445_Assignment_1

Tucker Harris

2023-10-05

Chapter 8 Question 1

```
vec_a<-c( 2, 4, 6 )
vec_b<-c( 8, 10, 12 )
vec_c<-vec_a + vec_b

vec_c
```

```
## [1] 10 14 18
```

Chapter 8 Question 2

```
vec_d <- c( 14, 20 )
vec_d + vec_a
```

```
## Warning in vec_d + vec_a: longer object length is not a multiple of shorter
## object length
```

```
## [1] 16 24 20
```

```
# This adds va[1] + vd[1], va[2] + vd[2], va[3] + vd[1]

# The shorter vector merely repeats once it runs out of elements

# the warning is merely saying that one is shorter than the other
```

Chapter 8 Question 3

```
vec_a + 5
```

```
## [1]  7  9 11
```

```
# This merely added a 5 to each element within vec_a, there is no warning
# Because there is no vector end bound breached.
```

Chapter 8 Question 4

```
seq_vec <- seq(1,5,1)
colon_vec <- 1:5

seq_vec
```

```
## [1] 1 2 3 4 5
```

```
colon_vec
```

```
## [1] 1 2 3 4 5
```

Chapter 8 Question 5

```
even_seq <- seq(2, 20, 2)
even_col <- 2 * 1:10

even_seq
```

```
##  [1]  2  4  6  8 10 12 14 16 18 20
```

```
even_col
```

```
##  [1]  2  4  6  8 10 12 14 16 18 20
```

Chapter 8 Question 6

```
seq(0,1,length.out=21)
```

```
##  [1] 0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70
## [16] 0.75 0.80 0.85 0.90 0.95 1.00
```

```
# Can also do this

seq(0,1,0.05)
```

```
##  [1] 0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70
## [16] 0.75 0.80 0.85 0.90 0.95 1.00
```

```
# But less efficient, as sometimes it may not be as nice of a number
# As 0.05, so using the length.out makes the computer figure out the math for you
```

Chapter 8 Question 7

```
rep( c( 2, 4, 8 ), 3 )
```

```
## [1] 2 4 8 2 4 8 2 4 8
```

Chapter 8 Question 8

```
 rep( c( 2, 4, 8 ), each = 4)
```

```
##  [1] 2 2 2 2 4 4 4 4 8 8 8 8
```

Chapter 8 Question 10

```
M <- matrix( seq( 2, 30, 2 ), nrow = 3, byrow=TRUE )
M
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    2    4    6    8   10
## [2,]   12   14   16   18   20
## [3,]   22   24   26   28   30
```

```
a <- seq( 2, 10, 2 )
b <- seq( 12, 20, 2 )
c <- seq( 22, 30, 2 )


M <- rbind(a,b,c)

M
```

```
##   [,1] [,2] [,3] [,4] [,5]
## a    2    4    6    8   10
## b   12   14   16   18   20
## c   22   24   26   28   30
```

Chapter 8 Question 12

```
df <- data.frame(name= c('Alice','Bob','Charlie','Daniel'),
                 Grade = c(6,8,NA,9))

df[ -which(  is.na(df$Grade) ), ]
```

```
##     name Grade
## 1  Alice     6
## 2    Bob     8
## 4 Daniel     9
```

```
df[  which( !is.na(df$Grade) ), ]
```

```
##     name Grade
## 1  Alice     6
## 2    Bob     8
## 4 Daniel     9
```

The first method manually places values and column names into a new matrix. First way that NA is taken out of the matrix is by merely subtracting the df$Grade values with the na value. The second way uses the ! (not) operator. That says display all the values that are not "na"

Chapter 8 Question 14

```
x = c(4,5,6,7,8,9,10)
y = c(34,35,41,40,45,47,51)
slope = 2.82
p.value = 0.000131

my.test <- list( x = x, y = y, Slope = slope, P = p.value )

my.test
```

```
## $x
## [1]  4  5  6  7  8  9 10
##
## $y
## [1] 34 35 41 40 45 47 51
##
## $Slope
```

```
## [1] 2.82
##
## $P
## [1] 0.000131
```

```
my.test[2]
```

```
## $y
## [1] 34 35 41 40 45 47 51
```

```
my.test['P']
```

```
## $P
## [1] 0.000131
```

Chapter 9 Question 1

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readxl)
library(googlesheets4)
```

Use file –> import dataset to bring the "Raw Data" page into R Studio. I also told it to skip rows 1:4, as those are either headers or empty.

```
tree <- read_excel('Example_5.xls', range = "A5:C36", sheet = 2 )

str( tree )
```

```
## tibble [31 x 3] (S3: tbl_df/tbl/data.frame)
##  $ Girth : num [1:31] 8.3 8.6 8.8 10.5 10.7 10.8 11 11 11.1 11.2 ...
##  $ Height: num [1:31] 70 65 63 72 81 83 66 75 80 75 ...
##  $ Volume: num [1:31] 10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 ...
```

The three columns and 31 rows are displayed appropriately.

Chapter 9 Question 2

File -> Import Dataset, then select Example_3.xls

```
tesla <- read_excel('Example_3.xls', range = "A1:L34", na = "-9999", sheet = 2 )


tail(tesla)
```

```
## # A tibble: 6 x 12
##   model          mpg cyl   disp     hp  drat    wt  qsec vs       am  gear carb
```

```
##   <chr>        <dbl> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <chr> <dbl> <dbl> <chr>
## 1 Lotus Europa  30.4 4     95.0~   113  3.77  1.51  16.9 1         1     5 2
## 2 Ford Panter~  15.8 8     351     264  4.22  3.17  14.5 0         1     5 4
## 3 Ferrari Dino  19.7 6     145     175  3.62  2.77  15.5 0         1     5 6
## 4 Maserati Bo~  15   8     301     335  3.54  3.57  14.6 0         1     5 8
## 5 Volvo 142E    21.4 4     121     109  4.11  2.78  18.6 1         1     4 2
## 6 Tesla Model~  98   NA    NA      778 NA     4.94  10.4 NA        0     1 NA
```