Homework 3

Sam Reeves

1. Data Exploration

We have been given a set of 466 observations concerning areas in a city, and each corresponds to a True or False value for the area experiencing elevated crime levels. We have made a predictive model to categorize new areas as high crime or not high crime.

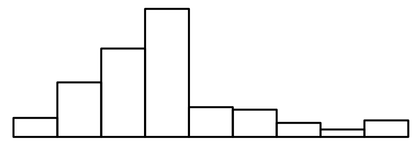
The variable "black" is missing from the data. I assume this is the modification indicated in the initial filenames. This is not a problem, I would also throw this out. Leaving out this variable could protect against dangerous side effects of an overzealous model fit. However, without this information included, we can not actively safeguard against racist models.

```
## large_zone  ind_acres    charles        nox      rooms        age   dist_emp
##  "numeric"  "numeric"  "integer"  "numeric"  "numeric"  "numeric"  "numeric"
##    hw_dist   full_tax    ptratio low_status median_val     target
##  "integer"  "integer"  "numeric"  "numeric"  "numeric"  "integer"
```

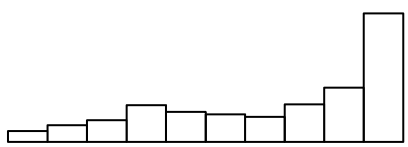
```
##    large_zone       ind_acres        charles             nox
##  Min.   :  0.00   Min.   : 0.460   Min.   :0.00000   Min.   :0.3890
##  1st Qu.:  0.00   1st Qu.: 5.145   1st Qu.:0.00000   1st Qu.:0.4480
##  Median :  0.00   Median : 9.690   Median :0.00000   Median :0.5380
##  Mean   : 11.58   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
##      rooms            age            dist_emp         hw_dist
##  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
##     full_tax        ptratio       low_status       median_val
##  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
```

Combing through exploratory data analysis, it seems there are no missing values. The information concerning large zones (large_zone) and median home value (median_val) seem quite skew in the same direction. The other values appear to have fairly reasonable distributions.

