MSDS 6306: Live session Unit 01 Assignment

1. Basic Math
   1. Log of a positive number.

|  |
| --- |
| R Code |
| # Basic Math  # a. The log of a positive number  pos\_num = log(10) # the default of log is exponental log.  pos\_num # display exp log of 10. |
| Screen Capture |
| /Users/giovanni/Desktop/Screen Shot 2018-05-14 at 6.36.06 PM.png |

* 1. The default base for log() is exponential, the following example will be log(10) in base 10.

|  |
| --- |
| R Code |
| # Basic Math  # b. The log of a positive number base 10  pos\_num = log(10, base=10) # the log base 10.  pos\_num # display log base 10 of 10. |
| Screen Capture |
| /Users/giovanni/Desktop/Screen Shot 2018-05-14 at 6.45.49 PM.png |

* 1. The log of negative 10 is:

|  |
| --- |
| R Code |
| # Basic Math  # c. The log of a negative number base 10  pos\_num = log(-10, base=10) # the log base 10.  pos\_num # display log base 10 of -10. |
| Screen Capture |
| /Users/giovanni/Desktop/Screen Shot 2018-05-14 at 6.49.19 PM.png |
| The logarithmic function approaches zero but never attains it and is never negative. |

* 1. The square root of 25.

|  |
| --- |
| R Code |
| #  # d. The square root of 25  sqrt25 = sqrt(25) # square root function  sqrt25 # display squqre root of 24 |
| Screen Capture |
| /Users/giovanni/Desktop/Screen Shot 2018-05-14 at 6.59.25 PM.png |

1. Random Number Generations
   1. Here is a vector of 15 standard normal random variables, including the mean and standard deviation of this vector.

|  |
| --- |
| R Code |
| # a. Generate a vector of 15 standard normal random variables,  # including the mean and standard deviation of this vector.  my\_vec = rnorm(15) # generate 15 normally distributed random numbers  my\_vec # display resulting vector  mean(my\_vec) #display mean  sd(my\_vec) #display standard deviation |
| Screen Capture |
| Screen%20Shot%202018-05-14%20at%208.23.51%20PM.png |

* 1. Changing the mean to 10 and the standard deviation to 2.

|  |
| --- |
| R Code |
| # b. Generate a vector of 15 standard normal random variables,  # with a mean of 10 and sd of 2.  # including the mean and standard deviation of this vector.  my\_vec = rnorm(15, mean=10, sd=2) # generate 15 normally distributed random numbers  my\_vec # display resulting vector  mean(my\_vec) #display mean  sd(my\_vec) #display standard deviation |
| Screen Capture |
| Screen%20Shot%202018-05-14%20at%208.27.25%20PM.png |

* 1. The mean and the standard deviation may not be close to the mean and sd we selected in the rnorm() function due to the small sample of 15. If this is reprocessed with a larger sample size, for example 1000 it will approx. the actual 10 and 2. See this sample run

|  |
| --- |
| R Code |
| # C. Generate a vector of 1000 standard normal random variables,  # with a mean of 10 and sd of 2.  # including the mean and standard deviation of this vector.  my\_vec = rnorm(1000, mean=10, sd=2) # generate 15 normally distributed random numbers  #my\_vec # display resulting vector  mean(my\_vec) #display mean  sd(my\_vec) #display standard deviation |
| Screen Capture |
| Screen%20Shot%202018-05-14%20at%208.35.24%20PM.png |

1. Vector Operations.
   1. Weights of 6 Individuals are 60 kg, 72 kg, 57 kg, 90 kg, 95 kg, and 72 kg
   2. Heights of the 6 individuals in meters are 1.80, 1.85, 1.72, 1.90, 1.74, 1.91.
   3. Create these vectors in R Language
   4. Create a scatterplot of weight vs. Height. How do we interpret the scatterplot
   5. Calculate the BMI for each individual. (BMI = weight/height^2
   6. Calculate the mean weight
   7. Subtract the mean from each value of weight.
   8. Sum the results. Now you know why we square the deviations from the mean to determine the standard deviation.

|  |
| --- |
| R CODE |
| #3. Vector Operations  # a. Weights of 6 Individuals are 60 kg, 72 kg, 57 kg, 90 kg, 95 kg, and 72 kg  # b. Heights of the 6 individuals in meters are 1.80, 1.85, 1.72, 1.90, 1.74, 1.91.  # c. Create these vectors in R Language  wgt = c(60,72,57,90,95,72) # vector of weights  wgt  hgt = c(1.80,1.85,1.72,1.90,1.74,1.91) # vector of heights  hgt  # d. Create a scatterplot of weight vs. Height. How do we interpret the scatterplot  plot(wgt, hgt, main="height vs weight") # simple scatterplot  # e. Calculate the BMI for each individual. (BMI = weight/height^2  BMI = wgt / hgt^2  BMI  # f. Calculate the mean weight  mean\_wgt = mean(wgt)  mean\_wgt  # g. Subtract the mean from each value of weight.  delta\_wgt = wgt - mean\_wgt  delta\_wgt  # h. Sum the results. Now you know why we square the deviations from the mean to determine the standard deviation.  sum\_delta\_wgt = sum(delta\_wgt)  sum\_delta\_wgt |
| Capture Screen |
| Screen%20Shot%202018-05-14%20at%209.00.13%20PM.png |
| Scatterplot (item d.) |
| Screen%20Shot%202018-05-14%20at%209.08.19%20PM.png |

The reason we square the delta values is to remove the negative values so we can determine the standard deviation values, otherwise if the delta’s are center around 0 then we will have a value center at zero as well.

1. Your data science profile.

|  |
| --- |
| R CODE |
| #4 Your data science profile.  # Enter your data science profile into R as a data frame with two columns.  # Call it your first name. The categories are computer programming, math,  # statistics, machine learning, domain expertise, communication and presentation skills,  # and data visualization. Your ranking for each the category 1 to 5,  # with 5 as best  #  # Create a bar graph of your data science profile.  #  categories = c("Prgmng","Math","Stats","MachLearn",  +"DomExp","Coms&Pres","DataVis")  ranking = c(5,4,4,3,3,2,3)  terry = data.frame(categories,ranking) # create the data.frame  terry # display profile  hist(terry$ranking) # create histogram |
| SCREEN CAPTURE |
| Screen%20Shot%202018-05-14%20at%209.50.00%20PM.png |
| GRAPH / PLOT |
| Screen%20Shot%202018-05-14%20at%209.50.26%20PM.png |
|  |

2. Install swirl package and complete modules 1-7.