

Assignment1

Kyle Bambling

2023-10-05

Question 1

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
```

Question 2

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
```

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

It takes the first element from the longer vector and uses that for the third addition calculation.

Question 3

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?

```
vec_a + 5
```

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

It adds 5 to each element of the vector.

Question 4

Generate the vector of integers $\{1, 2, \dots, 5\}$ in two different ways.

- First using the `seq()` function
- Using the `a:b` shortcut.

```
#part a
seq(1,5)
```

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

```
#part b
1:5
```

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

Question 5

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 multiple it by 2.*

```
#part a
seq(2, 20, by = 2)
```

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

```
#part b
1:10 * 2
```

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

Question 6

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

Question 7

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
```

Question 8

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=4)
```

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

Question 10

In this problem, we will work with the matrix

$$\begin{bmatrix} 2 & 4 & 6 & 8 & 10 \\ 12 & 14 & 16 & 18 & 20 \\ 22 & 24 & 26 & 28 & 30 \end{bmatrix}$$

- 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.
 - ii. Create the same matrix by some combination of multiple `seq()` commands and either the `rbind()` or `cbind()` command.
- b) Extract the second row out of M.
- c) Extract the element in the third row and second column of M.

```
#part a i
matrix( seq(2,30, by=2), nrow = 3, byrow = TRUE)

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

# ii
M <- rbind(seq(2,10, by=2), seq(12,20, by=2), seq(22,30, by=2))

#part b
M[2,]

## [1] 12 14 16 18 20

#part c
M[3,2]

## [1] 24
```

Question 12

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) ), ]
```

The first one finds which rows have the NA value and displays the other rows, the second one finds and displays every row that does not have the value NA.

Question 14

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

```
#part a
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.value=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.value
## [1] 0.000131
```

```
#part b
my.test[2]
```

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

```
#part c
my.test[['p.value']]
```

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

Chapter 9

Question 1

Download from GitHub the data file `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.

```
cherry_tree <- read_excel('Example_5.xls', sheet='RawData', range='A5:C36')
head(cherry_tree)
```

```
## # A tibble: 6 x 3
##   Girth Height Volume
##   <dbl>   <dbl>   <dbl>
## 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
```

Question 2

Download from GitHub the data file `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.

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

```
## # 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