# Assignment #1: Learning R

**SANDEEP DASARI** **UIN - 829002252**

**1. Data generation and matrix indexing**

**(1) Generate a vector with 25 elements and each element independently follows a normal distribution (with mean =0 and sd=1);**

**Solution:**

```
> set.seed(1000)
> data1 = rnorm(25,mean=0,sd=1)
> data1
 [1] -0.44577826 -1.20585657  0.04112631  0.63938841 -0.78655436 -0.38548930 -0.47586788  0.71975069
 [9] -0.01850562 -1.37311776 -0.98242783 -0.55448870  0.12138119 -0.12087232 -1.33604105  0.17005748
[17]  0.15507872  0.02493187 -2.04658541  0.21315411  2.67007166 -1.22701601  0.83424733  0.53257175
[25] -0.64682496
```

**(2) Reshape this vector into a 5 by 5 matrix in two ways (arranged by row and column);**

**Solution:** a) Matrix arranged by row

```
> matrix(data1,nrow=5, ncol=5,byrow=TRUE)
          [,1]       [,2]       [,3]       [,4]       [,5]
[1,] -0.4457783 -1.2058566 0.04112631  0.63938841 -0.7865544
[2,] -0.3854893 -0.4758679 0.71975069 -0.01850562 -1.3731178
[3,] -0.9824278 -0.5544887 0.12138119 -0.12087232 -1.3360410
[4,]  0.1700575  0.1550787 0.02493187 -2.04658541  0.2131541
[5,]  2.6700717 -1.2270160 0.83424733  0.53257175 -0.6468250
```

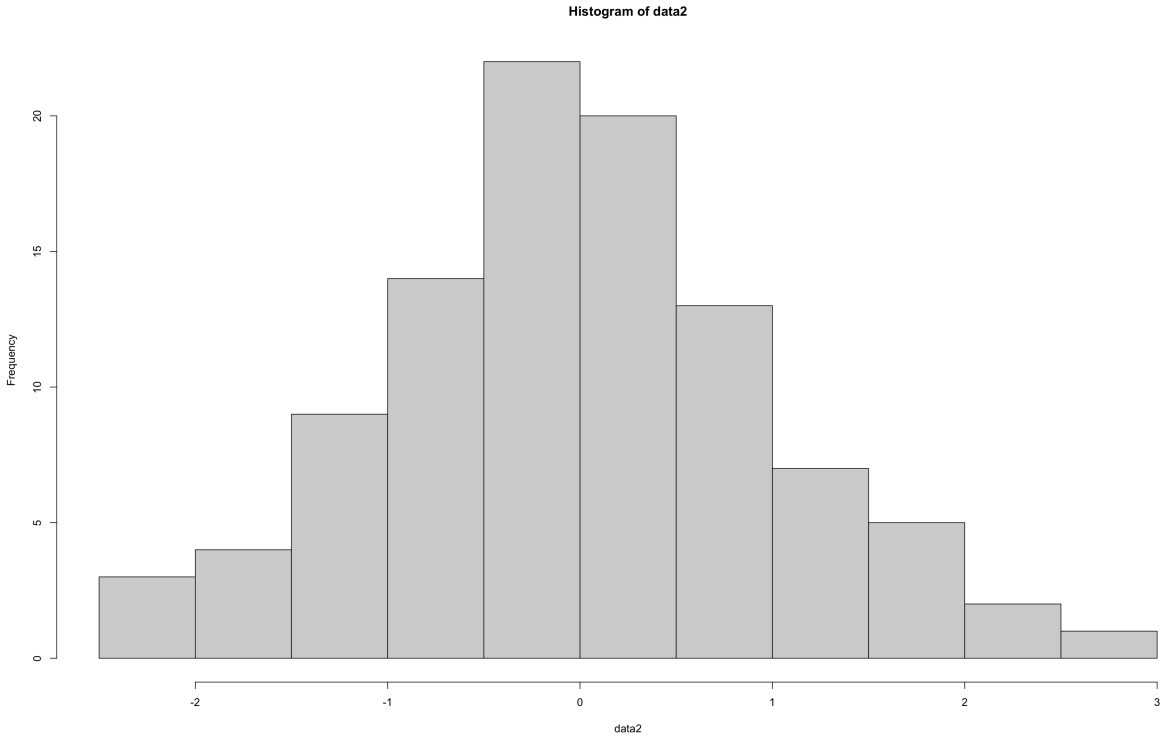
b) Matrix arranged by column

```
> matrix(data1,nrow=5, ncol=5,byrow=FALSE)
          [,1]       [,2]       [,3]       [,4]       [,5]
[1,] -0.44577826 -0.38548930 -0.9824278  0.17005748  2.6700717
[2,] -1.20585657 -0.47586788 -0.5544887  0.15507872 -1.2270160
[3,]  0.04112631  0.71975069  0.1213812  0.02493187  0.8342473
[4,]  0.63938841 -0.01850562 -0.1208723 -2.04658541  0.5325717
[5,] -0.78655436 -1.37311776 -1.3360410  0.21315411 -0.6468250
```

