Kevin Gong
kg2445
HW01
Stat W4240
Section 2

Homework 1

Problem #1 (James 2.8)

Part a)

```
> college <- read.csv('http://www-bcf.usc.edu/~gareth/ISL/College.csv')
> head(college)
                             X Private Apps Accept Enroll Top10perc Top25perc
1 Abilene Christian University     Yes 1660   1232    721        23        52
2           Adelphi University     Yes 2186   1924    512        16        29
3               Adrian College     Yes 1428   1097    336        22        50
4          Agnes Scott College     Yes  417    349    137        60        89
5    Alaska Pacific University     Yes  193    146     55        16        44
6            Albertson College     Yes  587    479    158        38        62
  F.Undergrad P.Undergrad Outstate Room.Board Books Personal PhD Terminal S.F.Ratio
1        2885         537     7440       3300   450     2200  70       78      18.1
2        2683        1227    12280       6450   750     1500  29       30      12.2
3        1036          99    11250       3750   400     1165  53       66      12.9
4         510          63    12960       5450   450      875  92       97       7.7
5         249         869     7560       4120   800     1500  76       72      11.9
6         678          41    13500       3335   500      675  67       73       9.4
  perc.alumni Expend Grad.Rate
1          12   7041        60
2          16  10527        56
3          30   8735        54
4          37  19016        59
5           2  10922        15
6          11   9727        55
```

Here, we read the College.csv data into a variable we'll call *college*. While the question asks us to call the data *college* (and we've done this in our code), for the purposes of saving space here, we'll use *head(college)* to examine the first few rows of *college*.

Part b)

R Data Editor

X	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board
Abilene Christian University	Yes	1660	1232	721	23	52	2885	537	7440	3300
Adelphi University	Yes	2186	1924	512	16	29	2683	1227	12280	6450
Adrian College	Yes	1428	1097	336	22	50	1036	99	11250	3750
Agnes Scott College	Yes	417	349	137	60	89	510	63	12960	5450
Alaska Pacific University	Yes	193	146	55	16	44	249	869	7560	4120
Albertson College	Yes	587	479	158	38	62	678	41	13500	3335
Albertus Magnus College	Yes	353	340	103	17	45	416	230	13290	5720
Albion College	Yes	1899	1720	489	37	68	1594	32	13868	4826
Albright College	Yes	1038	839	227	30	63	973	306	15595	4400
Alderson-Broaddus College	Yes	582	498	172	21	44	799	78	10468	3380
Alfred University	Yes	1732	1425	472	37	75	1830	110	16548	5406
Allegheny College	Yes	2652	1900	484	44	77	1707	44	17080	4440
Allentown Coll. of St. Francis de Sales	Yes	1179	780	290	38	64	1130	638	9690	4785
Alma College	Yes	1267	1080	385	44	73	1306	28	12572	4552
Alverno College	Yes	494	313	157	23	46	1317	1235	8352	3640
American International College	Yes	1420	1093	220	9	22	1018	287	8700	4780
Amherst College	Yes	4302	992	418	83	96	1593	5	19760	5300
Anderson University	Yes	1216	908	423	19	40	1819	281	10100	3520
Andrews University	Yes	1130	704	322	14	23	1586	326	9996	3090
Angelo State University	No	3540	2001	1016	24	54	4190	1512	5130	3592
Antioch University	Yes	713	661	252	25	44	712	23	15476	3336
Appalachian State University	No	7313	4664	1910	20	63	9940	1035	6806	2540
Aquinas College	Yes	619	516	219	20	51	1251	767	11208	4124
Arizona State University Main campus	No	12809	10308	3761	24	49	22593	7585	7434	4850
Arkansas College (Lyon College)	Yes	708	334	166	46	74	530	182	8644	3922
Arkansas Tech University	No	1734	1729	951	12	52	3602	939	3460	2650
Assumption College	Yes	2135	1700	491	23	59	1708	689	12000	5920
Auburn University-Main Campus	No	7548	6791	3070	25	57	16262	1716	6300	3933

First we view the data using *fix(college)*

Using the provided code, we rename the row names and delete the first column of name data:

```
rownames(college) <- college[,1]
college <- college[,-1]
```

Now the first few columns look like this:

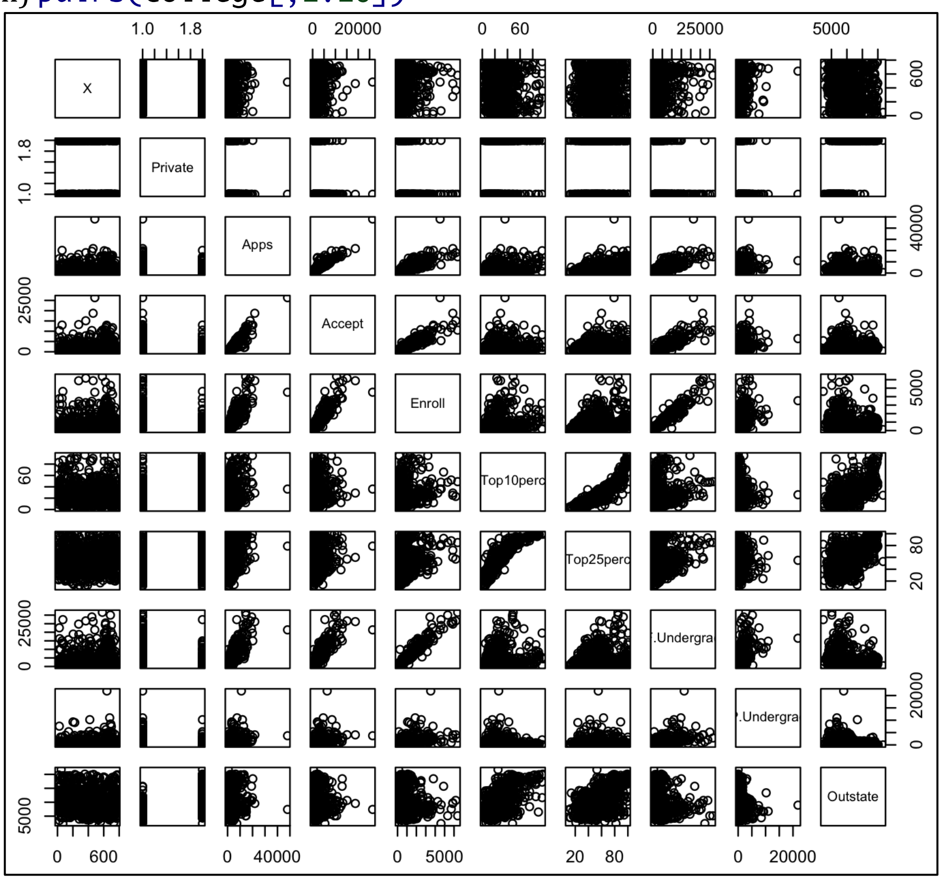
X	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad
Abilene Christian University	Yes	1660	1232	721	23	52	288
Adelphi University	Yes	2186	1924	512	16	29	268

