Homework 1 (IDS 575)

Homa Rashidisabet & Claudia Vesel

February 10, 2019

Homework 1

Question 5

What are the advantages and disadvantages of a very flexible (versus less flexible) approach for regression or classification? Under what circumstances might a more flexible approach be preferred to a less flexible approach? When might a less flexible approach be preferred?

The advantage of a very flexible approach is that it may be more robust in cases where a highly nonlinear model is necessary because it can generate a wide range of shapes to estimate the function. While this method can decrease the bias, and is preferred in cases where **prediction** is important, it can also lead to overfitting, or it may model the noise too closely. Also, the approach requires a large number of parameters, making it hard to interpret. A flexible model would thus perform better when the sample size is large and the number of predictors is small as it may decrease the likelihood of overfitting.

On the other hand, a more restrictive approach is preferred when **inference** is important. a simpler model will have a higher interpretability. While the fit will be more biased, it yields a more transparent relationship between individual predictors and their associate response. This is especially important when the variance of errors is large, or when the number of predictors is large, but the sample size is small.

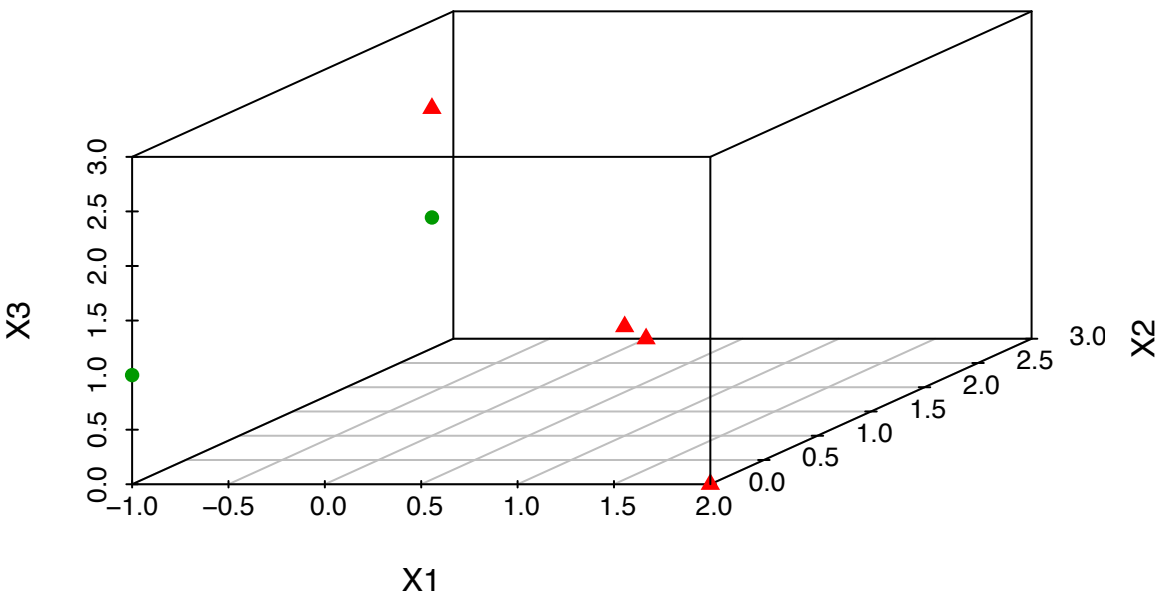
Question 7

The table below provides a training data set containing 6 observations, 3 predictors, and 1 qualitative response variable. Suppose we wish to use this data set to make a prediction for Y when X1 = X2 = X3 = 0 using K-nearest neighbors.

```
head(dataK)
```

```
##   X1 X2 X3     Y
## 1  0  3  0   Red
## 2  2  0  0   Red
## 3  0  1  3   Red
## 4  0  1  2 Green
## 5 -1  0  1 Green
## 6  1  1  1   Red
```

```
scatterplot3d(dataK[,1], dataK[,2], dataK[,3],main="Training Data",
              xlab = "X1",ylab = "X2",zlab = "X3",pch = shapes,color=colors)
```



Part a) Compute the Euclidean distance between each observation and the test point, X1 = X2 = X3 = 0.

```
library(foreach)
#a) Compute the Eucledian distance between (0,0,0) and all points
p1 = cbind(dataK$X1,dataK$X2,dataK$X3)
p2 = c(0,0,0)
euc.dist <- function(p1, p2) sqrt(sum((p1 - p2) ^ 2))

dist<-foreach(i = 1:nrow(p1), .combine = c ) %do% euc.dist(p1[i,],p2)
dataK$Distance<-dist

head(dataK)
```

```
##   X1 X2 X3     Y Distance
## 1  0  3  0   Red 3.000000
## 2  2  0  0   Red 2.000000
## 3  0  1  3   Red 3.162278
## 4  0  1  2 Green 2.236068
## 5 -1  0  1 Green 1.414214
## 6  1  1  1   Red 1.732051
```

Part b) and c): What is our prediction with K = 1? Why?

The code for K = 1 or K = 3 is the same. The smallest kth distances between the point $x_0 = [0, 0, 0]^T$ and the original data are counted. The color corresponding to the highest color becomes the prediction for the test point. The probability for each color can then be computed according to:

$$P(Y = Color|X = x_0) = \frac{1}{k}\sum_{i=1}^{N} I(y_i = Color)$$

```
## Loading required package: Rcpp
```

```
## Loading required package: RcppZiggurat
```

```
##
```

```
## Attaching package: 'reshape'

## The following objects are masked from 'package:plyr':
##
##     rename, round_any
```

```
k = 1 #Set K
threshold = nth(dataK$Distance, k) #smallest kth distance
dataK.sub <- subset(dataK, Distance <= threshold) ##Extract rows<= threshold
closestPts<-count(dataK.sub, "Y")
closestPts$prob<-closestPts$freq/sum(closestPts$freq)
#head(closestPts)
```

Thus, for $\mathbf{k = 1}$ the closest point is green, hence the conditional probability for **green** is 1.

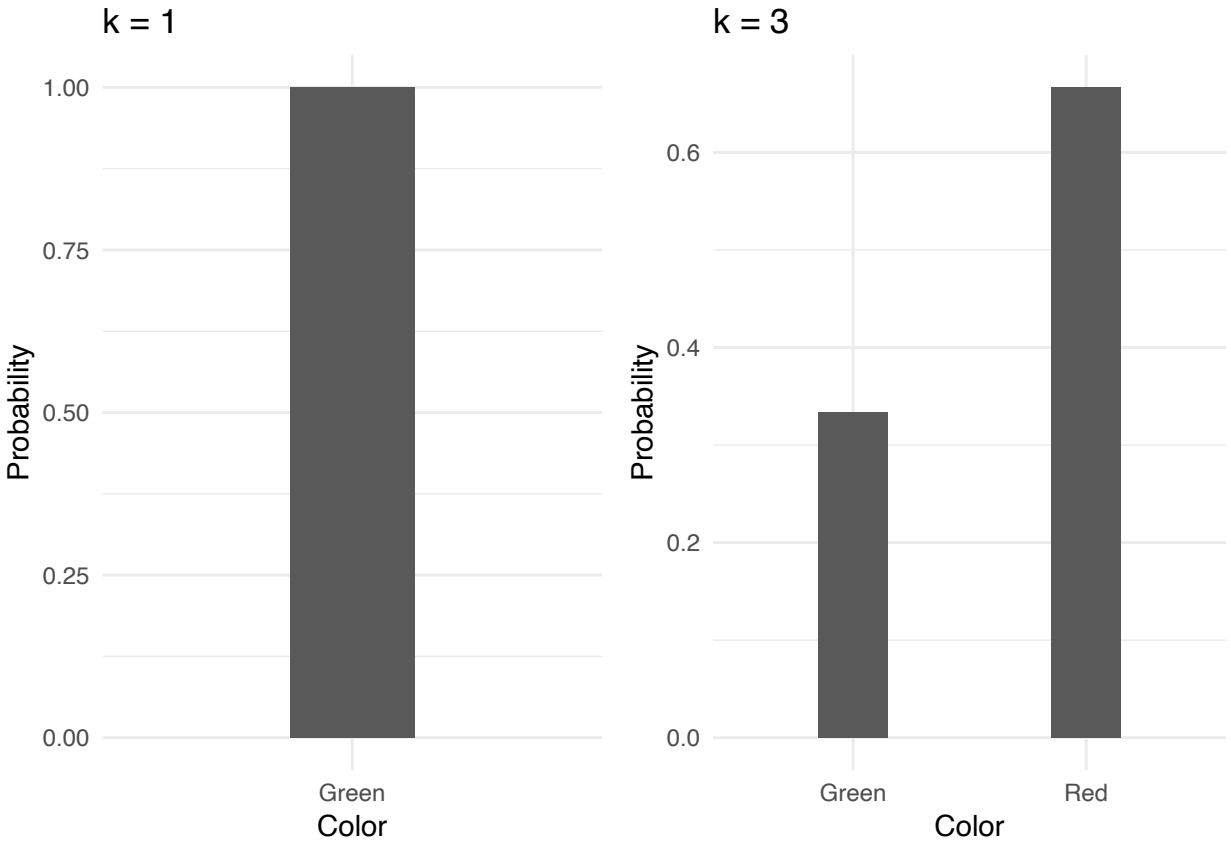
$$P(Y = red|X = x_0) = \frac{1}{1}\sum_{i=1}^{N} I(y_i = red) = 0$$

$$P(Y = green|X = x_0) = \frac{1}{1}\sum_{i=1}^{N} I(y_i = green) = 1$$

If $\mathbf{k = 3}$, then two of the closest points are red and the third one is green. Thus, there is a 2/3 probability that the test point is also **red**.

$$P(Y = red|X = x_0) = \frac{1}{3}\sum_{i=1}^{N} I(y_i = red) = \frac{2}{3}$$

$$P(Y = green|X = x_0) = \frac{1}{3}\sum_{i=1}^{N} I(y_i = green) = \frac{1}{3}$$



Part d) If the Bayes decision boundary in this problem is highly nonlinear, then would we expect the best value for K to be large or small ? Why ?

A highly nonlinear Bayesian boundary implies a high data variance and as K increases the model becomes less flexible so it would not be able to capture its trend. Thus a smaller K value should be chosen.

