Logs\_Mean\_SD\_Vector\_Operations

*Celia Taylor - Live Session Unit 2 Assignment Doing Data Science MSDS 6306, section 402, Due 5/18/2016 by 10 p.m. Central*

*May 18, 2016*

This R code calculates Logarithms, the Mean, Standard Deviations, and Vector Operations.

Calculation Questions (using R)

* Use R to calculate the following:
* The log of a positive number
* What is the default base for the log function? Calculate the log of your previous number with a different base.
* The log of a negative number (explain the answer).

Calculate the log(20). Answer is 2.995732.

## [1] 2.995732

Default base for the log function is the natural Log. Calculate the log of #####your previous number with a different base.

Executing the log(20) with base 2. Answer is 4.321928.

## [1] 4.321928

Executing the log(20) with base 10. Answer is 1.30103.

## [1] 1.30103

Executing the log of a negative number.

Executing the log of -20. Answer is NaN, which means not a number, i.e. undefined.

## Warning in log(-20): NaNs produced

## [1] NaN

Mean and Standard Deviation calculations using a Vector

* Create a vector of 15 standard normal random variables. Calculate its mean and SD.
* Change the mean to 10 and the SD to 2 and recalculate the vector of 15 random normal variables. Calculate its mean and SD.
* Why are the means and SD not exactly the same as the means and SDs specified in the function?

Create a vector of 15 standard normal random variables. Each time this command will generate a different vector of random variables, so contents and resulting calculations will be different each time.

## [1] -0.39545568 -0.07948196 -0.84801716 -0.08750076 -1.70797758

## [6] -1.22141002 1.78349838 0.62727775 -0.99226612 -1.73036077

## [11] -0.92371664 -0.57137354 1.27186948 -0.28538908 -1.10514246

Calculate the vector’s mean and Standard Deviation (sd)

## [1] -0.4176964

## [1] 1.011883

Change the mean to 10 and the SD to 2 and recalculate the vector of 15 random normal variables. Calculate its mean and SD.

## [1] 5.451872 10.511362 10.925206 7.354012 11.532404 9.971252 7.585224

## [8] 11.118228 12.234747 9.656200 10.747065 12.707651 11.028437 11.057765

## [15] 13.189542

## [1] 10.33806

## [1] 2.102245

The means and SD are not exactly the same as the means and SDs specified in the function as per <http://stackoverflow.com/questions/18919091/generate-random-numbers-with-fixed-mean-and-sd>, “When generating random numbers in R using rnorm (or runif etc.), they seldom have the exact mean and SD as the distribution they are sampled from”.

Vector Operations

* Enter these weight and height vectors into R. The weights of 6 individuals in kg are 60, 72, 57, 90, 95, 72 The heights of 6 individuals in m are 1.80, 1.85, 1.72, 1.90, 1.74, 1.91
* Create a scatterplot of weight vs. height. Interpret the scatterplot.
* Calculate the BMI for each individual (BMI = weight in kg divided by the square of the height in m)
* Calculate the mean for weight
* Subtract the mean from each value of weight
* Sum the result. Now you know why we square the deviations from the mean to calculate a standard deviation!

Enter these weight and height vectors into R.

## [1] 60 72 57 90 95 72

## [1] 1.80 1.85 1.72 1.90 1.74 1.91

Create a scatterplot of weight vs. height. Interpret the scatterplot.

The two plots show that the data is not really skewed on one side because the boxplots are fairly even with their 2nd and 3rd quartile and upper and lower tails. I am used to seeing scatterplots with dozens or even hundreds of data points for geological formations. This very small data set is not very dramatic.

Calculate the BMI for each individual (BMI = weight in kg divided by the square of the height in m)

## [1] 18.51852 21.03725 19.26717 24.93075 31.37799 19.73630

Calculate the mean for weight

Subtract the mean from each value of weight

Sum the result. Now you know why we square the deviations from the mean to calculate a standard deviation!

## [1] 74.33333

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

## [1] 2.842171e-14

Because the value is so small and approaching zero.

The code is as follows:

---

title: "Logs\_Mean\_SD\_Vector\_Operations"

author: "Celia Taylor - Live Session Unit 2 Assignment Doing Data Science MSDS 6306, section 402, Due 5/18/2016 by 10 p.m. Central"

date: "May 18, 2016"

output:

html\_document:

keep\_md: true

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

```

### This R code calculates Logarithms, the Mean, Standard Deviations, and Vector Operations.

### Calculation Questions (using R)

- Use R to calculate the following:

- The log of a positive number

- What is the default base for the log function? Calculate the log of your previous number with a different base.

- The log of a negative number (explain the answer).

#####Calculate the log(20). Answer is 2.995732.

```{r l1g1, echo = FALSE}

log(20)

```

#####Default base for the log function is the natural Log. Calculate the log of #####your previous number with a different base.

#####Executing the log(20) with base 2. Answer is 4.321928.

```{r l2g2, echo = FALSE}

log(20, 2)

```

#####Executing the log(20) with base 10. Answer is 1.30103.

