BDA 551 Post-Class Homework #1

Feray Ece Topcu

June 30, 2018

Question 1

1. This exercise involves the Boston housing data set.

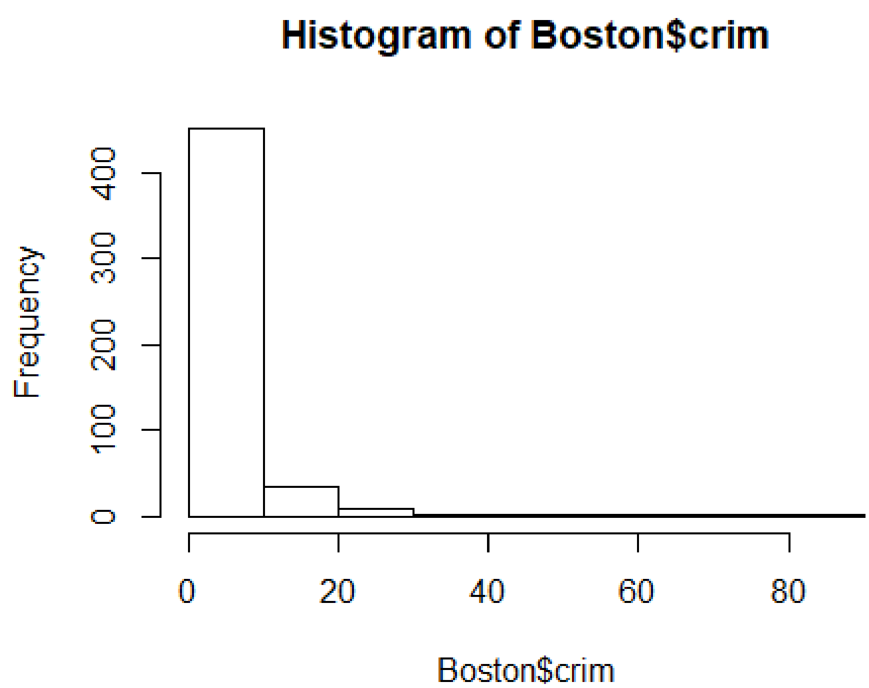
   a) To begin, load in the Boston data set. The Boston data set is part of the MASS library in R.How many rows are in this data set? How many columns? What do the rows and columns represent?

- Dimension and properties of Columns:

```
library(MASS)
#Boston
dim(Boston)

## [1] 506  14

str(Boston)

## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.5
24 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

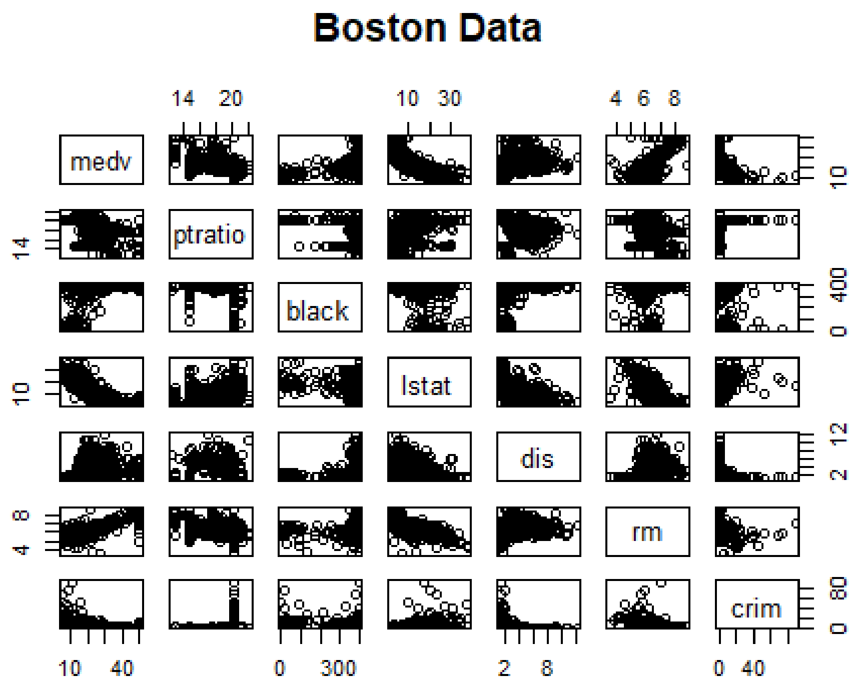
- Explanation of columns and Histogram of Crim Rate:

```
?Boston

## starting httpd help server ... done

hist(Boston$crim)
```



(b) Make some pairwise scatterplots of the predictors (columns) inthis data set. Describe your findings.

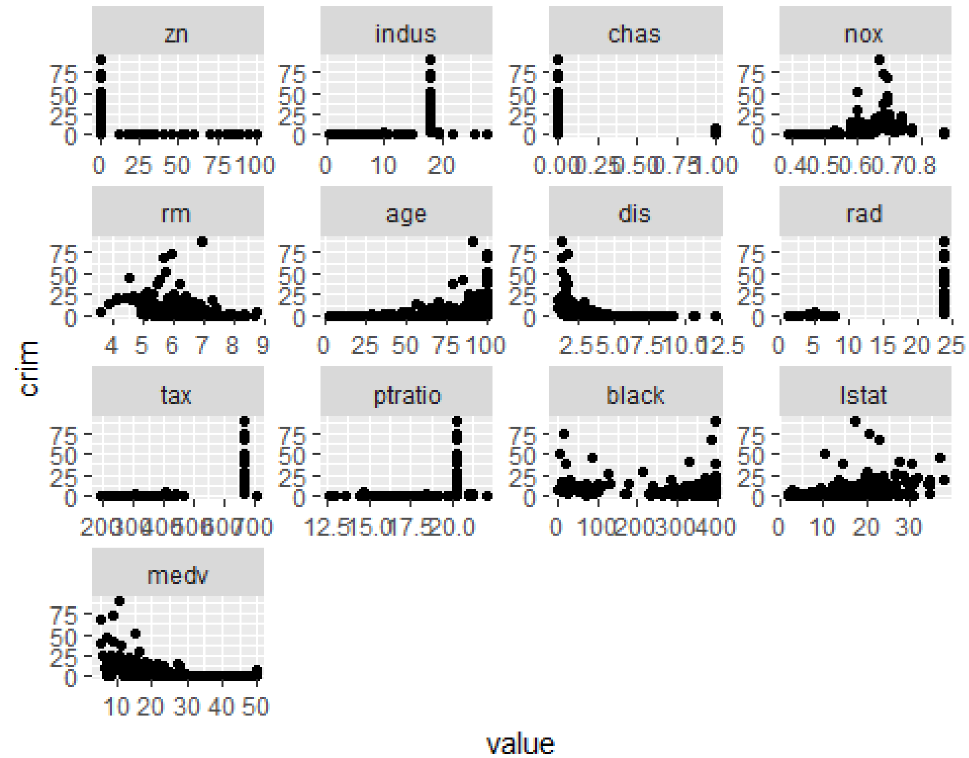
```
library(ggplot2)

#ggplot(data = Boston, aes(x = medv)) +  geom_histogram()

pairs(~ medv + ptratio + black + lstat + dis + rm + crim, data = Boston, main
= "Boston Data")
```



lstat, dis and rm are good linear variables while crim seems not linear, in fact the relationship is quite complicated.

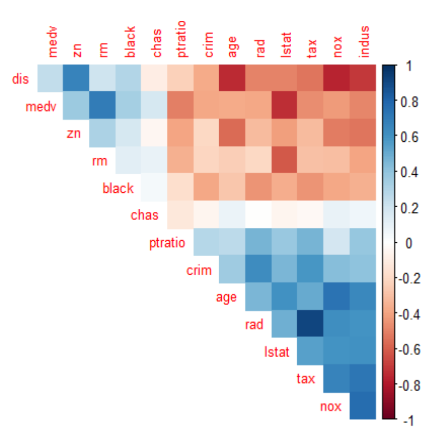
(c) Are any of the predictors associated with per capita crime rate? If so, explain the relationship.

```
library(reshape2)
# plot each feature against crim rate
bosmelt <- melt(Boston, id="crim")
ggplot(bosmelt, aes(x=value, y=crim))+
  facet_wrap(~variable, scales="free")+
  geom_point()
```



Due to correlation matrix; there is indeed an association between the per capita crime rate (crim) and the other predictors. Especially; dis,medv****,Black****,rad and tax but rad and tax highly correlated so, it is better to select one of them as predictor.

```r
library(corrplot)

## corrplot 0.84 loaded

#plot correlation:
M<-cor(Boston)
corrplot(M, diag = FALSE, order = "FPC",
         tl.pos = "td", tl.cex = 0.8, method = "color",type="upper")
```

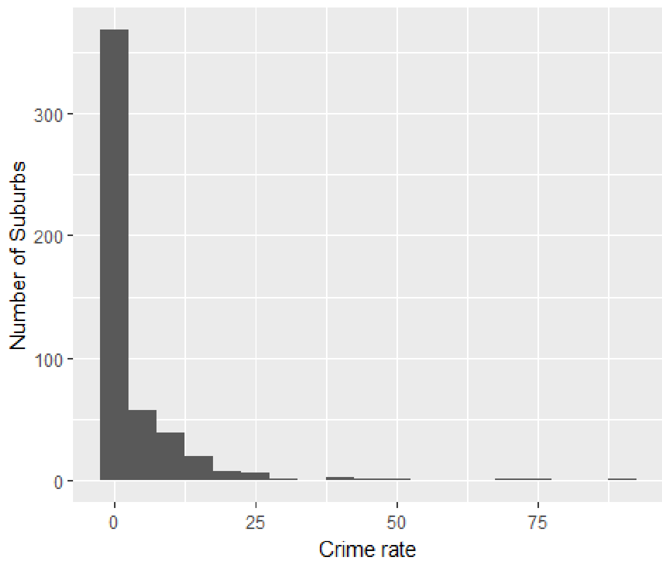


(d) Do any of the suburbs of Boston appear to have particularly high crime rates? Tax rates? Pupil-teacher ratios? Comment on the range of each predictor.

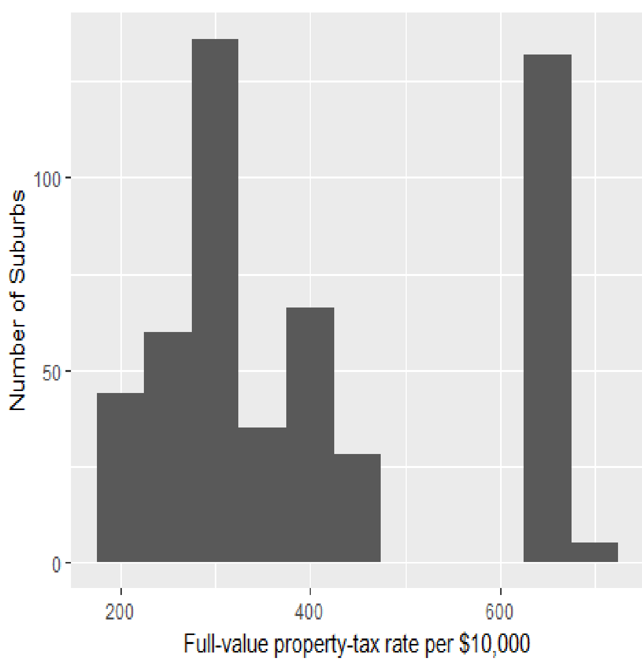
```
summary(Boston$crim)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.00632  0.08204  0.25651  3.61352  3.67708 88.97620