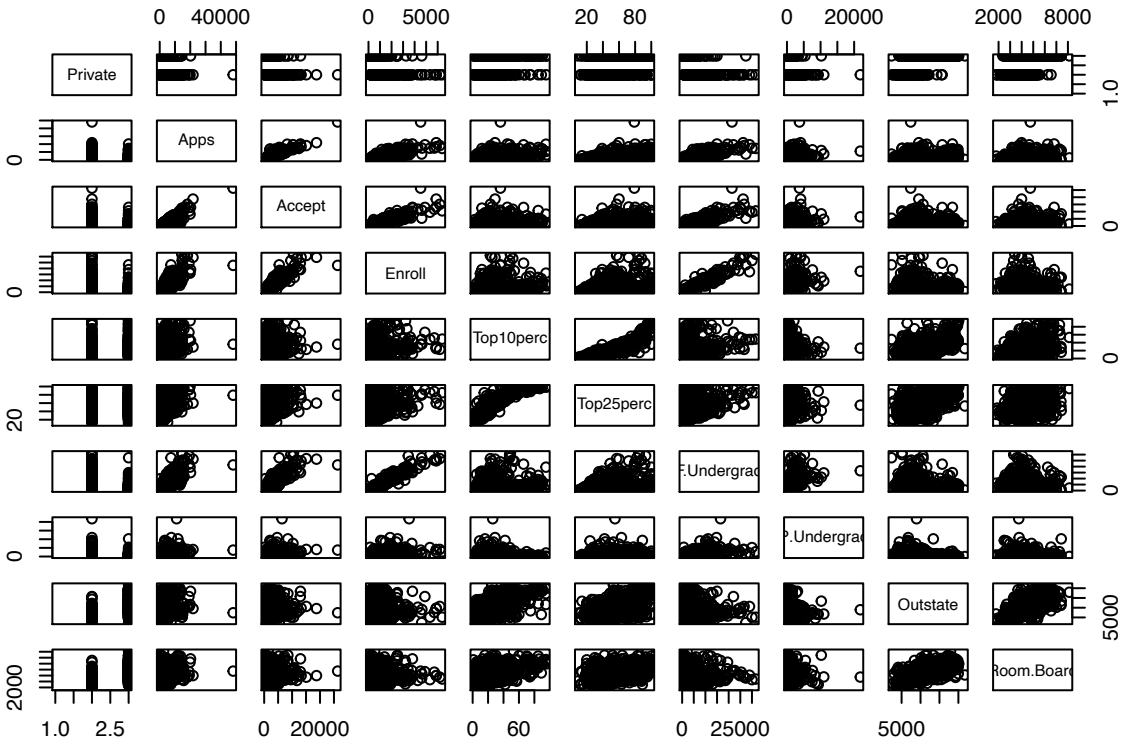
Question 8

Load the College.csv data set which contains variables for 777 universities and colleges in the US.

Part a) and b) Load the data and remove the name of the colleges

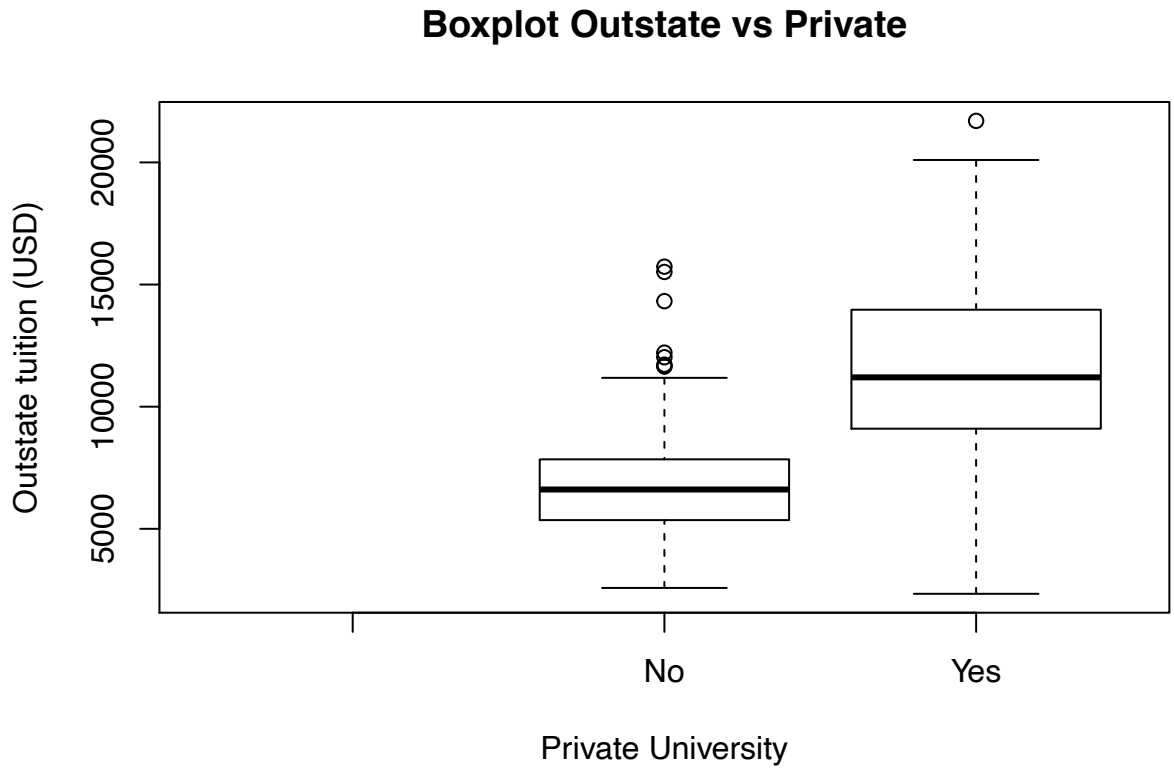
```
#Read data set
college<-read.csv('College.csv')
#fix(college)
#Assign the names of colleges
#rownames(college) = college[,1]
#fix(college)
#Get rid of names
college = college[,-1]
head(college)
```

```
##   Private Apps Accept Enroll Top10perc Top25perc F.Undergrad P.Undergrad
## 1     Yes 1660   1232    721        23        52        2885         537
## 2     Yes 2186   1924    512        16        29        2683        1227
## 3     Yes 1428   1097    336        22        50        1036          99
## 4     Yes  417    349    137        60        89         510          63
## 5     Yes  193    146     55        16        44         249         869
```



Part c(iii): Use the plot() function to produce side-by-side boxplots of Outstate versus Private universities.

```
#plots instate vs out of state
plot(college[,1],college[,9],xlab = "Private University",
     ylab ="Outstate tuition (USD)", main="Boxplot Outstate vs Private")
```



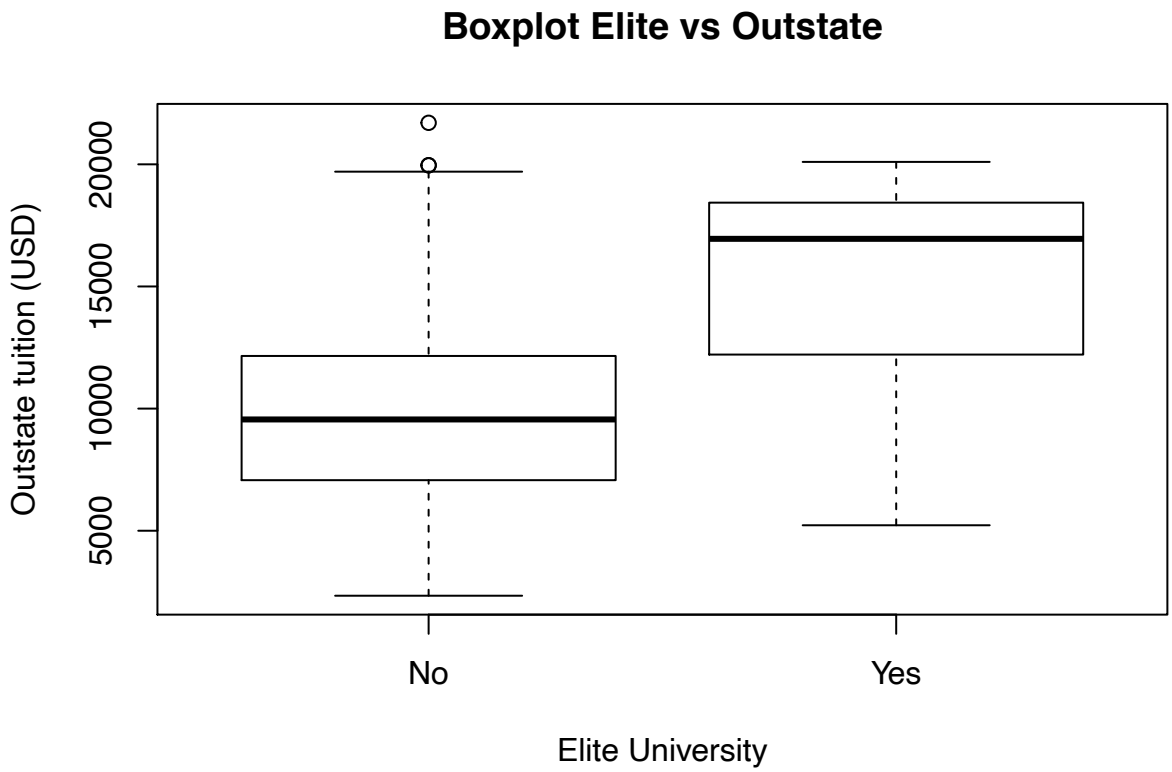
Part c(iv): Create a new variable called Elite and bin universities based on whether ot not the proportion

of students coming from the top 10% of their highschool classes exceeds 50%.

```
#Binning Top 10%
Elite = rep("No",nrow(college)) #construct a qualitative variable initialized w No
Elite[college$Top10perc>50] = "Yes"  #If Top10>50 students then change No to yes
Elite = as.factor(Elite)             #SetElite to a factor type of data instead of character
college$Elite = Elite                #Add the Elite column to the college data set
summary(Elite) #No = 699; Yes = 78
```

```
##  No Yes
## 706  78
```

```
plot(college$Elite,college$Outstate,xlab = "Elite University",
     ylab ="Outstate tuition (USD)", main="Boxplot Elite vs Outstate")
```



Part c(vi): Use hist() to produce histograms with different numbers of bvins for a few of the quantitative variables.