Part c)

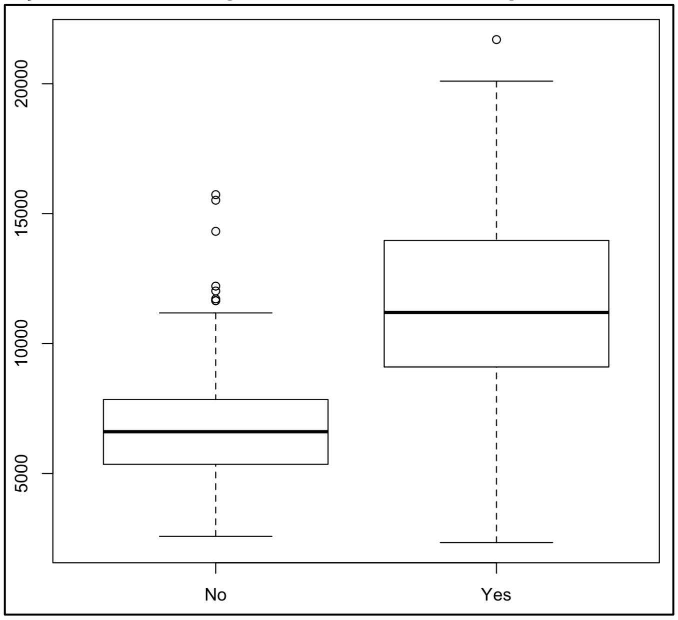
i) `summary(college)`

```
> summary(college)
                            X        Private        Apps           Accept          Enroll        Top10perc       Top25perc      F.Undergrad     P.Undergrad        Outstate    
 Abilene Christian University:  1   No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00   Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340  
 Adelphi University          :  1   Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00   1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320  
 Adrian College              :  1             Median : 1558   Median : 1110   Median : 434   Median :23.00   Median : 54.0   Median : 1707   Median :  353.0   Median : 9990  
 Agnes Scott College         :  1             Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56   Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441  
 Alaska Pacific University   :  1             3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00   3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925  
 Albertson College           :  1             Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00   Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700  
 (Other)                     :771                                                                                                                                           
   Room.Board       Books           Personal         PhD            Terminal       S.F.Ratio      perc.alumni        Expend        Grad.Rate     
 Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00   Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186   Min.   : 10.00  
 1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00   1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751   1st Qu.: 53.00  
 Median :4200   Median : 500.0   Median :1200   Median : 75.00   Median : 82.0   Median :13.60   Median :21.00   Median : 8377   Median : 65.00  
 Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66   Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660   Mean   : 65.46  
 3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00   3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830   3rd Qu.: 78.00  
 Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00   Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233   Max.   :118.00  
```

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



iii) `plot(college$Private,college$Outstate)`



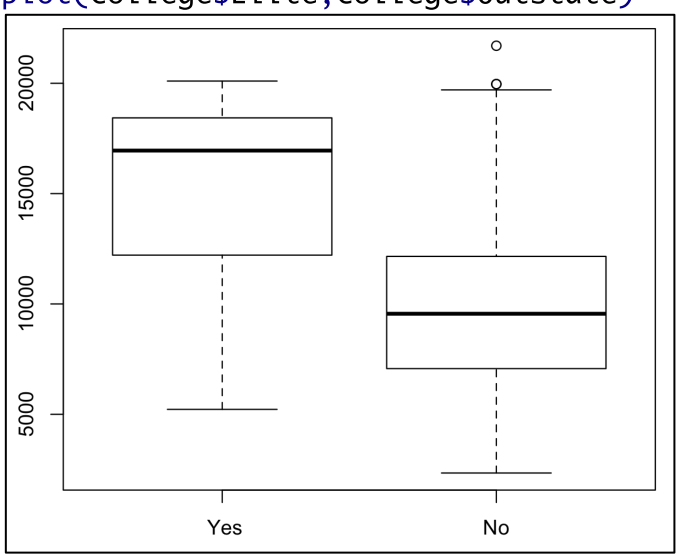
We can see that private universities have significantly higher median out-of-state tuitions than public universities.

iv) `summary(college)`

```
  Elite
  Yes: 78
 No  :699
```

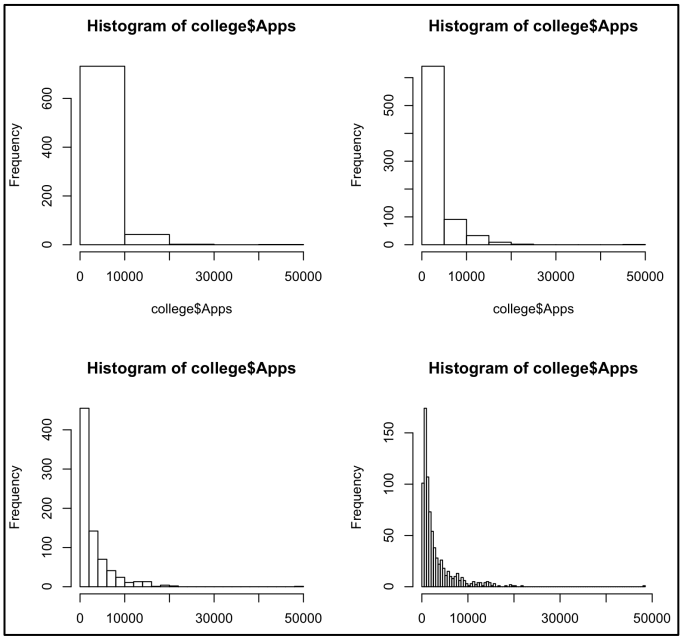
We see there are 78 elite universities, where an elite university is defined as one where the proportion of incoming students coming from the top 10% of their high school classes exceeds 50%.

