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);

Below screen shot shows the code to generate a random normal distribution with mean = 0 and sd = 1

```
> x<-rnorm(25, mean = 0, sd = 1)
> x
 [1] -2.52665091  0.35305752  2.16162627  1.04957598  0.98175334  1.03703786
 [7] -1.48244638  1.70160227  0.69773325  0.06486547 -0.04144346 -1.25631532
[13] -1.01017916  1.29820834  0.02050608 -0.01751723 -1.00995929  0.57670298
[19] -1.13447859 -0.16431072  0.76168275 -0.32346163  1.80257667 -0.92718806
[25] -0.23880424
```

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

By row:

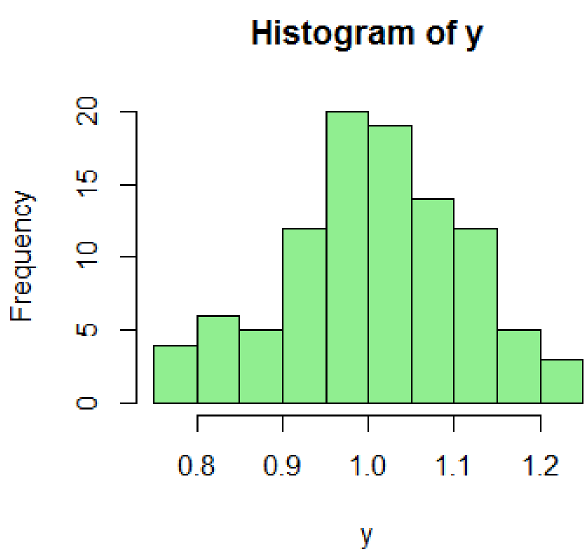
```
> matrix(data = x, nrow = 5, ncol = 5, byrow = T)
            [,1]       [,2]      [,3]       [,4]        [,5]
[1,] -2.52665091  0.3530575  2.161626  1.0495760  0.98175334
[2,]  1.03703786 -1.4824464  1.701602  0.6977332  0.06486547
[3,] -0.04144346 -1.2563153 -1.010179  1.2982083  0.02050608
[4,] -0.01751723 -1.0099593  0.576703 -1.1344786 -0.16431072
[5,]  0.76168275 -0.3234616  1.802577 -0.9271881 -0.23880424
```

By column:

```
> matrix(data = x, nrow = 5, ncol = 5, byrow = F)
           [,1]        [,2]        [,3]        [,4]       [,5]
[1,] -2.5266509  1.03703786 -0.04144346 -0.01751723  0.7616827
[2,]  0.3530575 -1.48244638 -1.25631532 -1.00995929 -0.3234616
[3,]  2.1616263  1.70160227 -1.01017916  0.57670298  1.8025767
[4,]  1.0495760  0.69773325  1.29820834 -1.13447859 -0.9271881
[5,]  0.9817533  0.06486547  0.02050608 -0.16431072 -0.2388042
```

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

```
> y<-rnorm(100, mean = 1, sd = 0.1)
> z<-hist(y, col="lightgreen")
```



(4) The above plot is a histogram of the frequencies of the vectors generated using *rnorm*. It can be observed that the histogram is bell shaped with mean as 1 and sd as 0.1.

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))

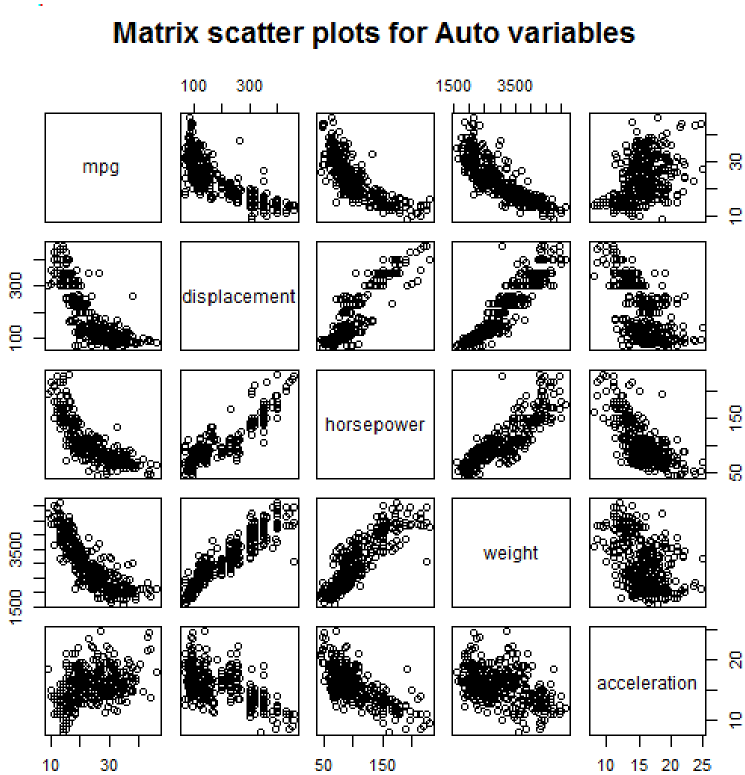
After installing the package ISLR, we load the ISLR library which contains the *Auto* dataset. Below screenshot shows the variables in the dataset.

```
> data(Auto)
> names(Auto)
[1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"
[6] "acceleration" "year"         "origin"       "name"
```

3. Make a scatterplot between 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?

The code to create the matrix of scatterplot is:

```
> pairs(~ mpg + displacement + horsepower + weight + acceleration, Auto,
+ main = "Matrix scatter plots for Auto variables")
```



From the matrix plot, we can see correlation between the various variables. We can see correlation in the below table.

