1 Introduction

The R programming language was created in the early 1990s. It is based upon the S programming language that was developed at Bell Laboratories in the 1970s. Once you learn R syntax, it is relatively easy to also program in S. The main difference is that R is free and open source software. A considerable feature in R is that if you do not like how something is implemented, you can easily write a package to make R do things the way you want. Since there are countless R packages and different implementations, the question is not "Can I do this in R?", but more commonly "Which of these implementations should I use?".

R is a scripting language, meaning that your code does not need to be compiled before you run it. The main focus in R is to help you analyze data and it is mainly used by statisticians. Hereby, R is not suited for programming that requires you to access the inner workings of a computer. R is an imperative language in its core (you write a script that does calculations one after another). Additionally, R supports object oriented programming (data and functions are combined inside classes) and functional programming (functions are first-class objects; you can treat them like any other variable and call them recursively). This mix of different programming styles means that your R code can resemble several other languages. You can write imperative code that looks like C, use reference classes to write object-oriented code that looks like C# or Java. Due to the things stated above, there is often more than one way to perform a single task in R.

In this course we use an integrated development environment (IDE) called RStudio. Note, that you can also program R using the basic R interface. However, RStudio gives us some additional features compared to the basic R. For example, the plot windows are nicer than in the basic R.

2 Basics

1. Start RStudio. Select RStudio from the Start → All Programs → RStudio.

2. On the command line, type *demo(graphics)* and *demo(image)*. All the demos are given by *demo()*.

3. Type *help(plot)*. Note that *help()* and *help.search()* are useful functions when you are using R.

4. Create a new working directory for this course.

5. From RStudio, select Session → Set working directory → Choose directory and choose the the directory created before.

6. From RStudio, select File → New File → R script. Save the script to your working directory.

7. Copy the command from the console that changes your working directory. Place the command to the first row in your script.

8. Note that you can run an individual line from your script by pressing ctrl+r (Windows) or ctrl + enter (Linux).

3 Data Types

The are several basic data types in R. Note that incorrect data types may cause unwanted results. You can check the data type of a variable with the command: *class()*. The basic data types in R are numeric, integer, complex, logical and character.

9. Decimal values are called numerics in R. Assign the values 10.5 and 10 to variables a and b. Check the data types of a and b.

10. Use the command *is.integer()* to see if a and b are integers.

11. Use the *as.integer()* function to assign 3 to variable c and 3.7 to variable d. Check the data type and the result from *is.integer()*.

12. Assign a complex number to variable e, where the real part is 1 and the imaginary part is 2.

13. Assign a logical value TRUE to variable f.

14. Assign your first name to variable name and your age to variable age.

15. Use the function *paste()* to combine the variables name and age.

16. Convert the vector a into a character using *as.character().*

17. Use the commands *ls()* and *ls.str()* to view all the variables that you have created.

4 Vectors

Some functions to get you started: *c(), rep(), length(), sum(), paste(), sort(), mean(), sqrt(), abs(), order(), cumprod().* Use *help()* to see how they work. Before starting create an R-script where you can save your work.

18. Create the following vectors: (vectors are created using the function *c().* The function gets its name from the word concatenate. However, *seq* and *:* are sometimes more useful.)

    (a) (1,2,3,...,29,30)

    (b) (30,29,...,2,1)

    (c) (1,2,3,...,19,20,19,18,...,2,1)

    (d) (1,2,3) and assign it to the variable a.

    (e) (1,2,3,1,2,3,..., 1,2,3) where there are 5 occurrences of 1.

    (f) (1,2,3,1,2,3,...,,1,2,3,1) where there are 6 occurrences of 1, 5 occurrences of 2 and 5 occurrences of 3.

    (g) (1,1,...,1,2,2,...,2,3,3,...,3) where there are 10 occurrences of 1, 20 occurrences of 2 and 30 occurrences of 3.

    (h) $\left(2, \frac{2^2}{2}, \frac{2^3}{3}, \ldots, \frac{2^{20}}{20}\right)$

    (i) (“label 1”, “label 2”, ..., “label 30”)

19. Calculate the sum of the vectors (1,2,...,5) and (1,2,...,15). What happens?

20. Calculate the following

    (a) $\sum_{i=1}^{25} \left(\frac{2^i}{i} + \frac{3^i}{i^2} \right)$

    (b) $\sum_{i=1}^{20} (i^2 + 2i^3)$

21. Use the following command to create two vectors:

    ```
    set.seed(12)
    x <- sample(0:500,250,replace=T)
    y <- sample(0:500,250,replace=T)
    ```

    (a) Create the vector $(y_2 - x_1, ..., y_n - x_{n-1})$.