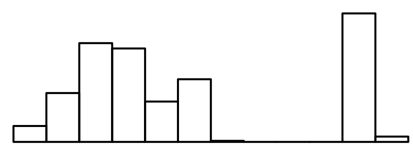
Large Zoning



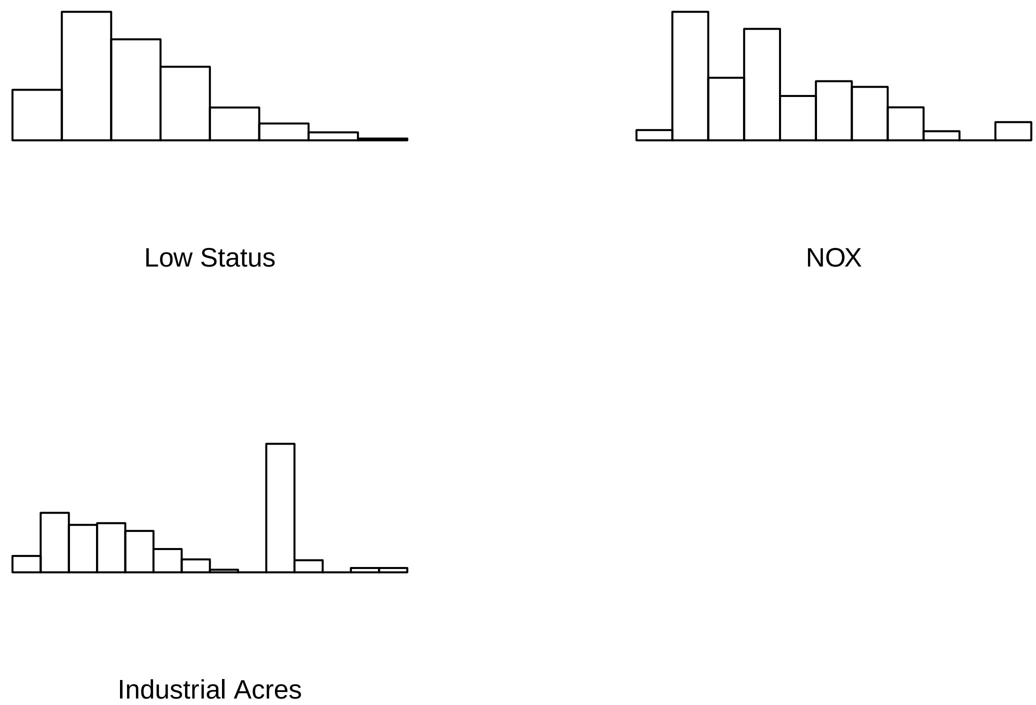
Median Value



Age



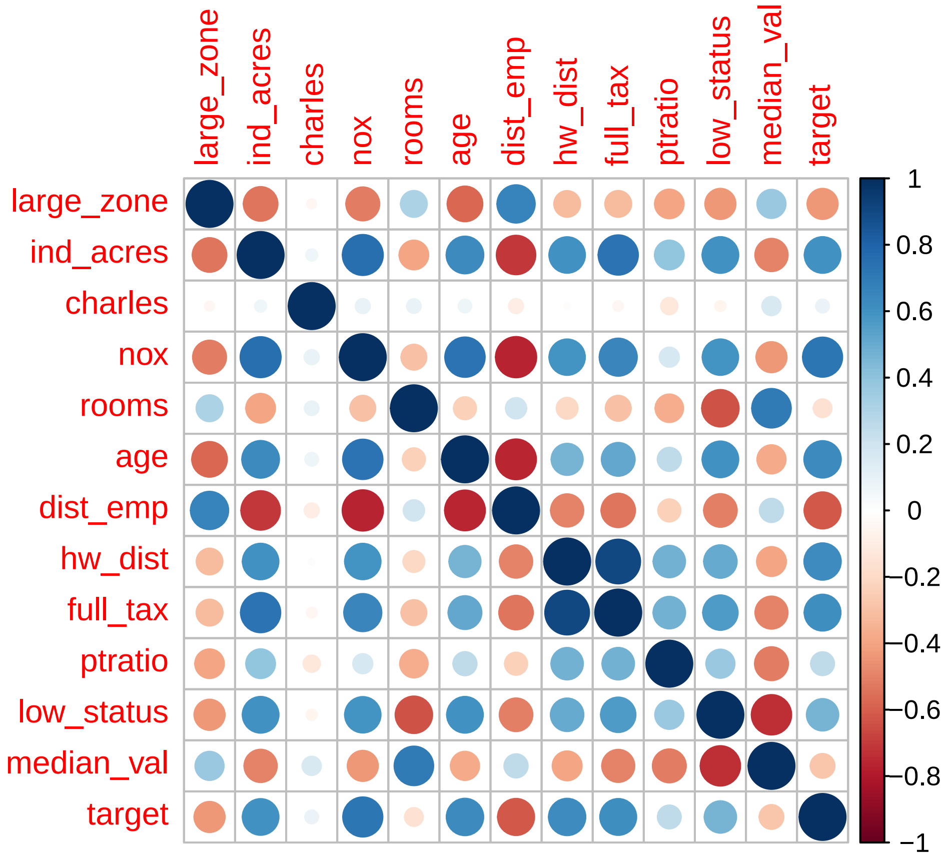
Full Tax Value



The full value property tax data is bimodal, as are age and industrial acreage. All of these observations stand to reason, and it looks like our dataset is pretty clean and good without alterations. Since the dependent variable is binary, we should be looking for bimodal frequency patterns.

We tried applying a log transformation to median_val, but it didn't make it through the stepwise selection process.

Let's look at the correlations to examine some more relationships among the independents.



Here we see that the greatest correlations are among the pairs is highway-distance/full-tax-value. Nitrogen compounds in the air correlate heavily with industrial acreage, age, and the target variable. This stands to reason. Distance to the Charles river and number of rooms are not particularly correlated with the target variable, and distance to employment centers, large zoning, and median value are uncorrelated to the target.

