R for Beginners

Benjamin Utting

Introduction

This tutorial assumes basic knowledge of statistics, including descriptive statistics (mean, median, mode, etc.), types of data (qualitative/quantitative, discrete/continuous, ordered/ordinal), and statistical tests (e.g. t-test, chi-square test).

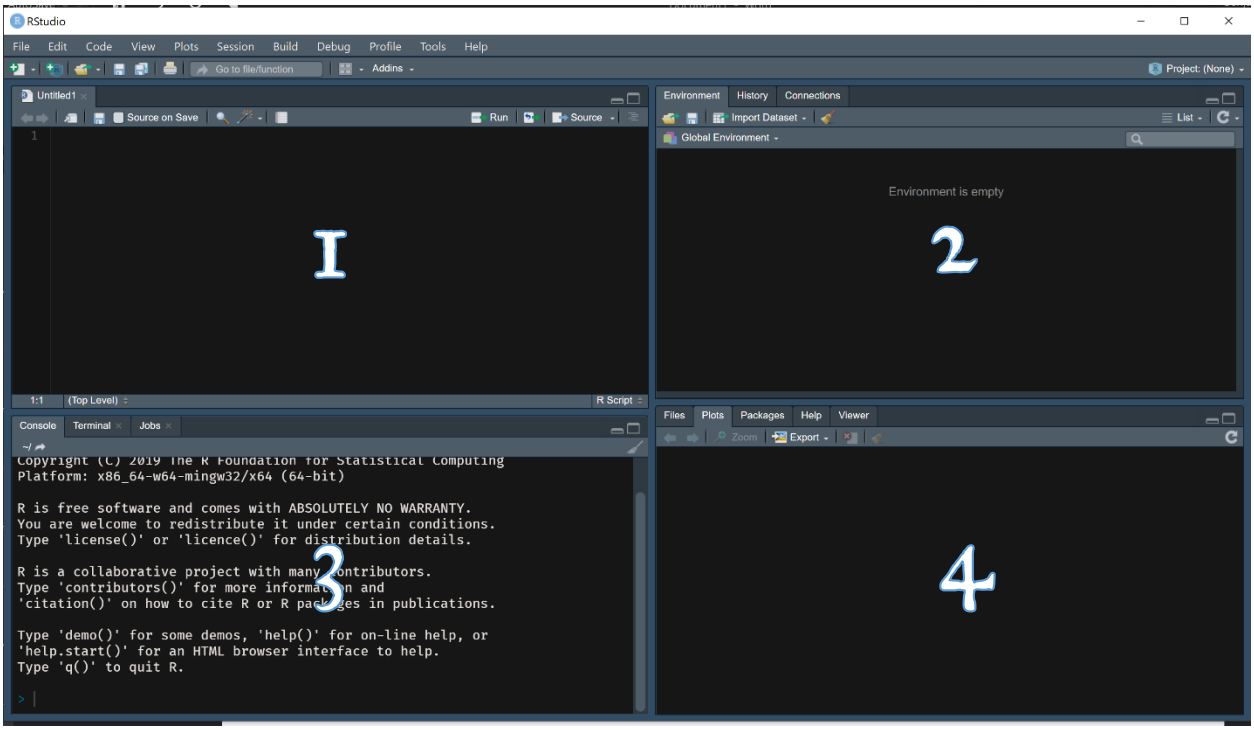
You will need a computer with internet access.

Software to have installed

- R (https://www.r-project.org/)
- RStudio (https://www.rstudio.com/)
    - RStudio is a GUI (graphical user interface) that makes R more easily accessible. It is not necessary to have RStudio, but it makes things easier.

*If you are having trouble installing R or RStudio, you can find YouTube tutorials on how to make things work. A major part of coding is knowing how to find help on the internet.