summary(Boston$tax)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   187.0   279.0   330.0   408.2   666.0   711.0

summary(Boston$ptratio)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12.60   17.40   19.05   18.46   20.20   22.00
```

Considering that the median and maximum crime rate values are respectively about 0.26% and 89%, we can see that there are some neighborhoods where the crime rate is extremely high:

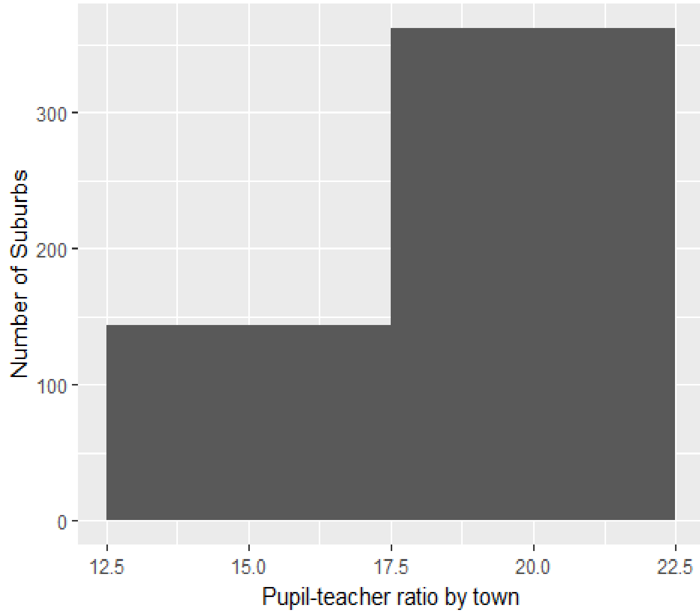
```
library(dplyr)

boston <-Boston
boston %>% filter(crim >=10) %>% summarise(count=n(), perc=n()/dim(boston)[1]
)   # 54 , ~11%

##   count      perc
## 1    54 0.1067194

# 11% of the neighborhood’s have crime rates above 10%
boston %>% filter(crim >=50) %>% summarise(count=n(), perc=n()/dim(boston)[1]
)

##   count        perc
## 1     4 0.007905138

# 0.8% of the neighborhoods have crim rates above 50%

#graph
qplot(Boston$crim, binwidth=5 , xlab = "Crime rate", ylab="Number of Suburbs"
)
```




```
qplot(Boston$tax, binwidth=50 , xlab = "Full-value property-tax rate per $10,
000", ylab="Number of Suburbs")
```




```
qplot(Boston$ptratio, binwidth=5, xlab ="Pupil-teacher ratio by town", ylab="
Number of Suburbs")
```



Based on the histogram of the Tax rates, there are few neighborhoods where rates are relative higher.

```
boston %>% filter(tax >=600) %>% summarise(count=n(), perc=n()/dim(boston)[1]
)

##   count     perc
## 1   137 0.270751

# 73% of the neighborhood pay under $600
```

(e) How many of the suburbs in this data set bound the Charles River

```
boston%>%
  group_by(chas) %>%
  summarise(Count = n()) %>%
  arrange(desc(Count))

## # A tibble: 2 x 2
##    chas Count
##   <int> <int>
## 1     0   471
## 2     1    35
```

(f) What is the median pupil-teacher ratio among the towns in this data set?

```
summary(boston$ptratio)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12.60   17.40   19.05   18.46   20.20   22.00
```

The median pupil-teacher ratio is 19 pupils for each teacher.

(g) Which suburb of Boston has lowest median value of owner- occupied homes? What are the values of the other predictors for that suburb,and how do those values compare to the overall ranges for those predictors? Comment on your findings.

```
selection <- Boston[order(Boston$medv),]
selection[1,]
```

```
##        crim zn indus chas   nox    rm age    dis rad tax ptratio black
## 399 38.3518  0  18.1    0 0.693 5.453 100 1.4896  24 666    20.2 396.9
##     lstat medv
## 399 30.59    5
```

Suburban 399 has the lowest median.

Suburd #399: * Crime is very high compared to median and average rates of all Boston neighborhoods. * No residential land zoned for lots over 25,000 sq.ft. This applies to more than half of the neighborhoods in Boston * Proportion of non-retail business acres per town is very high compared to most suburbs. * This suburd is not one of the suburbs that bound the Charles river. * Nitrogen oxides concentration (parts per 10 million) is one of the highest. * Average number of rooms per dwelling is one of the lowest * Highest proportion of owner proportion of owner-occupied units built prior to 1940. * One of the lowest weighted mean of distances to five Boston employment centres. * Highest index of accessibility to radial highways. * One of the highest full-value property-tax rate per $10,000. * One of the highest pupil-teacher ratio by town * Highest value for 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town. * One of the highest lower status of the population (percent) * Lowest median value of owner-occupied homes in $1000s. Based on the list above, suburb 399 can be classified as one of the least desirable places to live in Boston.

(h) In this data set, how many of the suburbs average more than seven rooms per dwelling? More than eight rooms per dwelling? Commenton the suburbs that average more than eight rooms per dwelling.

```
boston_rms = boston %>% filter(rm>7)
boston_rms%>%
  summarise(Count = n()) %>%
  arrange(desc(Count))
```

```
##   Count
## 1    64
```

There are 64 suburbs with more than 7 rooms per dwelling.

```
boston_rms%>%
  summarise(Count = n()) %>%
  arrange(desc(Count))
```

```
##   Count
## 1    64
```

There are 13 suburbs with more than 8 rooms per dwelling.