2. Data Preparation

Because full_tax and hw_dist are so highly correlated, it seemed logical to combine them. It also seemed useful to combine nox and ind_acres for the same reason. I also added a combined low_status and dist_emp term. Each of these pairs was multiplied together and added to the independent terms in the logistic regression.

I didn't feel it was important to bin any of this data. After mutations, I converted every value to a value between 0 and 1. I've had good success with this in the past.

3. Build Models

For building the models, I used a stepwise approach. That means, we regress each independent variable to the target, one at a time, and we throw out all variables that have a p value greater than 0.15. Of the remaining variables, the one with the smallest p value will be added, and others progressively mixed in.

We tried this first on the raw, untransformed data, and then on the transformed dataset. The transformed data performed better.

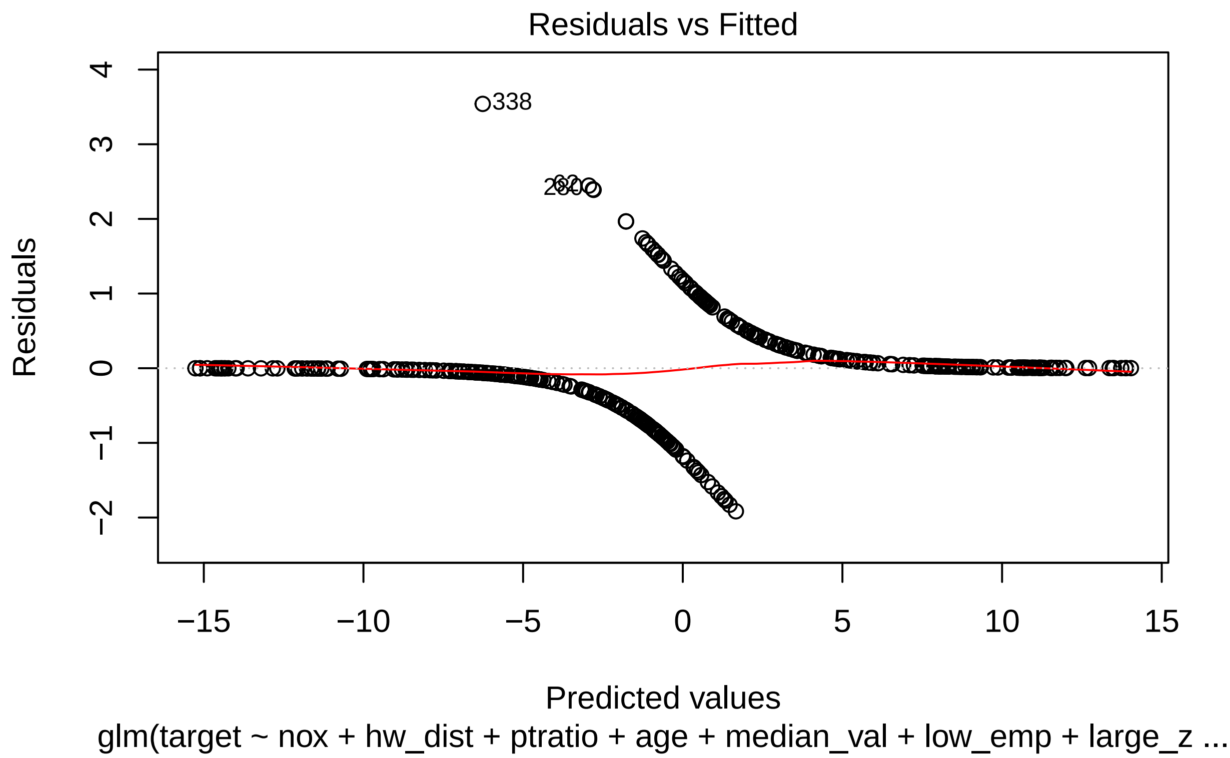
```
##
## Call:  glm(formula = target ~ nox + hw_dist + ptratio + age + median_val +
##     low_emp + large_zone + tax_dist + dist_emp, family = "binomial",
##     data = train.trans)
##
## Coefficients:
## (Intercept)          nox      hw_dist      ptratio          age   median_val
##     -17.945       21.026       29.293        3.575        2.461        6.780
##     low_emp   large_zone     tax_dist     dist_emp
##       3.436       -7.983      -22.894        4.656
##
## Degrees of Freedom: 465 Total (i.e. Null);  456 Residual
## Null Deviance:       645.9
## Residual Deviance: 193.3     AIC: 213.3
```