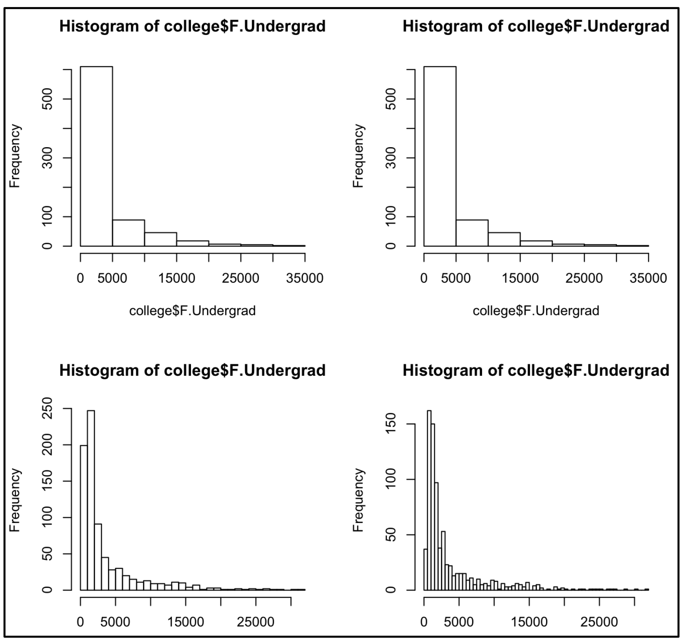
`plot(college$Elite,college$Outstate)`



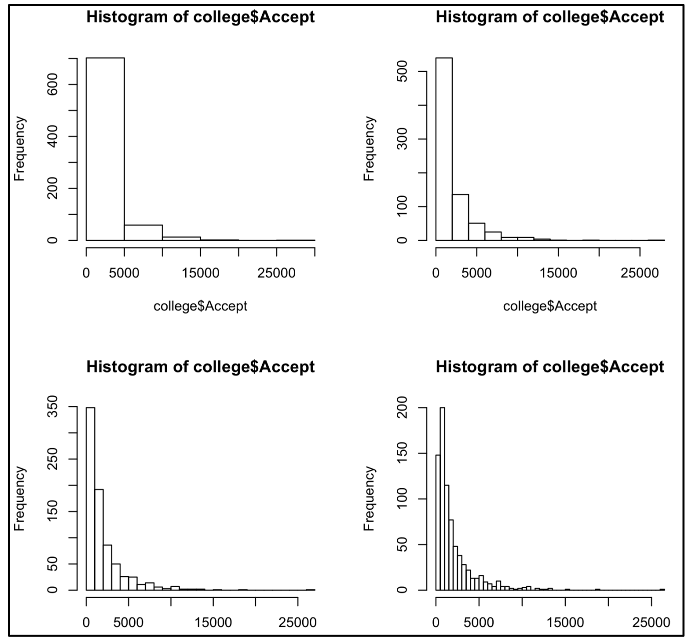
We can see that elite universities have significantly higher median out-of-state tuitions than non-elite universities.

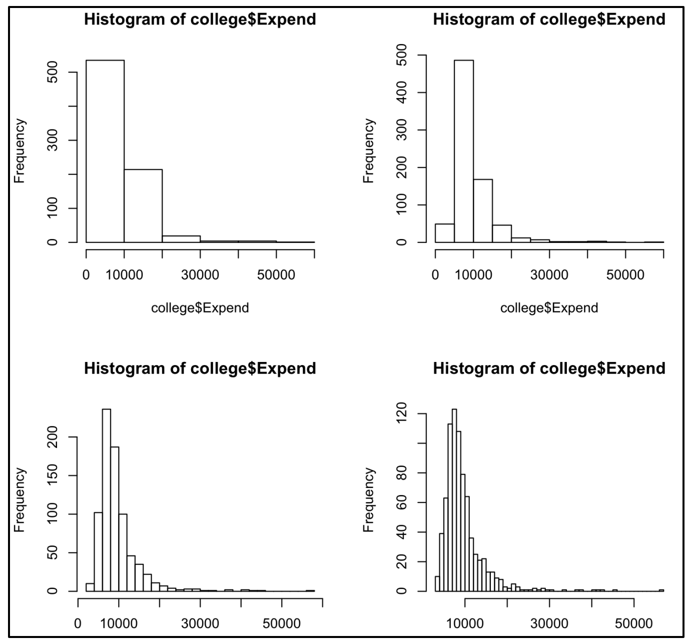
v)

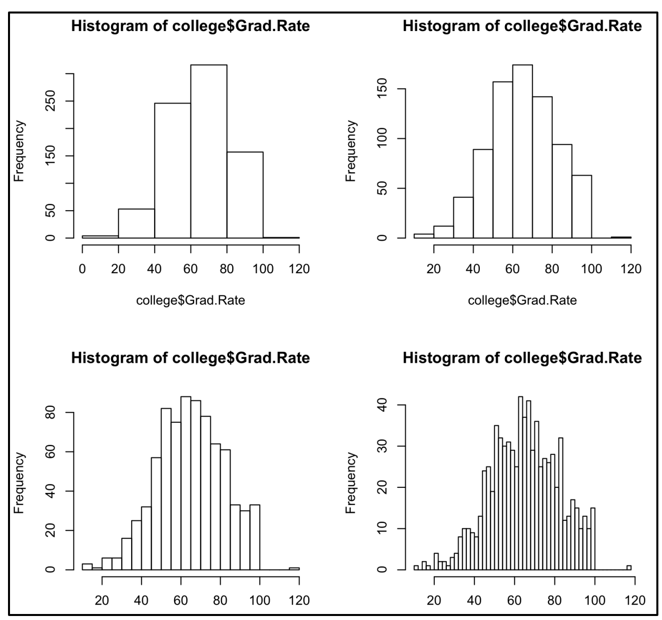




Histograms of # applications received, # applicants accepted, # full-time undergraduates, instructional expenditure per student, and graduate rate. The frequencies we used are 5, 10, 25, and 75.







vi) From the histograms in part v, we see that the mean graduation rate for all universities appears to lie between 60% and 80%, while the expense per student has a median of just under $10,000 with a long right tail. From our boxplots, we can see that elite and private universities have significantly higher median out-of-state tuitions than non-elite universities.

Problem #2 (James 2.9)

Part a)

```
sapply(auto,class)
```

```
sapply(auto,class)
      mpg  cylinders displacement   horsepower       weight acceleration         year       origin         name
"numeric"  "integer"    "numeric"    "integer"    "integer"    "numeric"    "integer"    "integer"     "factor"
```

The *name* predictor is qualitative, while all other predictors are quantitative.

Part b)

```
sapply(auto[,1:8],range,na.rm=TRUE)
```

```
> sapply(auto[,1:8],range,na.rm=TRUE)
      mpg cylinders displacement horsepower weight acceleration year origin
[1,]  9.0         3           68         46   1613          8.0   70      1
[2,] 46.6         8          455        230   5140         24.8   82      3
```

The ranges for the quantitative predictors is shown above. This was done by applying the *range()* function to the first 8 predictors.

Part c)

```
sapply(auto[,1:8],mean,na.rm=TRUE)
sapply(auto[,1:8],sd,na.rm=TRUE)
```

```
> sapply(auto[,1:8],mean,na.rm=TRUE)
        mpg    cylinders displacement   horsepower       weight acceleration         year       origin
  23.515869     5.458438   193.532746   104.469388  2970.261965    15.555668    75.994962     1.574307
> sapply(auto[,1:8],sd,na.rm=TRUE)
        mpg    cylinders displacement   horsepower       weight acceleration         year       origin
  7.8258039    1.7015770  104.3795833   38.4911599  847.9041195    2.7499953    3.6900049    0.8025495
```

The mean and standard deviation of each quantitative predictor is shown above. This was done by applying the *mean()* and *sd()* functions to the quantitative predictors.