**(3) Similarly, generate another vector with 100 elements and plot its histogram.**

**Solution:**

```
> set.seed(100)
> data2 = rnorm(100,mean=0,sd=1)
> data2
 [1] -0.50219235  0.13153117 -0.07891709  0.88678481  0.11697127  0.31863009 -0.58179068  0.71453271
 [9] -0.82525943 -0.35986213  0.08988614  0.09627446 -0.20163395  0.73984050  0.12337950 -0.02931671
[17] -0.38885425  0.51085626 -0.91381419  2.31029682 -0.43808998  0.76406062  0.26196129  0.77340460
[25] -0.81437912 -0.43845057 -0.72022155  0.23094453 -1.15772946  0.24707599 -0.09111356  1.75737562
[33] -0.13792961 -0.11119350 -0.69001432 -0.22179423  0.18290768  0.41732329  1.06540233  0.97020202
[41] -0.10162924  1.40320349 -1.77677563  0.62286739 -0.52228335  1.32223096 -0.36344033  1.31906574
[49]  0.04377907 -1.87865588 -0.44706218 -1.73859795  0.17886485  1.89746570 -2.27192549  0.98046414
[57] -1.39882562  1.82487242  1.38129873 -0.83885188 -0.26199577 -0.06884403 -0.37888356  2.58195893
[65]  0.12983414 -0.71302498  0.63799424  0.20169159 -0.06991695 -0.09248988  0.44890327 -1.06435567
[73] -1.16241932  1.64852175 -2.06209602  0.01274972 -1.08752835  0.27053949  1.00845187 -2.07440475
[81]  0.89682227 -0.04999577 -1.34534931 -1.93121153  0.70958158 -0.15790503  0.21636787  0.81736208
[89]  1.72717575 -0.10377029 -0.55712229  1.42830143 -0.89295740 -1.15757124 -0.53029645  2.44568276
[97] -0.83249580  0.41351985 -1.17868314 -1.17403476
```



**(4) Explain the plots in your own words.**

The histogram is plotted for the data2 values to the frequency. The mean of the data is almost zero in the histogram. The histogram is plotted with class intervals of 0.5 units. The data ranges from -2.5 to 3, hence the histogram makes use of 3-(-2.5) /2 =. 11 class intervals of 0.5 units each. Most number of values are between -0.5 to +0.5, with the mean of the data being 0.

**2. Upload 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))**

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

#A data frame with 392 observations on the following 9 variables.

#`Mpg` (miles per gallon)
#`Cylinders` (Number of cylinders between 4 and 8)
#`Displacement` (Engine displacement (cu. inches)
#`Horsepower` (Engine horsepower)
#`Weight` (Vehicle weight (lbs.))
#`Acceleration` (Time to accelerate from 0 to 60 mph (sec.))
#`Year` (Model year (modulo 100))
#`Origin` (Origin of car (1. American, 2. European, 3. Japanese))
#`Name` (Vehicle name)

