DS413613Homework4KEY

James Dickens

2/12/2022

# Homework 4 KEY  
  
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.5

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

## v ggplot2 3.3.3 v purrr 0.3.4  
## v tibble 3.1.2 v dplyr 1.0.5  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## Warning: package 'forcats' was built under R version 4.0.5

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

library(dplyr)  
  
1

## [1] 1

# a) 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. 6 points  
  
iris

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1 5.1 3.5 1.4 0.2 setosa  
## 2 4.9 3.0 1.4 0.2 setosa  
## 3 4.7 3.2 1.3 0.2 setosa  
## 4 4.6 3.1 1.5 0.2 setosa  
## 5 5.0 3.6 1.4 0.2 setosa  
## 6 5.4 3.9 1.7 0.4 setosa  
## 7 4.6 3.4 1.4 0.3 setosa  
## 8 5.0 3.4 1.5 0.2 setosa  
## 9 4.4 2.9 1.4 0.2 setosa  
## 10 4.9 3.1 1.5 0.1 setosa  
## 11 5.4 3.7 1.5 0.2 setosa  
## 12 4.8 3.4 1.6 0.2 setosa  
## 13 4.8 3.0 1.4 0.1 setosa  
## 14 4.3 3.0 1.1 0.1 setosa  
## 15 5.8 4.0 1.2 0.2 setosa  
## 16 5.7 4.4 1.5 0.4 setosa  
## 17 5.4 3.9 1.3 0.4 setosa  
## 18 5.1 3.5 1.4 0.3 setosa  
## 19 5.7 3.8 1.7 0.3 setosa  
## 20 5.1 3.8 1.5 0.3 setosa  
## 21 5.4 3.4 1.7 0.2 setosa  
## 22 5.1 3.7 1.5 0.4 setosa  
## 23 4.6 3.6 1.0 0.2 setosa  
## 24 5.1 3.3 1.7 0.5 setosa  
## 25 4.8 3.4 1.9 0.2 setosa  
## 26 5.0 3.0 1.6 0.2 setosa  
## 27 5.0 3.4 1.6 0.4 setosa  
## 28 5.2 3.5 1.5 0.2 setosa  
## 29 5.2 3.4 1.4 0.2 setosa  
## 30 4.7 3.2 1.6 0.2 setosa  
## 31 4.8 3.1 1.6 0.2 setosa  
## 32 5.4 3.4 1.5 0.4 setosa  
## 33 5.2 4.1 1.5 0.1 setosa  
## 34 5.5 4.2 1.4 0.2 setosa  
## 35 4.9 3.1 1.5 0.2 setosa  
## 36 5.0 3.2 1.2 0.2 setosa  
## 37 5.5 3.5 1.3 0.2 setosa  
## 38 4.9 3.6 1.4 0.1 setosa  
## 39 4.4 3.0 1.3 0.2 setosa  
## 40 5.1 3.4 1.5 0.2 setosa  
## 41 5.0 3.5 1.3 0.3 setosa  
## 42 4.5 2.3 1.3 0.3 setosa  
## 43 4.4 3.2 1.3 0.2 setosa  
## 44 5.0 3.5 1.6 0.6 setosa  
## 45 5.1 3.8 1.9 0.4 setosa  
## 46 4.8 3.0 1.4 0.3 setosa  
## 47 5.1 3.8 1.6 0.2 setosa  
## 48 4.6 3.2 1.4 0.2 setosa  
## 49 5.3 3.7 1.5 0.2 setosa  
## 50 5.0 3.3 1.4 0.2 setosa  
## 51 7.0 3.2 4.7 1.4 versicolor  
## 52 6.4 3.2 4.5 1.5 versicolor  
## 53 6.9 3.1 4.9 1.5 versicolor  
## 54 5.5 2.3 4.0 1.3 versicolor  
## 55 6.5 2.8 4.6 1.5 versicolor  
## 56 5.7 2.8 4.5 1.3 versicolor  
## 57 6.3 3.3 4.7 1.6 versicolor  
## 58 4.9 2.4 3.3 1.0 versicolor  
## 59 6.6 2.9 4.6 1.3 versicolor  
## 60 5.2 2.7 3.9 1.4 versicolor  
## 61 5.0 2.0 3.5 1.0 versicolor  
## 62 5.9 3.0 4.2 1.5 versicolor  
## 63 6.0 2.2 4.0 1.0 versicolor  
## 64 6.1 2.9 4.7 1.4 versicolor  
## 65 5.6 2.9 3.6 1.3 versicolor  
## 66 6.7 3.1 4.4 1.4 versicolor  
## 67 5.6 3.0 4.5 1.5 versicolor  
## 68 5.8 2.7 4.1 1.0 versicolor  
## 69 6.2 2.2 4.5 1.5 versicolor  
## 70 5.6 2.5 3.9 1.1 