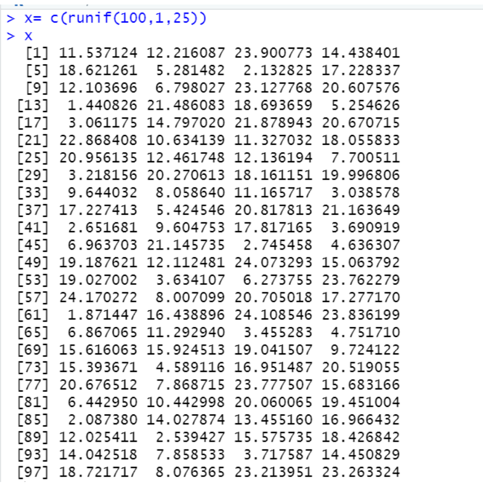
1. Vectors & Matrices

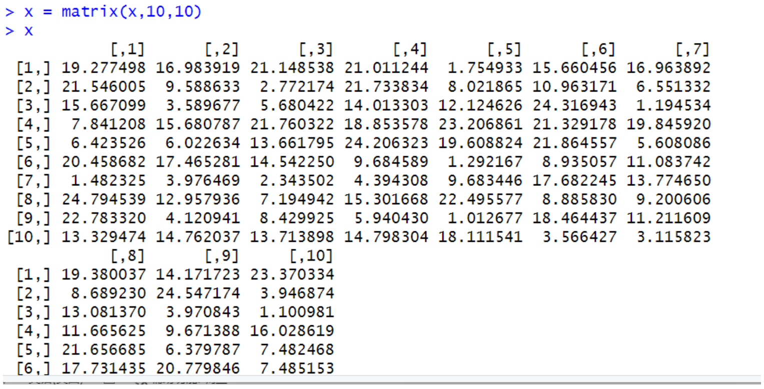
a. Create a vector with 100 elements with randomly generated real numbers ranging from 1 to 25

(Hint: use runif() function).



```
> x= c(runif(100,1,25))
> x
  [1] 11.537124 12.216087 23.900773 14.438401
  [5] 18.621261  5.281482  2.132825 17.228337
  [9] 12.103696  6.798027 23.127768 20.607576
 [13]  1.440826 21.486083 18.693659  5.254626
 [17]  3.061175 14.797020 21.878943 20.670715
 [21] 22.868408 10.634139 11.327032 18.055833
 [25] 20.956135 12.461748 12.136194  7.700511
 [29]  3.218156 20.270613 18.161151 19.996806
 [33]  9.644032  8.058640 11.165717  3.038578
 [37] 17.227413  5.424546 20.817813 21.163649
 [41]  2.651681  9.604753 17.817165  3.690919
 [45]  6.963703 21.145735  2.745458  4.636307
 [49] 19.187621 12.112481 24.073293 15.063792
 [53] 19.027002  3.634107  6.273755 23.762279
 [57] 24.170272  8.007099 20.705018 17.277170
 [61]  1.871447 16.438896 24.108546 23.836199
 [65]  6.867065 11.292940  3.455283  4.751710
 [69] 15.616063 15.924513 19.041507  9.724122
 [73] 15.393671  4.589116 16.951487 20.519055
 [77] 20.676512  7.868715 23.777507 15.683166
 [81]  6.442950 10.442998 20.060065 19.451004
 [85]  2.087380 14.027874 13.455160 16.966432
 [89] 12.025411  2.539427 15.575735 18.426842
 [93] 14.042518  7.858533  3.717587 14.450829
 [97] 18.721717  8.076365 23.213951 23.263324
```