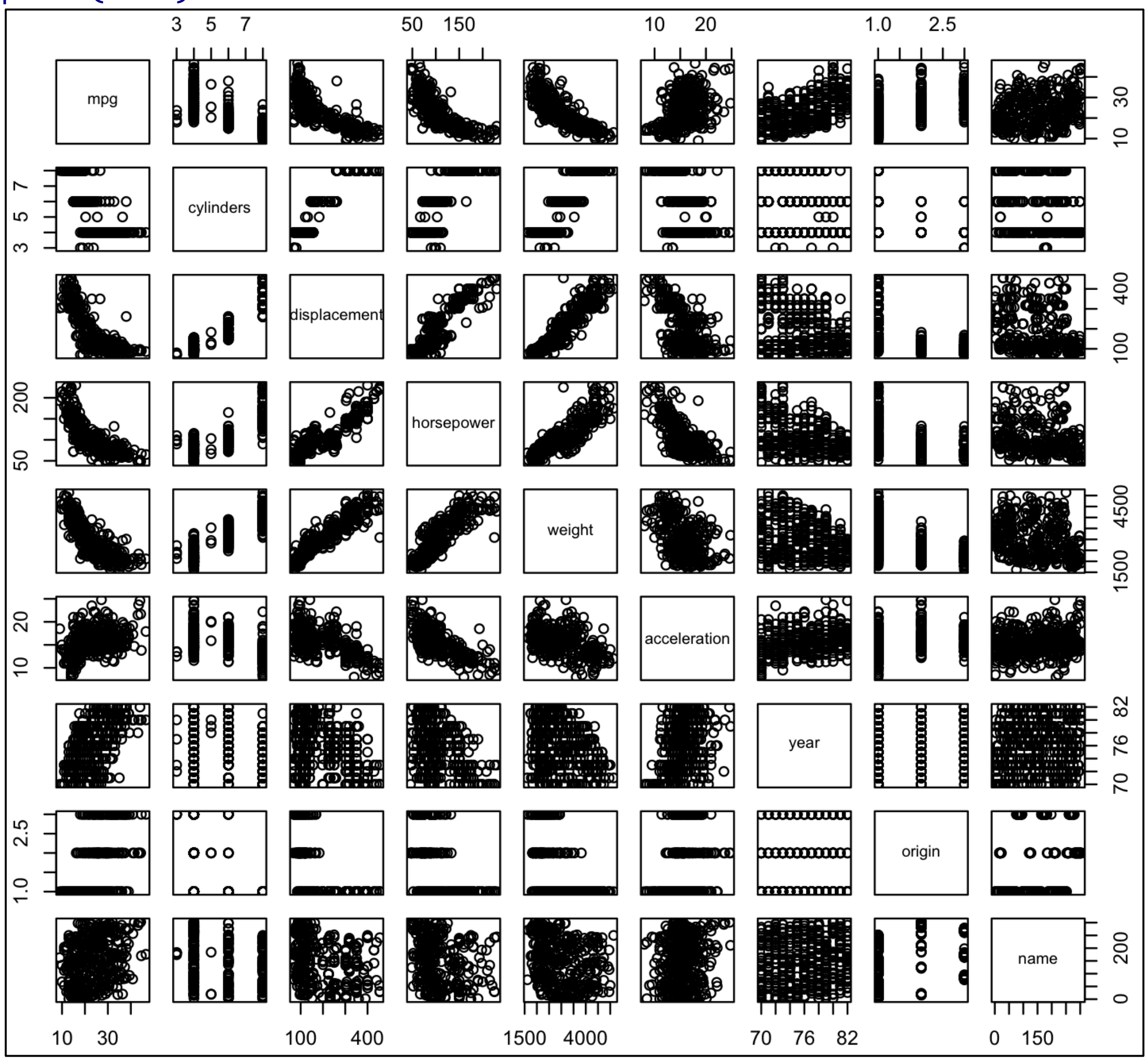
Part d)

```
auto = auto[-c(10:85),]
sapply(auto[,1:8],range,na.rm=TRUE)
sapply(auto[,1:8],mean,na.rm=TRUE)
sapply(auto[,1:8],sd,na.rm=TRUE)
```

```
> sapply(auto[,1:8],range,na.rm=TRUE)
      mpg cylinders displacement horsepower weight acceleration year origin
[1,] 11.0         3           68         46   1649          8.5   70      1
[2,] 46.6         8          455        230   4997         24.8   82      3
> sapply(auto[,1:8],mean,na.rm=TRUE)
        mpg   cylinders displacement   horsepower       weight acceleration         year       origin
  24.438629    5.370717   187.049844   100.955836  2933.962617    15.723053    77.152648     1.598131
> sapply(auto[,1:8],sd,na.rm=TRUE)
        mpg   cylinders displacement   horsepower       weight acceleration         year       origin
  7.9081842   1.6534857   99.6353853   35.8955668  810.6429384    2.6805138    3.1112298    0.8161627
```

Part e)

```
pairs(auto)
```



Part f)

From our scatterplot matrix in part e, we can see that *mpg* appears to have somewhat negative linear relationships with the predictors *displacement, horsepower,* and *weight*. The negative relationships indicate that mpg is higher when a car's displacement, horsepower, and weight are lower. Thus, these three predictors could be useful in predicting *mpg* since they appear have fairly strong negative relationships with *mpg*.

Problem #3 (James 2.10)

Part a)

