Assignment #1: Learning R

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);

```
> x <- c(rnorm(25,mean = 0, sd = 1))
> x
 [1] -2.176195697 -1.273559708 -2.153812846 -0.647352745  0.254992228  0.349052913 -0.483227471
 [8] -0.846714466  1.110071427  1.734635674 -2.104590979 -0.012113859 -1.121084094 -0.821566061
[15] -0.811302374 -0.736196404 -0.188993750 -0.722942271 -0.075694155 -0.009242935 -0.208302404
[22] -0.518254785  0.290460909 -0.036842213  0.226055919
>
```

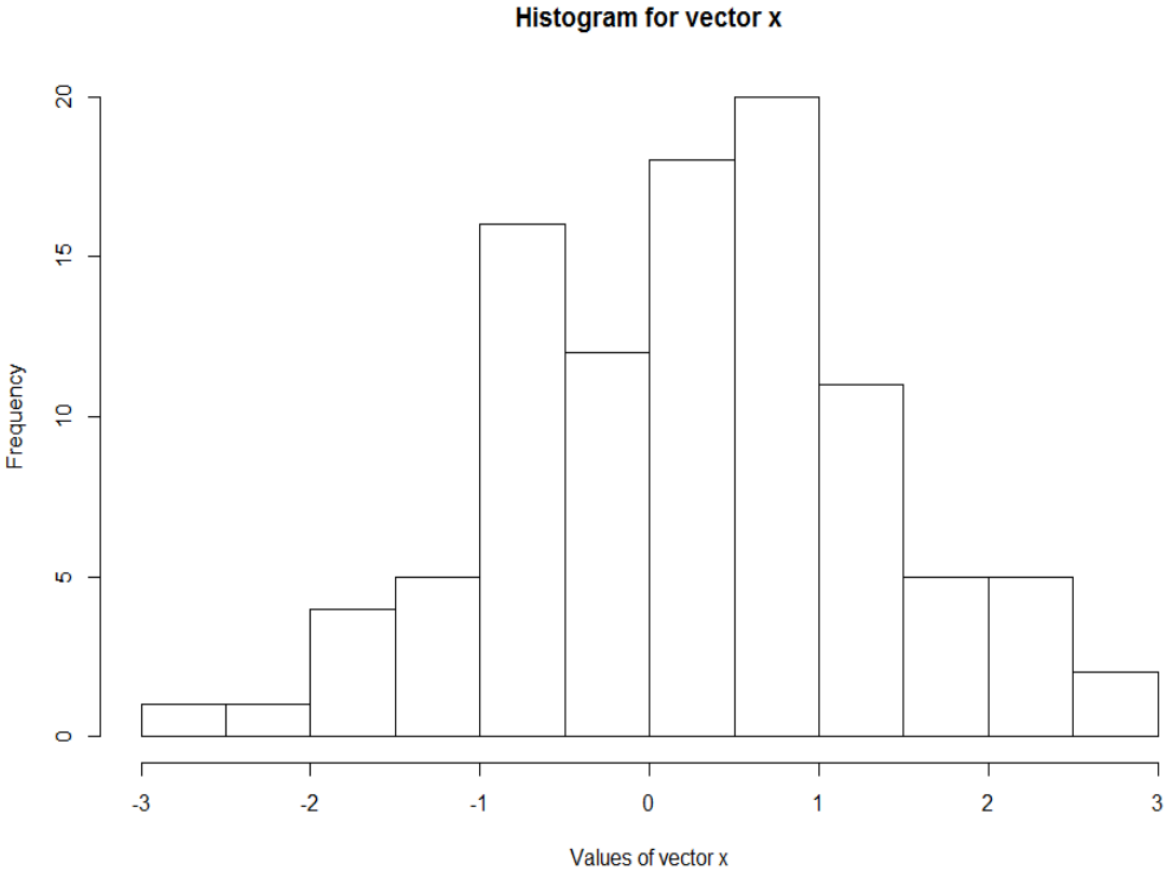
(2) Reshape this vector into a 5 by 5 matrix in two ways (arranged by row and column);

```
> matrix(x, nrow = 5, ncol = 5)
           [,1]       [,2]        [,3]         [,4]        [,5]
[1,] -2.1761957  0.3490529 -2.10459098 -0.736196404 -0.20830240
[2,] -1.2735597 -0.4832275 -0.01211386 -0.188993750 -0.51825478
[3,] -2.1538128 -0.8467145 -1.12108409 -0.722942271  0.29046091
[4,] -0.6473527  1.1100714 -0.82156606 -0.075694155 -0.03684221
[5,]  0.2549922  1.7346357 -0.81130237 -0.009242935  0.22605592
> matrix(x, nrow = 5, ncol = 5, byrow = TRUE)
           [,1]        [,2]       [,3]        [,4]         [,5]
[1,] -2.1761957 -1.27355971 -2.1538128 -0.64735275  0.254992228
[2,]  0.3490529 -0.48322747 -0.8467145  1.11007143  1.734635674
[3,] -2.1045910 -0.01211386 -1.1210841 -0.82156606 -0.811302374
[4,] -0.7361964 -0.18899375 -0.7229423 -0.07569416 -0.009242935
[5,] -0.2083024 -0.51825478  0.2904609 -0.03684221  0.226055919
>
```

