1. a. Positive number was randomly selected as 15 and below is the R code (and the result) to get the log of the number:

> x<- c(15)

> log(x)

[1] 2.70805

b. Default base for log function is 10. Below is the log calculated with base 2:

> x <-(15)

> log(x,2)

[1] 3.906891

c. Log of a negative number give the below result. This is because the base to the power anything would not return a negative number and hence we get ‘Not a Number’ warning message.

> x <-c(-15)

> log(x)

[1] NaN

Warning message:

In log(x) : NaNs produced

d. Positive number selected was 10. Below is the code to get the square root:

> x <- c(10)

> sqrt(x)

[1] 3.162278

1. a. Below is the code to generate a vector with 15 random numbers and calculate mean and SD:

> x <- runif(n = 15, min = 1, max = 100)

> mean(x)

[1] 49.14144

> sd(x)

[1] 34.02246

b. Below is the code to change the mean to 10 and SD to 2 and regenerate the vector & calculate the mean and SD :

> XM= mean(x)

> XM

[1] 49.14144

> XM <- 10 #assign 10 to mean

> XM

[1] 10

> XSD=sd(x)

> XSD<-2 #assign 2 to standard deviation

> XSD

[1] 2

> x <- runif(n = 15, min = 1, max = 100)

> x

[1] 52.787672 74.250322 1.140588 51.538921 49.626590 54.024468 78.697971

[8] 36.340003 84.559949 97.088440 51.934777 78.510706 7.306501 3.479953

[15] 22.205494

> mean(x)

[1] 49.56616

> sd(x)

[1] 30.46877

C. Changing the values of the mean and standard derivation variables , XM & XSD in this case to 10 and 2 doesn’t really change the calculated mean and standard deviation of the vector.

1. C. Below is the code to create weight and height vector in r. w denotes weight vector and h denotes height vector.

> w <-c(60, 72, 57, 90, 95, 72)

> w

[1] 60 72 57 90 95 72

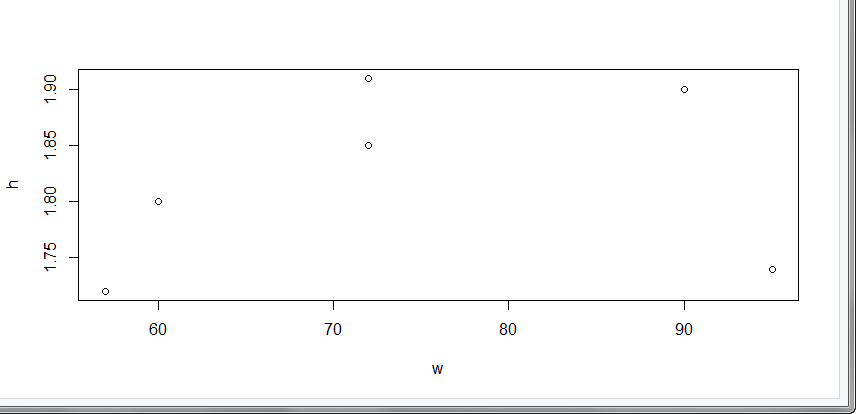
> h <-c(1.80, 1.85, 1.72, 1.90, 1.74, 1.91)

> h

[1] 1.80 1.85 1.72 1.90 1.74 1.91

d. Plot scatterplot of weight vs height (w vs h)

plot(w,h)



Looking at the above scatter plot, it can be inferred to some extent that as height increases the weight tends to increase. However, there is an exception where someone short ( h=1.74 m) may be over weight (w=95 kg).

e. Below code calculates BMI :

> hs= h^2

> hs

[1] 3.2400 3.4225 2.9584 3.6100 3.0276 3.6481

> bmi=w/hs

> bmi

[1] 18.51852 21.03725 19.26717 24.93075 31.37799 19.73630

> bmi=w/hs # bmi

> bmi

[1] 18.51852 21.03725 19.26717 24.93075 31.37799 19.73630

f. Mean of weight

> mean(w)

[1] 74.33333

g. Subtracting mean from weight

> XM=mean(w) #assign mean of weight to variable XM

> w-XM #mean minus XM

[1] -14.333333 -2.333333 -17.333333 15.666667 20.666667 -2.333333

h. Summing the result from g.

> sum(w-XM)

[1] 2.842171e-14

1. Below is the code for DS profile :

> Neha <- data.frame("Seq"=1:7, "category"=c("computer programming", "math", "statistics", "machine learning", "domain expertise", "communication and presentation skills", "data visualization"), "Ranking"=c(2,3,2,1,2,4,4)) #Create dataframe for category and ranking

> Neha

Seq category Ranking

1 1 computer programming 2

2 2 math 3

3 3 statistics 2

4 4 machine learning 1

5 5 domain expertise 2

6 6 communication and presentation skills 4

7 7 data visualization 4

> library(ggplot2) #import ggplot2 for bar graph

Registered S3 methods overwritten by 'ggplot2':

method from

[.quosures rlang

c.quosures rlang

print.quosures rlang

> p<-ggplot(Neha, aes(category,Ranking)) #create bar chart

> p +geom\_bar(stat = "identity")