versicolor  
## 71 5.9 3.2 4.8 1.8 versicolor  
## 72 6.1 2.8 4.0 1.3 versicolor  
## 73 6.3 2.5 4.9 1.5 versicolor  
## 74 6.1 2.8 4.7 1.2 versicolor  
## 75 6.4 2.9 4.3 1.3 versicolor  
## 76 6.6 3.0 4.4 1.4 versicolor  
## 77 6.8 2.8 4.8 1.4 versicolor  
## 78 6.7 3.0 5.0 1.7 versicolor  
## 79 6.0 2.9 4.5 1.5 versicolor  
## 80 5.7 2.6 3.5 1.0 versicolor  
## 81 5.5 2.4 3.8 1.1 versicolor  
## 82 5.5 2.4 3.7 1.0 versicolor  
## 83 5.8 2.7 3.9 1.2 versicolor  
## 84 6.0 2.7 5.1 1.6 versicolor  
## 85 5.4 3.0 4.5 1.5 versicolor  
## 86 6.0 3.4 4.5 1.6 versicolor  
## 87 6.7 3.1 4.7 1.5 versicolor  
## 88 6.3 2.3 4.4 1.3 versicolor  
## 89 5.6 3.0 4.1 1.3 versicolor  
## 90 5.5 2.5 4.0 1.3 versicolor  
## 91 5.5 2.6 4.4 1.2 versicolor  
## 92 6.1 3.0 4.6 1.4 versicolor  
## 93 5.8 2.6 4.0 1.2 versicolor  
## 94 5.0 2.3 3.3 1.0 versicolor  
## 95 5.6 2.7 4.2 1.3 versicolor  
## 96 5.7 3.0 4.2 1.2 versicolor  
## 97 5.7 2.9 4.2 1.3 versicolor  
## 98 6.2 2.9 4.3 1.3 versicolor  
## 99 5.1 2.5 3.0 1.1 versicolor  
## 100 5.7 2.8 4.1 1.3 versicolor  
## 101 6.3 3.3 6.0 2.5 virginica  
## 102 5.8 2.7 5.1 1.9 virginica  
## 103 7.1 3.0 5.9 2.1 virginica  
## 104 6.3 2.9 5.6 1.8 virginica  
## 105 6.5 3.0 5.8 2.2 virginica  
## 106 7.6 3.0 6.6 2.1 virginica  
## 107 4.9 2.5 4.5 1.7 virginica  
## 108 7.3 2.9 6.3 1.8 virginica  
## 109 6.7 2.5 5.8 1.8 virginica  
## 110 7.2 3.6 6.1 2.5 virginica  
## 111 6.5 3.2 5.1 2.0 virginica  
## 112 6.4 2.7 5.3 1.9 virginica  
## 113 6.8 3.0 5.5 2.1 virginica  
## 114 5.7 2.5 5.0 2.0 virginica  
## 115 5.8 2.8 5.1 2.4 virginica  
## 116 6.4 3.2 5.3 2.3 virginica  
## 117 6.5 3.0 5.5 1.8 virginica  
## 118 7.7 3.8 6.7 2.2 virginica  
## 119 7.7 2.6 6.9 2.3 virginica  
## 120 6.0 2.2 5.0 1.5 virginica  
## 121 6.9 3.2 5.7 2.3 virginica  
## 122 5.6 2.8 4.9 2.0 virginica  
## 123 7.7 2.8 6.7 2.0 virginica  
## 124 6.3 2.7 4.9 1.8 virginica  
## 125 6.7 3.3 5.7 2.1 virginica  
## 126 7.2 3.2 6.0 1.8 virginica  
## 127 6.2 2.8 4.8 1.8 virginica  
## 128 6.1 3.0 4.9 1.8 virginica  
## 129 6.4 2.8 5.6 2.1 virginica  
## 130 7.2 3.0 5.8 1.6 virginica  
## 131 7.4 2.8 6.1 1.9 virginica  
## 132 7.9 3.8 6.4 2.0 virginica  
## 133 6.4 2.8 5.6 2.2 virginica  
## 134 6.3 2.8 5.1 1.5 virginica  
## 135 6.1 2.6 5.6 1.4 virginica  
## 136 7.7 3.0 6.1 2.3 virginica  
## 137 6.3 3.4 5.6 2.4 virginica  
## 138 6.4 3.1 5.5 1.8 virginica  
## 139 6.0 3.0 4.8 1.8 virginica  
## 140 6.9 3.1 5.4 2.1 virginica  
## 141 6.7 3.1 5.6 2.4 virginica  
## 142 6.9 3.1 5.1 2.3 virginica  
## 143 5.8 2.7 5.1 1.9 virginica  
## 144 6.8 3.2 5.9 2.3 virginica  
## 145 6.7 3.3 5.7 2.5 virginica  
## 146 6.7 3.0 5.2 2.3 virginica  
## 147 6.3 2.5 5.0 1.9 virginica  
## 148 6.5 3.0 5.2 2.0 virginica  
## 149 6.2 3.4 5.4 2.3 virginica  
## 150 5.9 3.0 5.1 1.8 virginica

nrow(iris)

