

# R Exercises

Cafer Bakac

01 05 2023

## R Exercises

In this exercise sheet, I have bundled a couple of exercises that I think would be helpful for you to develop your R skills. I have divided the quizzes into different parts so that it becomes easier to follow. Please get in touch with me if you guys want to have further exercises :)

### Vectors

1. Create the following vectors:

- a. (4,6,3,4,6,3,...,4,6,3,4) where there are 11 occurrences of 4
- b. (4,4,...,4,6,6,...,6,3,3,...,3) where there are 10 occurrences of 4, 20 occurrences of 6, and 30 occurrences of 3.

2. Predict the output of each when executed in R. Please do not use R for this at the beginning. Try to predict the output and then, execute the code in R to see if your prediction was correct. This exercise aims to check if you understood how the vectors are working.

a.

```
nums = c(5, 8, 3, 2, 7, 4) + c(1, 2, 3)
nums[c(2:4)]
```

b.

```
nums = c(5, 8, 3, 2, 7, 4) - c(3, 0, 1)
nums[-c(2:4)]
```

c.

```
nums = c(3, 1, 5, 4) + c(2)
nums[c(1,3)]
```

d.

```
nums = c(9,7,6,2,8)
nums[length(nums)]
```

## Matrices

1. The following code creates in R a matrix.

```
m <- rbind(c(11,12,13,14,15),
           c(21,22,23,24,25),
           c(31,32,33,34,35),
           c(41,42,43,44,45),
           c(51,52,53,54,55))
```

Using the above, write R code to generate the following sub-matrices of m: a. the sub-matrix of elements in rows 3, 4, or 5 and in columns 1 or 2 b. the sub-matrix formed by deleting columns 2 and 4 c. the sub-matrix of rows whose first column element has an even ten's digit

## Lists

1. Use R to express the following list in order by component name

```
my.list <- list(b=3,a="bob",c=c(1,2,3),f=TRUE,e=c("x","y","z"),d=37)
```

2. Create a list with the factor named grades made from the vector c("A","C","B","B","D","A"); the vector named nums identical to c(3,2.1,3.3,4,1.5,4.9). Then do the following:
  - a. Write R code to display the elements of grades that correspond to elements of nums that are greater than or equal to 3.
  - b. Add a new member to the list, name it days. Add the weekdays to the newly created member from Monday through Saturday.

## Data Frames

1. Create a data frame named dframe in accordance with the following table. Note, person should be a vector of strings, while sex and funny should be factors.

person	sex	funny
Stan	<i>M</i>	High
Francine	<i>F</i>	Med
Steve	<i>M</i>	Low
Roger	<i>M</i>	High
Abigail	<i>F</i>	High
Klaus	<i>M</i>	Med

Now do or find the following: a. Add an age column where the ages of Stan, Francine, Steve, Roger, Hayley, and Klaus are 41, 41, 15, 1600, 21, and 60, respectively. (Yes, Roger is very old.)

- Reorder the columns of the data frame so that they occur in the following order: person, age, sex, funny.
- Find the mean age with including and excluding Roger.
- Find the standard deviation of the age including and excluding Roger.
- Select these people who are “high” on funny variable and store it in a new data.frame

## Loops

- Loop through a vector (lets say: `vec <- seq(1,10, by=.5)`) and print only the odd numbers.
- Using the same vector, loop through it and print only the odd numbers that the remainder is zero when divided by 3.
- A coin is flipped and one of the results “heads”, “tails”, “on edge” is stored in the variable coin. Write code that prints “It will be sunny” if the coin landed “heads”, prints “It will be cloudy” if the coin landed “tails”, and prints “There will be a blizzard” if the coin landed “on edge”. I want you toss the coin 30 times.

Use the following code for this exercise:

```
coin <- c('Heads', 'Tails', 'Edge') # the coin and possible outcomes
set.seed(1257)
sample(coin, size = 4, replace = TRUE) # this tosses the coin with replacement four times, prints outcom
```

## Functions

- Write a function  $U(n)$  that calculates the following:

$$U(n) = \begin{cases} 7n + 3 & \text{if } n \text{ is a multiple of 3} \\ \frac{7n + 2}{3} & \text{if } n \text{ has remainder 1 upon division by 3} \\ \frac{n - 2}{3} & \text{if } n \text{ has remainder 2 upon division by 3} \end{cases}$$

Figure 1: function

- Suppose a jar contains  $r$  red and  $b$  blue marbles, and  $n$  marbles are selected without replacement. Write an R function, `marbles.probability(r,b,n,x)`, that returns the probability of getting exactly  $x$  red marbles, similar to the function shown below:

```
marbles.probability(8,9,5,2)
0.3800905
```