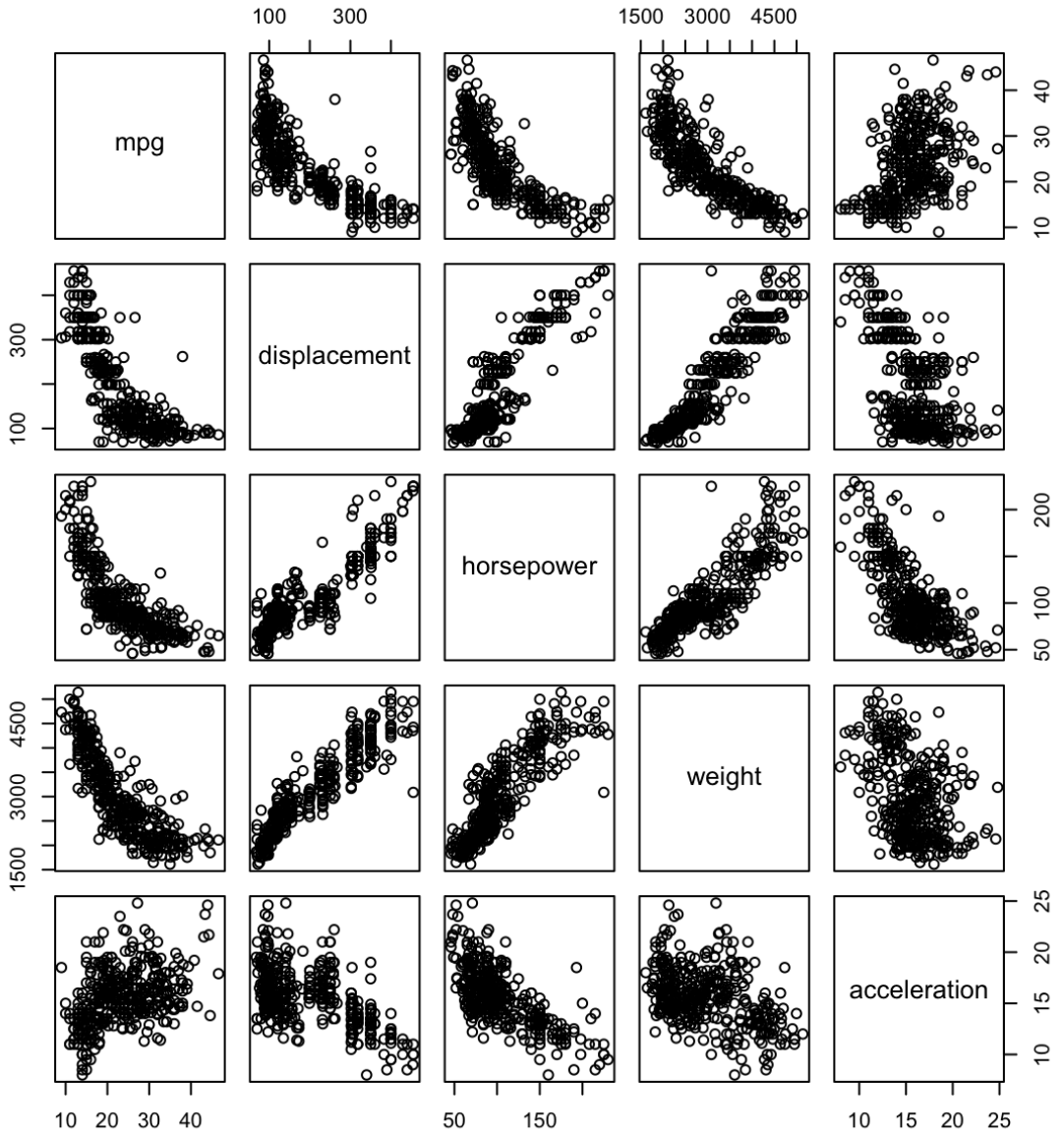
**3. Make a scatterplot between any two of the following variables (try to plot all scatterplots in one figure; hint: use pairs() command): "mpg", "displacement", "horsepower", "weight", "acceleration". By observing the plots, do you think the two variables in each scatterplot are *correlated*? If so, how?**

```
> pairs(Auto[,c(1,3,4,5,6)])
#selecting the variables from the data which are to be plotted

> cor(Auto[,c(1,3,4,5,6)])
                mpg displacement horsepower     weight acceleration
mpg          1.0000000   -0.8051269 -0.7784268 -0.8322442    0.4233285
displacement -0.8051269    1.0000000  0.8972570  0.9329944   -0.5438005
horsepower   -0.7784268    0.8972570  1.0000000  0.8645377   -0.6891955
weight       -0.8322442    0.9329944  0.8645377  1.0000000   -0.4168392
acceleration  0.4233285   -0.5438005 -0.6891955 -0.4168392    1.0000000
```

