Applied Data Mining: Homework 1

Anshul Doshi

Anshul Doshi (ad3222)
STAT UN3106 Section 001
Assignment: 1

Problem 1: James 2.8

a.) Load College.csv

```
college <- read.csv("College.csv", header = TRUE)
```

b.) Viewing data using the fix() function

```
rownames(college) = college[,1]
fix(college)

#Need to eliminate the row.names column
college = college[,-1]
fix(college)
```

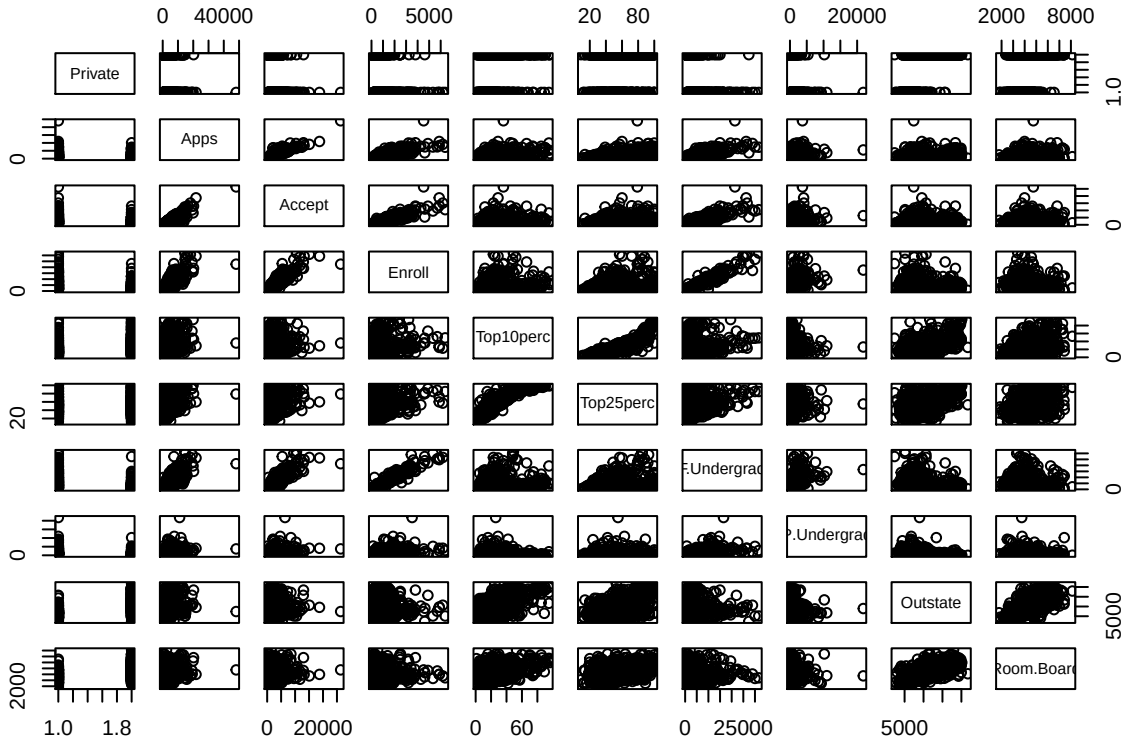
c.) Different plots of the college data

```
# i. summary
summary(college)
```

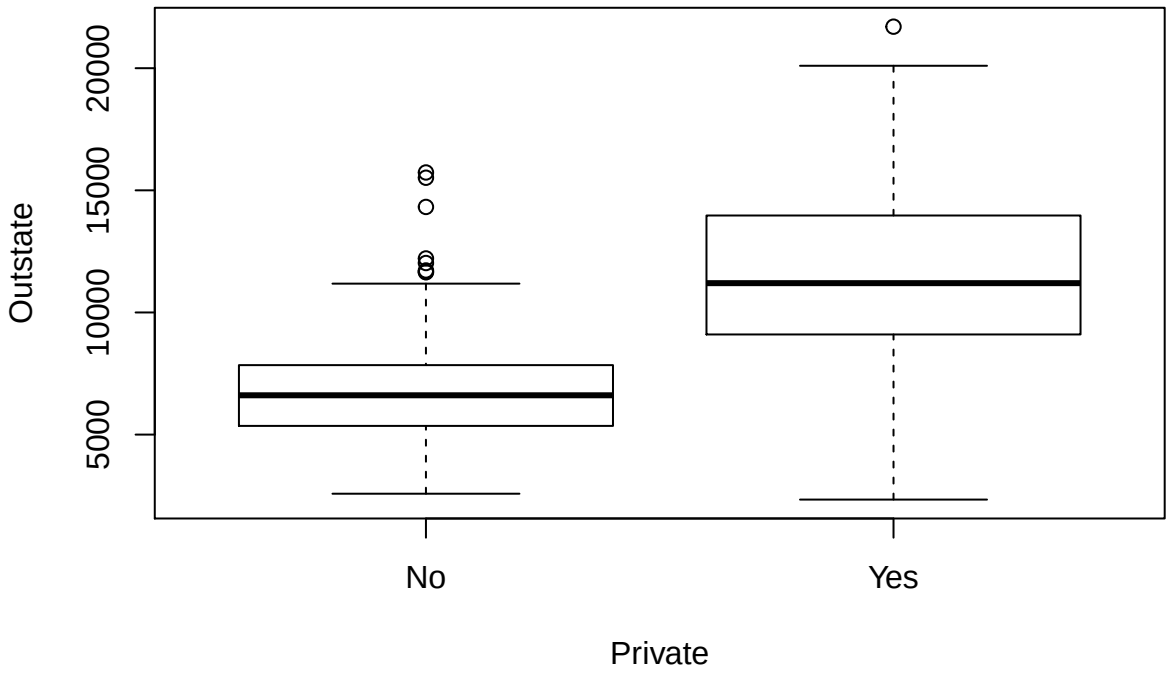
```
##  Private        Apps           Accept          Enroll       Top10perc
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
##            Median : 1558   Median : 1110   Median : 434   Median :23.00
##            Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
##            3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
##            Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00
##    Top25perc      F.Undergrad     P.Undergrad         Outstate
##  Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
##  1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
##  Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
##  Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
##  3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
##  Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
##    Room.Board       Books           Personal         PhD
##  Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00
##  1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
##  Median :4200   Median : 500.0   Median :1200   Median : 75.00
##  Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66
##  3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
##  Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
##     Terminal       S.F.Ratio      perc.alumni        Expend
##  Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186
##  1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751
##  Median : 82.0   Median :13.60   Median :21.00   Median : 8377
##  Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660
```

```
##  3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830
##  Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233
##    Grad.Rate
##  Min.   : 10.00
##  1st Qu.: 53.00
##  Median : 65.00
##  Mean   : 65.46
##  3rd Qu.: 78.00
##  Max.   :118.00
```

```r
# ii. pairs
pairs(college[,1:10])
```