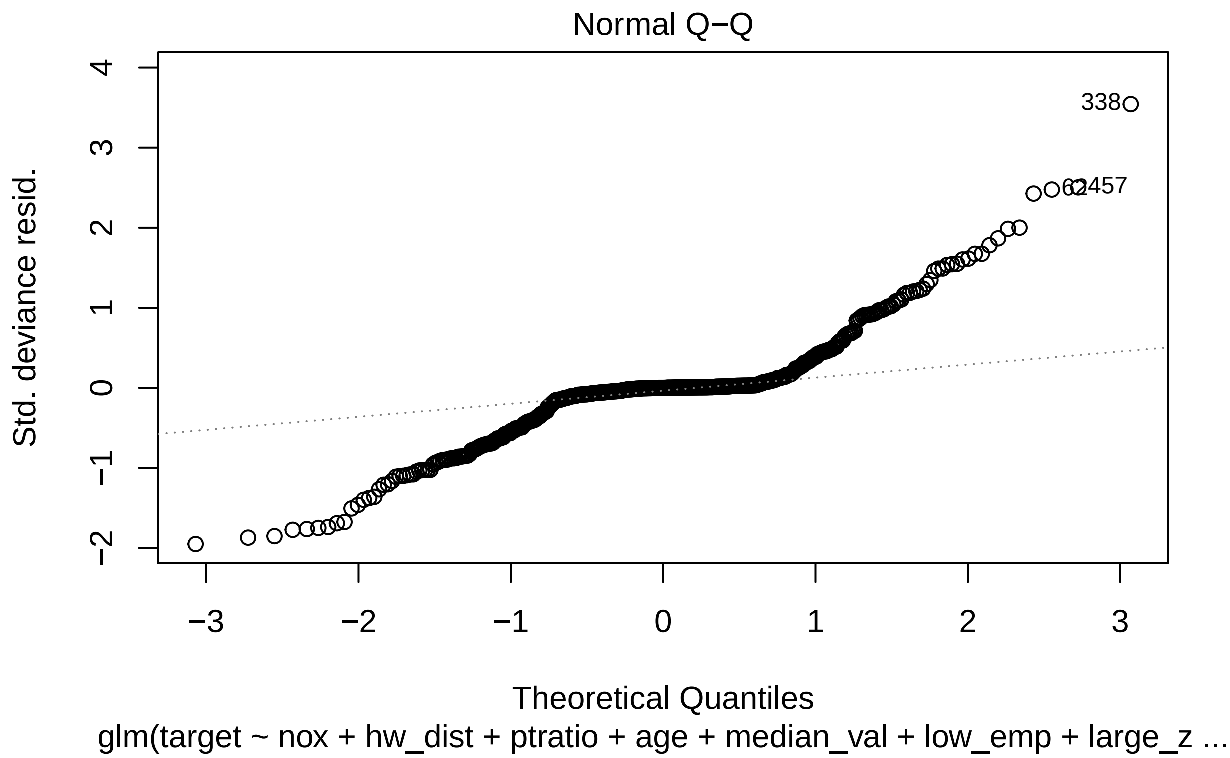
The coefficients of this model suggest a few things. High values for nox and distance from a highway, coupled with low values for median value, ptratio, and the combined full tax status and distance from the highway, together signal a high crime area. There are other significant factors at play, but these are the most significant factors. It makes sense.

