Lab 1

Statistical Inference, Spring 98

In this Lab, you will learn some basic concepts and features of R language. Throughout this, you will read tiny tips and clues about each of the topics; then you should answer the related questions. Some questions may be easy and some would be trickier. The main objective of this material is to motivate you for exploration, so feel free to search the net (e.g. StackOverflow, StackExchange, etc.) and mine the knowledge yourself. Also note that, to be brief and useful, it is assumed that you are familiar with classic programming languages fundamentals.

a) Variables and Types

In R, you can use the operator "=" or "<-" for value assignment. Below is an example:

```
> x = 3
> y = 2
> a <- x + y
```

1- What is the difference between these two?

Operators can be used within the variables or values of the same type. Sometimes, we call these types as "Classes".

2- What are the five basic atomic classes of R?

If you try to use one operator within two variables/values of different types, R tries to cast the type of variables; sometimes it is possible and sometimes it is not.

3- Make an example that R does successful casting and an unsuccessful one (preceding an error) when using the operator "+".

4- Note the following script. First, we try to sum over two values "FALSE" and "TRUE" and put the result in a variable named "sum". Then we try to see the result. Why is the result not in the Boolean format? What is the type of the variable "sum"? How do you know it? Try to show the value of the variable "sum" in the Boolean format and indicate the related function.

```
> sum <- FALSE + TRUE
> sum
[1] 1
```

b) Data structures

5- What are the main data structures of R? Name at least five of them.

c) Vectors

Vector is the basic data structure in R. It is commonly said that everything in R is a vector (it is somehow true). With the good knowledge of how vectors work, you can guess the basic behavior of other data structures, even if you did not confront them before.

6- How we can create a vector? What is the function for creating vectors is named after?

7- Can you put values 5 (integer), "Truck" (character) and FALSE (logical) all in one vector? Do you get an error? Briefly explain what happens.

8- Try to create a numeric vector consisting of all integers from 10 to 150 (sequence). Name it "first_feature".

Assume that the values in "first_feature" are values observed from a sample of some population. Now, you can see why R is basically a statistical language:

9- Find the sample size of "first_feature".

10- What are the mean, variance, std. deviation and median of "first_feature"?

11- Plot this vector against its indices. Include the plot in your report.

12- Create a vector consisting of 15^{th} to 27^{th} elements of vector "first_feature". Name it "x".

13- Create vector "y" that has the 15^{th}, 18^{th}, 21^{st}, 22^{nd} and 53^{rd} elements of the vector "first_feature".

14- Now we have two numeric vectors, named "x" and "y". Create a vector "z" that binds all elements in "x" and "y".

15- Create the "string_1" vector with 6 elements as follows. Find how many of these elements contain the letter 'a' or 'l'. Do not use "for" and "if" statements.

```
> string_1 <- c("barney","robin","ted","lily","marshall","tracy")
```

d) Factors

R can handle categorical variables as well as numerical variables. Let us see how it works:

16- What are factors? What's the difference between factors and vectors in R?

We have created a factor named "second_feature" from a vector consists of geographical directions.

```
> directions <- c("West","East","East","South","West","West")
> second_feature <- factor(directions)
```

17- Try to change the first element of factor "second_feature" to value "North". Is it possible? Why?

To find levels of factors, there is a function called "levels()". It returns a vector of character type (why?) which contains all values of categories.

```
> levels(second_feature)
[1] "East"  "South" "West"
```

18- Add a new level "North" to the factor "second_feature".

19- Now try to change first element of factor "second_feature" to value "North" again. Report what happens and the reason for that.

e) Missing values

In reality, there are many cases that observations from some samples are not available.

20- How do we represent unavailable values in R?
21- Change the value of the first element of vector "first_feature" to unavailable.
22- Now try to calculate the mean of "first_feature". What happens? Why? Suggest a solution.
23- Assume that we have a vector with some unavailable values. Print out which indices have the unavailable value. (Tip: "which()" function would be useful, also make some statements about what it does.)
24- What is the "NULL" value in R? Can it be used as an unavailable value? What are the use cases?

f) Lists

25- What are lists in R? What's the difference between R lists (generic vectors) and R atomic vectors?

26- A nested list has been created below. Find out what's the difference between index R lists with "[ ]" and "[[ ]]" notations. Explain. (Tip: Notice the data type and class that each one returns.)

```
> list_1 <- list(7, 8, 9, list(6, 4, "abc"))
```

g) Naming

In R, we can add names to objects. Consider the named list created below:

```
> named_list <- list(a=1, b=2, c=3, d=c(4,5,6,7))
```

27- Compare results of the following commands. Can we say that "$" operator is just a syntax sugar? Why?

```
> named_list["b"]
```

```
> named_list[["b"]]
```

```
> named_list$b
```

h) Data Frames

Data frame is the most important data structure in R. In summary, data frame is a list of [named] atomic vectors with some constraints over the data structure. All atomic vectors in a data frame should have the same length.
A data frame could be seen as a collection of features (variables) sampled from the population, where each row represents one observation (a sample point) and each column represents a feature.

There is a data frame named "Orange" loaded by default in R. Answer the following questions:

28- How many features does this data frame have? (Find it using a function, not manually!)

29- How many of sample points does this data frame have? (Again, find out using a function.)

30- Extract feature "circumference" as an atomic vector into a variable named "f3". You can do this using the feature's name or its index; explain both methods.

31- What type of data structure is "f3"? Explain. Apply function table to "f3" and interpret the result.

32- Which variables in "Orange" are categorical variables? How can you make sure?

33- Extract all features of the sample point indexed by "29". Put it into a variable named "s29".

34- What type of data structure is "s29"? Explain.

35- Extract all observations that have the value 3 for their variable "Tree".

36- Extract features "Tree" and "age" only for the first 10 observations, and put them all in one data structure.

37- Find the median of feature "age" without extracting the feature into another variable.

i) Export and Import

Many datasets in statistical researches are loaded into the computers using a file with "CSV" format.

38- Create a new data frame that consists of the last 15 sample points of "Orange". Name the variable "df_1"

39- Export "df_1" into a csv file, named "df_1.csv". Use R commands, don't use R Studio convenient features.

40- Now, import "df_1.csv" file as a data frame into a variable "df_2".

j) Notes

Find and think about the following tips and questions. Answering them would be important to increase your knowledge and expertise in R; but it is not necessary to include the answers in your report:

- Note the options of the functions you have used for exporting and importing csv files. Find out use cases for each one.
- There is a rich data structure in R named matrix. It has a variety of functions and features, which makes it a powerful mathematical structure. It is widely used for implementing machine learning algorithms; but we're not covering it in this course.
- In some cases the data extracted from a data frame might be converted automatically to a different data structure. Find out the cases, and find out how to convert between data frame and vectors bidirectionally.
- Sometimes factors could behave in a way that you are not expected. The easy way is to convert them to vectors; but it is usually not a good idea. Try to learn how to work with factors correctly. Don't sweep it under the carpet!
- Find out how coercions occur in R. That helps you not to be surprised.
- Try to learn R language operators. There are cases that you can use operators instead of functions.
- Using loops in R is as bad as using "goto" in C. There are many functions such as apply family, aggregation family, etc. to avoid using loops. Be creative and try to avoid using loops as much as possible.