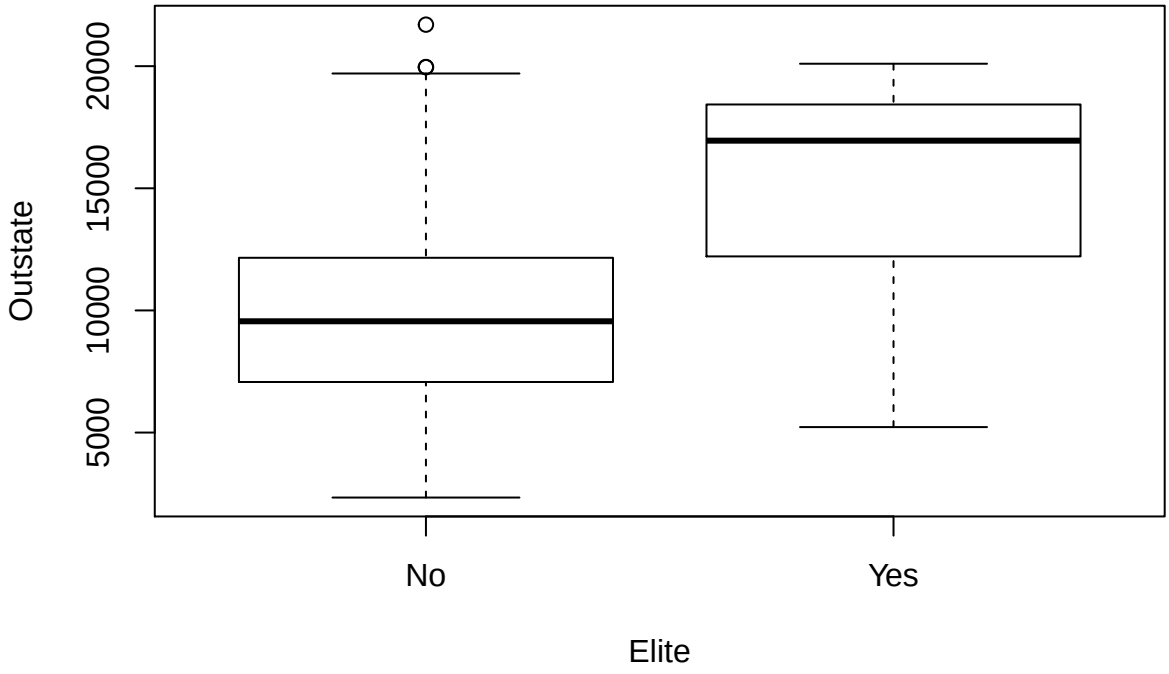

```r
# iii. plot of Outstate vs. Private
plot(college$Private, college$Outstate, xlab = "Private", ylab = "Outstate")
```



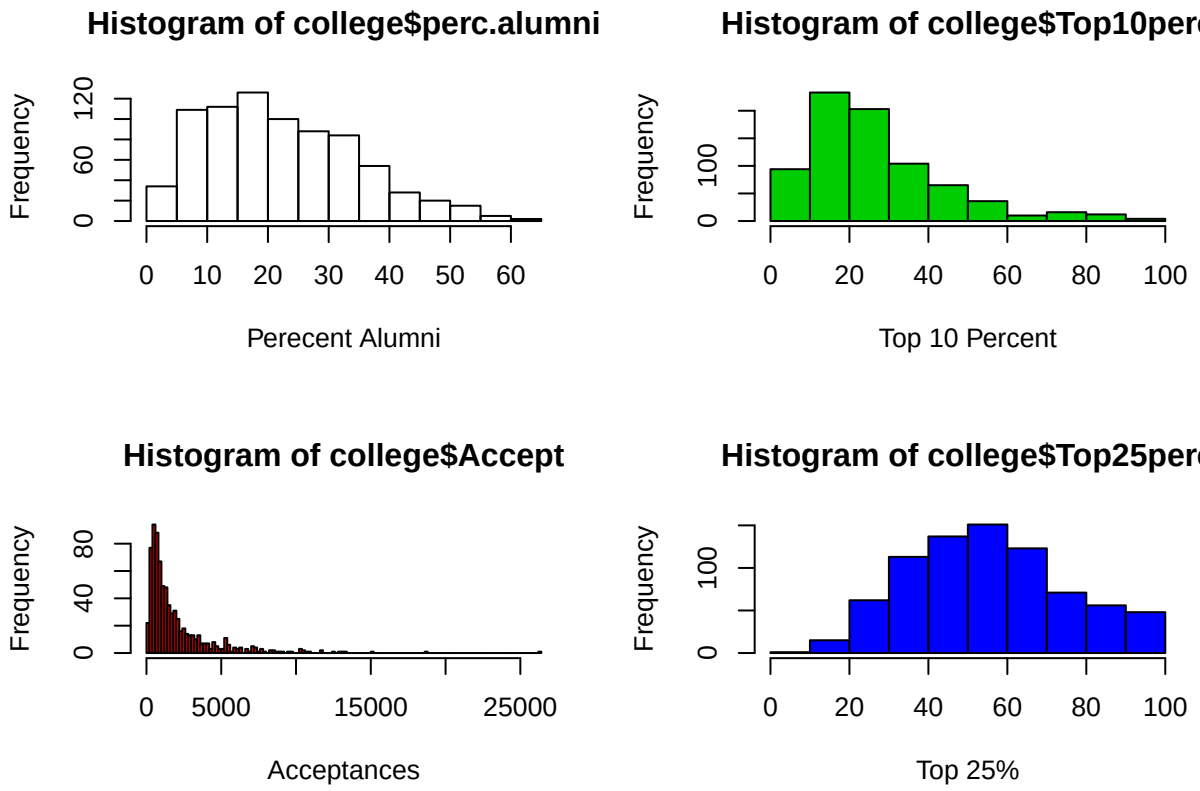

```
# iv. plot of Outstate vs. Elite
Elite=rep("No",nrow(college))
Elite[college$Top10perc >50]="Yes"
Elite=as.factor(Elite)
college=data.frame(college ,Elite)
summary(college$Elite)
```

```
##  No Yes
## 699  78
```