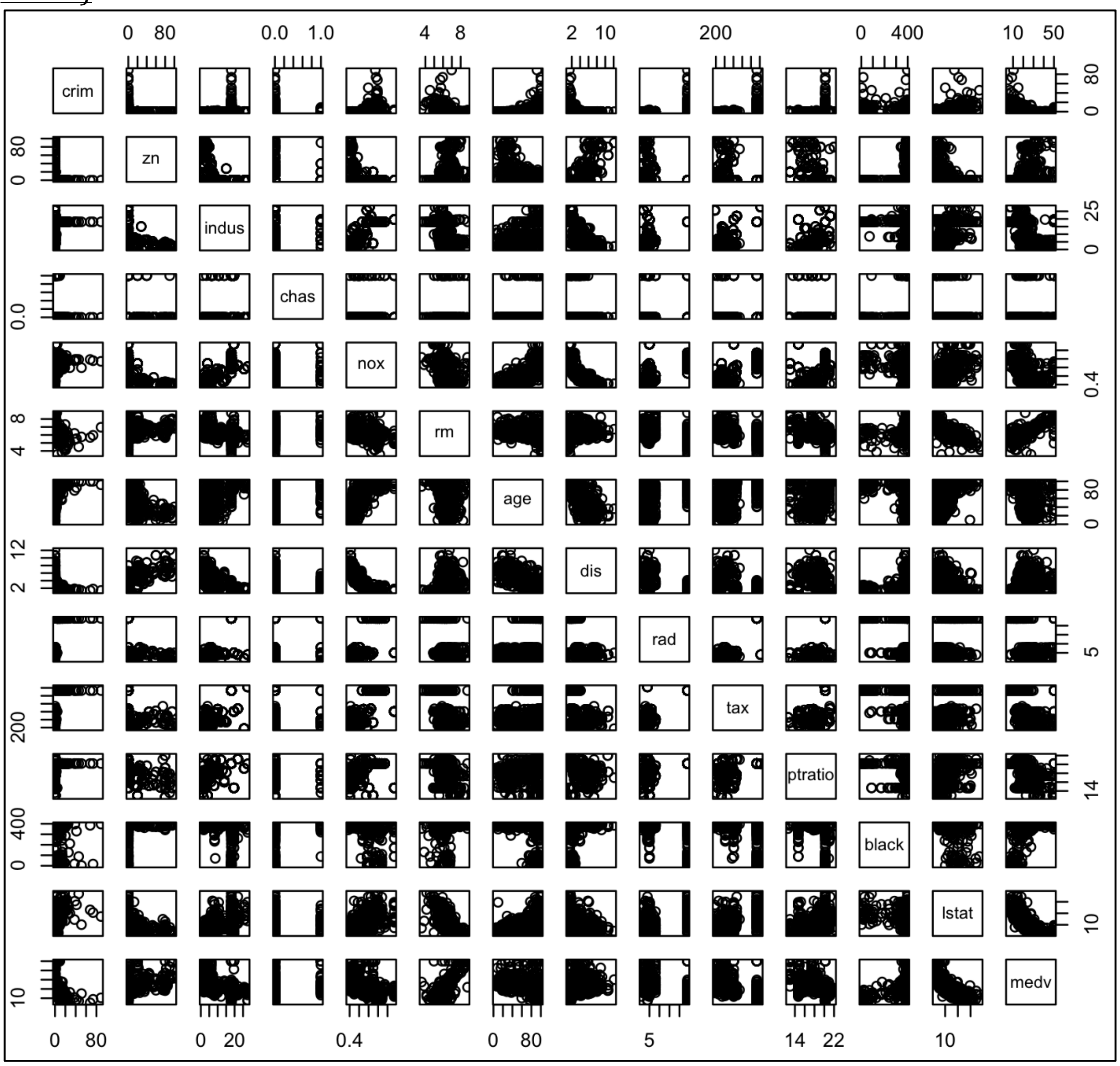
```
?Boston
```

Description

The `Boston` data frame has 506 rows and 14 columns.

The rows represent different tracts/towns/suburbs of Boston, while the columns represent different predictors/indicators.

Part b)



From a quick glance at the scatterplot matrix, there don't appear to be any especially strong linear correlations between the predictors that immediate stand out. Perhaps the most apparent ones are *rm* (average number of rooms per dwelling) correlating negatively with *lstat* (lower status of the population as a percent), *rm* correlating positively with *medv* (median value of owner-occupied homes), and a negative correlation between *lstat* and *medv*. These correlations seem to make sense since larger homes (which have more rooms than smaller homes) are probably higher in value and lower-status families are less likely to be able to afford these homes.

Part c)

The predictor for per capita crime rate is *crim*. From the scatterplot matrix, some more notable relationships suggest that:

- crime is higher among tracts that bound the Charles River than those that don't (*chas*)
- crime is higher where there are above average concentrations of nitrogen oxides (*nox*)
- crime appears to be exponentially correlated with the proportion of owner-occupied units built prior to 1940 (*age*)
- crime appears to be negatively exponentially correlated with the weighted mean of distances to five Boston employment centers (*dis*)
- crime appears to be negatively exponentially correlated with the median value of owner-occupied homes (*medv*)

Part d)

```
sapply(Boston,range,na.rm=TRUE)
sapply(Boston,mean,na.rm=TRUE)
sapply(Boston,sd,na.rm=TRUE)
summary(Boston)
```

```
> sapply(Boston,range,na.rm=TRUE)
         crim  zn indus chas   nox    rm   age     dis rad tax ptratio  black lstat medv
[1,]  0.00632   0  0.46    0 0.385 3.561   2.9  1.1296   1 187    12.6   0.32  1.73    5
[2,] 88.97620 100 27.74    1 0.871 8.780 100.0 12.1265  24 711    22.0 396.90 37.97   50
> sapply(Boston,mean,na.rm=TRUE)
        crim          zn       indus        chas         nox          rm         age         dis         rad         tax     ptratio       black       lstat 
  3.61352356 11.36363636 11.13677866  0.06916996  0.55469506  6.28463439 68.57490119  3.79504269  9.54940711 408.23715415 18.45553360 356.67403162 12.65306324 
        medv 
 22.53280632 
> sapply(Boston,sd,na.rm=TRUE)
       crim          zn       indus        chas         nox          rm         age         dis         rad         tax     ptratio       black       lstat        medv 
  8.6015451  23.3224530   6.8603529   0.2539940   0.1158777   0.7026171  28.1488614   2.1057101   8.7072594 168.5371161   2.1649455  91.2948644   7.1410615   9.1971041 
```

```
> summary(Boston)
      crim                zn             indus            chas              nox               rm             age              dis              rad              tax       
 Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000   Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130   Min.   : 1.000   Min.   :187.0  
 1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000   1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100   1st Qu.: 4.000   1st Qu.:279.0  
 Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000   Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207   Median : 5.000   Median :330.0  
 Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917   Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795   Mean   : 9.549   Mean   :408.2  
 3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000   3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188   3rd Qu.:24.000   3rd Qu.:666.0  
 Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000   Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.000   Max.   :711.0  
    ptratio          black            lstat             medv      
 Min.   :12.60   Min.   :  0.32   Min.   : 1.73   Min.   : 5.00  
 1st Qu.:17.40   1st Qu.:375.38   1st Qu.: 6.95   1st Qu.:17.02  
 Median :19.05   Median :391.44   Median :11.36   Median :21.20  
 Mean   :18.46   Mean   :356.67   Mean   :12.65   Mean   :22.53  
 3rd Qu.:20.20   3rd Qu.:396.23   3rd Qu.:16.95   3rd Qu.:25.00  
 Max.   :22.00   Max.   :396.90   Max.   :37.97   Max.   :50.00  
```

We see that the crime rate (*crim*) ranges from 0.00632 to 88.97 per capita, the full-value property-tax rate (*tax*) ranges from 187 to 711 per $10,000, and the pupil-teacher ratio (*ptratio*) ranges from 12.6 to 22. Of these three predictors, the pupil-teacher ratio seems to have a particularly small range, with even the most crowded classes not exceeding 22 students per teacher on average. This may be due to good state requirements for class sizes. Meanwhile, the property tax rate and especially the crime rate appear to vary drastically. The very large range of the crime rate suggests that some suburbs are clearly more dangerous than others.

```
> which.max(Boston$crim)
[1] 381
> which.max(Boston$tax)
[1] 489
> which.max(Boston$ptratio)
[1] 355
```

We find that suburb #381 has the highest crime rate, suburb #489 has the highest tax rate, and suburb #355 has the highest pupil-teacher ratio.