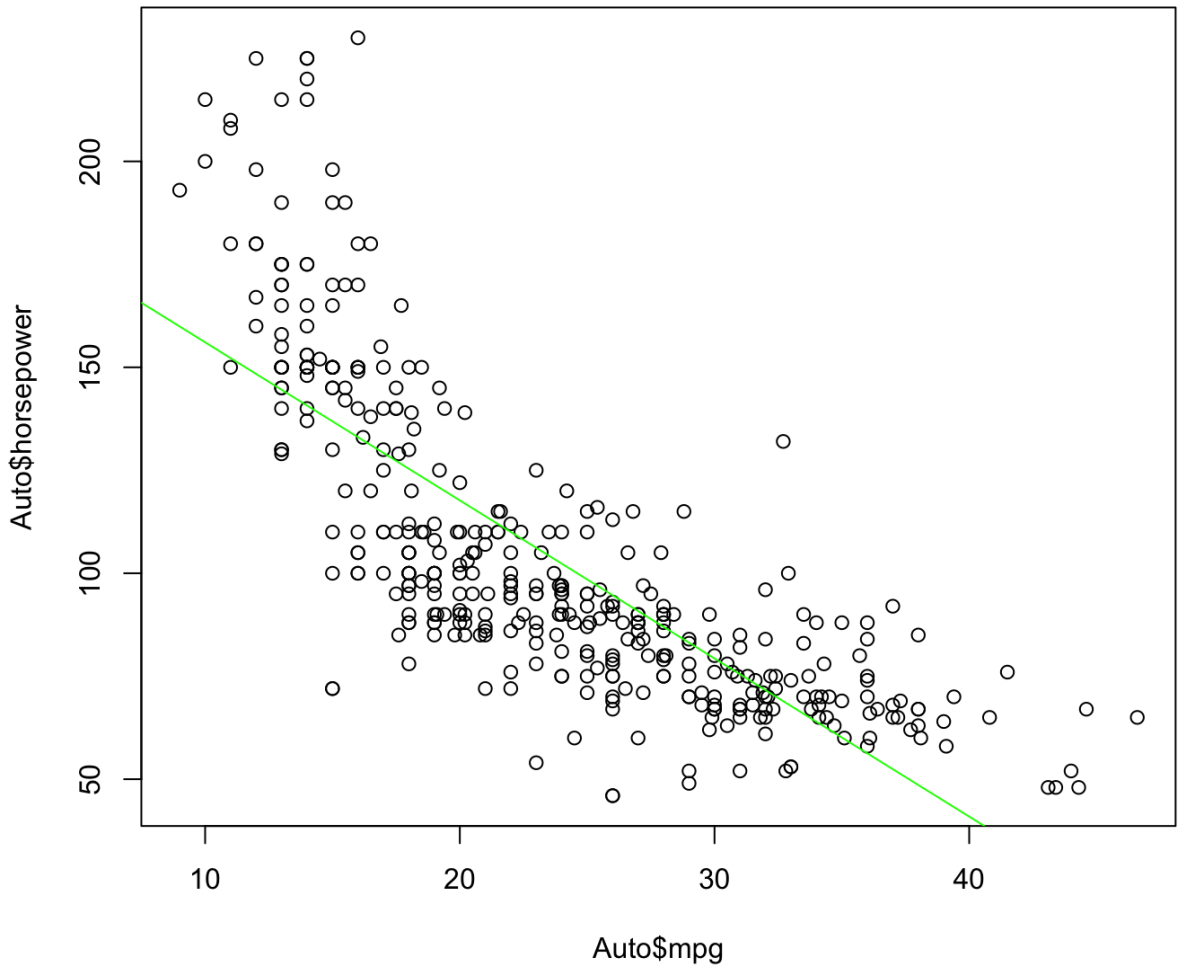
#A positive correlation (ex. Horsepower vs displacement) means the variables increase or decrease together; A negative correlation (ex. Mpg vs displacement) means if one variable increases the other decreases. So, looking at the above results and the scatter plots we can say that while some pairs have positive correlation, and some have negative correlation.

Displacement and weight have a very high positive correlation, mpg and weight have the highest negative correlation. Weight vs acceleration and mpg vs acceleration have the least correlations, it can be inferred from the plots and the correlation values.