```
rm8 <- subset(Boston, rm>8)
summary(rm8)
```

```
##       crim               zn             indus             chas       
##  Min.   :0.02009   Min.   : 0.00   Min.   : 2.680   Min.   :0.0000  
##  1st Qu.:0.33147   1st Qu.: 0.00   1st Qu.: 3.970   1st Qu.:0.0000  
##  Median :0.52014   Median : 0.00   Median : 6.200   Median :0.0000  
##  Mean   :0.71879   Mean   :13.62   Mean   : 7.078   Mean   :0.1538  
##  3rd Qu.:0.57834   3rd Qu.:20.00   3rd Qu.: 6.200   3rd Qu.:0.0000  
##  Max.   :3.47428   Max.   :95.00   Max.   :19.580   Max.   :1.0000  
##       nox               rm             age             dis       
##  Min.   :0.4161   Min.   :8.034   Min.   : 8.40   Min.   :1.801  
##  1st Qu.:0.5040   1st Qu.:8.247   1st Qu.:70.40   1st Qu.:2.288  
##  Median :0.5070   Median :8.297   Median :78.30   Median :2.894  
##  Mean   :0.5392   Mean   :8.349   Mean   :71.54   Mean   :3.430  
##  3rd Qu.:0.6050   3rd Qu.:8.398   3rd Qu.:86.50   3rd Qu.:3.652  
##  Max.   :0.7180   Max.   :8.780   Max.   :93.90   Max.   :8.907  
##       rad              tax           ptratio          black      
##  Min.   : 2.000   Min.   :224.0   Min.   :13.00   Min.   :354.6  
##  1st Qu.: 5.000   1st Qu.:264.0   1st Qu.:14.70   1st Qu.:384.5  
##  Median : 7.000   Median :307.0   Median :17.40   Median :386.9  
##  Mean   : 7.462   Mean   :325.1   Mean   :16.36   Mean   :385.2  
##  3rd Qu.: 8.000   3rd Qu.:307.0   3rd Qu.:17.40   3rd Qu.:389.7  
##  Max.   :24.000   Max.   :666.0   Max.   :20.20   Max.   :396.9  
##      lstat           medv     
##  Min.   :2.47   Min.   :21.9  
##  1st Qu.:3.32   1st Qu.:41.7  
##  Median :4.14   Median :48.3  
##  Mean   :4.31   Mean   :44.2  
##  3rd Qu.:5.12   3rd Qu.:50.0  
##  Max.   :7.44   Max.   :50.0
```

Question 2

Using the Boston data set, fit classification models in order to predict whether a given suburb has a crime rate above or below the median. Explore logistic regression, LDA, and QDA models using various sub- sets of the predictors. Describe your findings.

- From coll matrix: we know dis,medv,Black,rad,tax but rad and tax highly correlated.

Logistic Regression:

- Prepare data: Mark crim column due to its value. (convert it into categorical variable for classification algo.'s)

```
#attach(Boston)
#Boston
boston <- Boston
#dim(boston)
```

```r
boston$crim <- ifelse(boston$crim>=0.520,1, 0) #bigger than median or not. (c
onvert cat variable!)
```

Logistic Regression: crim and tax

```r
glm.fit=glm(crim~tax,data=boston,family=binomial) #fit

#summarY:
summary(glm.fit)

##
## Call:
## glm(formula = crim ~ tax, family = binomial, data = boston)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6355  -0.6218  -0.3924   0.3289   2.0775
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.269807   0.418306  -12.60   <2e-16 ***
## tax          0.012252   0.001064   11.52   <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: 687.46  on 505  degrees of freedom
## Residual deviance: 404.10  on 504  degrees of freedom
## AIC: 408.1
##
## Number of Fisher Scoring iterations: 5

coef(glm.fit)

## (Intercept)         tax
## -5.26980719  0.01225194

#Conf.Matrix:
glm.probs=predict(glm.fit,type="response")
#length(glm.probs)

glm.pred=rep(0,length(glm.probs))
glm.pred[glm.probs>.5]=1

table(glm.pred,boston$crim)

##
## glm.pred   0   1
##        0 272  72
##        1  23 139
```

```r
#Accuracy:
mean(glm.pred==boston$crim)  # 0.812253

## [1] 0.812253
```

- This model is not bad due to accuracy and p value of tax.
- AIC: 408.1, Accuracy=0.812253

Logistic Regression: crim and dis,medv,Black,tax

```r
glm.fit=glm(crim~tax+dis+medv+black,data=boston,family=binomial) #fit

#summarY:
summary(glm.fit)

##
## Call:
## glm(formula = crim ~ tax + dis + medv + black, family = binomial,
##     data = boston)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5897  -0.4773  -0.1749   0.3852   2.3110
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  6.124355   2.389330   2.563 0.010371 *
## tax          0.008354   0.001208   6.915 4.68e-12 ***
## dis         -0.706143   0.119856  -5.892 3.82e-09 ***
## medv         0.029853   0.017222   1.733 0.083021 .
## black       -0.021598   0.006017  -3.590 0.000331 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 687.46  on 505  degrees of freedom
## Residual deviance: 310.93  on 501  degrees of freedom
## AIC: 320.93
##
## Number of Fisher Scoring iterations: 7

coef(glm.fit)

##  (Intercept)          tax          dis         medv        black
##  6.124354933  0.008354061 -0.706142731  0.029853253 -0.021598424

#Conf.Matrix:
glm.probs=predict(glm.fit,type="response")
#length(glm.probs)
```

```
glm.pred=rep(0,length(glm.probs))
glm.pred[glm.probs>.5]=1

table(glm.pred,boston$crim)

##
## glm.pred   0   1
##        0 276  39
##        1  19 172

#Accuracy:
mean(glm.pred==boston$crim)  # 0.8853755

## [1] 0.8853755
```

- This model is better due to accuracy and p values of predictors. tax, dis and black are important predictors on predicting crim due to p-values.
- AIC: 320.93, Accuracy=0.8853755

Logistic Regression: crim and all predictors