Part e)

```
> sum(Boston$chas==1)
[1] 35
```

We find that 35 of the suburbs in this data set are bound by the Charles river.

Part f)

```
> 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 among the towns in this dataset is 19.05.

Part g)

```
> which.min(Boston$medv)
[1] 399
```

The suburb with the lowest median value of owner-occupied homes is suburb #399.

The values of the other predictors for suburb #399 are shown below:

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

Compared with the overall ranges of the predictors (shown below), suburb #399 appears to have a far above average crime rate, a higher than average proportion of non-retail business acres per town, a greater than average nitrogen oxide concentration, lower than average number of rooms per dwelling, the highest proportion of owner-occupied units built prior to 1940, the highest proportion of blacks, and a much higher proportion of lower-status individuals. These predictor values make sense since suburb #399 appears to be a much poorer suburb compared to most other Boston suburbs, and thus the predictors general associated with lower socioeconomic situations are respectively higher for this suburb.

```
> summary(Boston)
      crim                zn             indus            chas              nox               rm             age              dis              rad              tax       
 Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000   Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130   Min.   : 1.000   Min.   :187.0  
 1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000   1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100   1st Qu.: 4.000   1st Qu.:279.0  
 Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000   Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207   Median : 5.000   Median :330.0  
 Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917   Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795   Mean   : 9.549   Mean   :408.2  
 3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000   3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188   3rd Qu.:24.000   3rd Qu.:666.0  
 Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000   Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.000   Max.   :711.0  
    ptratio          black            lstat            medv      
 Min.   :12.60   Min.   :  0.32   Min.   : 1.73   Min.   : 5.00  
 1st Qu.:17.40   1st Qu.:375.38   1st Qu.: 6.95   1st Qu.:17.02  
 Median :19.05   Median :391.44   Median :11.36   Median :21.20  
 Mean   :18.46   Mean   :356.67   Mean   :12.65   Mean   :22.53  
 3rd Qu.:20.20   3rd Qu.:396.23   3rd Qu.:16.95   3rd Qu.:25.00  
 Max.   :22.00   Max.   :396.90   Max.   :37.97   Max.   :50.00  
```

Part h)

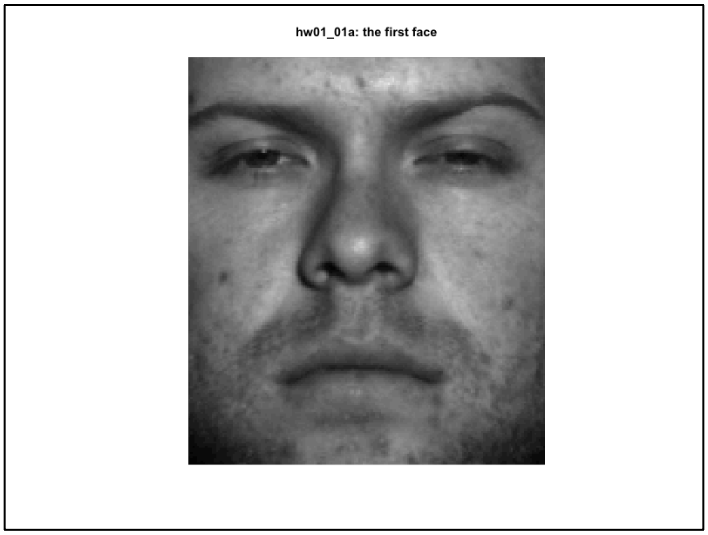
```
> sum(Boston$rm>7)
[1] 64
> sum(Boston$rm>8)
[1] 13
```

We find that 64 suburbs average more than 7 rooms per dwelling, and 13 suburbs average more than 8 rooms per dwelling. Of the suburbs in this latter group (see below), we find that they generally exhibit predictor scores characteristic of more affluent neighborhoods, such as generally having higher than average median values of owner-occupied homes and lower than average proportion of lower status individuals. This suggests that these neighborhoods are on average more affluent and socioeconomically well-off than the average Boston suburb.

```
> which(Boston$rm>8)
 [1]  98 164 205 225 226 227 233 234 254 258 263 268 365
> Boston[c(98,164,205,225,226,227,233,234,254,258,263,268,365),]
       crim zn indus chas    nox    rm  age    dis rad tax ptratio  black lstat medv
98  0.12083  0  2.89    0 0.4450 8.069 76.0 3.4952   2 276    18.0 396.90  4.21 38.7
164 1.51902  0 19.58    1 0.6050 8.375 93.9 2.1620   5 403    14.7 388.45  3.32 50.0
205 0.02009 95  2.68    0 0.4161 8.034 31.9 5.1180   4 224    14.7 390.55  2.88 50.0
225 0.31533  0  6.20    0 0.5040 8.266 78.3 2.8944   8 307    17.4 385.05  4.14 44.8
226 0.52693  0  6.20    0 0.5040 8.725 83.0 2.8944   8 307    17.4 382.00  4.63 50.0
227 0.38214  0  6.20    0 0.5040 8.040 86.5 3.2157   8 307    17.4 387.38  3.13 37.6
233 0.57529  0  6.20    0 0.5070 8.337 73.3 3.8384   8 307    17.4 385.91  2.47 41.7
234 0.33147  0  6.20    0 0.5070 8.247 70.4 3.6519   8 307    17.4 378.95  3.95 48.3
254 0.36894 22  5.86    0 0.4310 8.259  8.4 8.9067   7 330    19.1 396.90  3.54 42.8
258 0.61154 20  3.97    0 0.6470 8.704 86.9 1.8010   5 264    13.0 389.70  5.12 50.0
263 0.52014 20  3.97    0 0.6470 8.398 91.5 2.2885   5 264    13.0 386.86  5.91 48.8
268 0.57834 20  3.97    0 0.5750 8.297 67.0 2.4216   5 264    13.0 384.54  7.44 50.0
365 3.47428  0 18.10    1 0.7180 8.780 82.9 1.9047  24 666    20.2 354.55  5.29 21.9
```

Problem #4

Part a)

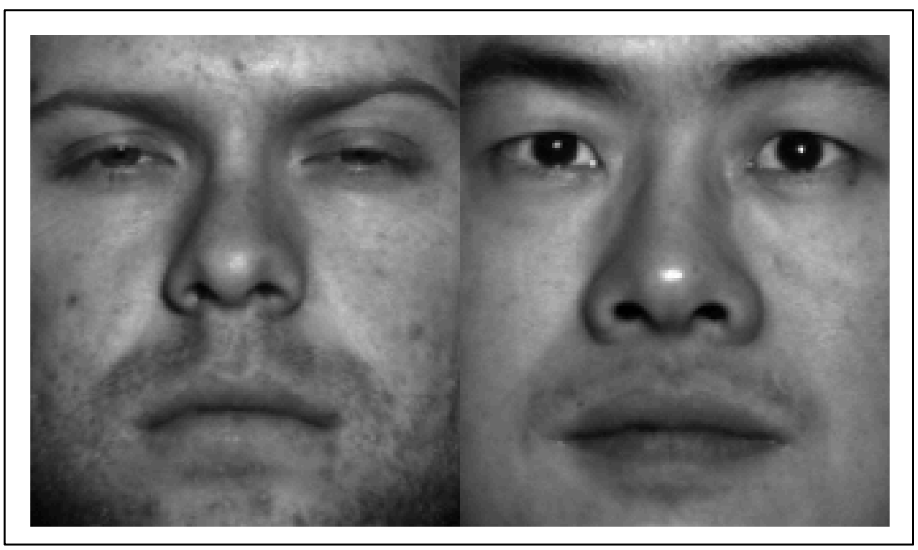


This is the picture of the face for face_01.

