

445 Assignment One

Calder Evans

6 October 2023

Chapter Eight

Exercise One

Create a vector of three elements (2,4,6) and name that vector `vec_a`. Create a second vector, `vec_b`, that contains (8,10,12). Add these two vectors together and name the result `vec_c`.

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

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

Exercise Two

Create a vector, named `vec_d`, that contains only two elements (14,20). Add this vector to `vec_a`. What is the result and what do you think R did (look up the recycling rule using Google)? What is the warning message that R gives you?

```
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
```

- (a) The result is a vector with three numbers. The first and second values are computed relatively easily. The first value in the printed vector results from adding the first value of vector d with the first value of vector a. The second value in the printed vector results from adding the second value of vector d with the second value of vector a. The third number “recycles” the first value of vector d and adds it to the third value of vector a.
- (b) The warning message is reminding you that the vectors are not the same size.

Exercise Three

Next add 5 to the vector `vec_a`. What is the result and what did R do? Why doesn't it give you a warning message similar to what you saw in the previous problem?

```
5 + vec_a
```

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

- a) The result adds 5 to every value in the a vector
- b) This sum does not give you a error like the previous problem because it is computing vector algebra, where you add 5 to every value.

Exercise Four

Generate the vector of integers $\{1, 2, \dots, 5\}$ in two different ways. a) First using the `seq()` function b) Using the `a:b` shortcut.

```
seq( from=1, to=5, by=1)
```

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

```
1:5
```

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

Exercise Five

Generate the vector of even numbers $\{2, 4, 6, \dots, 20\}$ a) Using the `seq()` function and b) Using the `a:b` shortcut and some subsequent algebra. *Hint: Generate the vector 1-10 and then multiply it by 2.*

```
seq( from=2, to=20, by=2)
```

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

```
2 * 1:10
```

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

Exercise Six

Generate a vector of 21 elements that are evenly placed between 0 and 1 using the `seq()` command and name this vector `x`.

```
x <- seq( from=0, to=1, by=0.05)
x
```

```
## [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
```

Exercise Seven

Generate the vector $\{2, 4, 8, 2, 4, 8, 2, 4, 8\}$ using the `rep()` command to replicate the vector `c(2,4,8)`.

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

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

Exercise Eight

Generate the vector $\{2, 2, 2, 2, 4, 4, 4, 4, 8, 8, 8, 8\}$ using the `rep()` command. You might need to check the help file for `rep()` to see all of the options that `rep()` will accept. In particular, look at the optional argument `each=`.

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

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

Exercise Ten

- a) Create the matrix in two ways and save the resulting matrix as M.
i. Create the matrix using some combination of the `seq()` and `matrix()` commands.

```
M <- matrix(seq( from=2, to=30, by=2), nrow=3, ncol=5)
M

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

- ii. Create the same matrix by some combination of multiple ``seq()`` commands and either the ``rbind()`` or ``cbind()`` command.

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

- b) Extract the second row out of M.

```
M[2,]

## [1] 12 14 16 18 20
```

- c) Extract the element in the third row and second column of M.

```
M[3,2]

## c
## 24
```

Exercise Twelve

The following code creates a `data.frame` and then has two different methods for removing the rows with NA values in the column `Grade`. Explain the difference between the two.

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

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

- (a) The subtraction sign in the first method subtracts the “NA” values from the data set. The exclamation point in the second method is a “not” operator and it takes out all the values that aren’t “NA”

Exercise Fourteen

Create and manipulate a list. a) Create a list named my.test with elements + $x = c(4,5,6,7,8,9,10)$ + $y = c(34,35,41,40,45,47,51)$ + $slope = 2.82$ + $p.value = 0.000131$

```
mytest <- list( 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)

mytest
```

```
## $x
## [1] 4 5 6 7 8 9 10
##
## $y
## [1] 34 35 41 40 45 47 51
##
## $slope
## [1] 2.82
##
## $p.value
## [1] 0.000131
```

b) Extract the second element in the list.

```
mytest[[2]]
```

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

c) Extract the element named `p.value` from the list.

```
mytest$p.value
```

```
## [1] 0.000131
```

Chapter Nine

Exercise One

Download from GitHub the data file [Example_5.xls] (https://github.com/dereksonderegger/444/raw/master/data-row/Example_5.xls). Open it in Excel and figure out which sheet of data we should import into R. At the same time figure out how many initial rows need to be skipped. Import the data set into a data frame and show the structure of the imported data using the `str()` command. Make sure that your data has $n = 31$ observations and the three columns are appropriately named. If you make any modifications to the data file, comment on those modifications.

```
library(readxl)
data.trees <- data.frame(read_excel('Example_5.xls',
                                   sheet='RawData',
                                   range="A5:C36"))

head(data.trees)
```

```
##   Girth Height Volume
## 1   8.3     70   10.3
## 2   8.6     65   10.3
## 3   8.8     63   10.2
## 4  10.5     72   16.4
## 5  10.7     81   18.8
## 6  10.8     83   19.7
```

Exercise Two

Download from GitHub the data file [Example_3.xls] (https://github.com/dereksonderegger/444/raw/master/data-row/Example_3.xls). Import the data set into a data frame and show the structure of the imported data using the `tail()` command which shows the last few rows of a data table. Make sure the Tesla values are NA where appropriate and that both -9999 and NA are imported as NA values. If you make any modifications to the data file, comment on those modifications.

```
data.cars <- read_excel('Example_3.xls', sheet=2, range="A1:L34", na=c("NA", "-9999"))
tail(data.cars)
```

```
## # A tibble: 6 x 12
##   model      mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear  carb
##   <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Lotus Europa  30.4     4  95.1   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
```