Homework 4

Loc Le

2023-02-13

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.0 ✔ purrr 0.3.4   
## ✔ tibble 3.1.6 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.1 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(purrr)  
library(dplyr)

## 1a) Use the R function nrow to confirm that the iris data frame has 150 rows. Then use and show R code that features a map function to confirm that the iris data frame has 150 rows.

nrow(iris)

## [1] 150

map(iris, length)

## $Sepal.Length  
## [1] 150  
##   
## $Sepal.Width  
## [1] 150  
##   
## $Petal.Length  
## [1] 150  
##   
## $Petal.Width  
## [1] 150  
##   
## $Species  
## [1] 150

## 1b) Each column of the iris data frame has a unique number of values or objects. For example, the column Sepal.Length has 150 values but 35 of them are unique. Use and show R code that features a map function to find the number of unique values or objects for each column of the iris data frame.

map(iris, unique) %>%  
 map(length)

## $Sepal.Length  
## [1] 35  
##   
## $Sepal.Width  
## [1] 23  
##   
## $Petal.Length  
## [1] 43  
##   
## $Petal.Width  
## [1] 22  
##   
## $Species  
## [1] 3

## 2) Use and show R code that features a nested loop that will produce the 5 by 3 matrix shown below.

q2 = matrix(nrow = 3, ncol = 5)  
for(i in 1:nrow(q2))  
{  
 for(j in 1:ncol(q2))  
 {  
 q2[i,j] = j - i  
 }  
}  
print(q2)

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 0 1 2 3 4  
## [2,] -1 0 1 2 3  
## [3,] -2 -1 0 1 2

## 3) Use and show R code that will produce a tibble that features 10 randomly generated values that are normally distributed, with means of -10, 0, 10 and 100 respectfully. Run your code again, producing a second tibble, that confirms random values, hence the second table will not have the same values.

mu <- c(-10, 0, 10, 100)  
rnorm(10, mu)

## [1] -10.76907648 -0.33742027 8.71380163 99.19831995 -11.86099399  
## [6] 0.42283110 8.68410814 100.68739588 -11.07066766 0.04248973

rnorm(10, mu)

## [1] -10.9213905 -0.6746104 10.8655642 99.9785065 -8.9761644 -1.8981312  
## [7] 11.2302332 100.0667410 -10.0190181 -1.0574658

X <- list(12, 14, 15, 18, 19, 22,10,18,18)  
Mean <- list(16, 16, 16, 16, 16,16,16,16,16)  
sd <- list(2, 2, 2, 2, 2,2,2,2,2)

## 4a) In statistics, a z score indicates the standard deviation distance between the mean and a specific value of the data set. What formula is used to find a z score? Use and show R coding that features a map function to iteratively find z scores across the lists given above.

Z-score formula:

pmap\_dbl(list(X, Mean, sd), function(first, second, third) ((first - second) / third))

## [1] -2.0 -1.0 -0.5 1.0 1.5 3.0 -3.0 1.0 1.0

## 4b) The test statistic for a population mean is given by the formula ((X - mean)/s/sqrt(n)) Use and show R coding that features a map function to iteratively find test statistics for population means across the lists given above.

pmap\_dbl(list(X, Mean, sd), function(first, second, third) ((first - second) / (third / sqrt(3))))

## [1] -3.4641016 -1.7320508 -0.8660254 1.7320508 2.5980762 5.1961524 -5.1961524  
## [8] 1.7320508 1.7320508

V = c(10,15,17,22,32,38,42)

## 5a) Another purr package function is the keep( ) function. Research, explore, and use the keep( ) function to extract all number from the vector V given above that are less than 20

keep(V, V < 20)

## [1] 10 15 17

## 5b) Another purr package function is the discard( ) function. Research, explore, and use the discard( ) function to eliminate all numbers from the vector V given above that are less than 20

discard(V, V < 20)

## [1] 22 32 38 42

## 6) Another purr package function is the safely( ) function. Research, explore, and apply the safely( ) function to the given vector below as illustrated.

U = list(10,15,"mary",22,32,"james",42)  
map(U, safely(~ .x + 15))

## [[1]]  
## [[1]]$result  
## [1] 25  
##   
## [[1]]$error  
## NULL  
##   
##   
## [[2]]  
## [[2]]$result  
## [1] 30  
##   
## [[2]]$error  
## NULL  
##   
##   
## [[3]]  
## [[3]]$result  
## NULL  
##   
## [[3]]$error  
## <simpleError in .x + 15: non-numeric argument to binary operator>  
##   
##   
## [[4]]  
## [[4]]$result  
## [1] 37  
##   
## [[4]]$error  
## NULL  
##   
##   
## [[5]]  
## [[5]]$result  
## [1] 47  
##   
## [[5]]$error  
## NULL  
##   
##   
## [[6]]  
## [[6]]$result  
## NULL  
##   
## [[6]]$error  
## <simpleError in .x + 15: non-numeric argument to binary operator>  
##   
##   
## [[7]]  
## [[7]]$result  
## [1] 57  
##   
## [[7]]$error  
## NULL

The safely() function is used in this problem to add 15 to each element in the list wherever possible and if not, it will return an error message. This is only possible when the element in the list is numeric and when it is, the output will show the element + 15 under result and NULL under error. The first element in the list, for example, is a numeric 10 so the output had a result of 25 (10 + 25) and an error of NULL as there was no error. This is not possible when the element in the list is a character string and when it is, the output will show NULL under result and the error message under error. The third element in the list, for example, is the character string “mary” so the output had a result of NULL and an error stating that it is a non-numeric argument.