Homework 4

Aritra Samanta, Jorge Arauz Parada

2/5/2022

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.4 v dplyr 1.0.7  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 2.0.1 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(dplyr)  
library(purrr)

1. 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\_dbl(iris, function(x) length((x))) #each row has 150 observations.

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species   
## 150 150 150 150 150

1. 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\_dbl(iris, function(x) length(unique(x)))

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species   
## 35 23 43 22 3

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

z <- matrix( nrow = 3, ncol = 5)  
for (m in 1:3) {  
 for (n in 0:5) {  
 z[m, n] <- n-m  
 }  
}  
print(z)

## [,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.

tibble(  
 mean10n = rnorm(10, -10),  
 mean0 = rnorm(10,0),  
 mean10p = rnorm(10,10),  
 mean100 = rnorm(10,100)  
)

## # A tibble: 10 x 4  
## mean10n mean0 mean10p mean100  
## <dbl> <dbl> <dbl> <dbl>  
## 1 -12.2 1.30 10.0 99.2  
## 2 -11.4 0.0432 9.71 100.   
## 3 -10.5 0.140 9.64 99.8  
## 4 -9.48 -1.11 9.18 99.4  
## 5 -9.39 -0.809 10.3 102.   
## 6 -10.5 0.982 9.80 99.3  
## 7 -9.56 0.0633 9.54 100.   
## 8 -10.4 -1.23 7.63 98.7  
## 9 -9.88 -0.138 9.72 99.8  
## 10 -9.82 -0.451 10.7 101.

tibble(  
 mean10n = rnorm(10, -10),  
 mean0 = rnorm(10,0),  
 mean10p = rnorm(10,10),  
 mean100 = rnorm(10,100)  
)

## # A tibble: 10 x 4  
## mean10n mean0 mean10p mean100  
## <dbl> <dbl> <dbl> <dbl>  
## 1 -9.61 0.560 9.44 102.   
## 2 -10.3 -1.32 9.35 98.3  
## 3 -10.0 -1.47 10.4 99.8  
## 4 -9.33 0.600 10.2 99.9  
## 5 -9.57 -2.40 9.22 99.0  
## 6 -9.60 -1.30 9.39 97.6  
## 7 -10.5 0.383 9.77 100.   
## 8 -9.41 -1.20 10.9 99.6  
## 9 -9.24 0.0802 9.39 101.   
## 10 -11.4 -0.850 10.6 99.1

#running again to show randomness  
#assigning this to a variable will save the numbers also,losing the randomness.

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.

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

1. 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(9)))

## [1] -6.0 -3.0 -1.5 3.0 4.5 9.0 -9.0 3.0 3.0

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

V%>%  
 keep(V <20)

## [1] 10 15 17

b)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

V%>%  
 discard(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

In four or five sentences, explain the specific output for this problem and how the definition and the application of the safely( ) function is used.

In the output for this problem,each item in the list is added to 15, and then the errors for every item are also shown if applicable. For exampls, you cannot add james by 15, therefore the result is shown as null and the error tells us about the non numeric argument to binary operator. The safely function allows us to run code where the errors will show in the output instead of as a side message where the code will not run. ie. we can still evaluate the code as we know there are errors in it.