```
glm.fit=glm(crim~tax+dis+medv+black+zn+indus+chas+nox+rm+age+rad+ptratio+blac
k+lstat,data=boston,family=binomial) #fit

#summarY:
summary(glm.fit)

##
## Call:
## glm(formula = crim ~ tax + dis + medv + black + zn + indus +
##     chas + nox + rm + age + rad + ptratio + black + lstat, family = binomi
al,
##     data = boston)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.85874  -0.10524  -0.00077   0.00043   2.88043
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -42.160626   9.581892  -4.400 1.08e-05 ***
## tax          -0.010973   0.006654  -1.649 0.099111 .
## dis           1.188759   0.325535   3.652 0.000261 ***
## medv          0.248499   0.081913   3.034 0.002416 **
## black        -0.025259   0.007718  -3.273 0.001065 **
## zn           -0.087732   0.068324  -1.284 0.199121
## indus        -0.252269   0.080269  -3.143 0.001673 **
## chas          0.669611   0.886223   0.756 0.449902
```

```
## nox          83.620377  15.529261   5.385 7.26e-08 ***
## rm           -2.307175   0.867175  -2.661 0.007801 **
## age           0.084490   0.020420   4.138 3.51e-05 ***
## rad           0.600192   0.172130   3.487 0.000489 ***
## ptratio       0.412852   0.158510   2.605 0.009199 **
## lstat        -0.094821   0.062421  -1.519 0.128750
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 687.46  on 505  degrees of freedom
## Residual deviance: 139.89  on 492  degrees of freedom
## AIC: 167.89
##
## Number of Fisher Scoring iterations: 10
```

```
coef(glm.fit)
```

```
##  (Intercept)          tax          dis         medv        black
## -42.16062572  -0.01097349   1.18875856   0.24849873  -0.02525891
##           zn        indus         chas          nox           rm
##  -0.08773162  -0.25226850   0.66961137  83.62037688  -2.30717544
##          age          rad      ptratio        lstat
##   0.08449031   0.60019242   0.41285203  -0.09482087
```

```
#Conf.Matrix:
glm.probs=predict(glm.fit,type="response")
#length(glm.probs)

glm.pred=rep(0,length(glm.probs))
glm.pred[glm.probs>.5]=1

table(glm.pred,boston$crim)
```

```
##
## glm.pred   0   1
##        0 281  21
##        1  14 190
```

```
#Accuracy:
mean(glm.pred==boston$crim)  # 0.93083
```

```
## [1] 0.93083
```

- This model is the best. Accuracy is 0.93 and due to p-values, we ignored some important predictors such as nox, age and room on previous logistic regression models.
- AIC: 167.89, Accuracy=0.93083

- When we compare these three logistic regression models; we should select the third one, includes all predictors to predict crime rate, has the smallest AIC and highest accuracy.

Linear Discriminant Analysis

Linear Discriminant Analysis: crim and dis,medv,Black,tax

```
library(MASS)
lda.fit=lda(crim~tax+dis+medv+black,data=boston)

lda.fit

## Call:
## lda(crim ~ tax + dis + medv + black, data = boston)
##
## Prior probabilities of groups:
##        0        1
## 0.583004 0.416996
##
## Group means:
##        tax      dis     medv    black
## 0 308.9322 4.863315 24.89186 388.6363
## 1 547.0758 2.301487 19.23460 311.9874
##
## Coefficients of linear discriminants:
##                LD1
## tax    0.006016632
## dis   -0.296734712
## medv   0.010679906
## black -0.002616755

lda.preds=predict(lda.fit,type="response")
names(lda.preds)

## [1] "class"     "posterior" "x"

lda.class=lda.preds$class
table(lda.class,boston$crim)

##
## lda.class   0   1
##         0 274  43
##         1  21 168

mean(lda.class==boston$crim)

## [1] 0.8735178
```

Linear Discriminant Analysis: crim and all predictors

```
lda.fit=lda(crim~tax+dis+medv+black+zn+indus+chas+nox+rm+age+rad+ptratio+blac
k+lstat,data=boston)
```

```
lda.fit

## Call:
## lda(crim ~ tax + dis + medv + black + zn + indus + chas + nox +
##     rm + age + rad + ptratio + black + lstat, data = boston)
##
## Prior probabilities of groups:
##        0        1
## 0.583004 0.416996
##
## Group means:
##        tax      dis     medv    black        zn     indus       chas
## 0 308.9322 4.863315 24.89186 388.6363 18.677966  7.531051 0.06101695
## 1 547.0758 2.301487 19.23460 311.9874  1.137441 16.177962 0.08056872
##         nox       rm      age       rad  ptratio     lstat
## 0 0.4798736 6.396915 54.04983  4.345763 18.04576  9.812983
## 1 0.6593033 6.127654 88.88246 16.824645 19.02844 16.623791
##
## Coefficients of linear discriminants:
##                   LD1
## tax     -3.690944e-05
## dis      7.416012e-02
## medv     2.651362e-02
## black   -1.763019e-03
## zn       1.303495e-03
## indus    3.793903e-03
## chas    -2.691351e-01
## nox      9.078341e+00
## rm      -9.809320e-02
## age      1.353819e-02
## rad      1.012571e-01
## ptratio -4.186948e-02
## lstat   -1.004439e-02

lda.preds=predict(lda.fit,type="response")
names(lda.preds)

## [1] "class"     "posterior" "x"

lda.class=lda.preds$class
table(lda.class,boston$crim)

##
## lda.class   0   1
##         0 295  39
##         1   0 172

mean(lda.class==boston$crim)

## [1] 0.9229249
```

- Accuracy: 0.9229249

- Due to accuracy, LDA with all predictors is better than first LDA. However, Logistic regression with all predictors is better than LDA due to accuracy score.

Quadratic Discriminant Analysis

Quadratic Discriminant Analysis: crim and dis,medv,Black,tax