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

```
> x <- c(rnorm(100, mean = 0, sd =1 ))
> x
  [1]  2.39691624 -0.83947061 -0.05692103  0.08247683  0.56553971 -0.35224044 -0.50014198
  [8] -0.87313886 -2.51815156  1.11755037  1.52259604  0.22200819  1.32161672  0.55778214
 [15]  1.66401322 -0.52382995  1.32929575  0.02465134  1.40286370 -0.37306335 -0.10519968
 [22]  2.25296258  2.25742690  0.03380568 -0.11664167 -0.32359582  1.13505446  1.22567731
 [29]  1.46903804  0.79600199  1.38957670 -0.81772754  1.17881010  1.56954549  2.39102381
 [36] -0.20034237 -0.95845635  2.24750392 -0.65484017 -1.84728827  0.40498003  0.96992758
 [43] -1.36705830 -0.93245809  1.16070601  0.53884112 -1.07634501  0.09761738  2.56813620
 [50] -0.60816461  0.03978130  0.36157682  1.28057671  0.19750402  0.71014820  0.78551674
 [57] -0.15185780  0.05223915  0.51468629  0.47406112 -0.47289783  0.99157441  0.46134239
 [64]  0.17102674 -1.08669872 -0.27218703  0.23912655  2.67726633  0.98456713  0.09356256
 [71] -0.70752327  0.86092632  0.28354841  0.68240811  1.53072562  0.99206907 -1.48124123
 [78]  0.87430319  1.84553467 -2.33164384 -0.29363163 -1.95234061 -0.80162670 -1.51594595
 [85] -0.86295431  0.74711630 -1.96856279  0.21835063  0.94921252  0.98437551  0.85491951
 [92] -0.79100920 -1.03234066  0.43482845  0.74951117 -0.48582246  0.89623512 -0.71410261
 [99] -0.88812623 -0.50179369
> hist(x)
>
```

```
> hist(x, xlab = "Values of vector x", main = "Histogram for vector x")
>
```



(4) Provide screenshots of the R code used for the above questions as well as the plots in the report. Explain the plots in your own words.

Screenshots of R code has been provided above for each question.

Explanation for plot: The plot is a bell-shaped curve as we have taken the matrix to be of normal distribution. Approx. 68% entries are in the -1σ to 1σ. 27%entries are in -2σ to 2σ range and the rest are in the -3σ to 3σ. This follows the standard normal distribution.