Another section of elite colleges was obtained from the data, which covered the colleges that addmitted over 85 students from the top 10% of their highschool.

```
#Get super elite (over 85 students in the top 10%)
superElite = rep("No",nrow(college))
superElite[college$Top10perc>85] = "Yes"
superElite = as.factor(superElite)
#summary(superElite)
college2 = data.frame(college,superElite)
superEliteColleges<-college2[which(college2[,'superElite'] == "Yes"),]
Names = rownames(superEliteColleges) #names of colleges
head(superEliteColleges)
```

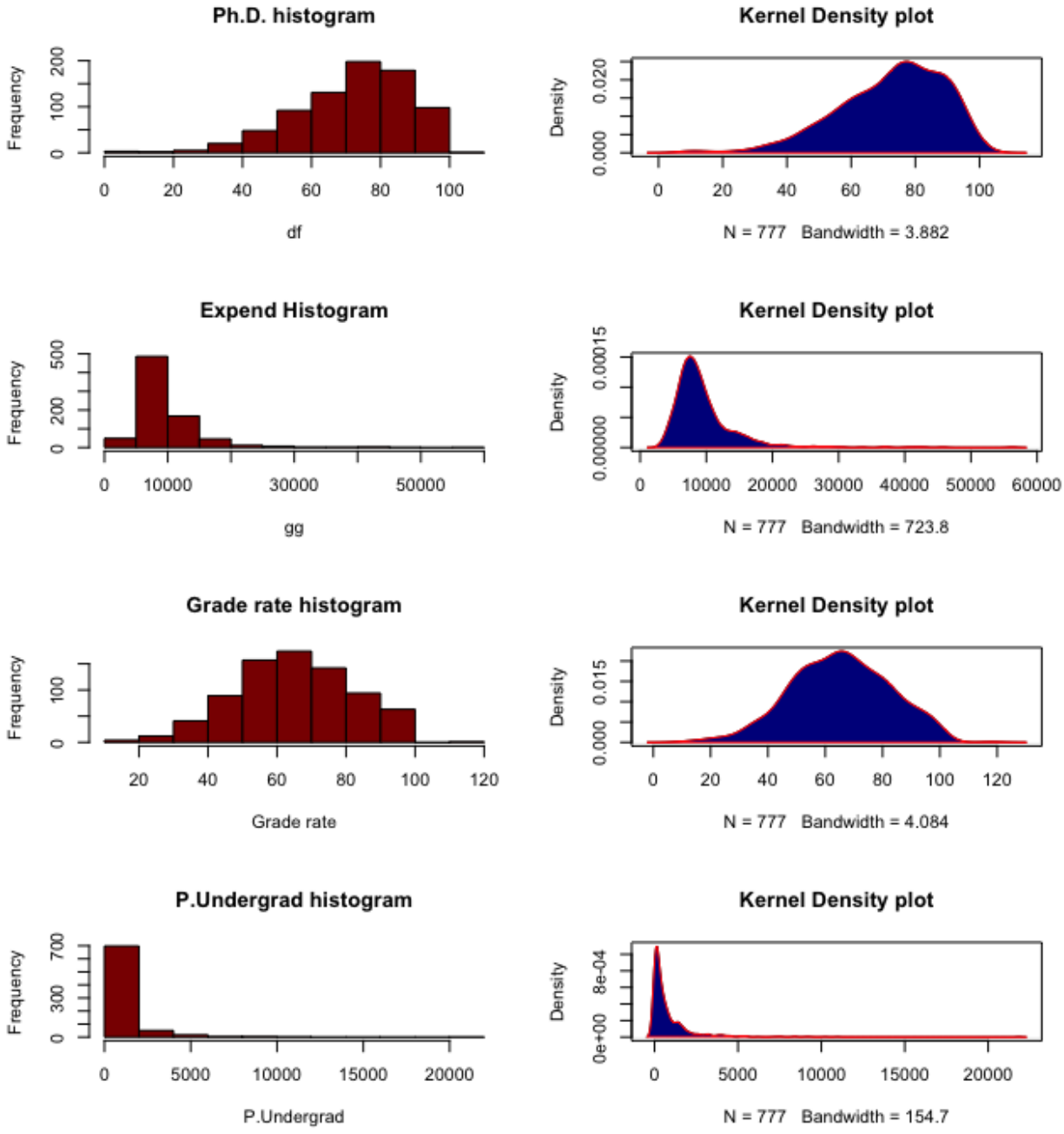
```
##     Private  Apps Accept Enroll Top10perc Top25perc F.Undergrad
```

```
## 71      Yes 12586   3239   1462        87        95        5643
## 159     Yes  8587   2273   1087        87        99        3918
## 175     Yes 13789   3893   1583        90        98        6188
## 223      No  7837   4527   2276        89        99        8528
## 251     Yes 13865   2165   1606        90       100        6862
## 252     Yes  1377    572    178        95       100         654
##     P.Undergrad Outstate Room.Board Books Personal PhD Terminal S.F.Ratio
## 71          349    19528       5926   720     1100  99      100       7.6
## 159          32    19545       6070   550     1100  95       99       4.7
## 175          53    18590       5950   625     1162  95       96       5.0
## 223         654     6489       4438   795     1164  92       92      19.3
## 251         320    18485       6410   500     1920  97       97       9.9
## 252           5    17230       6690   700      900 100      100       8.2
##     perc.alumni Expend Grad.Rate X.1 X.2 X.3 X.4 X.5 X.6 X.7 X.8 X.9 X.10
## 71           39  20440        97  NA  NA      NA
## 159          49  29619        98  NA  NA      NA
## 175          44  27206        97  NA  NA      NA
## 223          33  11271        70  NA  NA      NA
## 251          52  37219       100  NA  NA      NA
## 252          46  21569       100  NA  NA      NA
##     X.11 X.12 X.13 X.14 X.15 X.16 X.17 X.18 X.19 X.20 X.21 Elite
## 71                                                          Yes
## 159                                                         Yes
## 175                                                         Yes
## 223                                                         Yes
## 251                                                         Yes
## 252                                                         Yes
##     superElite
## 71         Yes
## 159        Yes
## 175        Yes
## 223        Yes
## 251        Yes
## 252        Yes
```

```r
#Applications vs acceptance data frame
ApsAcc = data.frame(Names,superEliteColleges$Apps,superEliteColleges$Accept)
colnames(ApsAcc) <-c("CollegeName","Applied","Accepted")

#Percent application vs acceptance
ApsAcc$percAcc <-((ApsAcc$Accepted/ApsAcc$Applied)*100)
```

Part c(v): For this section, we chose four predictors including Ph.D., expend, grade rate, and P.undergrad, and we ploted their histogram and distribution that shows each predictor's frequency and values as follows.

	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Terminal	S.F.Ratio	perc.alumni	Expend	Grad.Rate
min	81	72	35	1	9	139	1	2340	1780	96	250	8	24	2.5	0	3186	10
First quartile	776	604	242	15	41	992	95	7320	3597	470	850	62	71	11.5	13	6751	53
mean	3002	2019	780	28	56	3700	855	10441	4358	549	1341	73	80	14	23	9660	65
median	1558	1110	434	23	54	1707	353	9990	4200	500	1200	75	82	13.6	21	8377	65
third quartile	3624	2424	902	35	69	4005	967	12925	5050	600	1700	85	92	16.5	31	10830	78
max	48094	26330	6392	96	100	31643	21836	21700	8124	2340	6800	103	100	39.8	64	56233	118
mode	1006	452	295	20	60	1707	30	6550	4100	500	1000	77	96	12.1	10	7041	72

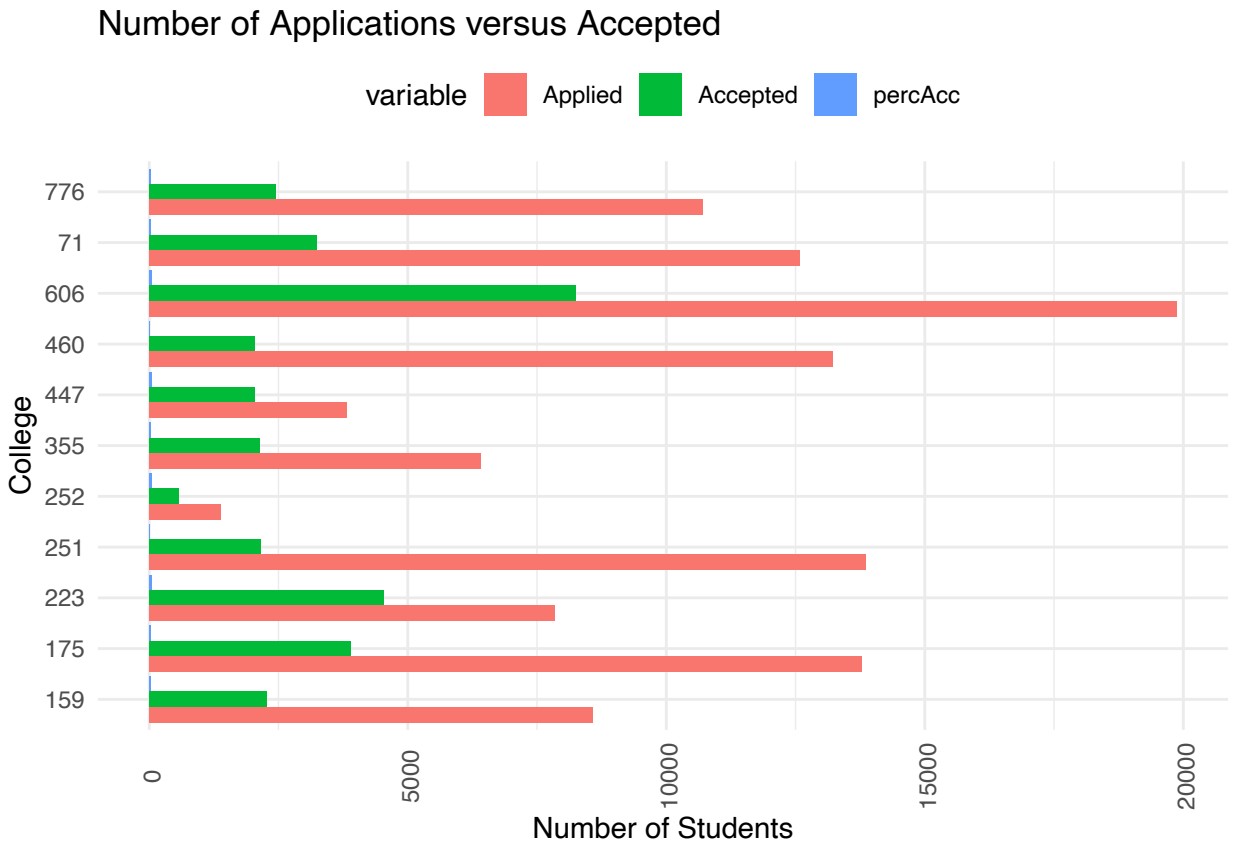



```r
college<-read.csv('College.csv')
# par(mfrow=c(4,2))
# hist(college[,14], breaks=12, col="dark red",xlab="df", main="Ph.D. histogram")
# d <- density(college[,14]);
# plot(d, "Kernel Density plot"); polygon(d, col="dark blue", border="red")
#
# hist(college[,18], breaks=12, col="dark red",xlab="gg", main="Expend Histogram")
# d <- density(college[,18]); plot(d, "Kernel Density plot");
# polygon(d, col="dark blue", border="red")
#
# hist(college[,19], breaks=12, col="dark red",xlab="Grade rate", main="Grade rate histogram")
```

