Klein\_Lab1\_Week2

Melanie Klein

2024-01-25

1. In three or four sentences, explain why constructing a Function to execute tasks is beneficial or advantageous

A function is beneficial because it reduces redundancy and repetitive calculations. Instead of having to write the calculation multiple times for different input values, defining a function allows you to write the calculation only once and call the function with various input values countless times. Additionally, writing code for the desired task once within the function reduces the risk of errors and mistakes. Lastly, functions make the program easier to read, easier to understand, and more efficient.

1. Write a function (using r code and structure demonstrated in class) to calculate a z score for a given observed value, a mean, and a standard deviation value. And then use your function to find a z score for the following problem. (Research the internet to find the formula used to calculate a z score). Observed value = 25.77, mean = 23.54, standard deviation = 2.442

z\_score\_calculate <- function(observed, mean, stddev) {  
 z\_score = (observed - mean) / stddev  
 return(z\_score)  
}  
  
z\_score\_calculate(25.77, 23.54, 2.442)

## [1] 0.9131859

1. Write a function (using r code and the structure demonstrated in class) to calculate the natural log of a number multiplied by the common log of the same number divided by the cube root of a given prime number. Use your function to find the answer if the number to be used for both log expressions is 32 and the given prime number is 11. Also use R code to round your answer to the nearest tenth

C <- function(num, prime) {  
 x = (log(num) \* log10(num)) / (prime^(1/3))  
 x = round(x, digits = 1)  
 return(x)  
}  
  
C(32, 11)

## [1] 2.3

1. Use and show R coding to calculate the standard deviation for each variable of the data table mtcars using the “Special For Loop Method” demonstrated in the class notes.

data(mtcars)  
  
output <- vector("double", ncol(mtcars))  
for (i in seq\_along(mtcars)) {  
 output[[i]] <- sd(mtcars[[i]])  
}  
  
output

## [1] 6.0269481 1.7859216 123.9386938 68.5628685 0.5346787 0.9784574  
## [7] 1.7869432 0.5040161 0.4989909 0.7378041 1.6152000