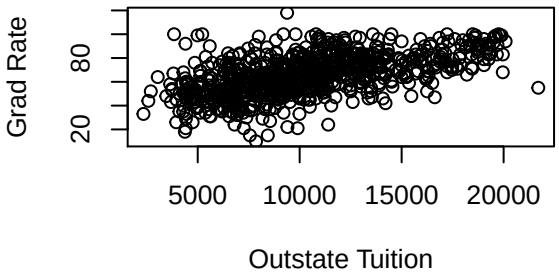
```
plot(college$Elite, college$Outstate, xlab = "Elite", ylab = "Outstate")
```

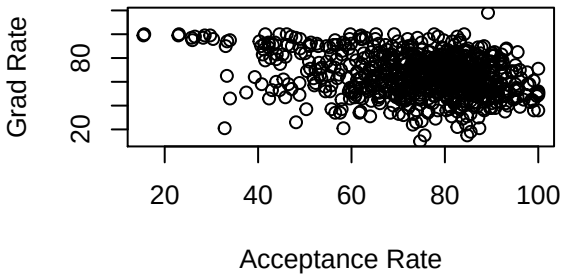


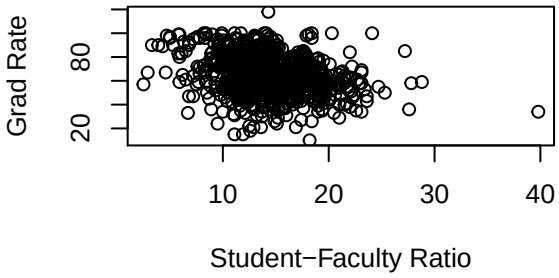
```r
# v. histograms
par(mfrow=c(2,2))
hist(college$perc.alumni, xlab = "Perecent Alumni")
hist(college$Top10perc, col=3, breaks = 10, xlab = "Top 10 Percent")
hist(college$Accept, col=2,breaks=100, xlab="Acceptances")
hist(college$Top25perc, col=4, xlab = "Top 25%")
```

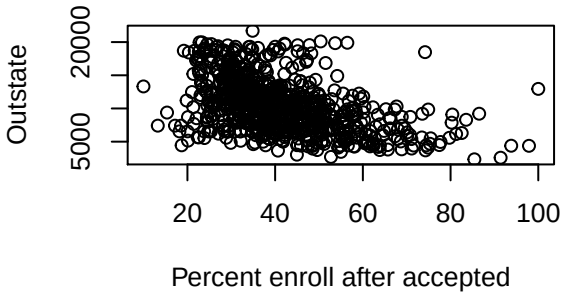



```r
# vi. Continue exploring the data
#Colleges with higher tuition seem to have higher graduation rate - grad rate higher than
#100 seems wrong
plot(college$Outstate, college$Grad.Rate,
     xlab = "Outstate Tuition", ylab = "Grad Rate")

#Shows colleges with low acceptance rates
#tend to have high gradutation rates
plot((college$Accept / college$Apps) * 100, college$Grad.Rate,
     xlab = "Acceptance Rate", ylab = "Grad Rate")

#Unusually, not a large correlation is shown between
#student faculty ratio and graduation rate
plot(college$S.F.Ratio, college$Grad.Rate,
     xlab = "Student-Faculty Ratio", ylab = "Grad Rate")

#The lower the tuition is, the more people seem to enroll after getting into a school
plot((college$Enroll / college$Accept)*100, college$Outstate,
     xlab = "Percent enroll after accepted", ylab = "Outstate")
```









Problem 2: James 2.9

```
#Omitting missing values
auto <- read.csv("Auto.csv", header = TRUE, na.strings = "?")
auto = na.omit(auto)
summary(auto)
```

```
##       mpg          cylinders      displacement     horsepower
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0
##
##      weight      acceleration        year           origin
##  Min.   :1613   Min.   : 8.00   Min.   :70.00   Min.   :1.000
##  1st Qu.:2225   1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000
##  Median :2804   Median :15.50   Median :76.00   Median :1.000
##  Mean   :2978   Mean   :15.54   Mean   :75.98   Mean   :1.577
##  3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000
##  Max.   :5140   Max.   :24.80   Max.   :82.00   Max.   :3.000
##
##                  name
##  amc matador       :  5
##  ford pinto        :  5
##  toyota corolla    :  5
##  amc gremlin       :  4
##  amc hornet        :  4
##  chevrolet chevette:  4
```

```
##  (Other)           :365
```

a.) Which of the predictors are quantitative, and which are qualiitative?