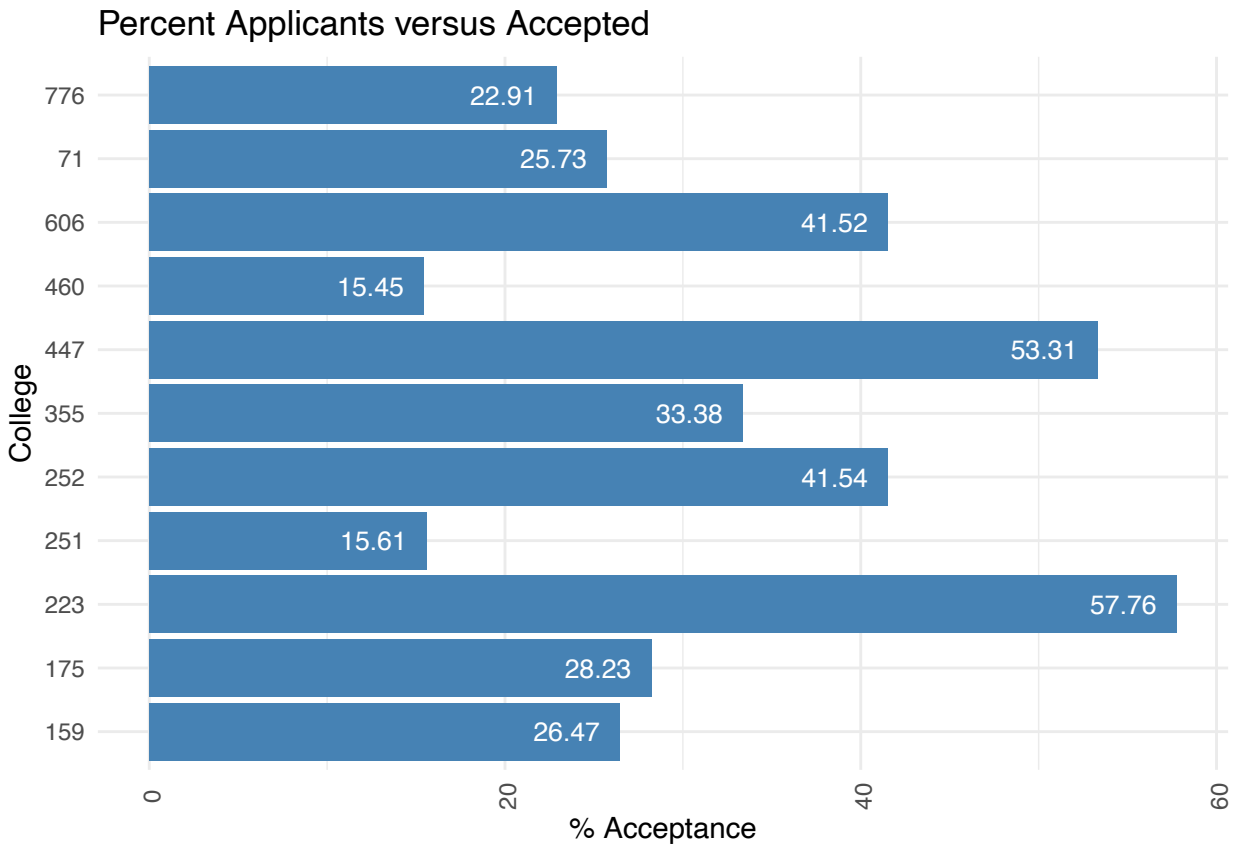
```
# d <- density(college[,19]); plot(d, "Kernel Density plot");
# polygon(d, col="dark blue", border="red")
#
# hist(college[,9], breaks=12, col="dark red",xlab="P.Undergrad", main="P.Undergrad histogram")
# d <- density(college[,9]); plot(d, "Kernel Density plot");
# polygon(d, col="dark blue", border="red")
```

In order to compare the acceptance rates in these elite colleges, the total number of submitted applicantions was compared to the total number of accepted students. Princeton and Harvard were found to have the smallest acceptance rates.

```
#Plotting
library(ggplot2)
library(reshape)
library(gridExtra)

#Plot 1: Number Apps vs Acceptance
ggplot(data=melt(ApsAcc, id ='CollegeName'),
       aes(x=CollegeName, y=value)) +
     geom_bar(aes(fill = variable),
              position = "dodge", stat="identity")+
     theme_minimal()+
     theme(text = element_text(size=11),
           axis.text.x  = element_text(angle=90, vjust=0.65))+
     theme(legend.position="top")+
     coord_flip()+
     ggtitle("Number of Applications versus Accepted")+
     xlab("College") + ylab("Number of Students")
```




```
#Plot 2: Percent applicants versus Applicaitons
ggplot(data.frame(ApsAcc$CollegeName,ApsAcc$percAcc),
       aes(x = ApsAcc$CollegeName, y = ApsAcc$percAcc))+
  geom_bar(stat="identity", fill="steelblue")+
  geom_text(aes(label=format(round(ApsAcc$percAcc, 2), nsmall = 2)),
            vjust=0.5, hjust = 1.3, color="white", size=3.5)+
  theme_minimal()+
  theme(text = element_text(size=11),
        axis.text.x  = element_text(angle=90, vjust=0.65))+
  ggtitle("Percent Applicants versus Accepted")+
  coord_flip()+
  xlab("College") + ylab("% Acceptance")
```



Question 10

Part a: *Mass library* is included in *r* and *Boston* data set is loaded.

```
library(MASS)

?Boston

BostonData <- data.frame(Boston);
colnames(BostonData) <- c('crim','zn','indus','chas','nox','rm','age',
                          'dis','rad','tax','ptratio','black','lstat','medv')
write.csv(BostonData, "Question10_BostonData.csv")
```

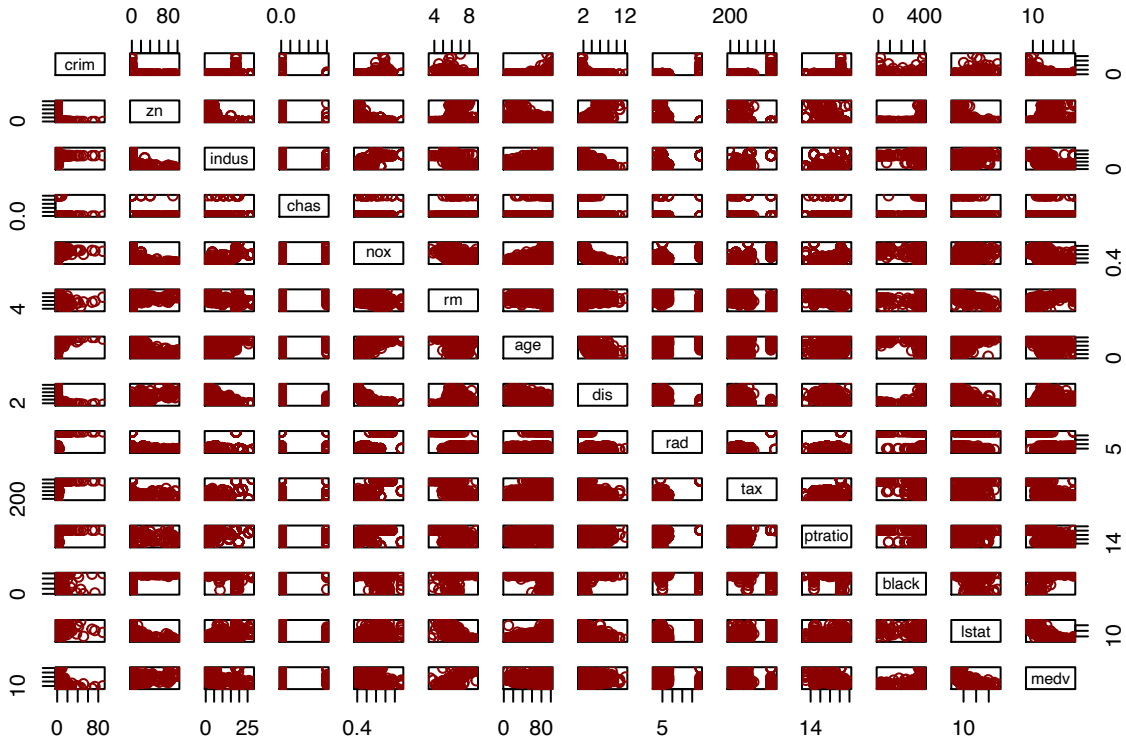
Housing Values in Suburbs of Boston **Format** rows corresponding to different formulations of the housing value equation and the columns corresponding to different formulations of the willingness-topay equation. Each row represents a new observation (suburb) each column is as follows: **crim** per capita crime rate by town. **zn** proportion of residential land zoned for lots over 25,000 sq.ft. **indus** proportion of non-retail business acres per town. **chas** Charles River dummy variable (= 1 if tract bounds river; 0 otherwise). **nox** nitrogen oxides concentration (parts per 10 million). **rm** average number of rooms per dwelling. **age** proportion of owner-occupied units built prior to 1940. **dis** weighted mean of distances to five Boston employment centres. **rad** index of accessibility to radial highways. **Tax** full-value property-tax rate per $10,000. **ptratio** pupil-teacher ratio by town. **black** 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town. **lstat** lower status of the population (percent). **medv** median value of owner-occupied homes in $1000s.