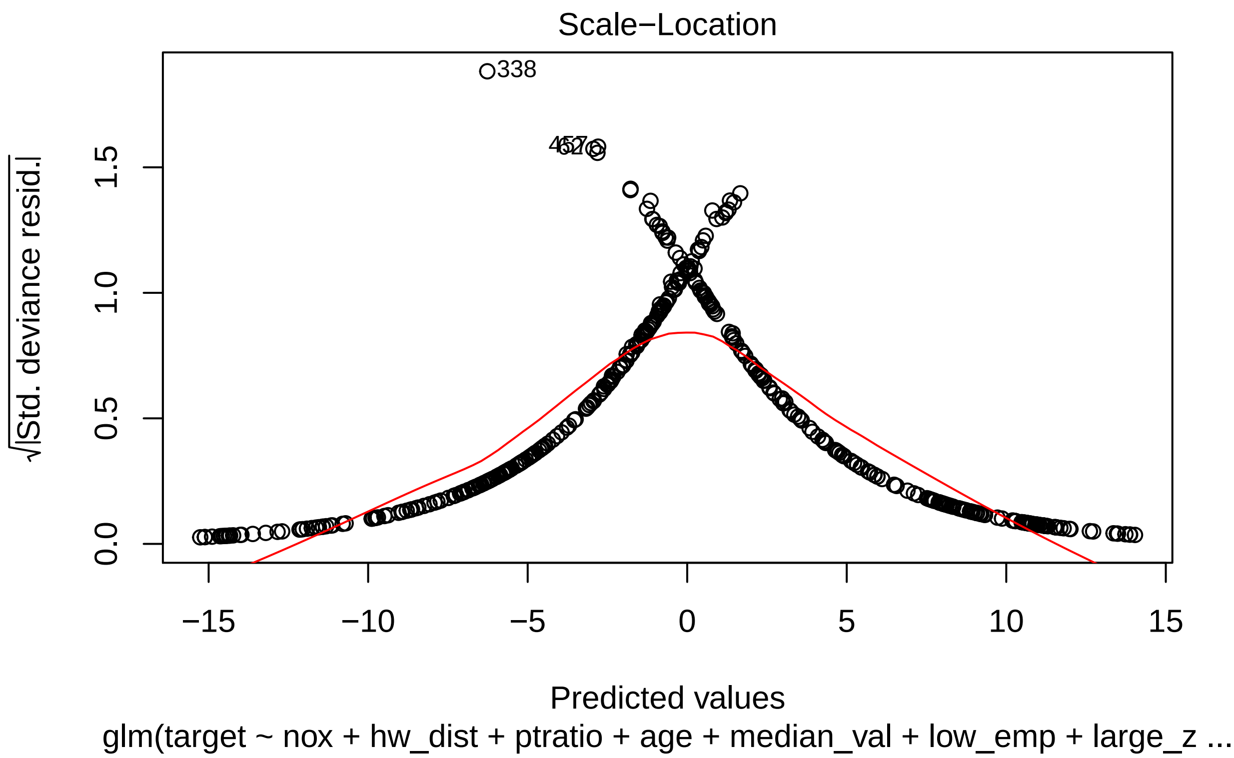
4. Select Models

The two final models scored similarly in AIC. The natural data model scored 215.32, and the transformed model scored 213.33. It seems that standardizing the data had almost no effect, and combining full_tax and hw_dist was slightly useful.

