hvalues <- influence(m)$hat
stanresDeviance <- residuals(m)/sqrt(1-hvalues)

plot(hvalues,stanresDeviance,ylab="Standardized Deviance Residuals",
     xlab="Leverage Values",ylim=c(-3,3),xlim=c(-0.05,0.7))

# NOTE: the '7' indicated here is found by adding 1 to the number of predictor variables
# used in the final model
abline(v=2 * 7 / nrow(hw3.t),lty=2)
```

```r
#.015

# Find outliers using ~ twice the average leverage
abline(v=2 * 14 / nrow(hw3.t),lty=2)
# .030

hw3.names <- as.character(seq(1:nrow(hw3.t)))



# need to click on potential outliers using the mouse and then click "finish" in the plot window
identify(hvalues, stanresDeviance, labels = hw3.names, cex=0.75)

# ![image](outliers.png)
```

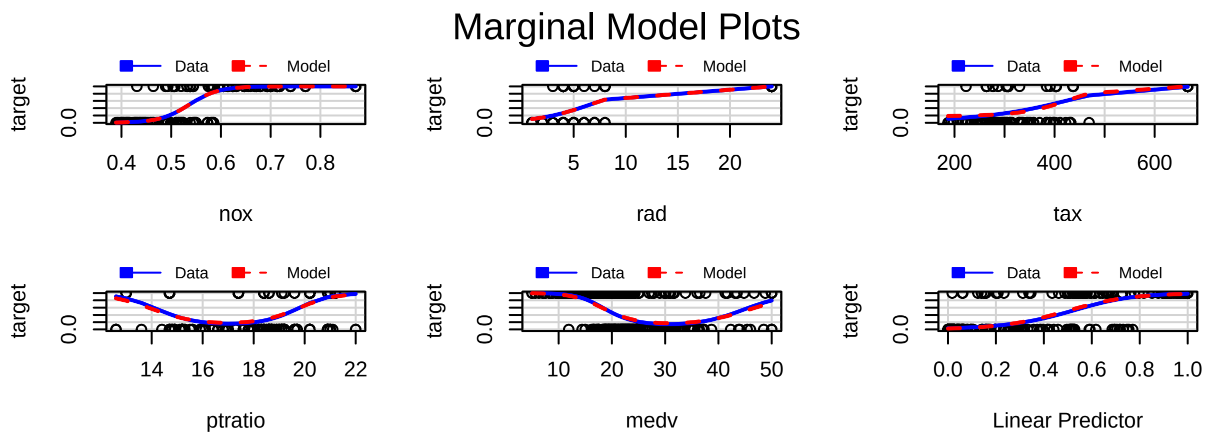
Results say remove rows 5,14,18,37,61,67,73,138,154,342,106,130,142,166,205,227,236,240,246,262,263,293,295,323,334,388,398

```r
# remove rows 5,14,18,37,61,67,73,138,154,342,106,130,142,166,205,227,236,240,246,262,263,293,295,323,3
hw3.re <- hw3.t[-c(5,14,18,37,61,67,73,138,154,342,106,130,142,166,205,227,236,240,246,262,263,293,295,3

# now rebuild
m.re <- glm(data = hw3.re, target ~ nox + rad + age + tax + ptratio + medv, family = binomial(link = "l
summary(m.re)
```

```
##
## Call:
## glm(formula = target ~ nox + rad + age + tax + ptratio + medv,
##     family = binomial(link = "logit"), data = hw3.re)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.71583  -0.20103  -0.01099   0.00421   2.84146
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -28.377499   4.406441  -6.440 1.19e-10 ***
## nox          33.737205   5.618203   6.005 1.91e-09 ***
## rad           0.700772   0.158747   4.414 1.01e-05 ***
## age1          1.291690   0.445345   2.900  0.00373 **
## tax          -0.007822   0.003645  -2.146  0.03187 *
## ptratio       0.370582   0.116400   3.184  0.00145 **
## medv          0.068884   0.033767   2.040  0.04135 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 608.58  on 438  degrees of freedom
## Residual deviance: 185.77  on 432  degrees of freedom
## AIC: 199.77
##
## Number of Fisher Scoring iterations: 9
```

```
# -------------------------
# marginal model plots
mmps(m.re,layout=c(4,3),key=TRUE)
```

```
## Warning in mmps(m.re, layout = c(4, 3), key = TRUE): Interactions and/or
## factors skipped
```