Part b: We plotted observation values for all permutations of each two features. The covariance matrix for the features is shown in the table to understand the dependence of features better. Based on both plot and

table, the prominent relations of each fixed feature with other features is given as follows:

Crime and *zn*, *dis* and *medv* have reverse relation while crime and age have direct relation. *zn* has reverse relation with *indus*, *nox* and *rad*, but it is highly correlated with *dis* and *medv*. Indus is highly (positively) correlated with *age*, *nox*, ptratio and istat, but negatively correlated with *dis*.

```
plot(Boston[1:14], col = "dark red")
```




```
summary(Boston)
```

```
##       crim                zn              indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000  
##       nox                rm              age              dis        
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
##       rad              tax           ptratio          black       
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32  
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38  
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44  
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67  
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23  
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90  
##      lstat            medv
```

Covariance Matrix

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
crim	74.0	-40.2	24.0	-0.1	0.4	-1.3	85.4	-6.9	46.8	844.8	5.4	-302.4	28.0	-30.7
zn	-40.2	543.9	-85.4	-0.3	-1.4	5.1	-373.9	32.6	-63.3	-1236.5	-19.8	373.7	-68.8	77.3
indus	24.0	-85.4	47.1	0.1	0.6	-1.9	124.5	-10.2	35.5	833.4	5.7	-223.6	29.6	-30.5
chas	-0.1	-0.3	0.1	0.1	0.0	0.0	0.6	-0.1	0.0	-1.5	-0.1	1.1	-0.1	0.4
nox	0.4	-1.4	0.6	0.0	0.0	0.0	2.4	-0.2	0.6	13.0	0.0	-4.0	0.5	-0.5
rm	-1.3	5.1	-1.9	0.0	0.0	0.5	-4.8	0.3	-1.3	-34.6	-0.5	8.2	-3.1	4.5
age	85.4	-373.9	124.5	0.6	2.4	-4.8	792.4	-44.3	111.8	2402.7	15.9	-702.9	121.1	-97.6
dis	-6.9	32.6	-10.2	-0.1	-0.2	0.3	-44.3	4.4	-9.1	-189.7	-1.1	56.0	-7.5	4.8
rad	46.8	-63.3	35.5	0.0	0.6	-1.3	111.8	-9.1	75.8	1335.8	8.8	-353.3	30.4	-30.6
tax	844	-1236	833	-1.5	13.0	-34.6	2402	-189	1335.8	28404	168.2	-6797	654.7	-726.3
ptratio	5.4	-19.8	5.7	-0.1	0.0	-0.5	15.9	-1.1	8.8	168.2	4.7	-35.1	5.8	-10.1
black	-302.4	373.7	-223.6	1.1	-4.0	8.2	-702.9	56.0	-353.3	-6797	-35.1	8334	-238	280
lstat	28.0	-68.8	29.6	-0.1	0.5	-3.1	121.1	-7.5	30.4	654.7	5.8	-238.7	51.0	-48.4
medv	-30.7	77.3	-30.5	0.4	-0.5	4.5	-97.6	4.8	-30.6	-726.3	-10.1	280.0	-48.4	84.6

Figure 1:

```
##  Min.   : 1.73   Min.   : 5.00
##  1st Qu.: 6.95   1st Qu.:17.02
##  Median :11.36   Median :21.20
##  Mean   :12.65   Mean   :22.53
##  3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :37.97   Max.   :50.00
```

Part c: In order to analyze the effect of the per capita crim rate with other predictors, we need to compute the covariance matrix and correlation matrix of the **Bostone** data set as follows.

```
CovMatrix = cov(Boston)
CovMatrix <- data.frame(CovMatrix)
row.names(CovMatrix) <- c('crim','zn','indus','chas','nox','rm','age',
                          'dis','rad','tax','ptratio','black','lstat','medv')
colnames(CovMatrix) <- c('crim','zn','indus','chas','nox','rm','age',
                         'dis','rad','tax','ptratio','black','lstat','medv')
write.csv(CovMatrix, "Question10_CovMxByCode.csv")
```

```
dimension = dim(Boston);  CorrelationValues <- c(); CorMx <- matrix(0, ncol = 14, nrow =14)
for (i in 1:dimension[2]){
  for (j in 1:dimension[2]){
    CorrelationValues[j] = cor(Boston[,i],Boston[,j])
  }
 CorMx[i,] = CorrelationValues;
}
CorMx <- data.frame(CorMx);
row.names(CorMx) <- c('crim','zn','indus','chas','nox','rm','age',
                      'dis','rad','tax','ptratio','black','lstat','medv')
colnames(CorMx) <- c('crim','zn','indus','chas','nox','rm','age',
                     'dis','rad','tax','ptratio','black','lstat','medv')
write.csv(CorMx, "Question10_CorrelationMx.csv")
```

As it is shown in the bellow plot, the feature of per capita crime rate is more correlated with index of accessibility to radial highways than other features. Their correlation is 0.6. On the other hand, the per capita crime rate is negatively more correlated with dis (weighted mean of distances to five Boston employment centres), black (1000(Bk - 0.63)ˆ2 where Bk is the proportion of blacks by town), medv(median value of

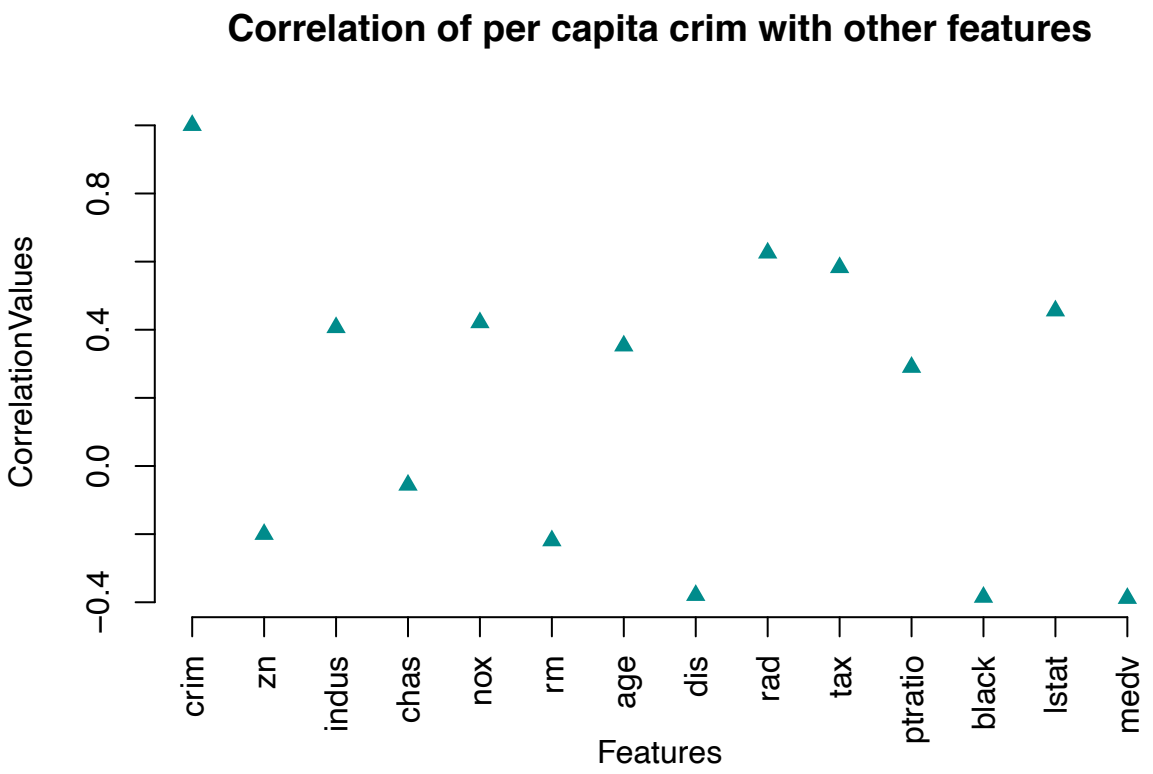
Correlation Matrix

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
crim	1	-0.2	0.407	-0.056	0.421	-0.219	0.353	-0.38	0.626	0.583	0.29	-0.385	0.456	-0.388
zn	-0.2	1	-0.534	-0.043	-0.517	0.312	-0.57	0.664	-0.312	-0.315	-0.392	0.176	-0.413	0.36
indus	0.407	-0.534	1	0.063	0.764	-0.392	0.645	-0.708	0.595	0.721	0.383	-0.357	0.604	-0.484
chas	-0.056	-0.043	0.063	1	0.091	0.091	0.087	-0.099	-0.007	-0.036	-0.122	0.049	-0.054	0.175
nox	0.421	-0.517	0.764	0.091	1	-0.302	0.731	-0.769	0.611	0.668	0.189	-0.38	0.591	-0.427
rm	-0.219	0.312	-0.392	0.091	-0.302	1	-0.24	0.205	-0.21	-0.292	-0.356	0.128	-0.614	0.695
age	0.353	-0.57	0.645	0.087	0.731	-0.24	1	-0.748	0.456	0.506	0.262	-0.274	0.602	-0.377
dis	-0.38	0.664	-0.708	-0.099	-0.769	0.205	-0.748	1	-0.495	-0.534	-0.232	0.292	-0.497	0.25
rad	0.626	-0.312	0.595	-0.007	0.611	-0.21	0.456	-0.495	1	0.91	0.465	-0.444	0.489	-0.382
tax	0.583	-0.315	0.721	-0.036	0.668	-0.292	0.506	-0.534	0.91	1	0.461	-0.442	0.544	-0.469
ptratio	0.29	-0.392	0.383	-0.122	0.189	-0.356	0.262	-0.232	0.465	0.461	1	-0.177	0.374	-0.508
black	-0.385	0.176	-0.357	0.049	-0.38	0.128	-0.274	0.292	-0.444	-0.442	-0.177	1	-0.366	0.333
lstat	0.456	-0.413	0.604	-0.054	0.591	-0.614	0.602	-0.497	0.489	0.544	0.374	-0.366	1	-0.738
medv	-0.388	0.36	-0.484	0.175	-0.427	0.695	-0.377	0.25	-0.382	-0.469	-0.508	0.333	-0.738	1

Figure 2:

owner-occupied homes in $1000s) with correlation value of -0.4. In addition, per capita crime rate is not correlated with chas (Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)) since its correlation value is very close to zero.

```
dimension = dim(Boston); CorrelationValues <- c();
for (j in 1:dimension[2]){
  CorrelationValues[j] = cor(Boston[,1],Boston[,j])
}
Features = c(1:14)
plot(CorrelationValues ~ Features, type = 'p', col="dark cyan", pch=17, axes=F)
axis(1, at=1:14, labels = c('crim','zn','indus','chas','nox','rm','age','dis','rad',
                            'tax','ptratio','black','lstat','medv'), las=2)
axis(2)
title(main = "Correlation of per capita crim with other features")
```




```
graphics.off(); par("mar"); par(mar=c(1,1,1,1));
```

```
## [1] 5.1 4.1 4.1 2.1
```

```
par(mfrow=c(5,3))
dimension = dim(Boston);  CorrelationValues <- c();
for (i in 1:dimension[2]){
  for (j in 1:dimension[2]){
    CorrelationValues[j] = cor(Boston[,i],Boston[,j])
  }
  Features = c(1:14)
  plot(CorrelationValues ~ Features, type = 'p', col="blue", pch=12, axes=F)
  axis(1, at=1:14, labels = c('crim','zn','indus','chas','nox','rm','age','dis','rad',
                              'tax','ptratio','black','lstat','medv'), las=2)
  axis(2)
  title(main = "Features correlation")
}
```

Part d: To figure out which suburbs of Boston appear to have particularly high crim rate, tax rates and the Pupil-teacher ratio, we computed the following distrbution of data for all predictors including these three features as we show in both following plots and table.

In the table the third quartile (75% of the crime rate) is 3.7 but the maximum crime rate is 89 which is particularly different from the third quartile. 124 suburbs has 3.7 crime rate which is equal to the third quartile of per capita crime distribution. Thus, to analyze particularly high crime rates in the following code, we selected 30 as the threshold in which only 8 suburbs will remain.

