**Ryan Paul**

**405**

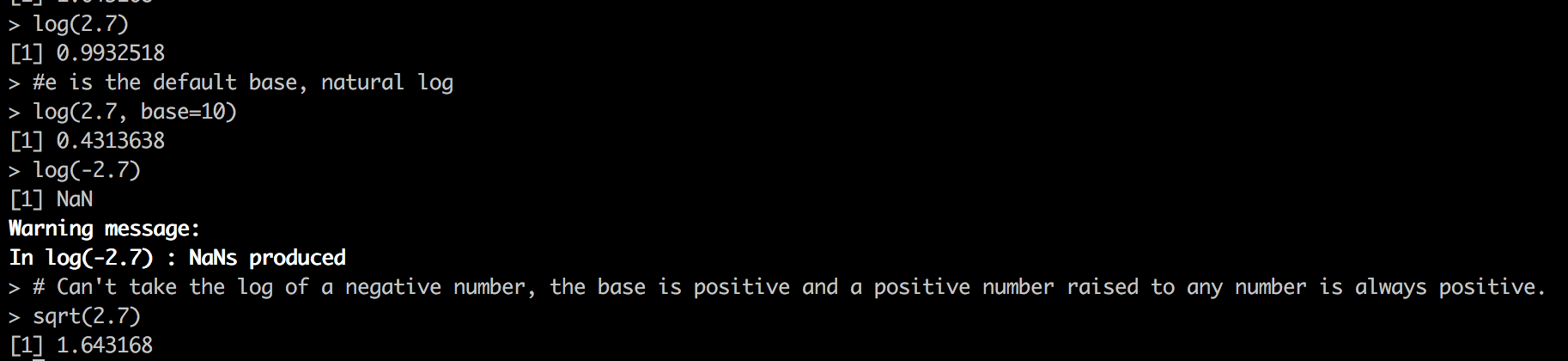
**8-30-2018**

**MSDS 6306: Doing Data Science**

# Live session Unit 01 assignment

1.

|  |
| --- |
| **a.** > log(2.71)  [1] 0.9969486  **b.** e is the default base, a natural log  > log(2.7, base=10)  [1] 0.4313638  **c.** You can't take the log of a negative number, the base is always positive and a positive number raised to any number is always positive.  > log(-2.7)  [1] NaN  **d.** > sqrt(2.7)  [1] 1.643168 |

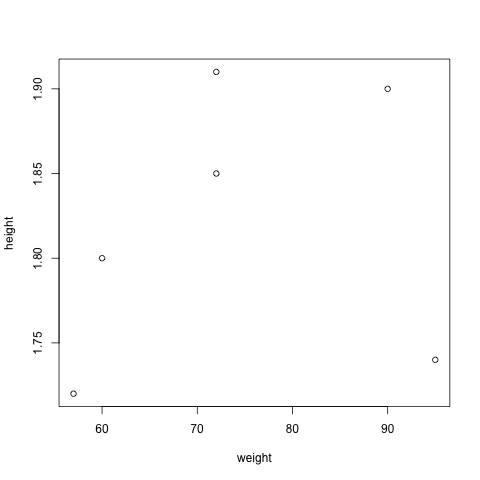
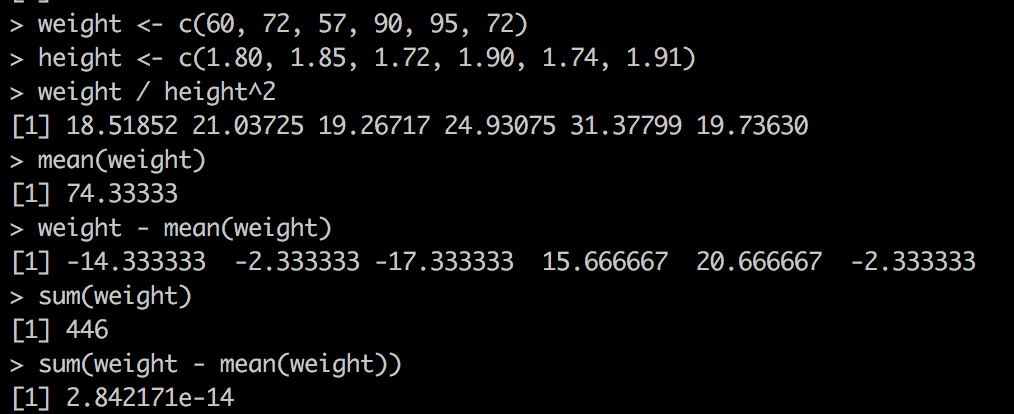


2.

|  |
| --- |
| **a.** > rnorm(15)  [1] 0.09573138 -1.28277650 1.28122483 1.53851715 0.02229286 -0.62870132  [7] -0.01615843 -2.18600691 0.97043184 0.28734783 1.69966857 -0.61100245  [13] 1.02590449 1.26379106 0.21793215  **b.** > questionb <- rnorm(15,10,2)  > mean(questionb)  [1] 9.621362  > sd(questionb)  [1] 1.795041  **c.** (I looked this up), since R is just grabbing random numbers, it can only converge toward the mean and standard deviation. It can’t draw random numbers that have an exact mean and standard deviation, you can only draw the numbers and make small corrections to the list of available numbers that can be picked as you try and create your desired population. |

3.

|  |
| --- |
| **c.** > weight <- c(60, 72, 57, 90, 95, 72)  > height <- c(1.80, 1.85, 1.72, 1.90, 1.74, 1.91)  **d.** > png(width=480,height=480,filename="health\_scatterplot.png")  > plot(weight,height)  > dev.off()  # Seems to be an upward trend of height as weight goes up  **e.** weight / height^2  [1] 18.51852 21.03725 19.26717 24.93075 31.37799 19.73630  **f.** > mean(weight)  [1] 74.33333  **g.** > weight - mean(weight)  [1] -14.333333 -2.333333 -17.333333 15.666667 20.666667 -2.333333  **h.** > sum(weight - mean(weight))  [1] 2.842171e-14 |



4.

|  |
| --- |
| Note: Maybe we can talk about how to sensibly display long names on the axis of a chart in this class.  > rating <- c(4,2,2,1,2,3,2)  > categories <- c("Computer Programming","Math","Statistics","Machine Learning","Agricultural Lending","Communication","Data Visualization")  > data\_science\_profile <- data.frame(categories, rating)  > png(filename="data\_science\_profile.png")  > barplot(data\_science\_profile$rating, names.arg = data\_science\_profile$categories, las=2, cex.names=0.4)  > dev.off() |