```
> class(face_01)
[1] "pixmapGrey"
attr(,"package")
[1] "pixmap"
> face_01@size
[1] 192 168
```

We find that face_01 has the class of pixmapGrey (meaning it is a greyscale image).
We find that the size of face_01 is 192 (height) by 168 (width) pixels.

Part b)


This is the picture of face_01 and face_02 side by side.

```
> range(faces_matrix)
[1] 0.007843137 1.000000000
```

We find that the range of pixel values for these two images ranges from 0.0078 to 1.0. In general, pixels for the class of objects pixmapGrey can range from 0 (minimum) to 1 (maximum), where 0 corresponds to the color black and 1 corresponds to the color white.

Part c)

The dir_list_1 and dir_list_2 lists capture some of the folder structure of the CroppedYale folder. dir_list_1 captures the first layer of folders under CroppedYale (such as yaleB01, yaleB02, etc.) and these correspond to different people that were photographed. dir_list_2 captures the files under each subject's folder in the first layer, and these correspond to the different lighting conditions used when photographing each subject.

```
> length(dir_list_1)
[1] 38
> length(dir_list_2)
[1] 2547
```

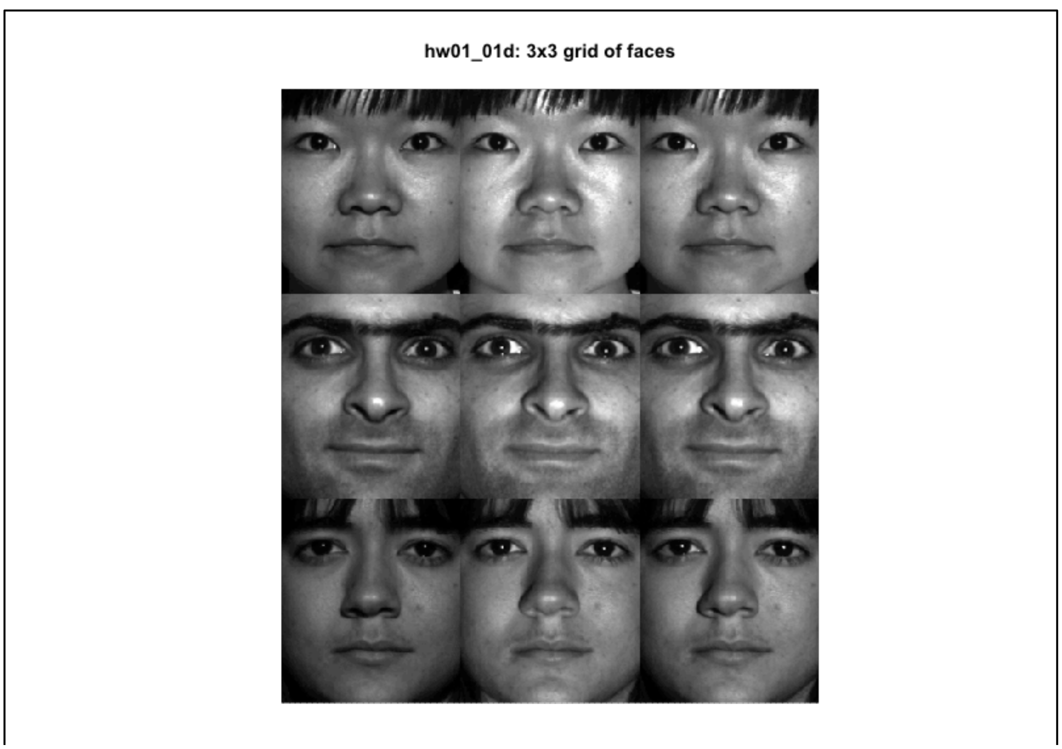
There are 38 elements in dir_list_1 (meaning 38 subjects photographed) and 2547 elements in dir_list_2 (meaning 2547 photos taken).

Some sample elements are shown below:

```
> head(dir_list_1)
[1] "yaleB01" "yaleB02" "yaleB03" "yaleB04" "yaleB05" "yaleB06"
> head(dir_list_2)
[1] "yaleB01/DEADJOE"                  "yaleB01/WS_FTP.LOG"                  "yaleB01/yaleB01_P00_Ambient.pgm"  "yaleB01/yaleB01_P00.info"
[5] "yaleB01/yaleB01_P00A-005E-10.pgm" "yaleB01/yaleB01_P00A-005E+10.pgm"
```

Part d)

Using nested for loops, rbind (binding rows), and cbind (binding columns), we create a matrix of our images and plot them to generate the following images:



Code

```
#############################
# Kevin Gong
# STAT W4240
# Homework 1 , Problem 1
# 2/5/14
#

#############################

#################
# Setup
#################

# make sure R is in the proper working directory
# note that this will be a different path for every machine
setwd("~/Dropbox/SIPA/Data Mining")

# first include the relevant libraries
# note that a loading error might mean that you have to
# install the package into your R distribution.
install.packages("ISLR")
library(ISLR)


#################
# Problem 1a
#################

#importing the data
college <- read.csv('http://www-bcf.usc.edu/~gareth/ISL/College.csv')
college
head(college)

#################
# Problem 1b
#################

#adding a column with university names
fix(college)
rownames(college) <- college[,1]
fix(college)

#eliminating the original column of university names
college <- college[,-1]
fix(college)

#################
# Problem 1c
#################


#
# Part i
#


#numerical summary of the variables in the data set
summary(college)
```

```
#
# Part ii
#

#scatterplot matrix of first 10 variables
pairs(college[,1:10])


#
# Part iii
#

#side-by-side boxplots of Private and Outstate
plot(college$Private,college$Outstate)

#
# Part iv
#

#generate new variable Elite based on whether more than 50% of incoming class come from
 the top 10% of their high school
Elite=rep("No",nrow(college ))
Elite[college$Top10perc >50]=" Yes"
Elite=as.factor(Elite)
college=data.frame(college ,Elite)

#we see there are 78 Elite universities
summary(college)

#side-by-side boxplots of Private and Outstate
plot(college$Elite,college$Outstate)