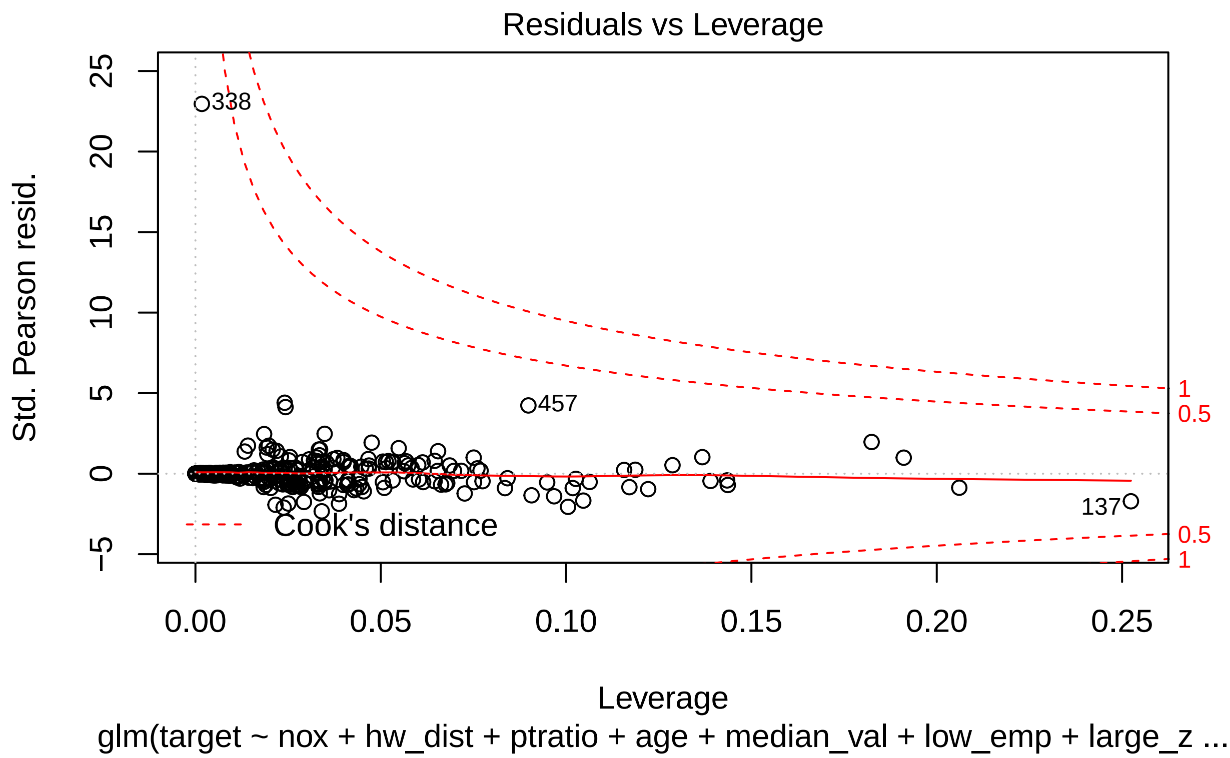
Examining the residuals, we can see that this model fits the training set extremely well. There are two clear and separate groups that are easily distinguishable. The residuals have relatively normal variance, and none of the outliers are beyond Cook's distance. Our fit line accurately recognizing most points, according to the Q-Q plot, and all the values that fall off the line are extreme in a specific desired direction.


Residuals vs Fitted
Residuals
4
3
2
1
0
−1
−2
338
−15
−10
−5
0
5
10
15
Predicted values
glm(target ~ nox + hw_dist + ptratio + age + median_val + low_emp + large_z ...


Normal Q−Q
Std. deviance resid.
Theoretical Quantiles
glm(target ~ nox + hw_dist + ptratio + age + median_val + low_emp + large_z ...
338
62457
−2
−1
0
1
2
3
4
−3
−2
−1
0
1
2
3