```
library(MASS)
lda.fit=qda(crim~tax+dis+medv+black,data=boston)

lda.fit

## Call:
## qda(crim ~ tax + dis + medv + black, data = boston)
##
## Prior probabilities of groups:
##        0        1
## 0.583004 0.416996
##
## Group means:
##        tax      dis     medv    black
## 0 308.9322 4.863315 24.89186 388.6363
## 1 547.0758 2.301487 19.23460 311.9874

lda.preds=predict(lda.fit,type="response")
names(lda.preds)

## [1] "class"     "posterior"

lda.class=lda.preds$class
table(lda.class,boston$crim)

##
## lda.class   0   1
##         0 287  50
##         1   8 161

mean(lda.class==boston$crim)  # 0.8853755

## [1] 0.8853755
```

Quadratic Discriminant Analysis: crim and all predictors

```
lda.fit=qda(crim~tax+dis+medv+black+zn+indus+chas+nox+rm+age+rad+ptratio+blac
k+lstat,data=boston)

lda.fit

## Call:
## qda(crim ~ tax + dis + medv + black + zn + indus + chas + nox +
##     rm + age + rad + ptratio + black + lstat, data = boston)
##
```

```
## Prior probabilities of groups:
##        0        1 
## 0.583004 0.416996 
## 
## Group means:
##        tax      dis     medv    black        zn     indus       chas
## 0 308.9322 4.863315 24.89186 388.6363 18.677966  7.531051 0.06101695
## 1 547.0758 2.301487 19.23460 311.9874  1.137441 16.177962 0.08056872
##         nox       rm      age       rad  ptratio     lstat
## 0 0.4798736 6.396915 54.04983  4.345763 18.04576  9.812983
## 1 0.6593033 6.127654 88.88246 16.824645 19.02844 16.623791
```

```
lda.preds=predict(lda.fit,type="response")
names(lda.preds)
```

```
## [1] "class"     "posterior"
```

```
lda.class=lda.preds$class
table(lda.class,boston$crim)
```

```
## 
## lda.class   0   1
##         0 295  25
##         1   0 186
```

```
mean(lda.class==boston$crim)   # 0.9505929
```

```
## [1] 0.9505929
```

- Due to accuracy, QDA with all predictors is better than first QDA and previos classification models but in any case, we need cross validation to avoid overfitting.

Question 3

- Try to predict per capita crime rate in the Boston data set.

*** (a) Try out some of the regression methods explored in the Chapter 6, such as best subset selection, the lasso, ridge regression, and PCR. Present and discuss results for the approaches that you consider. ***

- Call Ridge/Lasso library and mark x and y:

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-16
```

```
x=model.matrix(crim~.,boston)[,-1]  # matrix of all predictors.
y=boston$crim #predict y.
```

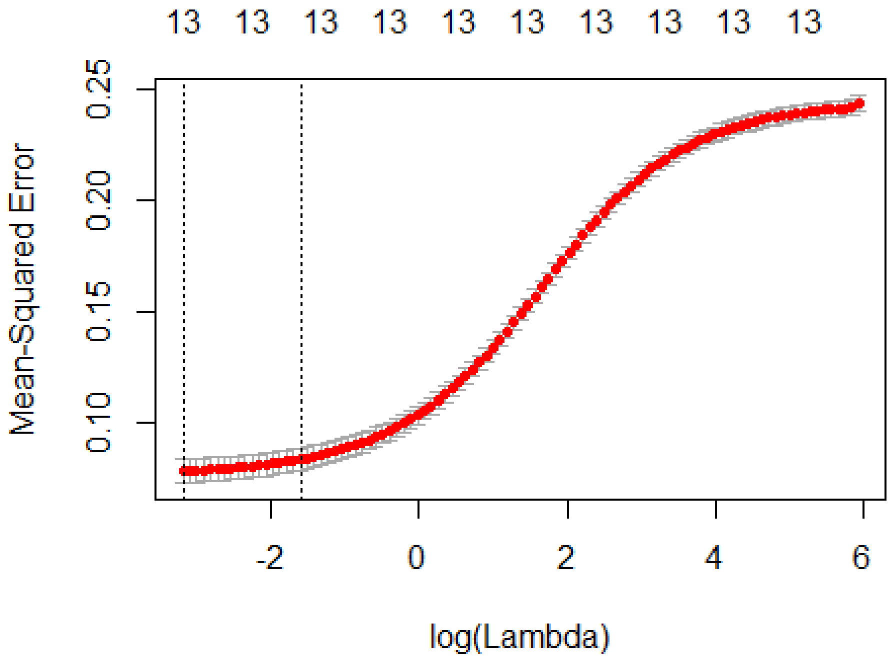
Ridge:

- Apply Ridge with 100 differend lambda:

```
#alpha=0 - ridge; 1 - lasso

grid=10^seq(10,-2,length=100)  # 100 different lambda.
ridge.mod=glmnet(x,y,alpha=0,lambda=grid)
dim(coef(ridge.mod))

## [1]  14 100

#ridge.mod$lambda[50] #  lambda= 11497.57
#coef(ridge.mod)[,50] # lambda=11497.57 iken Ã§Ä±kan coefficient deÄerleri.
sqrt(sum(coef(ridge.mod)[-1,50]^2)) # MSE.

## [1] 0.0001398968

#ridge.mod$lambda[60]
#coef(ridge.mod)[,60]
#sqrt(sum(coef(ridge.mod)[-1,60]^2))
predict(ridge.mod,s=50,type="coefficients")[1:14,]

##   (Intercept)            zn         indus          chas           nox
##  3.883510e-01 -7.323485e-05  4.228890e-04  7.220912e-04  3.111407e-02
##            rm           age           dis           rad           tax
## -1.198850e-03  1.016779e-04 -1.333703e-03  3.836242e-04  1.945867e-05
##       ptratio         black         lstat          medv
##  4.658888e-04 -2.132132e-05  3.043062e-04 -1.495093e-04
```