- Quantatative: mpg, cyclinders, displacements, horsepower, weight, acceleration, year
- Qualitative: origin, name

b.) What is therangeof each quantitative predictor? You can answer this using therange()function

```
sapply(auto[, 1:7], range)
```

```
##       mpg cylinders displacement horsepower weight acceleration year
## [1,]  9.0         3           68         46   1613          8.0   70
## [2,] 46.6         8          455        230   5140         24.8   82
```

c.) What is the mean and standard deviation of each quantitative predictor

```
#mean
sapply(auto[, 1:7], mean)
```

```
##          mpg    cylinders displacement   horsepower       weight
##    23.445918     5.471939   194.411990   104.469388  2977.584184
## acceleration         year
##    15.541327    75.979592
```

```
#stdev
sapply(auto[, 1:7], sd)
```

```
##          mpg    cylinders displacement   horsepower       weight
##     7.805007     1.705783   104.644004    38.491160   849.402560
## acceleration         year
##     2.758864     3.683737
```

d.) subsetting auto data

```
newAuto <- auto[-(10:85),1:7]

#range
sapply(newAuto[, 1:7], range)
```

```
##       mpg cylinders displacement horsepower weight acceleration year
## [1,] 11.0         3           68         46   1649          8.5   70
## [2,] 46.6         8          455        230   4997         24.8   82
```

```
#mean
sapply(newAuto[, 1:7], mean)
```

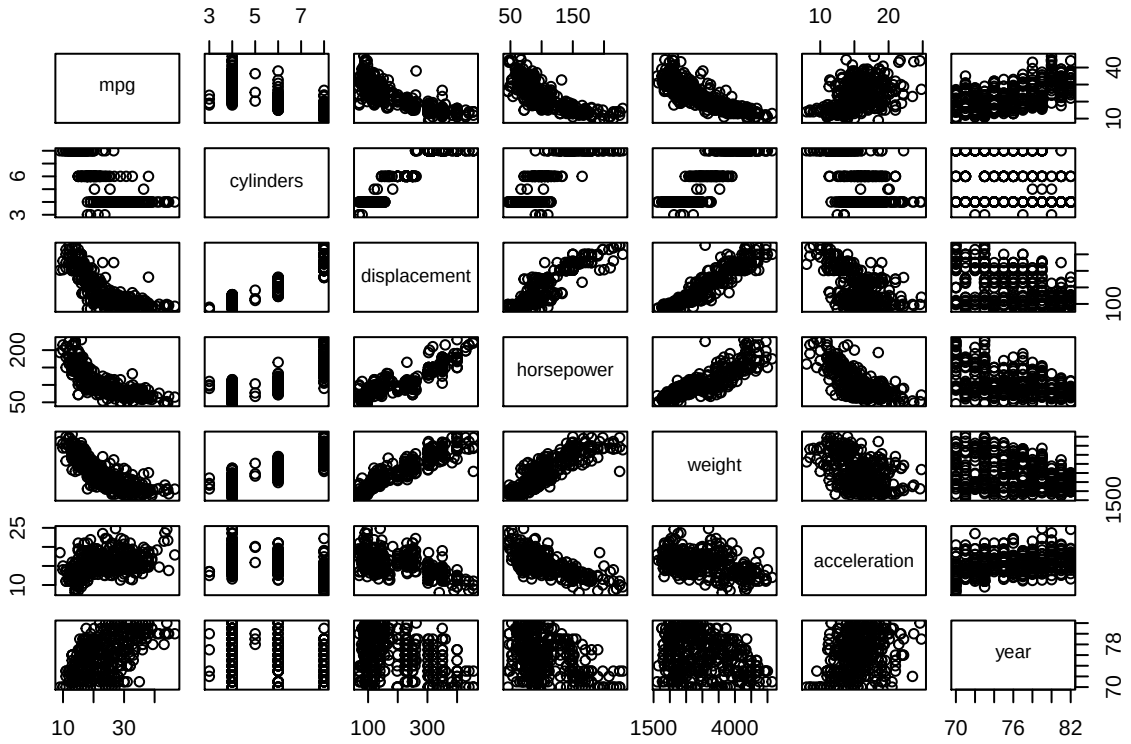
```
##          mpg    cylinders displacement   horsepower       weight
##    24.404430     5.373418   187.240506   100.721519  2935.971519
## acceleration         year
##    15.726899    77.145570
```

```
#stdev
sapply(newAuto[, 1:7], sd)
```

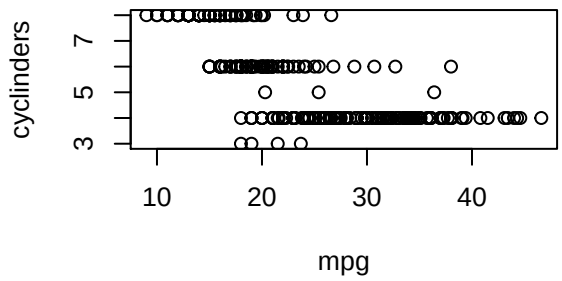
```
##          mpg    cylinders displacement   horsepower       weight
##     7.867283     1.654179    99.678367    35.708853   811.300208
## acceleration         year
##     2.693721     3.106217
```

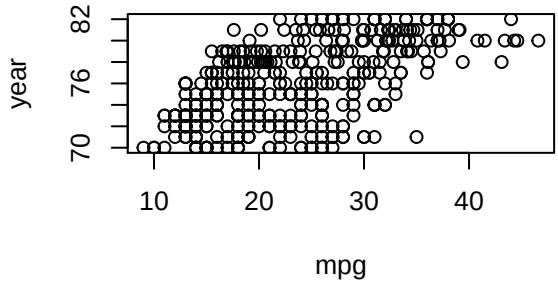
e.) Create some plots to show relationships