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)
> names(Auto)
[1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"       "acceleration"
[7] "year"         "origin"       "name"
> ?Auto
> dim(Auto)
[1] 392   9
> |
```

By this data we understood that Auto dataset contains 9 columns which are described with the command names(Auto).

Also, this dataset contains a total of 392 observations as evident from the dimension command operated om the dataset.

Meta data regarding the data set is generated using the ?Auto command:

Auto Data Set

Description

Gas mileage, horsepower, and other information for 392 vehicles.

Usage

```
Auto
```

Format

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

The orginal data contained 408 observations but 16 observations with missing values were removed.

Source

This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University. The dataset was used in the 1983 American Statistical Association Exposition.

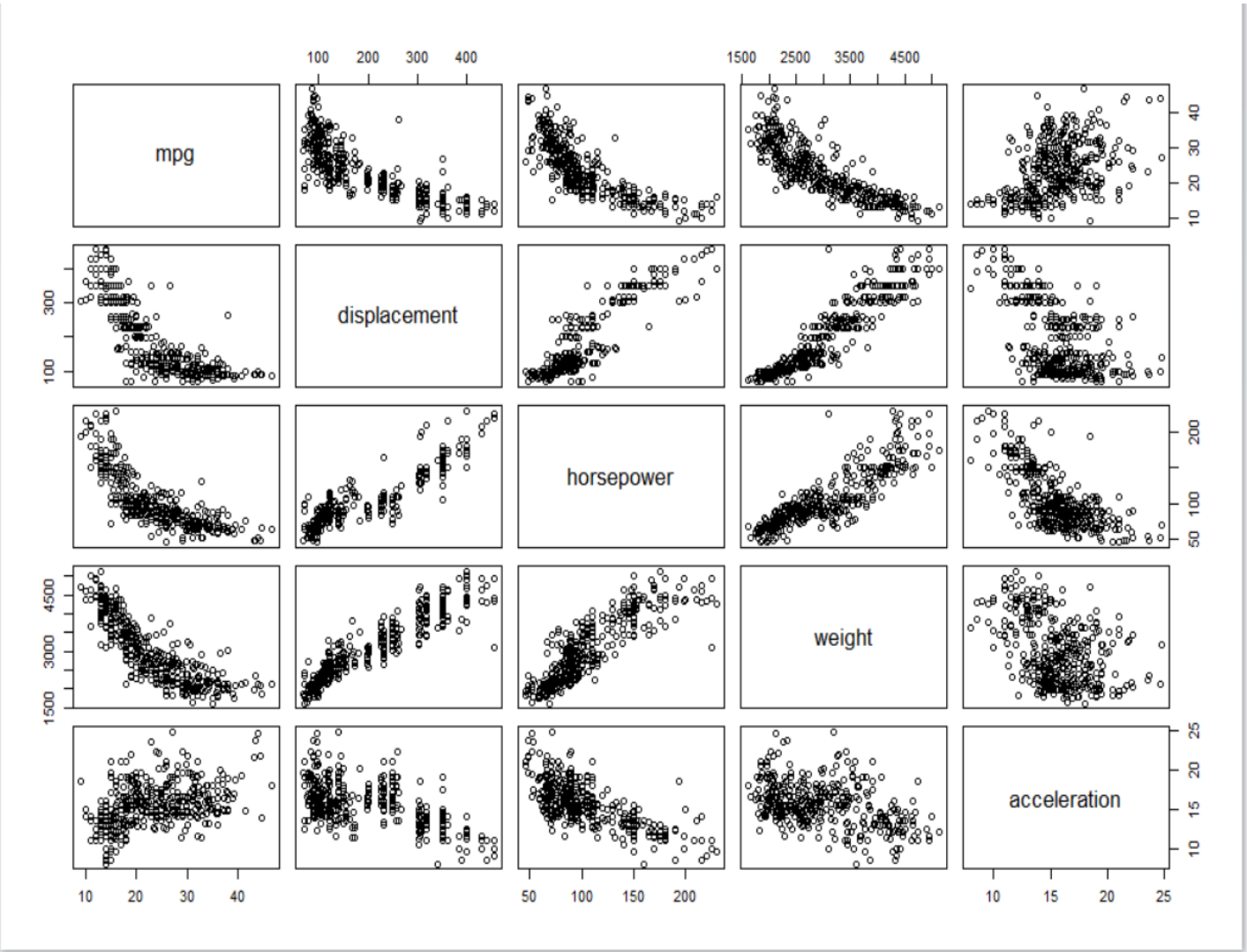
References

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013) *An Introduction to Statistical Learning with applications in R*, www.StatLearning.com, Springer-Verlag, New York

Examples