#
# Part v
#

par(mfrow=c(2,2))
hist(college$Apps, breaks=5)
hist(college$Apps, breaks=10)
hist(college$Apps, breaks=25)
hist(college$Apps, breaks=75)


par(mfrow=c(2,2))
hist(college$Accept, breaks=5)
hist(college$Accept, breaks=10)
hist(college$Accept, breaks=25)
hist(college$Accept, breaks=75)


par(mfrow=c(2,2))
hist(college$F.Undergrad, breaks=5)
hist(college$F.Undergrad, breaks=10)
hist(college$F.Undergrad, breaks=25)
hist(college$F.Undergrad, breaks=75)


par(mfrow=c(2,2))
```

```
hist(college$Expend, breaks=5)
hist(college$Expend, breaks=10)
hist(college$Expend, breaks=25)
hist(college$Expend, breaks=75)

par(mfrow=c(2,2))
hist(college$Grad.Rate, breaks=5)
hist(college$Grad.Rate, breaks=10)
hist(college$Grad.Rate, breaks=25)
hist(college$Grad.Rate, breaks=75)

#
# Part vi
#

mean(college$Expend[college$Elite=="Yes"])
summary(college$Expend,college$Elite=="No")
```

```
#############################
# Kevin Gong
# STAT W4240
# Homework 1 , Problem 2
# 2/5/14
#
#############################

#################
# Setup
#################

# make sure R is in the proper working directory
# note that this will be a different path for every machine
setwd("~/Dropbox/SIPA/Data Mining")

# first include the relevant libraries
# note that a loading error might mean that you have to
# install the package into your R distribution.
install.packages("ISLR")
library(ISLR)


#################
# Problem 1a
#################

#importing the data
auto <- read.csv('~/Dropbox/SIPA/Data Mining/Autodata3.csv')

#examine which ones are quantitative/qualitative
summary(auto)
sapply(auto,class)

#quantitative: mpg, cylinders, displacement, horsepower, weight, acceleration, year,
 origin
#qualitative: name
```

```
#################
# Problem 1b
#################

#range of each quantitative predictor
sapply(auto[,1:8],range,na.rm=TRUE)


#################
# Problem 1c
#################

#mean and standard deviation of each quantitative predictor
sapply(auto[,1:8],mean,na.rm=TRUE)
sapply(auto[,1:8],sd,na.rm=TRUE)


#################
# Problem 1d
#################

#removing rows 10 through 85
auto = auto[-c(10:85),]

#range, mean, and standard deviation of remaining observations of each quantitative
 predictor
sapply(auto[,1:8],range,na.rm=TRUE)
sapply(auto[,1:8],mean,na.rm=TRUE)
sapply(auto[,1:8],sd,na.rm=TRUE)


#################
# Problem 1e
#################

#restore removed rows
auto <- read.csv('~/Dropbox/SIPA/Data Mining/Autodata3.csv')

#comparing relationships between predictors via scatterplots
pairs(auto)
```

```
#############################
# Kevin Gong
# STAT W4240
# Homework 1 , Problem 3
# 2/5/14
#
# 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("~/Dropbox/SIPA/Data Mining")

# first include the relevant libraries
# note that a loading error might mean that you have to
# install the package into your R distribution.
install.packages("MASS")
library(MASS)


#################
# Problem 3a
#################

#count the number of rows in Boston
?Boston


#################
# Problem 3b
#################

#creating pairwise scatterplots
pairs(Boston)


#################
# Problem 3d
#################

#range, mean, and standard deviation of Boston predictors
sapply(Boston,range,na.rm=TRUE)
sapply(Boston,mean,na.rm=TRUE)
sapply(Boston,sd,na.rm=TRUE)
summary(Boston)

#find which have highest crime rates, tax rates, and pupil-teacher ratios
which.max(Boston$crim)
which.max(Boston$tax)
which.max(Boston$ptratio)




#################
# Problem 3e
#################

#counting the number of suburbs bordering the Charles River
sum(Boston$chas==1)
sum(Boston$chas==0)
sum(is.na(Boston$chas)) #check for missing data


#################
# Problem 3f
#################
```

```
#median pupil-teacher teacher among the towns
summary(Boston$ptratio)


#################
# Problem 3g
#################

#finding the lowest median value of owner-occupied homes
which.min(Boston$medv)

#values of the other predictors for suburb 399
Boston[399,]

#compare the predictor values of suburb 399 with overall predictor ranges
summary(Boston)


#################
# Problem 3h
#################

#finding the number of suburbs averaging more than 7 rooms per dwelling
sum(Boston$rm>7)

#finding the number of suburbs averaging more than 8 rooms per dwelling
sum(Boston$rm>8)

#compare the predictor values of the suburb averaging more than 8 rooms per dwelling
which(Boston$rm>8)
Boston[c(98,164,205,225,226,227,233,234,254,258,263,268,365),]
summary(Boston)






#############################
# Kevin Gong
# STAT W4240
# Homework 1 , Problem 4
# 2/5/14
#
# 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("~/Dropbox/SIPA/Data Mining")

# first include the relevant libraries
```

```r
# 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)

#################
# Problem 1a
#################

# paste or type in the given code here
face_01 = read.pnm(file = "CroppedYale/yaleB01/yaleB01_P00A-005E+10.pgm")

# 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)
dev.off()

# extract the class and size

#----- START YOUR CODE BLOCK HERE -----#
class(face_01)
face_01@size
#alternative method to get size
nrow(getChannels(face_01))
ncol(getChannels(face_01))
#----- END YOUR CODE BLOCK HERE -----#

#################
# Problem 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")
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 )

# plot to verify
plot(faces)

# find min and max values

#----- START YOUR CODE BLOCK HERE -----#
range(faces_matrix)
#alternative method to get min and max values
min(faces_matrix)
max(faces_matrix)
#----- END YOUR CODE BLOCK HERE -----#

#################
# Problem 1c
#################
```

```r
# 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)
length(dir_list_2)
#example elements
head(dir_list_1)
head(dir_list_2)
#----- END YOUR CODE BLOCK HERE -----#

#################
# Problem 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 -----#

pic_list = c( 05 , 11 , 31 )
view_list = c( 'P00A-005E+10' , 'P00A-005E-10' , 'P00A-010E+00')

faces_matrix = vector()
placeholder_matrix= vector()


for (i in 1:3) {

placeholder_matrix <- NULL

for (j in 1:3) {
 filename=
 sprintf("CroppedYale/%s/%s_%s.pgm",dir_list_1[pic_list[i]],dir_list_1[pic_list[i]],view_l
 ist[j])
print(filename)
face=read.pnm(file=filename)
pic_data <-getChannels(face)
placeholder_matrix=cbind(placeholder_matrix,pic_data)
}
faces_matrix=rbind(faces_matrix,placeholder_matrix)
}

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



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

```
title('hw01_01d: 3x3 grid of faces')
# save the result
filename = 'hw01_01d.png'
dev.copy(device=png, file=filename, height=600, width=800)
dev.off()

#################
# End of Script
#################
```