In addition, based on the above table, the tax rate distribution shows that 75% of the tax rates are about 666 and the maximum value is 711. Thus, we select 666 as the threshold for obtaining how many suburbs have particularly high tax rate. With the chosen threshold, 5 suburbs have very high tax rates.

In a same way, the pupil-teacher ratio by town distribution shows 75% of the data set are 20.2 while the

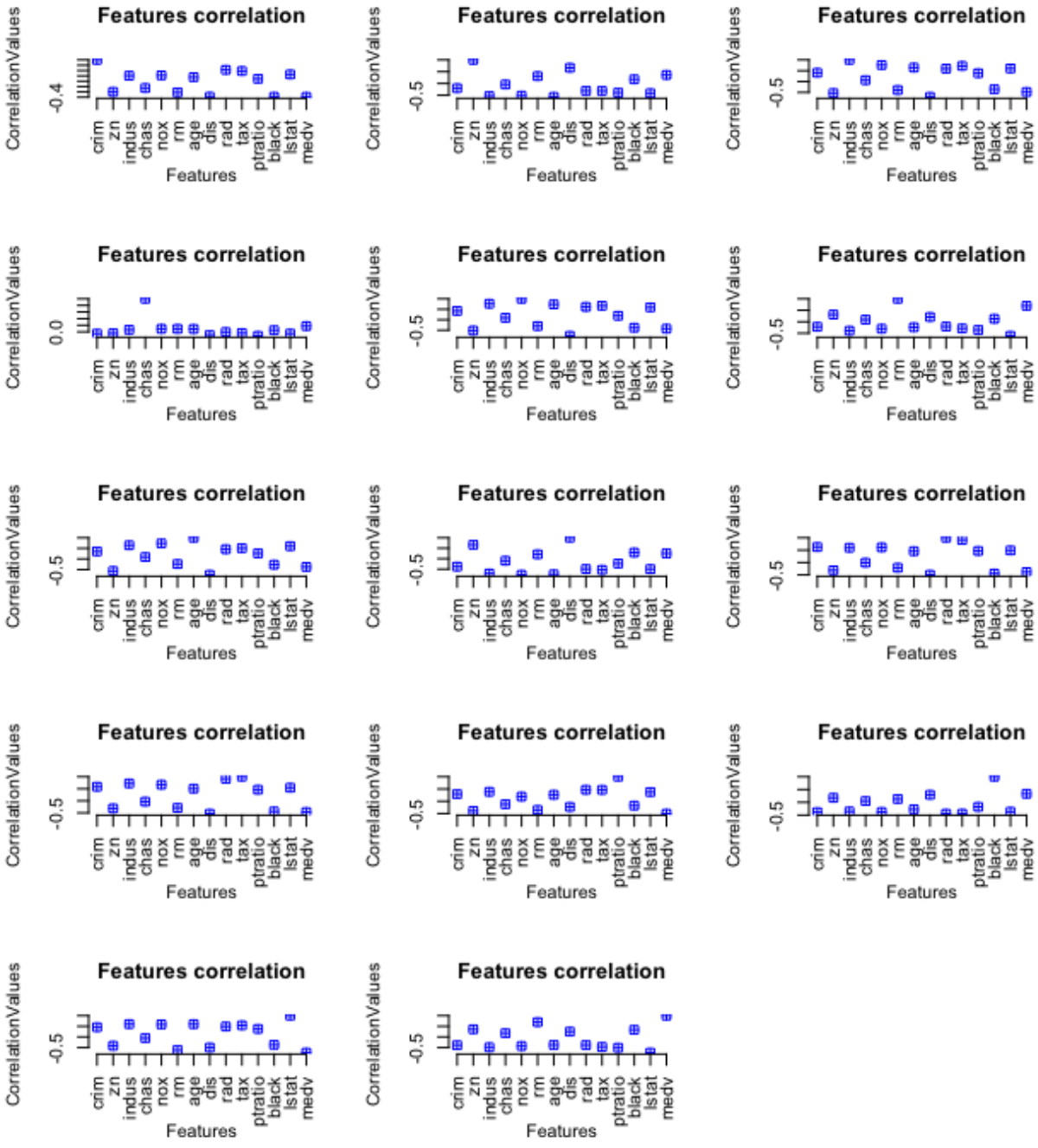

Features correlation
CorrelationValues
Features
crim
zn
indus
chas
nox
rm
age
dis
rad
tax
ptratio
black
lstat
medv

Figure 3:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
Min	0.0	0.0	0.5	0.0	0.4	3.6	2.9	1.1	1.0	187.0	12.6	0.3	1.7	5.0
Q1	0.1	0.0	5.2	0.0	0.4	5.9	45.0	2.1	4.0	279.0	17.4	375.4	7.0	17.0
Mean	3.6	11.4	11.1	0.1	0.6	6.3	68.6	3.8	9.5	408.2	18.5	356.7	12.7	22.5
median	0.3	0.0	9.7	0.0	0.5	6.2	77.5	3.2	5.0	330.0	19.1	391.4	11.4	21.2
Q3	3.7	12.5	18.1	0.0	0.6	6.6	94.1	5.2	24.0	666.0	20.2	396.2	17.0	25.0
Max	89.0	100.0	27.7	1.0	0.9	8.8	100.0	12.1	24.0	711.0	22.0	396.9	38.0	50.0
Std	8.6	544	47.1	0.06	0	0	792	4.4	76	28405	4.687	8335	51	84.59
Mode	0.0	0.0	18.1	0.0	0.5	5.7	100.0	3.5	24.0	666.0	20.2	396.9	8.1	50.0
Skewness	5.21	2.22	0.29	3.4	0.7	0	-0.6	1	1	0.66	-0.8	-2.8	0.9	1.1
Kurtosis	39.8	6.98	1.77	12.5	2.9	5	2.03	3.5	2.1	1.857	2.706	10.14	3.48	4.46

Figure 4:

maximum is 22. Regarding the mode value for ptratio which is same as its third quartile, best threshold for finding how many suburbs have particularly high ptratio can be 21.2. With threshold of 21, 18 suburbs will remain while by selecting 21.2 threshold only 2 suburbs will remain showing very high ptratios.

```
#----------------------------------------high crim
HighCrim <- data.frame(t(Boston[,1]))
NumofHighCrim=0
for (i in 1:length(HighCrim)){
  if (HighCrim[i] > 30){
    NumofHighCrim = NumofHighCrim + 1;
  }
}
print(NumofHighCrim)
```

```
## [1] 8
```

```
#----------------------------------------high tax
HighTax <- data.frame(t(Boston[,10]))
NumofHighTax=0
for (i in 1:length(HighTax)){
  if (HighTax[i] > 666){
    NumofHighTax = NumofHighTax + 1;
  }
}
print(NumofHighTax)
```

```
## [1] 5
```

```
#----------------------------------------high ptratio
Highptratio <- data.frame(t(Boston[,11]))
NumofHighptratio=0
for (i in 1:length(Highptratio)){
  if (Highptratio[i] > 21.2){
    NumofHighptratio = NumofHighptratio + 1;
  }
}
print(NumofHighptratio)
```

```
## [1] 2
```

Also all the explained information above can be seen in the following violin plot which shows a combo diagram

of box plot and kernel distribution of crim, tax ratio and ptratio in one plot. Also, before making these meaningful plots, we normalize the **Boston** data set because based on the above table, dispersion of each predictors for different observations is high.

```
library(moments)
library(vioplot)
```

```
## Loading required package: sm
```

```
## Package 'sm', version 2.2-5.4: type help(sm) for summary information
```

```
##
## Attaching package: 'sm'
```

```
## The following object is masked from 'package:MASS':
##
##     muscle
```

```
## Loading required package: zoo
```

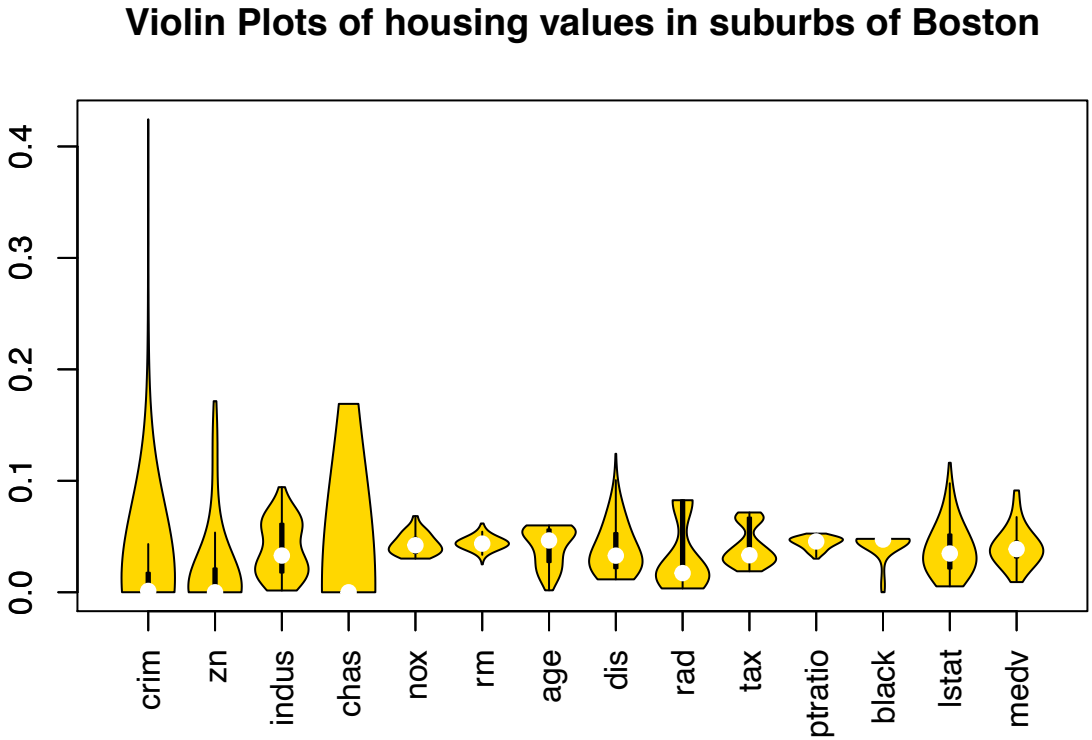
```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

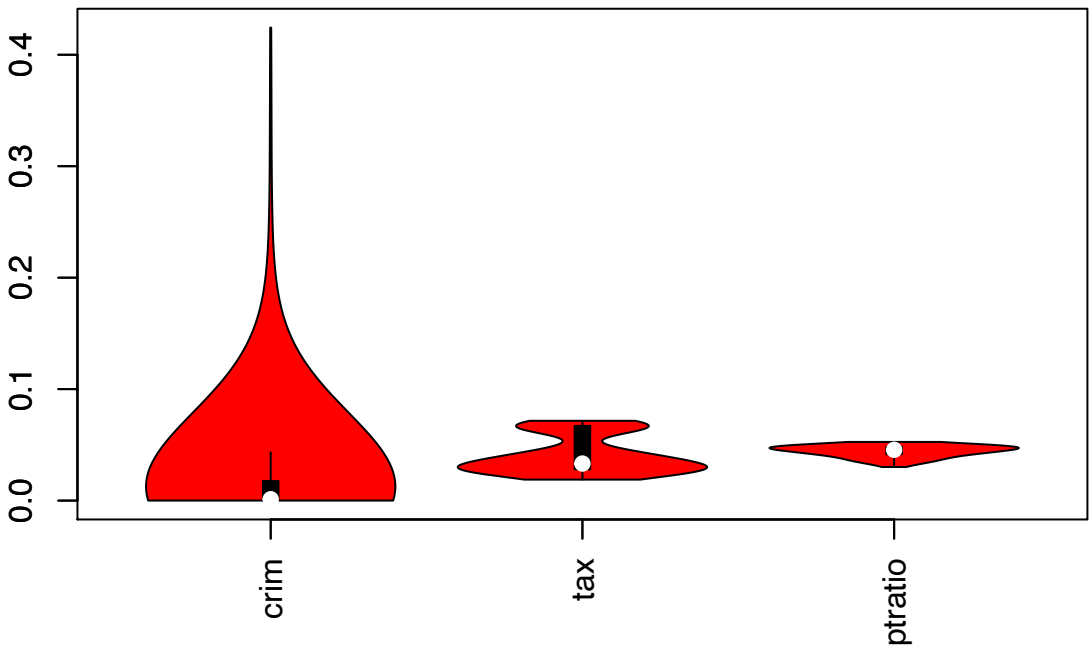
```
library(ggplot2)