```
pairs(Auto)
attach(Auto)
hist(mpg)
```

3. Make a scatterplot between every pair 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?

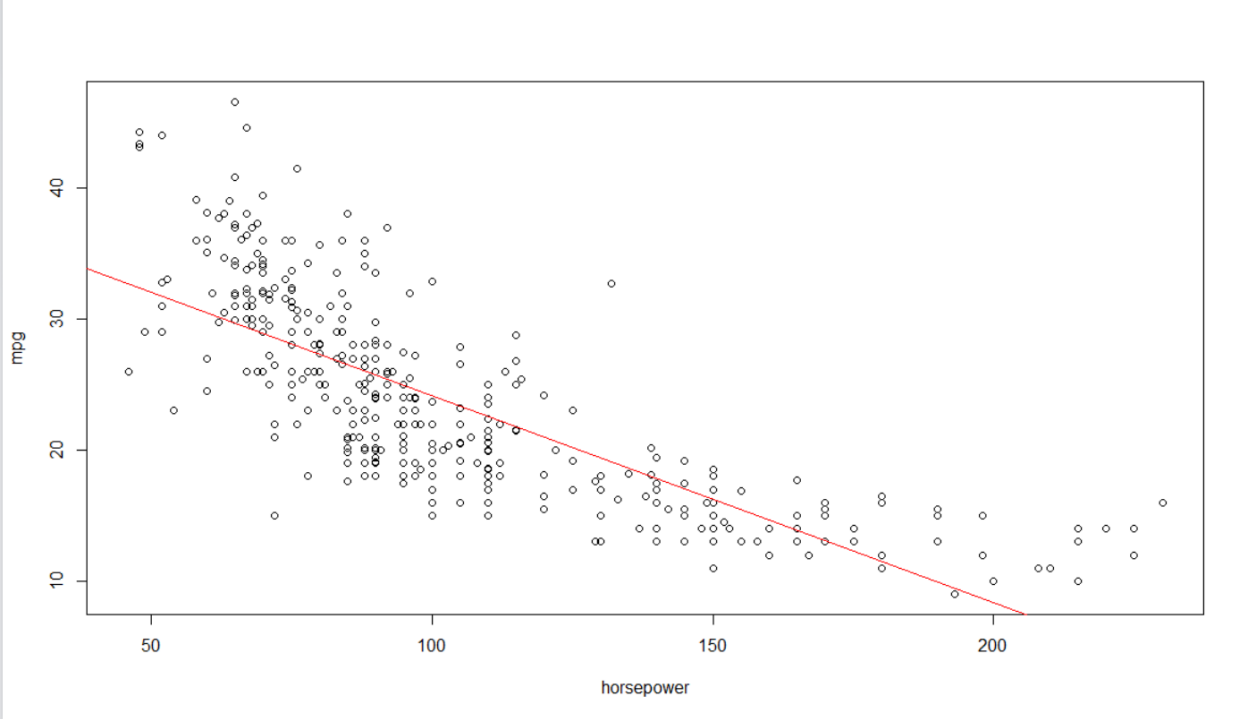


Relationships:

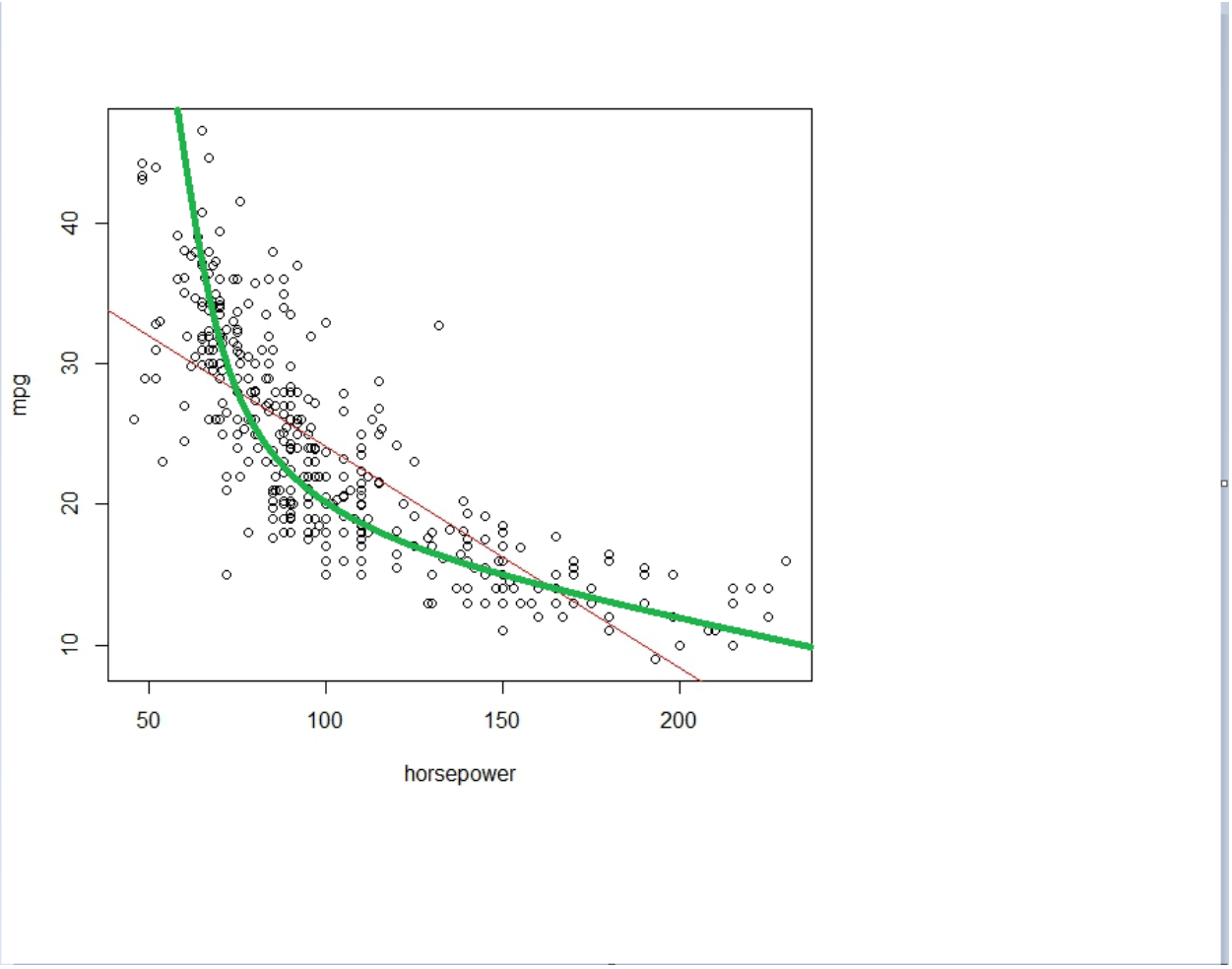
	Displacement	**Horsepower**	**Weight**	**Acceleration**
Mpg	Linear – negative	Linear – negative	Linear(skewed) – negative	No correlation. Scattered graph.
Displacement	X	Linear – Positive	Linear – Positive	No correlation. Scattered graph.
Horsepower	X	X	Linear – Positive	Very skewed linear - Negative.
Weight	X	X	X	No correlation. Scattered graph.

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

```
>
> fit <- lm(mpg~horsepower, data = Auto)
> plot(mpg~horsepower, data = Auto)
> abline(fit,col = 'Red')
>
```



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)



The plot is concentrated between horsepower range of ~60 - ~120. Hence linear model won't fit in properly for this data set. The plot in green looks close to what should be the actual fit, where it bends in the range of 60 -120 horsepower.