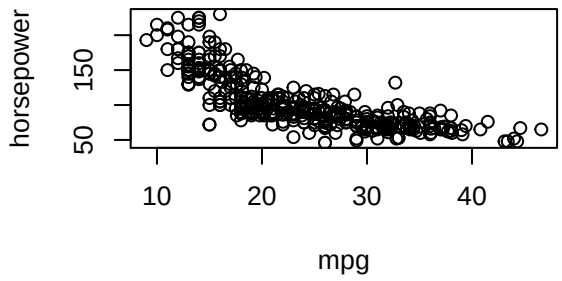
```
pairs(auto[,1:7])
```

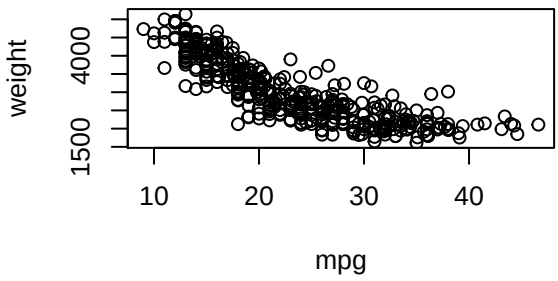



```
par(mfrow=c(2,2))
plot(auto$mpg, auto$cylinders, xlab = "mpg", ylab = "cyclinders")
plot(auto$mpg, auto$year, xlab = "mpg", ylab = "year")
plot(auto$mpg, auto$horsepower, xlab = "mpg", ylab = "horsepower")
plot(auto$mpg, auto$weight, xlab = "mpg", ylab = "weight")
```









The four plots above show - The more cyclinders, the less mpg - less mpg over time - More horsepower correlates to lower mpg - More weight correlates to lower mpg

f.) Are any of the variables useful in predicitng mpg

As the plots above show, most of the predicitors have some correlation with mpg meaning that they will indeed be useful in helping us predict mpg. we should be wary of overfitting due to too few observations as may be the case with mpg and name of car.

Problem 3: James 2.10

a.) Load the Boston data set

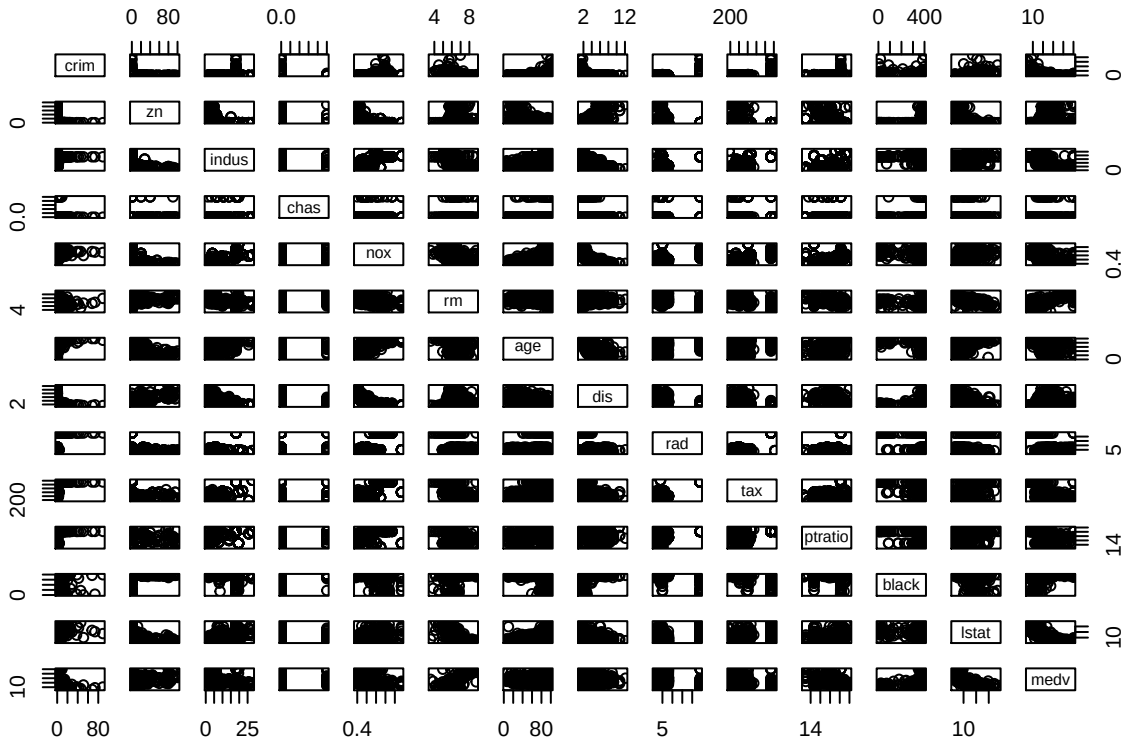
```
library(MASS)
?Boston
str(Boston)
```

```
## '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 0.524 ...
##  $ 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 ...
```

- Rows represent observations
- Columns represent features

b.) Make some pairwise scatterplots

```
pairs(Boston)
```

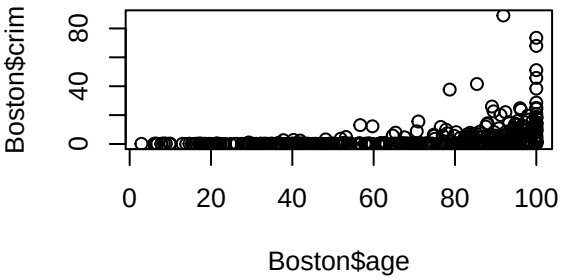


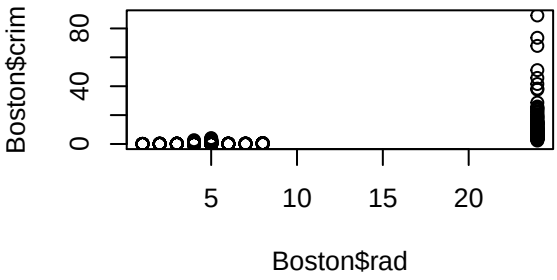
The scatter plots show that there is some correlation between predicitors. For example:

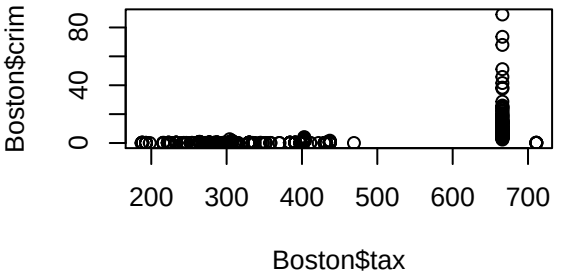
- Crime may be correlated with: age, rad, tax, ptration
- zn may be correlated with: indux, nox, age
- indus with age and dis etc. . .

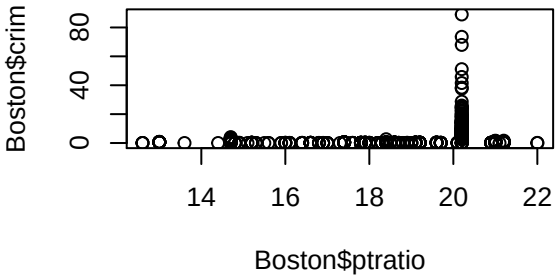
c.) Predictors associated with crime rate

```
#using observation from pairwise plot
par(mfrow=c(2,2))
plot(Boston$age, Boston$crim)
plot(Boston$rad, Boston$crim)
plot(Boston$tax, Boston$crim)
plot(Boston$ptratio, Boston$crim)
```







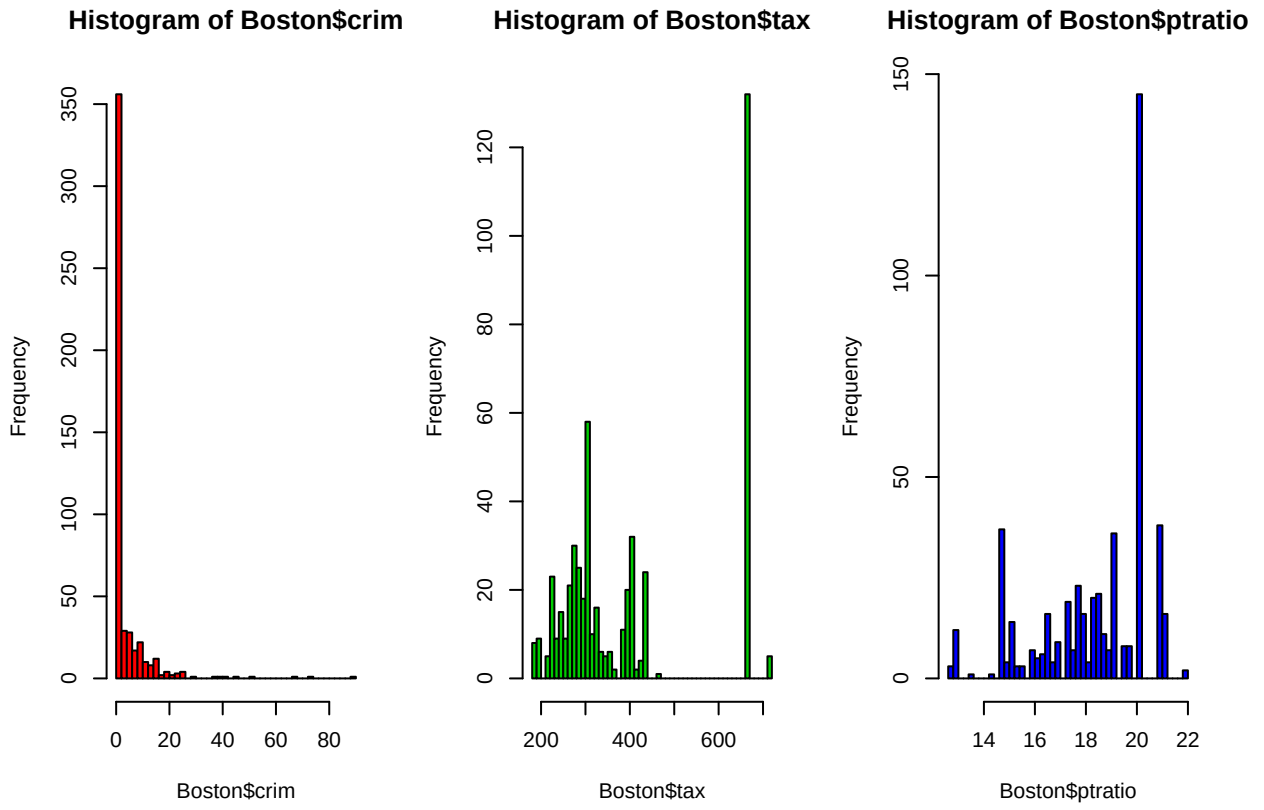


The above graphs indicates:

- Older homes are higher targeted
- High index to radial highways leads to higher crime
- Similarly, higher property tax leads to higher crime
- Higher pupil-teacher means higher crime

d.) Ranges of each predictor

```
par(mfrow=c(1,3))

hist(Boston$crim, col=2, breaks=50)
hist(Boston$tax, col=3, breaks=50)
hist(Boston$ptratio, col=4, breaks=50)
```




```
sapply(Boston[,1:14], range)
```

```
##          crim  zn indus chas   nox    rm   age     dis rad tax ptratio
## [1,]  0.00632   0  0.46    0 0.385 3.561   2.9  1.1296   1 187    12.6
## [2,] 88.97620 100 27.74    1 0.871 8.780 100.0 12.1265  24 711    22.0
##       black lstat medv
## [1,]   0.32  1.73    5
## [2,] 396.90 37.97   50
```