skw <- c(); kurt <- c()

myMatrixNormalizer <- function(A){
  l = dim(A)
  for (i in 1:l[2]){
    A[,i] = A[,i]/(norm(A[,i],type="2"));
    skw[i] = skewness(A[,i])
    kurt[i] = kurtosis(A[,i])
  }
  print(skw)  # Skewness of Bostone data
  print(kurt) # Kurtosis of Bostone data
  return(A)
}
Boston = myMatrixNormalizer(Boston)
```

```
##  [1]  5.2076524  2.2190631  0.2941463  3.3957993  0.7271442  0.4024147
##  [7] -0.5971856  1.0087788  1.0018335  0.6679683 -0.7999445 -2.8817983
## [13]  0.9037707  1.1048108
##  [1] 39.752786  6.979949  1.766782 12.531453  2.924136  4.861027  2.029986
##  [8]  3.471299  2.129479  1.857015  2.705884 10.143769  3.476545  4.468629
```

```
vioplot(Boston[,1], Boston[,2],Boston[,3],Boston[,4],Boston[,5],Boston[,6], Boston[,7],
        Boston[,8], Boston[,9], Boston[,10],Boston[,11], Boston[,12],Boston[,13],Boston[,14],
        names=F, col="gold")
axis(1, at=1:14, labels = c('crim','zn','indus','chas','nox','rm','age',
                            'dis','rad','tax','ptratio','black','lstat','medv')
     , las=2)
axis(2)
title("Violin Plots of housing values in suburbs of Boston")
```




```
vioplot(Boston[,1], Boston[,10],Boston[,11],names=F, col="red")
axis(1, at=1:3, labels = c('crim','tax','ptratio'), las=2); axis(2)
title("Violin Plots of crim, tax, ptratio")
```

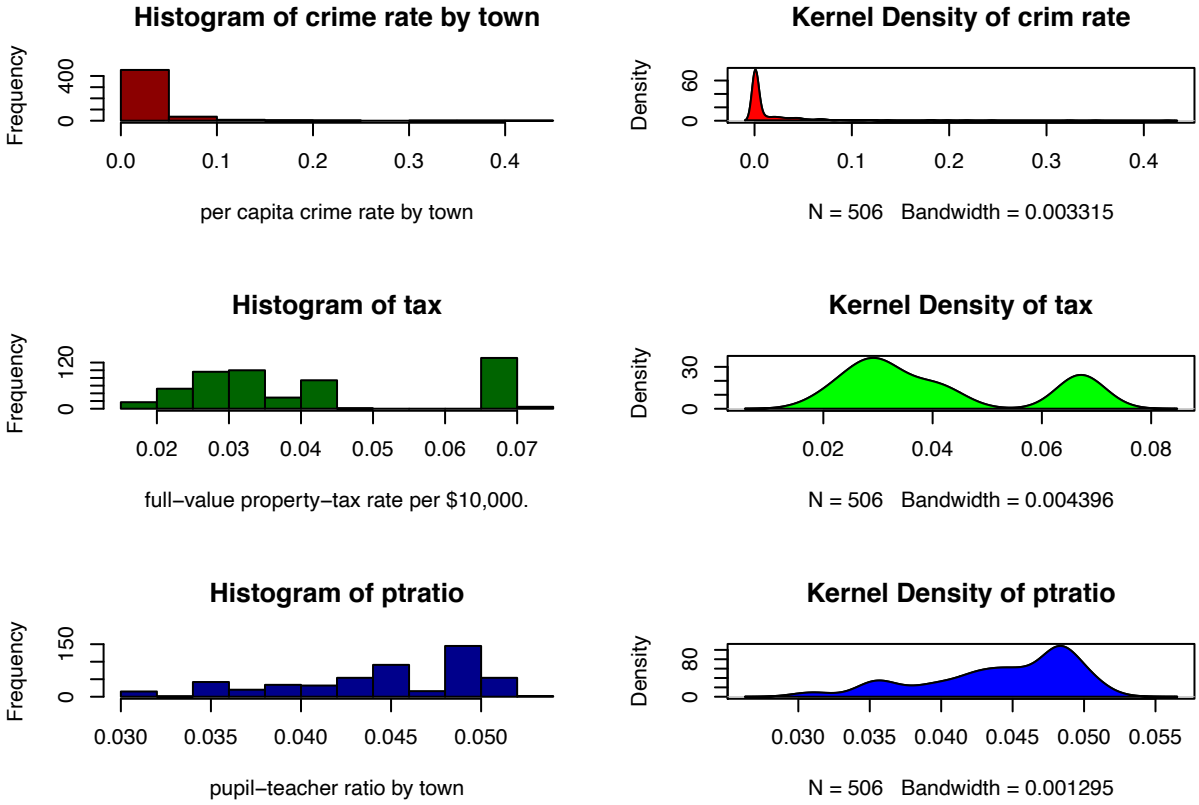


For a better understanding the distribution of crim rate, tax rate and ptratio, we show the histogram and the distribution of these three predictors in the following plots:

```
Boston = myMatrixNormalizer(Boston)
```

```
##  [1]  5.2076524  2.2190631  0.2941463  3.3957993  0.7271442  0.4024147
##  [7] -0.5971856  1.0087788  1.0018335  0.6679683 -0.7999445 -2.8817983
## [13]  0.9037707  1.1048108
##  [1] 39.752786  6.979949  1.766782 12.531453  2.924136  4.861027  2.029986
##  [8]  3.471299  2.129479  1.857015  2.705884 10.143769  3.476545  4.468629
```

```
par(mfrow=c(3,2))
#Histogram
hist(Boston[,1], breaks=12, col="dark red",
     xlab="per capita crime rate by town", main="Histogram of crime rate by town")
d <- density(Boston[,1]); plot(d, "Kernel Density of crim rate");
polygon(d, col="red", border="black")

hist(Boston[,10], breaks=12, col="dark green",
     xlab="full-value property-tax rate per $10,000.", main="Histogram of tax")
d <- density(Boston[,10]); plot(d, "Kernel Density of tax");
polygon(d, col="green", border="black")

hist(Boston[,11], breaks=12, col="dark blue",
     xlab="pupil-teacher ratio by town", main="Histogram of ptratio")
d <- density(Boston[,11]); plot(d, "Kernel Density of ptratio");
polygon(d, col="blue", border="black")
```



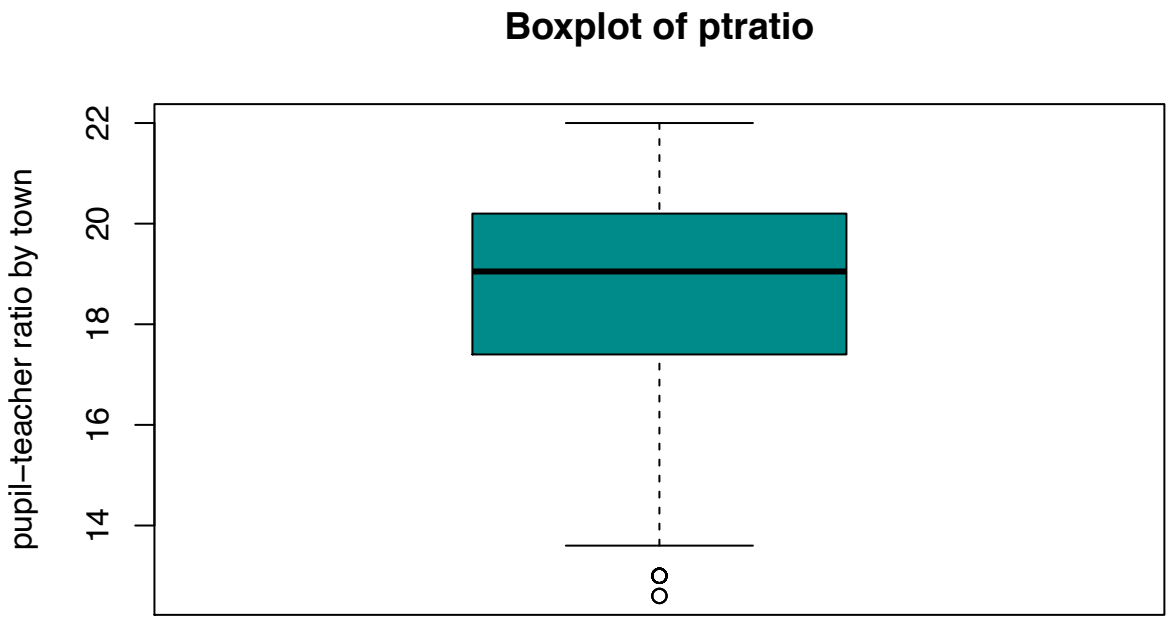
Part e: In the following code we cound how many observation in the data set for the *chas* feature is equal to one. These observations show which suburbs bound the Charles river.