```
#Figure 8.13 on page 291
par(mfrow=c(1,1))
hvalues <- influence(m.re)$hat
stanresDeviance <- residuals(m.re)/sqrt(1-hvalues)

plot(hvalues,stanresDeviance,ylab="Standardized Deviance Residuals",
     xlab="Leverage Values",ylim=c(-3,3),xlim=c(-0.05,0.7))

# NOTE: the '7' indicated here is found by adding 1 to the number of predictor variables
# used in the final model
abline(v=2 * 7 / nrow(hw3.re),lty=2)
#.015

# Find outliers using ~ twice the average leverage
abline(v=2 * 14 / nrow(hw3.re),lty=2)
# .030

hw3.names <- as.character(seq(1:nrow(hw3.re)))
```

```r
# need to click on potential outliers using the mouse and then click "finish" in the plot window
identify(hvalues, stanresDeviance, labels = hw3.names, cex=0.75)
```

STOP

Now run metrics

```r
# Coefficient Interpretation

# Logit model average marginal effects - use it to generate interpretable versions of coefficients
LogitScalar <- mean(dlogis(predict(m.re, type = "link")))
LogitScalar * coef(m.re)
```

```
##   (Intercept)           nox           rad          age1           tax
## -1.9302497579  2.2948192811  0.0476668470  0.0878612836 -0.0005320579
##       ptratio          medv
##  0.0252071403  0.0046855361
```

```r
# Logit model predicted probabilities - yields likelihood that each eval item is '+'
#
predprob.crime<- round(predict(m.re, type="response"), 2)
summary(predprob.crime)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.020   0.510   0.499   1.000   1.000
```

```r
# Percent correctly predicted values
# NOTE: Need to create variable 'Y' for this to work - set it to response variable
Y <- hw3.re[,14]

pred.crime <- round(fitted(m.re))

table(true = Y, pred = pred.crime)
```

```
##     pred
## true   0   1
##    0 194  26
##    1  19 200
```

```r
# t.r <- data.frame(table(true = Y, pred = pred.crime))
# t.r

# now use functions built in HW 2 to get required statistics
accuracy(Y, pred.crime)
```

```
## [1] 0.8974943
```

```r
classif.err.rate(Y, pred.crime)
```

```
## [1] 0.1025057
```

```r
precision(Y, pred.crime)
```

```
## [1] 0.8849558
```

```r
sensitivity(Y, pred.crime)
```

```
## [1] 0.913242
```

```r
specificity(Y, pred.crime)
```

```
## [1] 0.8818182
```

```r
F1.Score(Y, pred.crime)
```

```
## [1] 0.8988764
```

```r
# get AUC
rocCurve <- roc(response= Y, predictor= pred.crime)
auc(rocCurve)
```

```
## Area under the curve: 0.8975
```

Summary Table:

Metric	Value
Number of Predictors	7
AIC	199.77
Accuracy	0.8975
Classification Error Rate	0.1025
Precision	0.8850
Sensitivity	0.9132
Specificity	0.8818
F1 Score	0.9104
AUC	0.8989

Part 4. Select Models

```r
# need psych library for describe function
library(psych)
```

```r
# load training data + build model
hw3.t <- read.csv("https://raw.githubusercontent.com/spsstudent15/2016-02-621-W2/master/HW-3/621-HW3-Cl

hw3.t$zn <- factor(hw3.t$zn)
hw3.t$age <- factor(hw3.t$age)
```

```
m4 <- glm(data = hw3.t, target ~ nox + rad + age + tax + ptratio + medv, family = binomial(link = "logi
summary(m4)
```

```
# Now load evaluation data set and predict TARGET crime rate

# load EVAL data set
eval.d <- read.csv("https://raw.githubusercontent.com/spsstudent15/2016-02-621-W2/master/HW-3/621-HW3-C

eval.d$zn <- factor(eval.d$zn)
eval.d$age <- factor(eval.d$age)

# save original data
eval.2 <- eval.d

# now predict TARGET_WINS using m.4
pred.CR <- round(predict(m4, newdata=eval.2, type="response"))

# add predicted variables to TARGET_WINS variable
eval.2$target <- pred.CR
eval.d$target <- pred.CR
```

```
# write full model EVAL data to a CSV file
write.csv(eval.d, file = "C:/SQLData/HW3-PRED-EVAL-ALL_M_DATA.csv", row.names = FALSE)


describe(eval.d$target)
describe(hw3.t$target)
```