- Select best lambda is hard manually. So, Cross Validation is better:
- By default, the function performs ten-fold cross-validation.

```
set.seed(1)
cv.out <- cv.glmnet(x, y, alpha=0, nlambda=100, lambda.min.ratio=0.0001)
plot(cv.out)
```




```
best.lambda <- cv.out$lambda.min
best.lambda

## [1] 0.04135535
```

- use Ridge to predict with best lambda:

```
ridge.pred <- predict(ridge.mod, s=best.lambda, type="coefficients")[1:14, ]
```

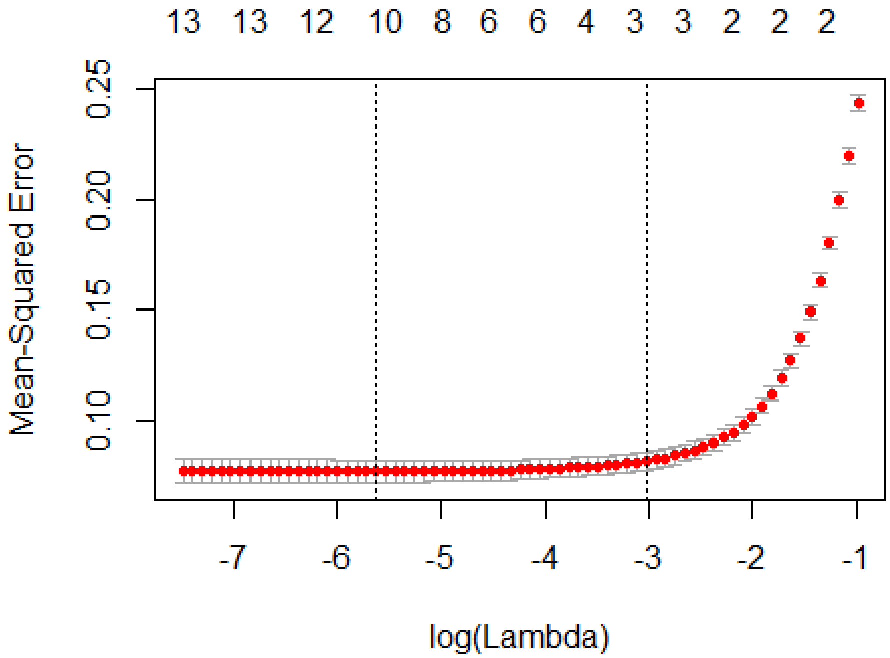
- Calculate MSE for Ridge with best lambda (train model with train data and ccalculate MSE for test data):

```
train=sample(1:nrow(x), nrow(x)/2)
test=(-train)
y.test=y[test]

ridge.mod=glmnet(x[train,],y[train],alpha=0,lambda=best.lambda, thresh=1e-12)
ridge.pred=predict(ridge.mod,s=best.lambda,newx=x[test,])
mean((ridge.pred-y.test)^2)  #0.084

## [1] 0.086315
```

Lasso:

- Try to find out best lambda on Lasso with ten-fold cross-validation: (when aplha=1 of glmnet, it's Lasso)

```
set.seed(1)
cv.out <- cv.glmnet(x, y, alpha=1, nlambda=100, lambda.min.ratio=0.0001)
plot(cv.out)
```




```
best.lambda <- cv.out$lambda.min
best.lambda

## [1] 0.003596878
```

- use Lasso to predict with best lambda:

```
predict(ridge.mod, s=best.lambda, type="coefficients")[1:14, ]

##   (Intercept)            zn         indus          chas           nox
## -0.5469884746  0.0002029699 -0.0002563473 -0.0887901426  1.4410963119
##            rm           age           dis           rad           tax
## -0.0013789536  0.0037190611 -0.0002823604  0.0142768969  0.0004946656
##       ptratio         black         lstat          medv
## -0.0211198023 -0.0003369088  0.0008004032  0.0039188820
```

- Calculate MSE for Lasso with best lambda (train model with train data and ccalculate MSE for test data):

```
ridge.mod=glmnet(x[train,],y[train],alpha=1,lambda=best.lambda, thresh=1e-12)
ridge.pred=predict(ridge.mod,s=best.lambda,newx=x[test,])
mean((ridge.pred-y.test)^2)  # 0.083

## [1] 0.08420347
```

- Due to MSE, Ridge is better than Lasso for this dataset for feature selection.

Best subset selection:

```
predict.regsubsets=function(object,newdata,id,...){  ## env a bir function ta
nÄ±mladÄ±k.
  form=as.formula(object$call[[2]])
  mat=model.matrix(form,newdata)
  coefi=coef(object,id=id)
  xvars=names(coefi)
  mat[,xvars]%*%coefi
}
```