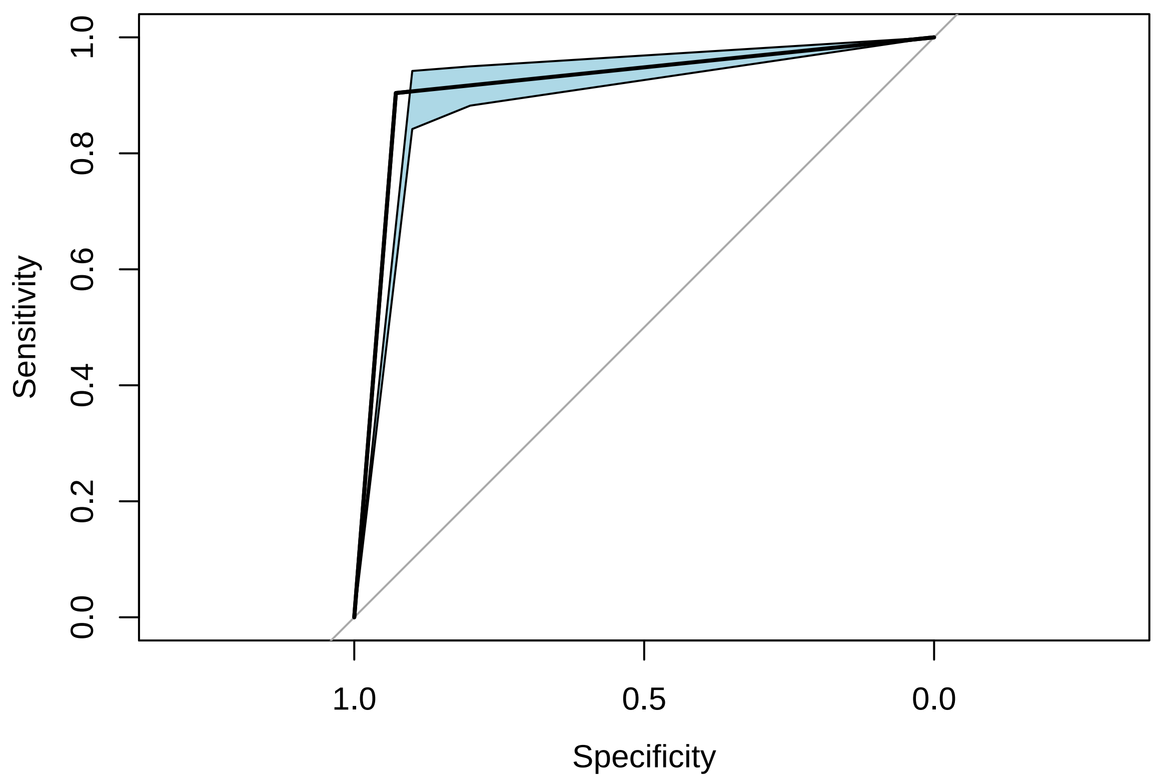
Scale−Location
338
457
452
√|Std. deviance resid.|
0.0
0.5
1.0
1.5
−15
−10
−5
0
5
10
15
Predicted values
glm(target ~ nox + hw_dist + ptratio + age + median_val + low_emp + large_z ...




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

## Setting direction: controls < cases

## Warning in plot.ci.se(sens.ci, type = "shape", col = "lightblue"): Low
## definition shape.
```




```
## Area under the curve: 0.9161
```

```
##       fit.values
## truth   0   1
##     0 220  17
##     1  22 207
```

```
(accuracy <- (true.pos + true.neg) / total)
```

```
## [1] 0.916309
```

```
(class.error <- 1 - accuracy)
```

```
## [1] 0.08369099
```

```
(precision <- true.pos / (true.pos + false.pos))
```

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

```
(sensitivity <- true.pos / (true.pos + false.neg))
```

```
## [1] 0.9241071
```

```
(specificity <- true.neg / (true.neg + false.pos))
```

```
## [1] 0.9090909
```

```
(f1 <- 2 * precision * sensitivity / (precision + sensitivity))
```

```
## [1] 0.9139073
```

Across the board, we can see that this model is effective. There is a chance it is overtrained, but I believe that chance is low. I would confidently use this model to predict new values.

After transforming the test set in the same way, we make these 40 predictions:

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
##  0  1  1  1  1  1  1  0  0  0  1  1  1  1  1  1  1  1  0  0  0  0  1  1  1  1
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40
##  0  1  1  1  1  1  1  1  1  1  1  1  1  1
```