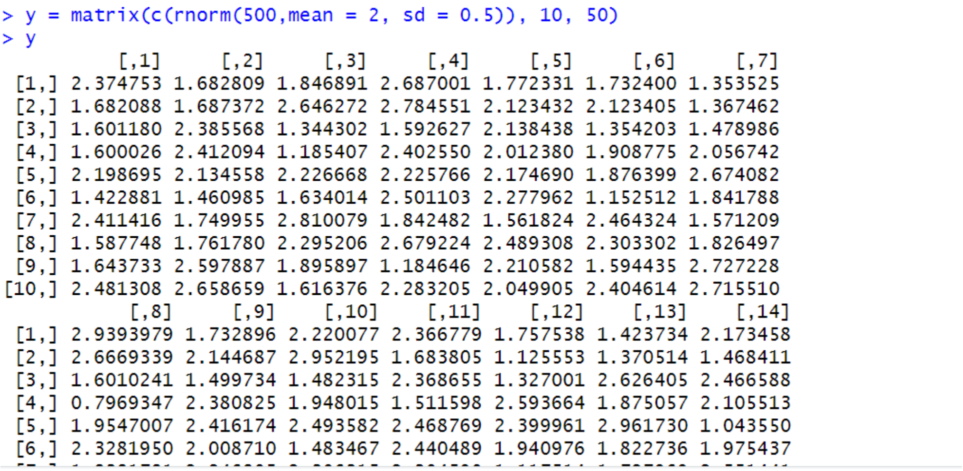
b. Reshape this vector into a 10 by 10 matrix reading by column.



```
> x = matrix(x,10,10)
> x
           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
 [1,] 19.277498 16.983919 21.148538 21.011244  1.754933 15.660456 16.963892
 [2,] 21.546005  9.588633  2.772174 21.733834  8.021865 10.963171  6.551332
 [3,] 15.667099  3.589677  5.680422 14.013303 12.124626 24.316943  1.194534
 [4,]  7.841208 15.680787 21.760322 18.853578 23.206861 21.329178 19.845920
 [5,]  6.423526  6.022634 13.661795 24.206323 19.608824 21.864557  5.608086
 [6,] 20.458682 17.465281 14.542250  9.684589  1.292167  8.935057 11.083742
 [7,]  1.482325  3.976469  2.343502  4.394308  9.683446 17.682245 13.774650
 [8,] 24.794539 12.957936  7.194942 15.301668 22.495577  8.885830  9.200606
 [9,] 22.783320  4.120941  8.429925  5.940430  1.012677 18.464437 11.211609
[10,] 13.329474 14.762037 13.713898 14.798304 18.111541  3.566427  3.115823
           [,8]      [,9]     [,10]
 [1,] 19.380037 14.171723 23.370334
 [2,]  8.689230 24.547174  3.946874
 [3,] 13.081370  3.970843  1.100981
 [4,] 11.665625  9.671388 16.028619
 [5,] 21.656685  6.379787  7.482468
 [6,] 17.731435 20.779846  7.485153
```

c. Similarly, generate another vector with 500 elements of standard normal distribution (with mean = 2 and sd =0.5) and plot its histogram. Reshape this into a 10 by 50 matrix reading by row.

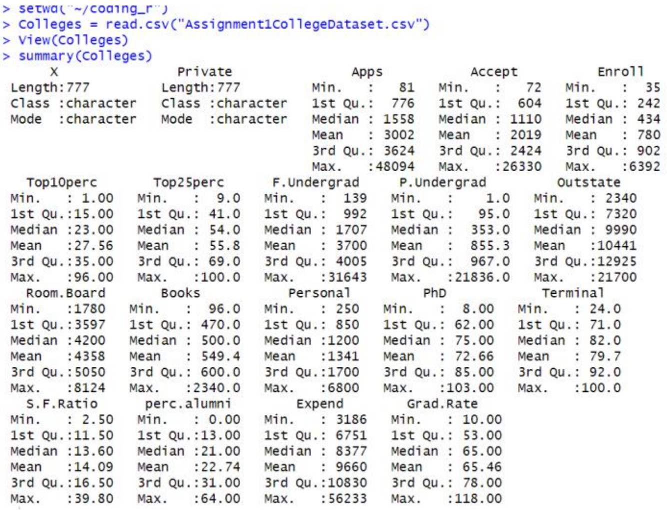
(Hint: use matrix() and rnorm() functions)



```
> y = matrix(c(rnorm(500,mean = 2, sd = 0.5)), 10, 50)
> y
          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
 [1,] 2.374753 1.682809 1.846891 2.687001 1.772331 1.732400 1.353525
 [2,] 1.682088 1.687372 2.646272 2.784551 2.123432 2.123405 1.367462
 [3,] 1.601180 2.385568 1.344302 1.592627 2.138438 1.354203 1.478986
 [4,] 1.600026 2.412094 1.185407 2.402550 2.012380 1.908775 2.056742
 [5,] 2.198695 2.134558 2.226668 2.225766 2.174690 1.876399 2.674082
 [6,] 1.422881 1.460985 1.634014 2.501103 2.277962 1.152512 1.841788
 [7,] 2.411416 1.749955 2.810079 1.842482 1.561824 2.464324 1.571209
 [8,] 1.587748 1.761780 2.295206 2.679224 2.489308 2.303302 1.826497
 [9,] 1.643733 2.597887 1.895897 1.184646 2.210582 1.594435 2.727228
[10,] 2.481308 2.658659 1.616376 2.283205 2.049905 2.404614 2.715510
           [,8]     [,9]    [,10]    [,11]    [,12]    [,13]    [,14]
 [1,] 2.9393979 1.732896 2.220077 2.366779 1.757538 1.423734 2.173458
 [2,] 2.6669339 2.144687 2.952195 1.683805 1.125553 1.370514 1.468411
 [3,] 1.6010241 1.499734 1.482315 2.368655 1.327001 2.626405 2.466588
 [4,] 0.7969347 2.380825 1.948015 1.511598 2.593664 1.875057 2.105513
 [5,] 1.9547007 2.416174 2.493582 2.468769 2.399961 2.961730 1.043550
 [6,] 2.3281950 2.008710 1.483467 2.440489 1.940976 1.822736 1.975437
```