    (b) Create the vector $(x_1 + 2x_2 - x_3, x_2 + 2x_3 - x_4, ..., x_{n-2} + 2x_{n-1} - x_n)$

    (c) Calculate $\sum_{i=1}^{n-1} \frac{e^{-x_{i+1}}}{x_i + 10}$

    (d) Pick out the values in y which are larger than 100.

    (e) What are the indices of y, where the element is larger than 100?

    (f) Create the vector $\left(|x_1 - \bar{x}|^{1/2}, ..., |x_n - \bar{x}|^{1/2}\right)$, where $\bar{x}$ is the sample mean of the vector x.

    (g) Sort the values of x and y into decreasing and increasing order.

    (h) Pick out the elements in x at index positions $2, 5, 8, 11, 14, ...$

22. Calculate the following sum:
    $1 + \frac{2}{3} + \left(\frac{2}{3}\frac{4}{5}\right) + \left(\frac{2}{3}\frac{4}{5}\frac{6}{7}\right) + ... + \left(\frac{2}{3}\frac{4}{5}\cdots\frac{20}{21}\right)$

23. What is the difference between the commands /and %/%?

24. Try calculating $1/0$ and $\sqrt{-1}$. What do the results mean?

5 Arrays and Matrices

Useful functions: *matrix(), %*%, solve(), t(), rnorm(), apply(), outer(), colSums(), rowSums(), rowMeans(), colMeans().*

25. Let matrix A be

$$A = \begin{pmatrix} 1 & 1 & 3 \\ 5 & 2 & 6 \\ -2 & -1 & -3 \end{pmatrix}.$$

    (a) Calculate A^3.

    (b) Replace the third column by the sum of the other columns.

26. Create a 15 × 10 matrix with elements simulated from the standard normal distribution. Set the seed to 12.

    (a) Calculate $B^T B$.

    (b) Calculate $(B^T B)^{-1}$.

    (c) Calculate the sample means from the rows and columns.

    (d) Find the number of entries in each row which are greater than 0.5.

    (e) Find those pairs of columns whose total sum is greater than 6.

27. Create the following matrix A. Your solution should not involve typing in all the entries of the matrix separately.

$$A = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 7 & 8 \end{pmatrix}$$

28. Use matrix A from exercise (27) to create the following matrix

$$B = \begin{pmatrix} A & A^T \\ A^T & A \end{pmatrix}.$$

29. Solve the following system of linear equations:

$$\begin{aligned} x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 &= 7 \\ 2x_1 + x_2 + 2x_3 + 3x_4 + 4x_5 &= -1 \\ 3x_1 + 2x_2 + x_3 + 2x_4 + 3x_5 &= -3 \\ 4x_1 + 3x_2 + 2x_3 + x_4 + 2x_5 &= 5 \\ 5x_1 + 4x_2 + 3x_3 + 2x_4 + x_5 &= 17. \end{aligned}$$

Solve the system by considering the matrix equation $Ax = y$. Use the special structure of A such that you can easily generalize the solution for a larger problem with the same structure, hence your solution should not involve typing every element of A separately.

30. In R, a list is a vector where each element can be of different type. Try creating a list where the first element is a matrix and the second is a vector. Assign a new value to one of the elements in the matrix that is contained in the list.

6 Functions

31. Suppose that you have a n-length vector x. Write the following functions that take x as an argument and returns:

    (a) Vector $(x_1, x_2^2, ..., x_n^n)$

    (b) Vector $\left(x_1, \frac{x_2^2}{2}, ..., \frac{x_n^n}{n}\right)$

32. Write a function that takes two arguments x and n, where $x \in \mathbb{R}$ and $n \in \mathbb{Z}^+$. The function should return:

$$1 + \frac{x}{1} + \frac{x^2}{2} + \frac{x^3}{3} + ... + \frac{x^n}{n}$$

33. Write a function that calculates the sample mean vector from a matrix A. Use that function to create a function that calculates the sample covariance matrix from a matrix A.

34. Create a function that takes a matrix A as an argument. The function should return a matrix where every odd element of the matrix is doubled and every even number of the matrix is left unchanged. You can assume that A only has integer values.