Appendix – Code

```
library(tidyverse)
library(corrplot)
library(My.stepwise)
library(pROC)


vars <- c('large_zone', 'ind_acres', 'charles', 'nox', 'rooms', 'age',
          'dist_emp', 'hw_dist', 'full_tax', 'ptratio', 'low_status',
          'median_val', 'target')

train <- read.csv('crime-training-data_modified.csv') %>%
  tibble() %>%
  setNames(vars)

eval <- read.csv('crime-evaluation-data_modified.csv') %>%
  tibble() %>%
  setNames(vars)

sapply(train, class)
summary(train[, vars])


par(mfrow = c(2,2))
hist(as.numeric(as.character(train$large_zone)),
  main = '', ylab = '', axes = FALSE, xlab = 'Large Zoning')
hist(as.numeric(as.character(train$median_val)),
  main = '', ylab = '', axes = FALSE, xlab = 'Median Value')
hist(as.numeric(as.character(train$age)),
  main = '', ylab = '', axes = FALSE, xlab = 'Age')
hist(as.numeric(as.character(train$full_tax)),
  main = '', ylab = '', axes = FALSE, xlab = 'Full Tax Value')

par(mfrow = c(2,2))
hist(as.numeric(as.character(train$low_status)),
  main = '', ylab = '', axes = FALSE, xlab = 'Low Status')
hist(as.numeric(as.character(train$nox)),
  main = '', ylab = '', axes = FALSE, xlab = 'NOX')
hist(as.numeric(as.character(train$ind_acres)),
  main = '', ylab = '', axes = FALSE, xlab = 'Industrial Acres')


ct <- cor(train)
corrplot(ct)


normalize <- function(x) {
  xmin <- min(x)
  xmax <- max(x)
  return(lapply(x, function(x) (x-xmin) / (xmax-xmin)))
}
```

```
vars.trans <- c(vars, 'log_val', 'nox_ind', 'tax_dist', 'low_emp')

train.trans <- train %>%
  mutate(log_val = log(median_val)) %>%
  mutate(nox_ind = nox * ind_acres) %>%
  mutate(tax_dist = full_tax * hw_dist) %>%
  mutate(low_emp = low_status * dist_emp) %>%
  transmute_all(normalize) %>%
  transmute_all(unlist) %>%
  tibble() %>%
  setNames(vars.trans)


My.stepwise.glm(Y = 'target', variable.list = vars,
                in.variable = 'NULL', data = train, sle = 0.15,
                sls = 0.15, myfamily = 'binomial', myoffset = 'NULL')

My.stepwise.glm(Y = 'target', variable.list = vars.trans,
                in.variable = 'NULL', data = train.trans, sle = 0.15,
                sls = 0.15, myfamily = 'binomial', myoffset = 'NULL')

(bm <- glm(target ~ nox + hw_dist + ptratio + age + median_val + low_emp +
             large_zone + tax_dist + dist_emp,
           train.trans,
           family = 'binomial'))

plot(bm)


truth <- bm$model$target
fit.values <- ifelse(bm$fitted.values >= 0.5, 1, 0)

proc_obj <- roc(truth, fit.values,
                smoothed = TRUE, ci = TRUE, ci.alpha = 0.9,
                stratified = FALSE, plot = TRUE)
sens.ci <- ci.se(proc_obj)

plot(sens.ci, type="shape", col="lightblue")


auc(proc_obj)

(conf_matrix <- table(truth, fit.values))
total <- 466

true.neg <- conf_matrix[1,1]
true.pos <- conf_matrix[2,2]
false.neg <- conf_matrix[1,2]
false.pos <- conf_matrix[2,1]

(accuracy <- (true.pos + true.neg) / total)

(class.error <- 1 - accuracy)
```

```
(precision <- true.pos / (true.pos + false.pos))

(sensitivity <- true.pos / (true.pos + false.neg))

(specificity <- true.neg / (true.neg + false.pos))

(f1 <- 2 * precision * sensitivity / (precision + sensitivity))


test <- eval %>%
  mutate(log_val = log(median_val)) %>%
  mutate(nox_ind = nox * ind_acres) %>%
  mutate(tax_dist = full_tax * hw_dist) %>%
  mutate(low_emp = low_status * dist_emp) %>%
  transmute_all(normalize) %>%
  transmute_all(unlist) %>%
  tibble() %>%
  setNames(vars.trans[-13])

pred <- predict(bm, test, type="response")
(pred <- ifelse(pred >= 0.5, 1, 0))
```