Data Analysis and Visualization Exercise 1

17 October 2019

Abstract

Basic R data structures

1 R Markdown

This is an R Markdown document. An R Markdown file is a combination of text and code cells. You might be familiar with this concept from python's jupyter notebook. Otherwise, this first task should familiarize you with this concept. What you are reading here is standard text.

```
## And here you see how to add some R code:
x <- 1:10
s <- sum(x)
print(paste('The sum of 1 to 10 is:', s))
```

[1] "The sum of 1 to 10 is: 55"

You can execute every single line within a code block by pressing `CTRL` and `ENTER` or you can knit the entire document into a pdf file by pressing the `Knit` button above.

2 Vectors

2.1 Question

First, create three named numeric vectors of size 10, 11 and 12 respectively in the following manner:

- One vector with the "colon" approach: *from:to*
- One vector with the `seq()` function: *seq(from, to)*
- And one vector with the `seq()` function and the `by` argument: *seq(from, to, by)*

For easier naming you can use the vectors `letters` or `LETTERS` which contain the latin alphabet in lowercase and capital, respectively. In order to select specific letters use e.g. `letters[1:4]` to get the first four letters. Check their types. What is the outcome? Where do you think the difference comes from?

Then combine all three vectors in a list called 'myList'. Check the attributes of the vectors and the list. What is the difference and why?

Hint: If the elements of a list have no names, we can access them with the double brackets and an index, e.g. `myList[[1]]`

```
# Please insert your solution here.
```

2.2 Question:

Assume you have a lookup table as `lookup <- c(a = "sun", b = "rain", c = "wind", u = NA)`. Generate the weekly weather predictions `c("sun","sun","rain", NA, "rain", "rain", "wind")` out of this lookup table.

```
lookup <- c(a = "sun", b = "rain", c = "wind", u = NA)
# Please insert your solution here.
```

3 Factors

3.1 Question:

What is the difference between a factor and a vector?

3.2 Question

Create a factor vector of length 30 with the three levels *Rita Repulsa*, *Lord Zedd* and *Rito Revolto* and equal length for each level. What happens if you replace the second element of the vector with *Shredder*? *Hint*: `gl()`

```
# Please insert your solution here.
```

3.3 Question:

Create the following 3 factor vectors f1, f2 and f3:

```
f1 <- factor(letters)
levels(f1) <- rev(levels(f1))
f2 <- rev(factor(letters))
f3 <- factor(letters, levels = rev(letters))
```

The function `rev` reverses the order of an order-able object. What is the difference between f1, f2 and f3? Why?

```
# Please insert your solution here.
```

4 Matrices

4.1 Question :

Create a 3 by 4 matrix that contains the numbers 1 to 12 and then convert it into a data frame.

```
# Please insert your solution here.
```

4.2 Question:

Create a 10 by 5 matrix which contains the numbers from 1 to 50 column-wisely. Name the rows as 'row_n' and columns as 'col_n'. Compute the mean and sum of each row and column. Add vector seq(60,100,10) as another row to the matrix.

Generate another matrix with the same dimensions, containing random numbers between 1 and 100. Subtract this matrix from the first one.

Plot the covariance matrix of the columns of the resulting matrix with spearman correlation coefficients.

Hint: Check the functions paste0(), colMeans(), rowMeans(), colSums(), rowSums(), sample(), cor() and corrplot() (in package 'corrplot')

```
# Please insert your solution here.
```

5 Lists and data.frames

5.1 From lists to data.frames

Take myList from the first question. Coerce it to a `data.frame` with `as.data.frame()`. Why does it fail and how can we fix it? What happened to the names?

```
# Please insert your solution here.
```

5.2 Creating data.frames

Create a data.frame with 26 rows like the one shown below. Only the first six and the last six rows are displayed.

Hint: Instead of the workaround with `list` you can also use simply `data.frame(column_name = column_vector, ...)`

```
# Please insert your solution here.
```

5.3 Attributes

Take again the `data.frame` from the previous question and make a copy of it.

- Change the row names and the column names of the data.frame to capital letters (or small letters, if they are already capital).
- Change the `class` attribute to *list*. What happens?
- Change it now to any name you like. What happens now? What happens if you remove the `class` attribute?

```
# Please insert your solution here.
```

5.4 Combining data.frames

Now take the previous data.frame and reproduce the following data.frame. Only the first and the last six rows are shown.