## [1] 150

# map method 1  
map\_dbl(iris, length)

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

# map method 2  
iris%>%  
 map\_dbl(length)

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

# b) 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. 6 points  
  
iris%>%  
 map(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. 6 points  
  
z <- matrix(nrow = 3, ncol = 5)  
 for (m in 1:3) {  
 for (n in 1:5) {  
 z[m,n] <- -1\*(m - n)  
 }  
 }  
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 6 points  
   
x<- rnorm(10,-10)  
x

## [1] -10.477432 -9.456436 -9.840646 -11.417427 -10.179196 -10.554178  
## [7] -11.809294 -11.945800 -11.129054 -9.388142

y<- rnorm(10,0)  
y

## [1] 1.1377265 -1.2622766 0.3711522 -0.9961190 0.2019653 0.8654880  
## [7] -0.8375730 1.0820320 1.3103234 0.3389271

z<- rnorm(10,10)  
z

## [1] 8.644221 10.827783 9.726023 11.030503 11.082138 10.520715 10.256253  
## [8] 10.307756 10.000502 10.727471

k<- rnorm(10,100)  
k

## [1] 100.74347 102.11791 100.99924 102.13145 100.05659 99.71955 99.97260  
## [8] 100.87443 100.99150 100.88638

tibble(x,y,z,k)

## # A tibble: 10 x 4  
## x y z k  
## <dbl> <dbl> <dbl> <dbl>  
## 1 -10.5 1.14 8.64 101.   
## 2 -9.46 -1.26 10.8 102.   
## 3 -9.84 0.371 9.73 101.   
## 4 -11.4 -0.996 11.0 102.   
## 5 -10.2 0.202 11.1 100.   
## 6 -10.6 0.865 10.5 99.7  
## 7 -11.8 -0.838 10.3 100.   
## 8 -11.9 1.08 10.3 101.   
## 9 -11.1 1.31 10.0 101.   
## 10 -9.39 0.339 10.7 101.

# 4  
# a) 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. 4 points  
  
X <- list(12, 14, 15, 18, 19, 22,10,18,18)  
X

## [[1]]  
## [1] 12  
##   
## [[2]]  
## [1] 14  
##   
## [[3]]  
## [1] 15  
##   
## [[4]]  
## [1] 18  
##   
## [[5]]  
## [1] 19  
##   
## [[6]]  
## [1] 22  
##   
## [[7]]  
## [1] 10  
##   
## [[8]]  
## [1] 18  
##   
## [[9]]  
## [1] 18

Mean <- list(16, 16, 16, 16, 16,16,16,16,16)  
Mean

## [[1]]  
## [1] 16  
##   
## [[2]]  
## [1] 16  
##   
## [[3]]  
## [1] 16  
##   
## [[4]]  
## [1] 16  
##   
## [[5]]  
## [1] 16  
##   
## [[6]]  
## [1] 16  
##   
## [[7]]  
## [1] 16  
##   
## [[8]]  
## [1] 16  
##   
## [[9]]  
## [1] 16

sd <- list(2, 2, 2, 2, 2,2,2,2,2)  
sd

## [[1]]  
## [1] 2  
##   
## [[2]]  
## [1] 2  
##   
## [[3]]  
## [1] 2  
##   
## [[4]]  
## [1] 2  
##   
## [[5]]  
## [1] 2  
##   
## [[6]]  
## [1] 2  
##   
## [[7]]  
## [1] 2  
##   
## [[8]]  
## [1] 2  
##   
## [[9]]  
## [1] 2

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

# b) 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. 6 points  
  
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

# 5  
V = c(10,15,17,22,32,38,42)  
# a) 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 4 points  
  
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 4 points  
  
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.  
# 4 points  
  
U = list(10, 15, "mary", 22, 32, "james", 42)  
U

## [[1]]  
## [1] 10  
##   
## [[2]]  
## [1] 15  
##   
## [[3]]  
## [1] "mary"  
##   
## [[4]]  
## [1] 22  
##   
## [[5]]  
## [1] 32  
##   
## [[6]]  
## [1] "james"  
##   
## [[7]]  
## [1] 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 of five sentences explain the specific output for this   
# and how the definition and the application of the safely() function is   
# used.  
  
# If The safely function is applied to a vector(List), if confirms that an  
# an operation can be executed on variable types that belong to the same   
# family of variables or not. If variables belong to the same family the   
# function outputs the answer(result) and NULL. NULL indicating that no error  
# ocurred. If variables do not belong to the same family, the safely function  
# outputs NULL for a result (indicating that there is no result) and a description  
# of the error.