- Try best subset selection with cross validation:

```
library(leaps)


regfit.full=regsubsets(crim~.,boston)
summary(regfit.full)

## Subset selection object
## Call: regsubsets.formula(crim ~ ., boston)
## 13 Variables  (and intercept)
##         Forced in Forced out
## zn          FALSE      FALSE
## indus       FALSE      FALSE
## chas        FALSE      FALSE
## nox         FALSE      FALSE
## rm          FALSE      FALSE
## age         FALSE      FALSE
## dis         FALSE      FALSE
## rad         FALSE      FALSE
## tax         FALSE      FALSE
## ptratio     FALSE      FALSE
## black       FALSE      FALSE
## lstat       FALSE      FALSE
## medv        FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          zn  indus chas nox rm  age dis rad tax ptratio black lstat medv
## 1  ( 1 ) " " " "   " "  "*" " " " " " " " " " " " "     " "   " "   " "
## 2  ( 1 ) " " " "   " "  "*" " " " " " " "*" " " " "     " "   " "   " "
## 3  ( 1 ) " " " "   " "  "*" " " " " " " "*" " " " "     " "   " "   "*"
## 4  ( 1 ) " " " "   " "  "*" " " "*" " " "*" " " " "     " "   " "   "*"
## 5  ( 1 ) " " " "   " "  "*" " " "*" " " "*" " " " "     "*"   " "   "*"
## 6  ( 1 ) " " " "   " "  "*" " " "*" "*" "*" " " " "     "*"   " "   "*"
## 7  ( 1 ) " " " "   " "  "*" " " "*" "*" "*" " " "*"     "*"   " "   "*"
## 8  ( 1 ) " " " "   "*"  "*" " " "*" "*" "*" " " "*"     "*"   " "   "*"

regfit.full=regsubsets(crim~.,data=boston,nvmax=19)
reg.summary=summary(regfit.full)
#reg.summary
which.max(reg.summary$adjr2)
```

```
## [1] 8

names(reg.summary)

## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"

reg.summary$rsq

##  [1] 0.5840526 0.6760848 0.6821002 0.6883987 0.6925228 0.6948550 0.6957132
##  [8] 0.6965903 0.6968859 0.6971528 0.6972367 0.6972752 0.6972761

# par(mfrow=c(2,2))
# plot(reg.summary$rss,xlab="Number of Variables",ylab="RSS",type="l")
# plot(reg.summary$adjr2,xlab="Number of Variables",ylab="Adjusted RSq",type=
"l")
# which.max(reg.summary$adjr2)
# points(11,reg.summary$adjr2[11], col="red",cex=2,pch=20)
# plot(reg.summary$cp,xlab="Number of Variables",ylab="Cp",type='l')
# which.min(reg.summary$cp)
# points(10,reg.summary$cp[10],col="red",cex=2,pch=20)
# which.min(reg.summary$bic)
# plot(reg.summary$bic,xlab="Number of Variables",ylab="BIC",type='l')
# points(6,reg.summary$bic[6],col="red",cex=2,pch=20)
# plot(regfit.full,scale="r2")
# plot(regfit.full,scale="adjr2")
# plot(regfit.full,scale="Cp")
# plot(regfit.full,scale="bic")
# coef(regfit.full,6)
#
# coef(regfit.full,8)
```

- Due to adjusted R-squared value (calculated by which.max(reg.summary$adjr2) code); 8. trial is the best. This trials select the columns: chas,nox,age,dis,rad,ptratio,black,medv.

(b) Propose a model (or set of models) that seem to perform well on this data set, and justify your answer. Make sure that you are evaluating model performance using validation set error, cross-validation, or some other reasonable alternative, as opposed to using training error.

- We know from Question 2, the best accuracy is generated by the QDA model with all predictors. When we try Ridge and Lasso, accuracy of them is not better than Logistic Regression, LDA and QDA with all predictors. So, we may try to generate a QDA model with predictors that selected by the best selection method.

Combine Best Selection and QDA

- QDA with crim and chas,nox,age,dis,rad,ptratio,black,medv while QDA with all predictors has the accuracy:0.950.

```
library(MASS)
lda.fit=qda(crim~chas+nox+age+dis+rad+ptratio+black+medv,data=boston)
```

```
lda.fit

## Call:
## qda(crim ~ chas + nox + age + dis + rad + ptratio + black + medv,
##     data = boston)
##
## Prior probabilities of groups:
##        0        1
## 0.583004 0.416996
##
## Group means:
##         chas       nox      age      dis       rad  ptratio    black
## 0 0.06101695 0.4798736 54.04983 4.863315  4.345763 18.04576 388.6363
## 1 0.08056872 0.6593033 88.88246 2.301487 16.824645 19.02844 311.9874
##       medv
## 0 24.89186
## 1 19.23460

lda.preds=predict(lda.fit,type="response")
names(lda.preds)

## [1] "class"     "posterior"

lda.class=lda.preds$class
table(lda.class,boston$crim)

##
## lda.class   0   1
##         0 292  31
##         1   3 180

mean(lda.class==boston$crim)  # 0.932

## [1] 0.9328063
```

- The result is perfect! We eliminate 5 columns due to the best selection method and accuracy of model is still higher than 90%.

Apply this model to train and test dataset:

- If we want to compare Ridge and Lasso with QDA with Best selection; we should check accuracy on splitted data.

```
dim(boston[train,])

## [1] 253  14

lda.fit=qda(crim~chas+nox+age+dis+rad+ptratio+black+medv,data=boston[train,])

#lda.fit
```

```
lda.predict=predict(lda.fit,boston[test,])



#mean((lda.predict$class-boston[test,])^2)


lda.class=lda.predict$class
table(lda.class,boston[test,]$crim)

##
## lda.class   0   1
##         0 157  19
##         1   0  77

mean(lda.class==boston[test,]$crim) #0.9407

## [1] 0.9249012
```

- Accuracy of Ridge on test dataset: 0.084
- Accuracy of Lasso on test dataset: 0.083
- Accuracy of QDA + Best selection on test dataset: 0.94

While we sample randomly train and test dataset, the accuracy values may differ for each run but it is clear that QDA + Best subset selection model is much more better than Ridge and Lasso due to accuracy.

(c) Does your chosen model involve all of the features in the data set? Why or why not?

According to some trials on Question 2 and 3, we see that the best model for this dataset is QDA. The model does not involve all the features because the best selection method shows us that with 9 important columns (chas,nox,age,dis,rad,ptratio,black,medv), model accuracy is still more than 90%. So, we select these important columns rather than all predictors for optimization of model and also execution time.