Hint: In order to combine to data.frames by column you can use `cbind(df1, df2, ...)`

```
# Please insert your solution here.
```

5.5 Operations on data.frames

Create the data.frame *df* using `df <- as.data.frame(matrix(runif(9e6), 3e3, 3e3))`

This will create a data.frame with 3000 columns and rows and a total of 9 million values.

Now compute the sum of any row, then compute the sum of any column. Measure the time for both operations. Why are the times different although the size is the same?

- **Hint 1:** The time is measured with the function `system.time(my_function_call)`
- **Hint 2:** The sum can be computed with the sum function `sum(my_vector)`
- **Hint 3:** Columns and rows are selected by single brackets. Rows: `df[row_number, ]`, Columns: `df[, column_number]`

```
# Please insert your solution here.
```

6 Subsetting

6.1 Question:

Use the data.frame you created from the 3 by 4 matrix in the earlier question (Question 3.1) for the next three questions.

1. Select the elements on the second row and the second and fourth columns.

```
# Please insert your solution here.
```

2. Set the rownames to "row1", "row2", "row3" and column names to "col1", "col2", "col3" and "col4". (Hint: use the function "paste0")

```
# Please insert your solution here.
```

3. Assign 0 to all the elements in columns "col3" and "col4".

```
# Please insert your solution here.
```

6.2 Question:

Considering x <- c("a"=1, "b"=2, "c"=3, "d"=4, "e"=5), select the third and fifth elements of x by using positive integers, negative integers, a logical vector, and a character vector.

```
# Please insert your solution here.
```

6.3 Question:

Why are vals[c(2, 5)] and vals[2, 5] different where `vals <- outer(1:5, 1:5, FUN = "/")`?

```
# Please insert your solution here.
```

6.4 Question:

Assume `x <- matrix(1:20, ncol=2)`. What is the difference between `x[1, , drop = T]` and `x[1, , drop = F]`? Now let `y <- as.data.frame(x)`. What is the difference between `y[,1]` , `y[[1]]` and `y[1]`?

```
# Please insert your solution here.
```

6.5 Question:

Now assume the weather in winter lookup table is a data frame as below and we have the predictions for the next week as stored in weeklyCast. How would you create "weeklyTable" by the use of rownames function? How would you create it using the `match` function? How would you order the rows of lookup table by 'desc' column?

```
lookup <- data.frame(
  averageTemperature = c(5, 7, 10, 0, 3),
  desc = c("cloudy", "rainy", "sunny", "snowy", "windy"),
  goodForSki = c(T, F, T, F, F)
)

weeklyCast <- c("rainy", "rainy", "cloudy", "windy", "snowy", "cloudy", "sunny")

weeklyTable <- data.frame(averageTemperature=c(7,7,5,3,0,5,10),
```

```
                                desc=c("rainy", "rainy", "cloudy", "windy", "snowy", "cloudy", "sunny"),
                                goodForSki=c(F,F,T,F,F,T,T))
```

```
        # Please insert your solution here.
```

6.6 Question:

For the next questions, consider the bigDF data.frame which has 1,500 columns and rows.

1. Select the even numbered columns named such as "Column_2", "Column_4", etc.

```
bigDF <- as.data.frame(matrix(0, ncol=1500, nrow=1500))
colnames(bigDF) <- paste0("Column_", 1:ncol(bigDF))
```

```
        # Please insert your solution here.
```

2. Select all the columns other than column 76.

```
        # Please insert your solution here.
```

3. Assign the number 1 to 500 randomly selected diagonal indices.

```
        # Please insert your solution here.
```

4. Retrieve the row and column indices of the elements which have been assigned a 1. Select the rows where columns Column_1 or Column_2 are 1 using the subset() function.

```
        # Please insert your solution here.
```

6.7 (Optional) Data.frames: Titanic

Compute the number of women who survived the Titanic. Start by loading the data into a data.frame using the following command:

```
tab <- read.csv("extdata/titanic.csv")
## Error in file(file, "rt"): cannot open the connection
```

```
        # Please insert your solution here.
```