DATA-613 HW 4

2023-02-09

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4  
## ✔ tibble 3.1.8 ✔ dplyr 1.0.9  
## ✔ tidyr 1.2.0 ✔ stringr 1.4.1  
## ✔ readr 2.1.2 ✔ forcats 0.5.2  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(purrr)  
library(dplyr)

# QUESTION 1  
 #a  
nrow(iris)

## [1] 150

map(iris, ~length(.x))

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

#b  
unique <- iris%>%  
 map\_dbl(n\_distinct)  
unique

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

# QUESTION 2  
matrix1 = matrix(nrow=3, ncol=5)   
for(i in 0:nrow(matrix1))   
{  
 for(j in 0:ncol(matrix1))   
 {  
 matrix1[i,j] = j-i   
 }  
}  
print(matrix1)

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

# QUESTION 3  
tribble( ~mean, ~n,  
 -10, 10,  
 0, 10,  
 -10, 10,  
 100, 10  
 ) -> parameters  
  
pmap(parameters, rnorm)

## [[1]]  
## [1] -10.303164 -8.335628 -10.240357 -9.938641 -9.947635 -10.172901  
## [7] -9.632622 -11.559014 -10.506239 -11.322065  
##   
## [[2]]  
## [1] 1.2708356 -0.5582302 -0.3568442 -1.3348193 -0.6389638 0.4212568  
## [7] 0.1693884 2.3707344 -2.1587129 0.5517289  
##   
## [[3]]  
## [1] -9.533414 -9.989531 -9.850989 -11.042472 -9.798604 -10.663091  
## [7] -10.336279 -11.657642 -9.483194 -9.490689  
##   
## [[4]]  
## [1] 99.84891 100.44236 97.98231 101.28252 100.22639 99.37696 100.03668  
## [8] 99.29860 100.20984 99.47926

pmap(parameters, rnorm)

## [[1]]  
## [1] -10.503597 -11.925999 -9.217486 -11.332450 -8.286664 -10.774424  
## [7] -9.818834 -11.476227 -10.213483 -9.360123  
##   
## [[2]]  
## [1] -0.016097725 -1.352201914 0.613947178 0.263219143 0.550131054  
## [6] 0.422498513 -1.360462275 1.161089890 0.001673724 1.552701881  
##   
## [[3]]  
## [1] -9.310416 -9.893505 -7.852088 -10.815814 -10.076733 -11.053588  
## [7] -9.572287 -9.667097 -10.964950 -10.182181  
##   
## [[4]]  
## [1] 100.75376 99.54756 99.06498 98.19333 99.41931 100.22301 100.23374  
## [8] 101.72724 98.89530 99.73926

# QUESTION 4  
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)  
  
 #a  
pmap\_dbl(list(X, Mean, sd), function(input, mu, sigma) (input-mu)/sigma)

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

#b\*\*\*  
pmap\_dbl(list(X, Mean, sd), function(input, mu, sigma)  
 (input-mu)/(sigma/sqrt(length(input))))

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

# QUESTION 5  
  
 #a  
V <- c(10, 15, 17, 22, 32, 38, 42)  
V%>%  
 keep(function(x) x < 20)

## [1] 10 15 17

#b  
V%>%  
 discard(function(x) x < 20)

## [1] 22 32 38 42

# QUESTION 6\*\*\*  
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 output for this function indicates the whether each value of the list  
 ## was able to be pass through the computation. It indicates the result of  
 ## of the computation if the value is numeric, and indicates the error as  
 ## NULL. If it was not possible to compute, the output indicates that there  
 ## was an error in computing, and points out the error. In this case, the  
 ## error was that both values were non-numeric.