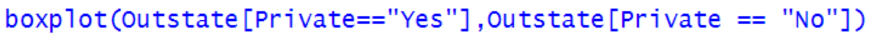
2. This exercise relates to the College data set, which can be found in the file College.csv. It contains a number of variables for 777 different universities and colleges in the US. Use the read.csv() function to read the data into R. Call the loaded data college. Make sure that you have the directory set to the correct location for the data.

a. Use the summary() function to produce a numerical summary of the variables in the data set.

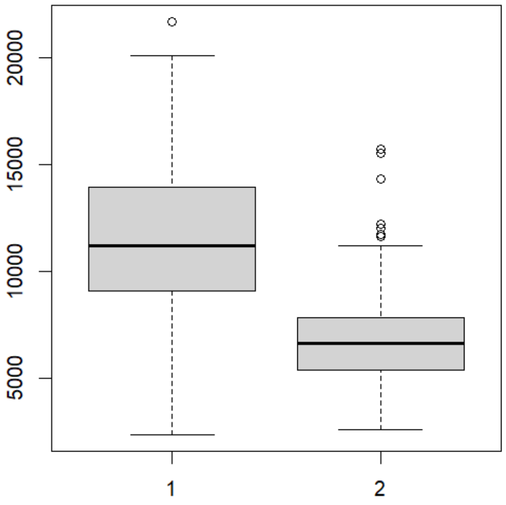


```
> setwd("~/coding_r")
> Colleges = read.csv("Assignment1CollegeDataset.csv")
> View(Colleges)
> summary(Colleges)
      X                Private               Apps           Accept          Enroll
 Length:777         Length:777         Min.   :   81   Min.   :   72   Min.   :  35
 Class :character   Class :character   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242
 Mode  :character   Mode  :character   Median : 1558   Median : 1110   Median : 434
                                       Mean   : 3002   Mean   : 2019   Mean   : 780
                                       3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902
                                       Max.   :48094   Max.   :26330   Max.   :6392
   Top10perc       Top25perc      F.Undergrad     P.Undergrad          Outstate
 Min.   : 1.00   Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
 1st Qu.:15.00   1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
 Median :23.00   Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
 Mean   :27.56   Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
 3rd Qu.:35.00   3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
 Max.   :96.00   Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
   Room.Board        Books           Personal         PhD             Terminal
 Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00   Min.   : 24.0
 1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00   1st Qu.: 71.0
 Median :4200   Median : 500.0   Median :1200   Median : 75.00   Median : 82.0
 Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66   Mean   : 79.7
 3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00   3rd Qu.: 92.0
 Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00   Max.   :100.0
   S.F.Ratio      perc.alumni        Expend         Grad.Rate
 Min.   : 2.50   Min.   : 0.00   Min.   : 3186   Min.   : 10.00
 1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751   1st Qu.: 53.00
 Median :13.60   Median :21.00   Median : 8377   Median : 65.00
 Mean   :14.09   Mean   :22.74   Mean   : 9660   Mean   : 65.46
 3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830   3rd Qu.: 78.00
 Max.   :39.80   Max.   :64.00   Max.   :56233   Max.   :118.00
```

b. Create a boxplot of variable "Outstate" as a function of "Private" (Yes for Private; No for Public).   Explain your interpretation of the boxplots.



```
boxplot(Outstate[Private=="Yes"],Outstate[Private == "No"])
```