```{r l3g3, echo = FALSE}

log(20, 10)

```

#####Executing the log of a negative number.

#####Executing the log of -20. Answer is NaN, which means not a number, i.e. undefined.

```{r l4g4, echo = FALSE}

log(-20)

```

###Mean and Standard Deviation calculations using a Vector

- Create a vector of 15 standard normal random variables. Calculate its mean and SD.

- Change the mean to 10 and the SD to 2 and recalculate the vector of 15 random normal variables. Calculate its mean and SD.

- Why are the means and SD not exactly the same as the means and SDs specified in the function?

#####Create a vector of 15 standard normal random variables. Each time this command will generate a different vector of random variables, so contents and resulting calculations will be different each time.

```{r V1r1, echo = FALSE}

v15StandardRdnVar <- rnorm(15)

v15StandardRdnVar

```

#####Calculate the vector's mean and Standard Deviation (sd)

```{r v2r2, echo = FALSE}

mean(v15StandardRdnVar)

sd(v15StandardRdnVar)

```

#####Change the mean to 10 and the SD to 2 and recalculate the vector of 15 random normal variables. Calculate its mean and SD.

```{r V3r3, echo = FALSE}

v15M10SD2 <-rnorm(15, mean = 10, sd = 2)

v15M10SD2

mean(v15M10SD2)

sd(v15M10SD2)

```

#####The means and SD are not exactly the same as the means and SDs specified in the function as per <http://stackoverflow.com/questions/18919091/generate-random-numbers-with-fixed-mean-and-sd>, "When generating random numbers in R using rnorm (or runif etc.), they seldom have the exact mean and SD as the distribution they are sampled from".

###Vector Operations

- Enter these weight and height vectors into R.

The weights of 6 individuals in kg are 60, 72, 57, 90, 95, 72

The heights of 6 individuals in m are 1.80, 1.85, 1.72, 1.90, 1.74, 1.91

- Create a scatterplot of weight vs. height. Interpret the scatterplot.

- Calculate the BMI for each individual (BMI = weight in kg divided by the square of the height in m)

- Calculate the mean for weight

- Subtract the mean from each value of weight

- Sum the result. Now you know why we square the deviations from the mean to calculate a standard deviation!

#####Enter these weight and height vectors into R.

```{r V4r4, echo = FALSE}

weights <- c(60, 72, 57, 90, 95, 72)

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

weights

heights

```

#####Create a scatterplot of weight vs. height. Interpret the scatterplot.

```{r V5r5, echo = FALSE}

library(car)

scatterplot(weights, heights)

plot(weights, heights)

```

#####The two plots show that the data is not really skewed on one side because the boxplots are fairly even with their 2nd and 3rd quartile and upper and lower tails. I am used to seeing scatterplots with dozens or even hundreds of data points for geological formations. This very small data set is not very dramatic.

#####Calculate the BMI for each individual (BMI = weight in kg divided by the square of the height in m)

```{r V6r6, echo = FALSE}

BMI <- (weights/(heights^2))

BMI

```

#####Calculate the mean for weight

#####Subtract the mean from each value of weight

#####Sum the result. Now you know why we square the deviations from the mean to calculate a standard deviation!

```{r V7r7, echo = FALSE}

mean(weights)

submean <- (weights - mean(weights))

submean

sum1 <- sum(submean)

sum1

```

#####Because the value is so small and approaching zero.

```{r e1t1, echo = FALSE}

#####The rest is documenting my environment as needed.

#sessionInfo()

## R version 3.2.5 (2016-04-14)

## Platform: x86\_64-w64-mingw32/x64 (64-bit)

## Running under: Windows 7 x64 (build 7601) Service Pack 1

##

## locale:

## [1] LC\_COLLATE=English\_United States.1252

## [2] LC\_CTYPE=English\_United States.1252

## [3] LC\_MONETARY=English\_United States.1252

## [4] LC\_NUMERIC=C

## [5] LC\_TIME=English\_United States.1252

##

## attached base packages:

## [1] stats graphics grDevices utils datasets methods base

##

## other attached packages:

## [1] car\_2.1-2

##

## loaded via a namespace (and not attached):

## [1] Rcpp\_0.12.4 knitr\_1.13 magrittr\_1.5

## [4] splines\_3.2.5 MASS\_7.3-45 lattice\_0.20-33

## [7] minqa\_1.2.4 stringr\_1.0.0 tools\_3.2.5

## [10] nnet\_7.3-12 pbkrtest\_0.4-6 parallel\_3.2.5

## [13] grid\_3.2.5 nlme\_3.1-125 mgcv\_1.8-12

## [16] quantreg\_5.21 MatrixModels\_0.4-1 htmltools\_0.3.5

## [19] yaml\_2.1.13 lme4\_1.1-12 digest\_0.6.9

## [22] Matrix\_1.2-4 nloptr\_1.0.4 formatR\_1.3

## [25] evaluate\_0.9 rmarkdown\_0.9.6 stringi\_1.0-1

## [28] SparseM\_1.7

```