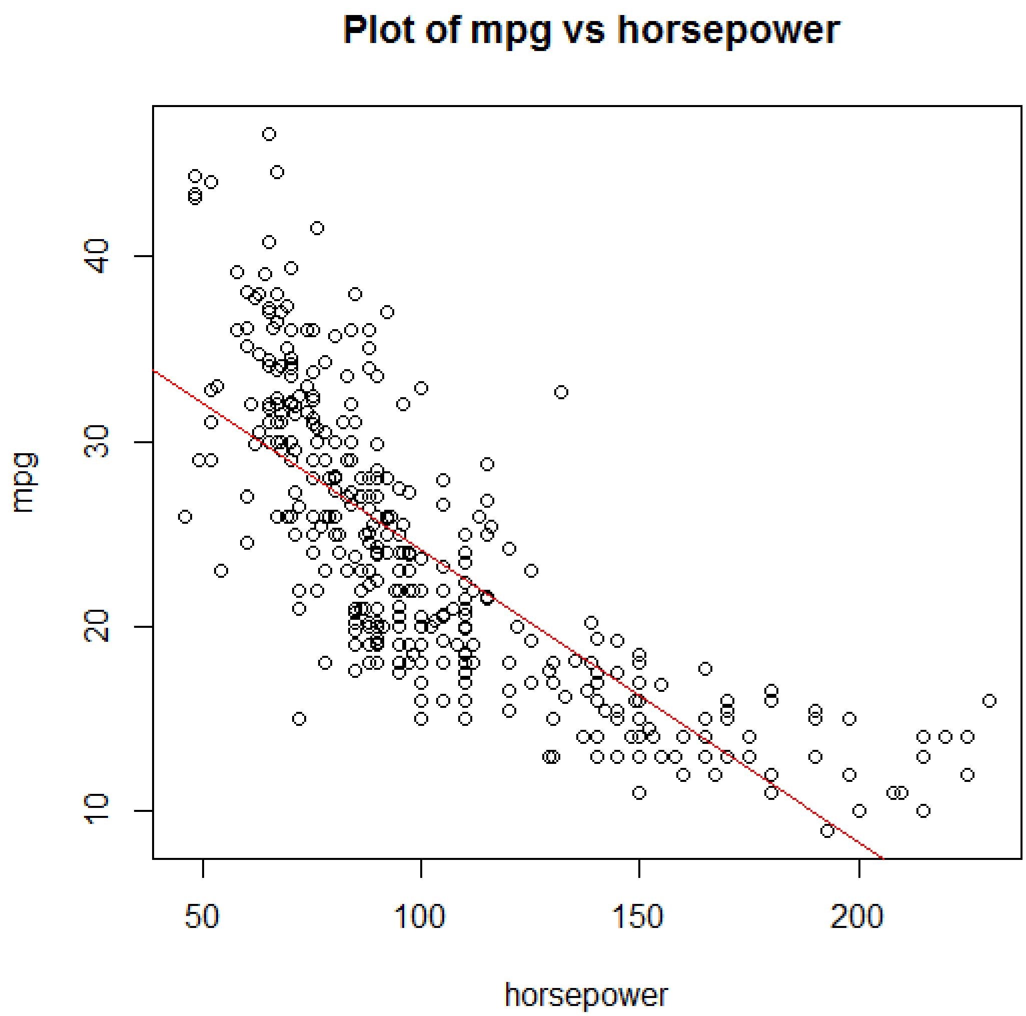
Correlation between variables	Type of correlation
mpg vs displacement	Negative
horsepower vs mpg	Negative
mpg vs weight	Negative
horsepower vs weight	Positive
horsepower vs acceleration	Negative
displacement vs weight	Positive
displacement vs horsepower	Positive

There are two types of correlation, positive and negative correlation. A positive correlation occurs when increase in one variable increases the other variable increases. A negative correlation occurs when increase in one variable increases the other variable decreases.

4. **Draw a line on the scatterplot of mpg vs. horsepower to represent relationship between the two variables.**

   Below is the code to create a scatterplot and a line to fit the points between mpg and horsepower:

```
> plot(Auto$horsepower,Auto$mpg,xlab="horsepower",ylab="mpg",
+ main=" Plot of mpg vs horsepower")
> abline(lm(Auto$mpg~Auto$horsepower), col="red") # regression line (y~x)
```



The red line is the linear fit for the points between mpg and horsepower. This line fits the points in accordance to the linear regression.

5. **Is there a better way to represent their relationship rather than the linear model you just drew?**

The blue line is a better way to represent the linear model. The blue line corresponds to a polynomial regression between the two variables.