**4. Draw a line on the scatterplot of mpg vs. horsepower to represent relationship between the two variables. (Hint: Search for "R Plot Line in Scatterplot)**

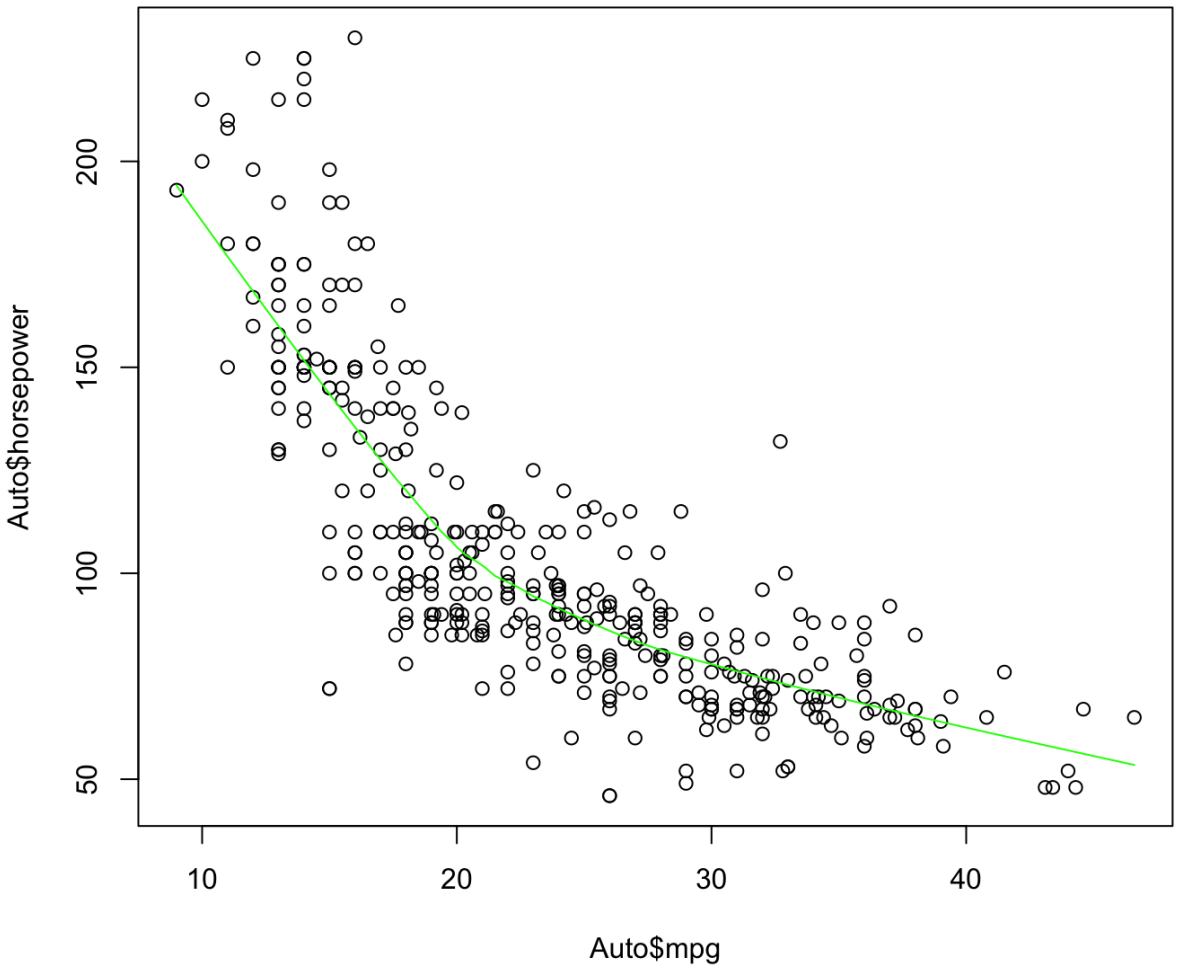
```
> plot(Auto$mpg,Auto$horsepower)
> abline(lm(horsepower ~ mpg, data = Auto), col = "green")
```



**5. Is there a better way to represent their relationship rather than the linear model you just drew? (No need to use mathematical formula. Just draw something on the figure)**

*> plot(Auto$mpg,Auto$horsepower)*

*> lines(lowess(Auto$mpg,Auto$horsepower), col = "green")*




#Yes, as the data is not following a linear pattern, a higher order equation can be used to better fit the data between mpg and horsepower. For example the lowess fit as shown above.