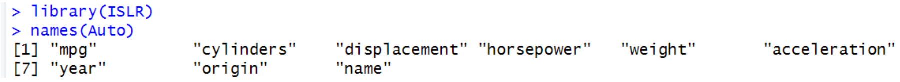


So, in the plot, number 1 is outstate-student number vs. private school. The average of outstate-student number in private school is about 12000. And in most private school, the outstate students' number is around 9000 to 14000. The limit of that is 2000 to 21000. There is only 1 outlier(school) which outstate student is over 20000.

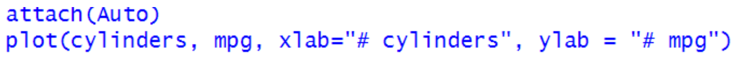
However, number 2 is outstate-student number vs. public school. The average of outstate-student number in public school is about 6500. And in most public school, the outstate students' number is around 9000 to 14000. The limit of that is 2000 to 12000. There are some outliers (schools) which outstate-student is over 12000.

In general, the average number of private outstate students is larger than the number of public school outstate students.

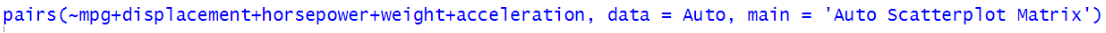
3. Load the Auto data set, which is in the ISLR library. Understand information about this data set by either ways we introduced in class (like "?Auto" and names(Auto))
a. Make a scatterplot between cylinders and mpg. Draw pairwise scatterplot between "mpg", "displacement", "horsepower", "weight", "acceleration" (try to plot all scatterplots in one figure; hint: use pairs() command). By observing the plots, do you think the two variables in each scatterplot are correlated? If so, how?



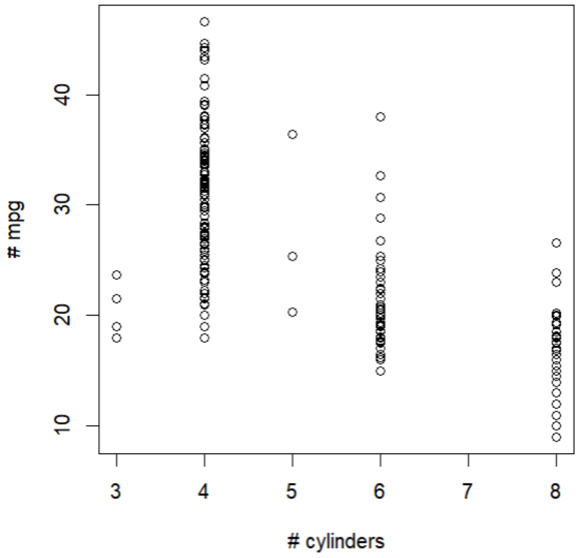
```
> library(ISLR)
> names(Auto)
[1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"       "acceleration"
[7] "year"         "origin"       "name"
```

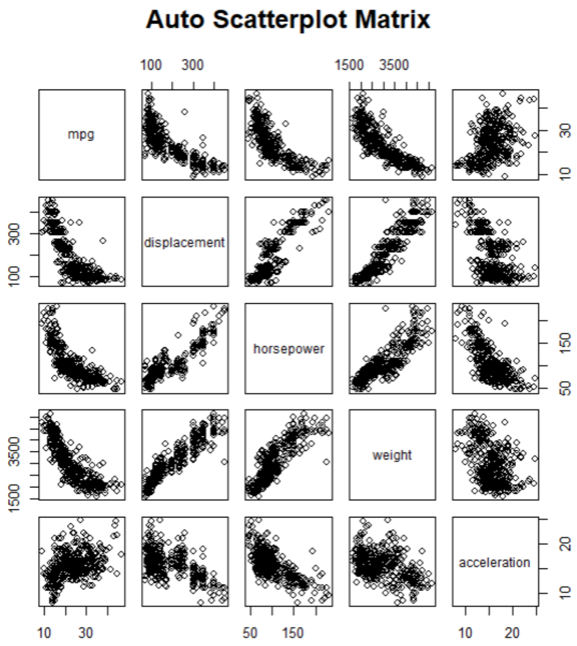


```
attach(Auto)
plot(cylinders, mpg, xlab="# cylinders", ylab = "# mpg")
```



```
pairs(~mpg+displacement+horsepower+weight+acceleration, data = Auto, main = 'Auto Scatterplot Matrix')
```





So, for mgp in this plot, it has negative trend relationship with displacement, horsepower and weight. But mgp does not have a very obvious relationship with acceleration.