```
CharlesRiver <- data.frame(t(Boston[,4]))
NumofBoundCharlesRiver=0
for (i in 1:length(CharlesRiver)){
  if (CharlesRiver[i] == 1){
    NumofBoundCharlesRiver = NumofBoundCharlesRiver + 1;
  }
}
print(NumofBoundCharlesRiver)
```

```
## [1] 0
```

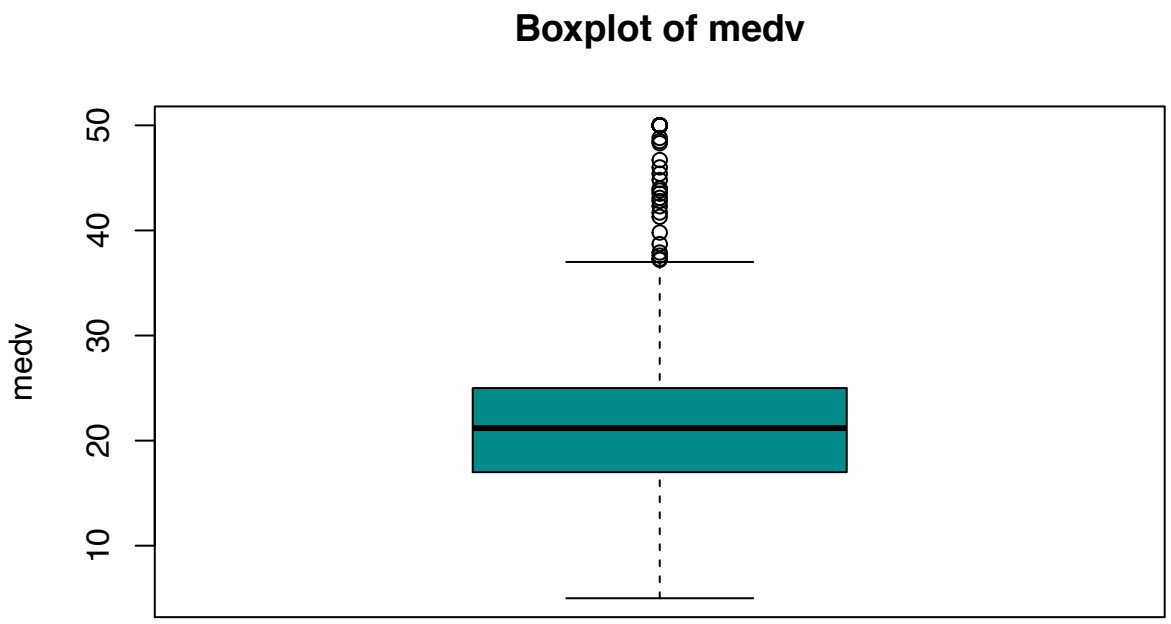
Part f: We showed the distribution of the pupil-teacher-ratio in the part *d* in both plot and table. In the following piece of code, we plot the box plot of the ptratio which shows minimum, first, second, third quartiles, and maximum of the pupil-teacher-ratio. The median for ptratio is equal to 19.

```
rm(Boston)
library(MASS)

boxplot(Boston[,11], main="Boxplot of ptratio", col= "dark cyan",
        ylab="pupil-teacher ratio by town")
```

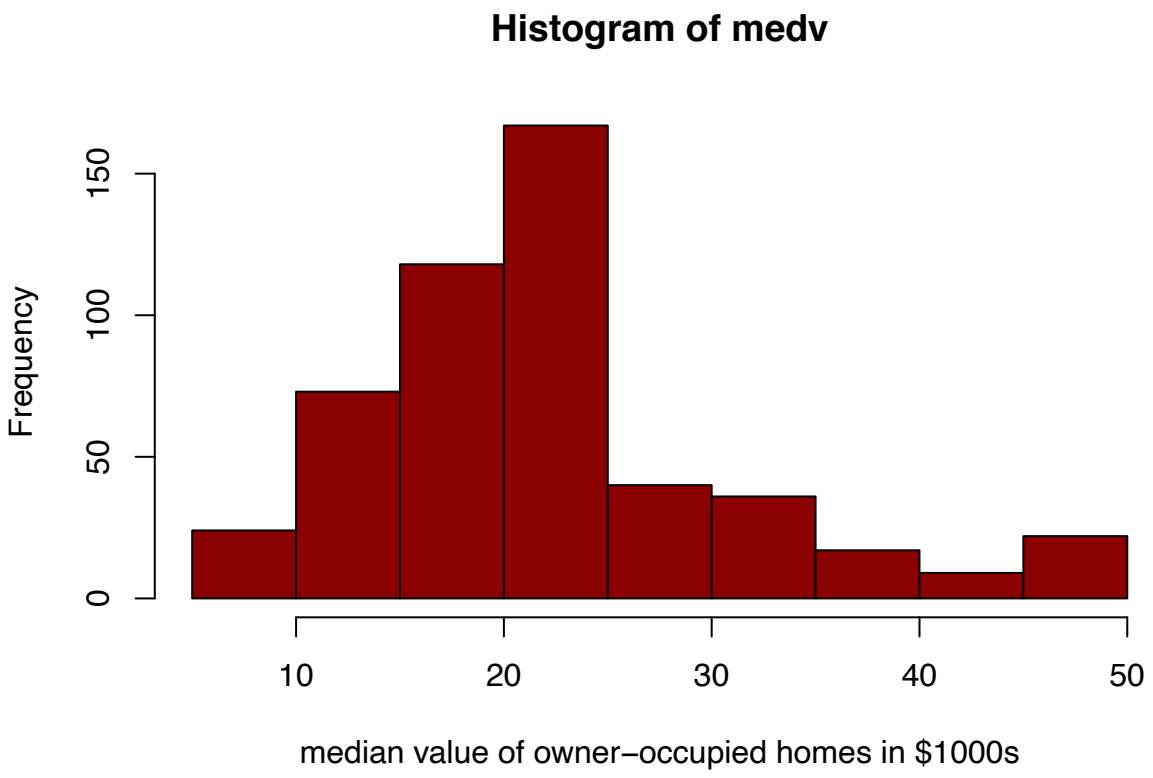


Part g: The minimum of medv is equal to 5 as we showed in *part a* tabels and violin plot. The max of crim rate is 89, while in the suburb with minimum medv, crime rate is 68 which is substantial, also as we expect proportion of residential land zoned for lots over 25,000 sq.ft is equal to zero. In addition proportion of non-retail business acres per town is in the third quartile of its disrtibution. in the minimum medv suburb, charles river is not bounded. Also, the portion of blacks by town is high and tax rate is high (less than maximum but in third quartile of its distribution). The nitrogen oxides concentration in this suburb with minimum medv is relatively high in its third quartile. pupil-teacher and lower status of the population ratio by town is also high and close to its maximum.

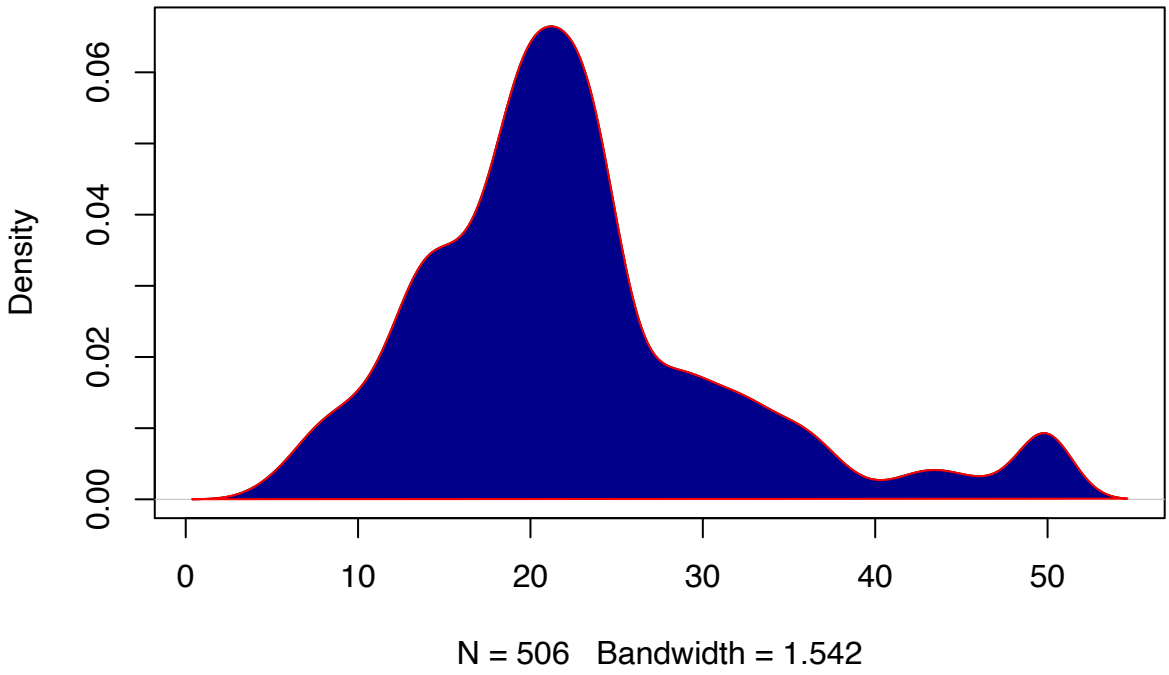
```
# min med = 5
#par(mfrow=c(3,1))
boxplot(Boston[,14], main="Boxplot of medv", col= "dark cyan",
        ylab="medv")
```




```
hist(Boston[,14], breaks=12, col="dark red",xlab="median value of owner-occupied homes in $1000s", main=
```




```
d <- density(Boston[,14]); plot(d, "Kernel Density of medv"); polygon(d, col="dark blue", border="red")
```




```
#values of other predictors
print(min(Boston[,14]))
```

```
## [1] 5
```

```
print(Boston[406,])
```

```
##        crim zn indus chas   nox    rm age    dis rad tax ptratio  black
## 406 67.9208  0  18.1    0 0.693 5.683 100 1.4254  24 666    20.2 384.97
##     lstat medv
## 406 22.98    5
```

Part h: In the following code we compute the number of suburbs which have average more than seven rooms per dwellings and then those which have average more than eight rooms per dwellings.

```
AverageNumberRooms  <- data.frame(t(Boston[,6]))
NumofrmMore7=0;NumofrmMore8=0;
for (i in 1:length(AverageNumberRooms)){
  if (AverageNumberRooms[i] > 7){
    NumofrmMore7 = NumofrmMore7 + 1;
  }
  if (AverageNumberRooms[i] > 8){
    NumofrmMore8 = NumofrmMore8 + 1;
  }
}
print(NumofrmMore7)
```

```
## [1] 64
```

```
print(NumofrmMore8)
```

```
## [1] 13
```