DATA 613 - HW 3

Britnie Smith and Alexis Threatt

1/28/2022

library(tidyverse)

*Vector1 <- (c( 10, 19, 121, 83, 63, 7, 77, 61, 51, 97, 123, 41)) Vector1* **1) For the vector given above, use and show two methods of R coding to extract the first element and the last element.**

Vector1 <- (c(10, 19, 121, 83, 63, 7, 77, 61, 51, 97, 123, 41))   
Vector1

## [1] 10 19 121 83 63 7 77 61 51 97 123 41

length(Vector1)

## [1] 12

Vector1[c(1,12)]

## [1] 10 41

Vector1[(-2:-11)]

## [1] 10 41

**2) For the vector given above, use and show two methods of R coding to extract all of the elements that are less than 60.**

Vector1[Vector1< 60]

## [1] 10 19 7 51 41

Vector1[!Vector1> 60]

## [1] 10 19 7 51 41

**3) For the vector given above, use and show two methods of R coding to extract all numbers that are not divisible by 2 or 3.**

Vector1[!Vector1 %%2 == 0 & !Vector1 %%3 == 0]

## [1] 19 121 83 7 77 61 97 41

Vector1[Vector1 %%2 > 0 & Vector1 %% 3 > 0]

## [1] 19 121 83 7 77 61 97 41

**4) Use and show two R coding methods to confirm that Vector1 does not have missing values** *Not sure if you wanted us to use vector1 or myList since you said Vector1 in the question but noted ’Use the [mylist] for problems 4-6. in the next section. SO we did both to be safe”*

Vector1[is.na(Vector1)]

## numeric(0)

Vector1[!is.na(Vector1)] ##Shows all values of vector 1 which shows that there are no NA values

## [1] 10 19 121 83 63 7 77 61 51 97 123 41

Vector1["NA"]

## [1] NA

myList <- list(TRUE, 12.35, 'pear', 48, c = 3:8, list(23, 'team'))  
myList

## [[1]]  
## [1] TRUE  
##   
## [[2]]  
## [1] 12.35  
##   
## [[3]]  
## [1] "pear"  
##   
## [[4]]  
## [1] 48  
##   
## $c  
## [1] 3 4 5 6 7 8  
##   
## [[6]]  
## [[6]][[1]]  
## [1] 23  
##   
## [[6]][[2]]  
## [1] "team"

myList[is.na(myList)]

## named list()

myList[!is.na(myList)] ##Shows all values of myList, which shows that there are no NA values.

## [[1]]  
## [1] TRUE  
##   
## [[2]]  
## [1] 12.35  
##   
## [[3]]  
## [1] "pear"  
##   
## [[4]]  
## [1] 48  
##   
## $c  
## [1] 3 4 5 6 7 8  
##   
## [[6]]  
## [[6]][[1]]  
## [1] 23  
##   
## [[6]][[2]]  
## [1] "team"

myList["NA"] ##Shows as Null becasue there are no NA values

## $<NA>  
## NULL

*myList <- list(TRUE, 12.35, “pear”, 48, c = 3:8, list(23, “team”)) myList* *(note: it is better to type the list into R studio or R markdown. Do not copy and paste) Use the list above for problems 4 – 6.*

**5) For the list given above, use and show R coding to confirm that “pear” is a character element.**

myList <- list(TRUE, 12.35, 'pear', 48, c = 3:8, list(23, 'team'))  
myList

## [[1]]  
## [1] TRUE  
##   
## [[2]]  
## [1] 12.35  
##   
## [[3]]  
## [1] "pear"  
##   
## [[4]]  
## [1] 48  
##   
## $c  
## [1] 3 4 5 6 7 8  
##   
## [[6]]  
## [[6]][[1]]  
## [1] 23  
##   
## [[6]][[2]]  
## [1] "team"

str(myList[[3]])

## chr "pear"

**6) For the list given above, use and show R coding to extract the first three elements of the list.**

myList[1:3]

## [[1]]  
## [1] TRUE  
##   
## [[2]]  
## [1] 12.35  
##   
## [[3]]  
## [1] "pear"

**7) Use the $ operator to extract the element “pear” from your list. Be sure to use and show required R code to produce the requested output.**

myList <- list(TRUE, 12.35, B= 'pear', 48, c = 3:8, list(23, 'team'))  
myList

## [[1]]  
## [1] TRUE  
##   
## [[2]]  
## [1] 12.35  
##   
## $B  
## [1] "pear"  
##   
## [[4]]  
## [1] 48  
##   
## $c  
## [1] 3 4 5 6 7 8  
##   
## [[6]]  
## [[6]][[1]]  
## [1] 23  
##   
## [[6]][[2]]  
## [1] "team"

myList$B

## [1] "pear"

**8) Use and show R code to write a function to solve the following quadratic equations by using the quadratic formula. (all equations have two real number solutions)** *a) x2- 3x - 28 = 0*  
*b) x2 + x - 30 = 0*  
*c) 3x2+ 14x + 8 = 0*   
*d) 2x2+11x = 6*

quadfunction <- function(a, b, c) {  
 x1 = (-b + sqrt((b)^2 -4\*a\*c))/2\*a  
 x2 = (-b - sqrt((b)^2 -4\*a\*c))/2\*a  
 print(c(x1,x2))  
}  
quadfunction(1,-3,-28)

## [1] 7 -4

quadfunction(1,1,-30)

## [1] 5 -6

quadfunction(3,14,8)

## [1] -6 -36

quadfunction(2,11,0)

## [1] 0 -22

**9) In your book (towards the end of chapter 16) a special set of vectors are defined as Augmented Vectors. One such augmented vector is a Tibble. Use and show R code that will produce the Tibble shown below. Do not simply type or copy and paste. You must show and use R coding that will output the tibble.**

tibble(  
 x = 1:10,  
 y = 10:1,  
 z = (y/2)  
)-> Table1  
Table1

## # A tibble: 10 x 3  
## x y z  
## <int> <int> <dbl>  
## 1 1 10 5   
## 2 2 9 4.5  
## 3 3 8 4   
## 4 4 7 3.5  
## 5 5 6 3   
## 6 6 5 2.5  
## 7 7 4 2   
## 8 8 3 1.5  
## 9 9 2 1   
## 10 10 1 0.5

**10) In statistics, the Interquartile Range is the difference between Q3 and Q1. Now show and use map function coding to find the Interquartile Range for each column of the tibble from.**

map\_dbl(Table1, IQR)

## x y z   
## 4.50 4.50 2.25