- Crime: Most cities have a small crime rate but the long tail indicates that a few places in Boston have a crime rate from 20 reaching out all the way to 88
- Tax: There is a divide between places with low tax rates and then those with very high taxes peaking at about 680.
- ptratio: slightly skewed toward higher ratios

e.) How many suburbs bound the Charles River?

```
sum(Boston$chas==1)
```

```
## [1] 35
```

f.) Median pupil-teacher ration

```
median(Boston$ptratio)
```

```
## [1] 19.05
```

g.) Which suburb has lowest median value of owner occupied homes?

```
lowMedv <- Boston[Boston$medv == min(Boston$medv),]
head(lowMedv)
```

```
##        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    20.2 396.90
## 406 67.9208  0  18.1    0 0.693 5.683 100 1.4254  24 666    20.2 384.97
##     lstat medv
## 399 30.59    5
## 406 22.98    5
```

```
sapply(Boston[,1:14], quantile)
```

```
##           crim    zn indus chas   nox     rm     age       dis rad tax
## 0%    0.006320   0.0  0.46    0 0.385 3.5610   2.900  1.129600   1 187
## 25%   0.082045   0.0  5.19    0 0.449 5.8855  45.025  2.100175   4 279
## 50%   0.256510   0.0  9.69    0 0.538 6.2085  77.500  3.207450   5 330
## 75%   3.677083  12.5 18.10    0 0.624 6.6235  94.075  5.188425  24 666
## 100% 88.976200 100.0 27.74    1 0.871 8.7800 100.000 12.126500  24 711
##      ptratio    black  lstat   medv
## 0%     12.60   0.3200  1.730  5.000
## 25%    17.40 375.3775  6.950 17.025
## 50%    19.05 391.4400 11.360 21.200
## 75%    20.20 396.2250 16.955 25.000
## 100%   22.00 396.9000 37.970 50.000
```

Comparing these predictors to ranges above:

- age and rad are at max
- crime, indux, nox, tax, ptration are high enough (75th percentile)
- zn, rm, dis are low

h.) Rooms per dwelling

```
sevRm <- Boston[Boston$rm >= 7,]
eightRm <- Boston[Boston$rm >= 8,]
nrow(sevRm[,])
```

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

```
nrow(eightRm[,])
```

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

```
summary(sevRm)
```

```
##       crim                zn             indus             chas        
##  Min.   : 0.00906   Min.   : 0.00   Min.   : 0.460   Min.   :0.000  
##  1st Qu.: 0.04502   1st Qu.: 0.00   1st Qu.: 2.460   1st Qu.:0.000  
##  Median : 0.09786   Median :20.00   Median : 3.970   Median :0.000  
##  Mean   : 0.97911   Mean   :28.17   Mean   : 5.776   Mean   :0.125  
##  3rd Qu.: 0.54289   3rd Qu.:45.00   3rd Qu.: 6.200   3rd Qu.:0.000  
##  Max.   :19.60910   Max.   :95.00   Max.   :19.580   Max.   :1.000  
##       nox               rm              age              dis        
##  Min.   :0.3940   Min.   :7.007   Min.   :  8.40   Min.   :1.202  
##  1st Qu.:0.4303   1st Qu.:7.183   1st Qu.: 36.00   1st Qu.:2.445  
##  Median :0.4880   Median :7.414   Median : 63.80   Median :3.495  
##  Mean   :0.5045   Mean   :7.570   Mean   : 60.64   Mean   :4.200  
##  3rd Qu.:0.5825   3rd Qu.:7.859   3rd Qu.: 85.03   3rd Qu.:5.463  
##  Max.   :0.7180   Max.   :8.780   Max.   :100.00   Max.   :9.223  
##       rad              tax           ptratio          black       
##  Min.   : 1.000   Min.   :193.0   Min.   :12.60   Min.   :354.3  
```

```
##  1st Qu.: 3.000   1st Qu.:244.8   1st Qu.:14.70   1st Qu.:384.9
##  Median : 5.000   Median :273.0   Median :17.40   Median :390.7
##  Mean   : 5.984   Mean   :312.2   Mean   :16.26   Mean   :388.3
##  3rd Qu.: 7.000   3rd Qu.:329.0   3rd Qu.:17.93   3rd Qu.:395.3
##  Max.   :24.000   Max.   :666.0   Max.   :20.20   Max.   :396.9
##      lstat            medv
##  Min.   : 1.730   Min.   :15.00
##  1st Qu.: 3.555   1st Qu.:32.98
##  Median : 4.775   Median :36.45
##  Mean   : 5.474   Mean   :38.40
##  3rd Qu.: 6.590   3rd Qu.:46.17
##  Max.   :16.740   Max.   :50.00
```

```
summary(eightRm)
```

```
##       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
```

The summary indicates that for at least 8 rooms, crime rates are much higher, lstat is lower, medv and age slightly higher, and lstat is lower.

Question 4:

The following code loads the eigenfaces data and performs a set of simple loading and plotting functions.