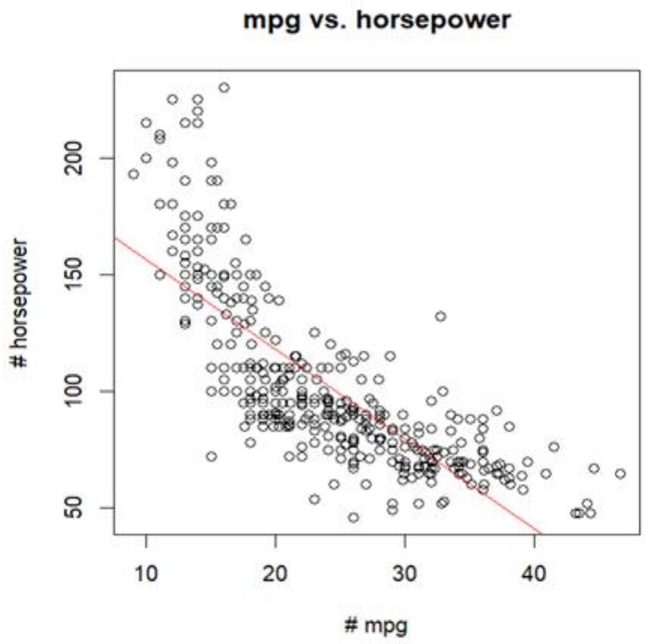
For displacement, it has positive trend relationship with horsepower and weight. But displacement has low negative trend relationship with acceleration.

For horsepower, it has strong positive trend relationship with weight and a strong negative relationship with acceleration.

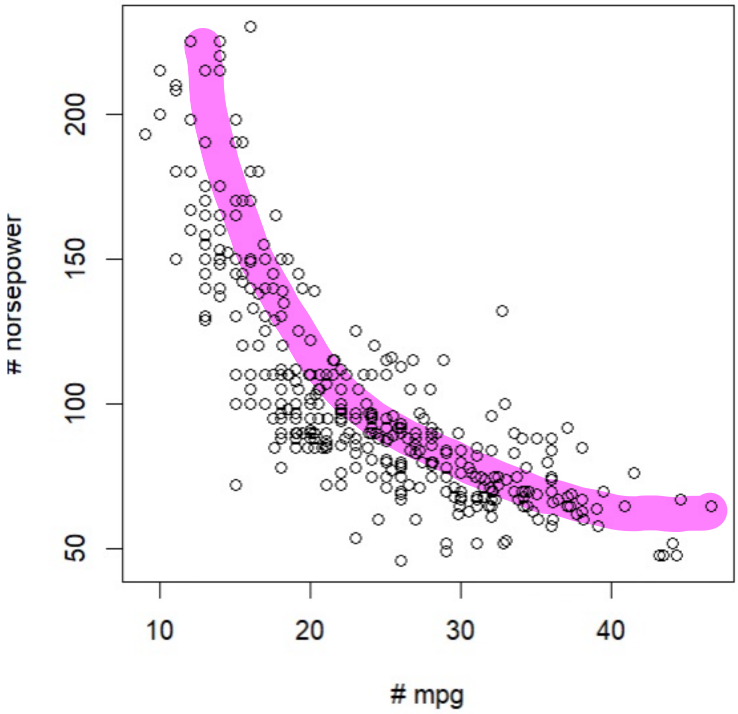
For weight, it dose not have a very obvious relationship with acceleration.

b. Create a scatterplot between mpg and horsepower. Draw a straight line on the scatterplot to represent relationship between the two variables.

(Hint: Search for “R Plot Line in Scatterplot”)



c. Is there a better way to represent their relationship rather than the linear model you just drew? (No need to use mathematical formula or plotting techniques. Just draw by hand on the scatterplot. We will learn non-linear fitting approaches later in the course.)


# norsepower
# mpg