Setup

Make sure R is in the proper working directory. Note that this will be a different path for every machine

```
setwd("~/Desktop/Spring2017/Applied Data Mining/Homeworks/Homework1")
```

First include the relevant libraries. Note that a loading error might mean that you have to install the package into your R distribution. From the command line, type {`install.packages("pixmap")`}

```
library(pixmap)
```

Part 1a

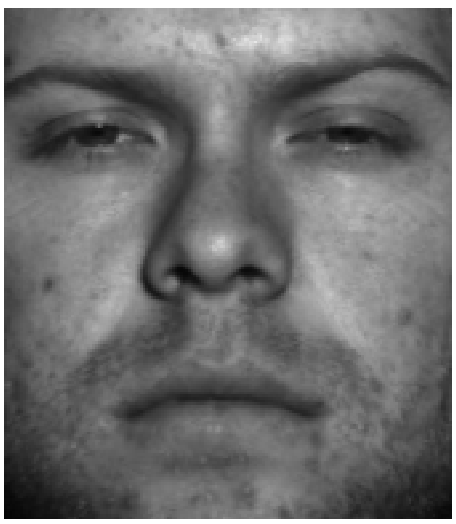
Paste or type in the given code here

```
face_01 = read.pnm(file = "CroppedYale/yaleB01/yaleB01_P00A-005E+10.pgm")
```

```
## Warning in rep(cellres, length = 2): 'x' is NULL so the result will be NULL
```

Plot the data, give it a title and save the result.

```
# now plot the data
plot(face_01)
# give it a nice title
title('hw01_01a: the first face')
```




```
# save the result
filename = 'hw01_01a.png'
dev.copy(device=png, file=filename, height=600, width=800)
```

```
## quartz_off_screen
##                 3
```

```
dev.off()
```

```
## pdf
##   2
```

Extract the class and size

```
#----- START YOUR CODE BLOCK HERE -----#
attr(face_01, "class")
```

```
## [1] "pixmapGrey"
## attr(,"package")
## [1] "pixmap"
```

```
attr(face_01,"size")
```

```
## [1] 192 168
```

```
#----- END YOUR CODE BLOCK HERE -----#
```

Part 1b

```
# make face_01 into a matrix with the given command
face_01_matrix = getChannels(face_01)

# load a second face
face_02 = read.pnm(file = "CroppedYale/yaleB02/yaleB02_P00A-005E+10.pgm")
```

```
## Warning in rep(cellres, length = 2): 'x' is NULL so the result will be NULL
```

```
face_02_matrix = getChannels(face_02)

# combine two faces into a single data matrix and make that a pixmap
faces_matrix = cbind( face_01_matrix , face_02_matrix )
faces = pixmapGrey( faces_matrix )
```

```
## Warning in rep(cellres, length = 2): 'x' is NULL so the result will be NULL
```

```
# plot to verify
plot(faces)
```

Find min and max values

```
#----- START YOUR CODE BLOCK HERE -----#
min(faces_matrix)
```

```
## [1] 0.007843137
```

```
max(faces_matrix)
```

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

```
#----- END YOUR CODE BLOCK HERE -----#
```

Part 1c

```
# get directory structure
dir_list_1 = dir(path="CroppedYale/",all.files=FALSE)
dir_list_2 = dir(path="CroppedYale/",all.files=FALSE,recursive=TRUE)
```

Find lengths

```
#----- START YOUR CODE BLOCK HERE -----#
length(dir_list_1)
```

```
## [1] 38
```

```
head(dir_list_1)
```

```
## [1] "yaleB01" "yaleB02" "yaleB03" "yaleB04" "yaleB05" "yaleB06"
```

```
length(dir_list_2)
```

```
## [1] 2547
```

```
head(dir_list_2)
```

```
## [1] "yaleB01/DEADJOE"                    "yaleB01/WS_FTP.LOG"
## [3] "yaleB01/yaleB01_P00_Ambient.pgm"    "yaleB01/yaleB01_P00.info"
## [5] "yaleB01/yaleB01_P00A-005E-10.pgm"   "yaleB01/yaleB01_P00A-005E+10.pgm"
```

```
#----- END YOUR CODE BLOCK HERE -----#
```

Part 1d

```
# the list of pictures (note the absence of 14 means that 31 corresponds to yaleB32)
pic_list = c( 05 , 11 , 31 )
view_list = c(  'P00A-005E+10' , 'P00A-005E-10' , 'P00A-010E+00')

# preallocate an empty list
pic_data = vector("list",length(pic_list)*length(view_list))
# initialize an empty matrix of faces data
faces_matrix = vector()
```

```
#----- START YOUR CODE BLOCK HERE -----#
for (i in c(1:length(pic_list)))
{
  i
  my_face_row = vector()
  for (j in c(1:length(view_list)))
  {
```

```
    j
    my_filename = sprintf("CroppedYale/%s/%s_%s.pgm",dir_list_1[pic_list[i]] ,
                          dir_list_1[pic_list[i]] , view_list[j])
    my_face = read.pnm(file = my_filename)
    my_face_matrix = getChannels(my_face)
    my_face_row = cbind(my_face_row, my_face_matrix)
  }
  faces_matrix = rbind(faces_matrix,my_face_row)
}
```

```
## Warning in rep(cellres, length = 2): 'x' is NULL so the result will be NULL

## Warning in rep(cellres, length = 2): 'x' is NULL so the result will be NULL

## Warning in rep(cellres, length = 2): 'x' is NULL so the result will be NULL

## Warning in rep(cellres, length = 2): 'x' is NULL so the result will be NULL

## Warning in rep(cellres, length = 2): 'x' is NULL so the result will be NULL

## Warning in rep(cellres, length = 2): 'x' is NULL so the result will be NULL

## Warning in rep(cellres, length = 2): 'x' is NULL so the result will be NULL

## Warning in rep(cellres, length = 2): 'x' is NULL so the result will be NULL

## Warning in rep(cellres, length = 2): 'x' is NULL so the result will be NULL
```

```
#----- END YOUR CODE BLOCK HERE -----#
```

Uncomment the code below once you completed the above code block.

```
# # now faces_matrix has been built properly.  plot and save it.
faces = pixmapGrey(faces_matrix)
```

```
## Warning in rep(cellres, length = 2): 'x' is NULL so the result will be NULL
```

```
plot(faces)
# # give it a nice title
title('hw01_01d: 3x3 grid of faces')
```

hw01_01d: 3x3 grid of faces

```
# # save the result
filename = 'hw01_01d.png'
dev.copy(device=png, file=filename, height=600, width=800)
```

```
## quartz_off_screen
##                 3
```

```
dev.off()
```

```
## pdf
##   2
```