Anatomy of RStudio



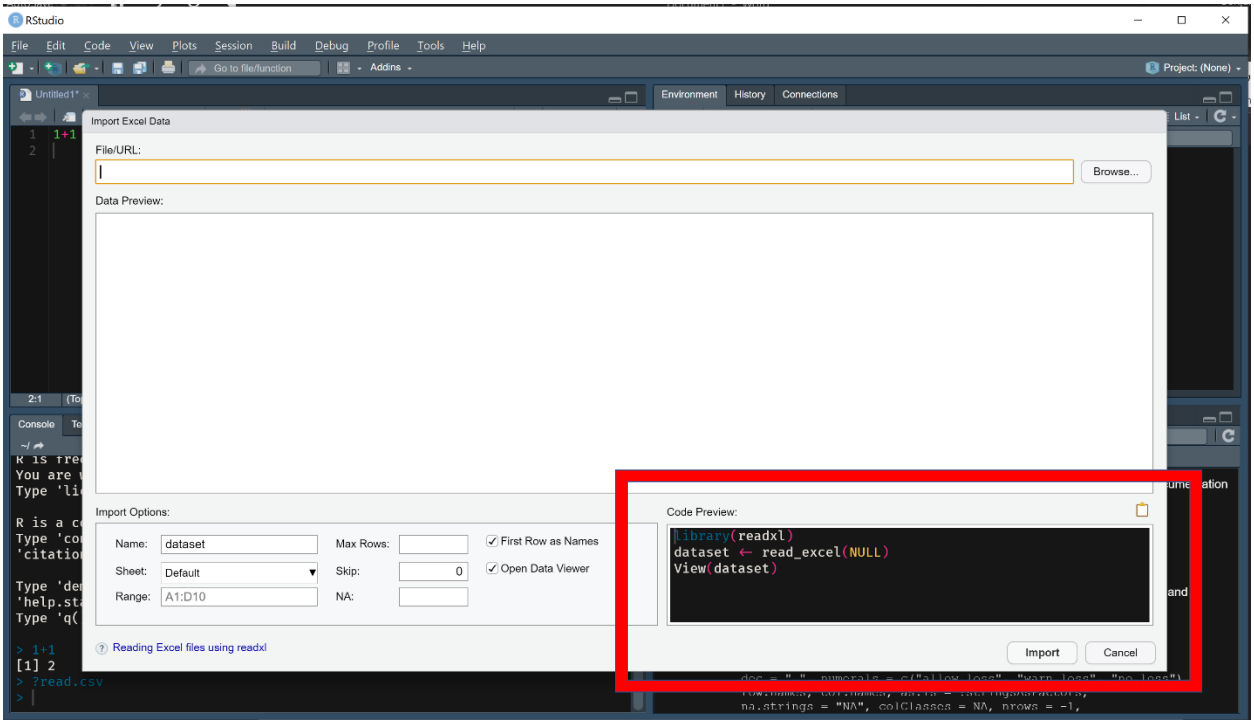
1. Source/Script
    a. This is where to write your code. You can save scripts as .r files for easy access and reproducibility.

2. Environment/History
    a. This is where you can see datasets loaded into R, subsetted data, and other variables that are currently defined.
3. Console
    a. This is where code from your script is run. You can write calculations directly into the console, but these are not saveable. Be sure to do the majority of your work in a script.
4. Files/Plots/Packages/Help
    a. See directories (where R is pulling files from), plots, packages, and help.

*Note that you can change the ordering of these panes to whatever works best for your workflow.

Importing data

- Know what type of file you have (e.g. .xlsx, .csv, etc.)
- For Excel files, using the 'Import Dataset' button will generate three short lines of code that you can copy into a script.



- If you have a .csv file, make sure your 'working directory' is set to the part of your computer where your data are stored, and use the command "`read.csv(yourdata.csv)`".
- Rename your dataset to something manageable using the "<-" operator (e.g. `d <- thisdatasethasareallylongname_copy2_cleaned`)

Data formats in R

R can interpret data in many ways. This is part of what makes R flexible for data analysis, but sometimes this flexibility can backfire. For example, if you have a list of lithic raw materials (e.g. 1, 2, 3, 4, 5), R can read this as a continuous (as opposed to a discrete) value. This can result in a lot of confusion. Before you begin any type of data exploration or statistical analysis, be sure to get your data formats right!

R has six main data types that you will most likely be working with. These are:

1. Character (`"apple", "orange", "pear"`)
2. Factor (`red, orange, yellow, green, blue`)
3. Numeric (`1, 2.1, 3.152, 4.102, 5.3110129`)
4. Integer (`1, 2, 3, 4, 5`)
5. Logical/Boolean (`TRUE, FALSE, TRUE, TRUE, TRUE`)
6. Complex (`e, i, pi`)

The first two (character and factor) are often interchangeable for statistical analyses, but it's common practice to convert your qualitative levels into factor data. Also, R will always read "T" as TRUE and "F" as FALSE.

R can store data in a number of different formats, but most of you will be using data tables in your analyses. Data tables include observations (rows) and variables (columns). Make sure your variable names are stored in the top row, and that they are 1) easy to use and 2) don't have spaces. The following is an example data table. Let's call it "mydata".

artifactID	site	bagNumber	trench	layer	length	width	thickness	rawMaterial
0001	HangTrong	0001	1	8609s3	25.3	10.2	2.0	limestone
0002	HangTrong	0001	1	8609s3	19.1	15.2	4.2	chert
0003	HangMoi	0002	3	9223	11.2	9.2	5.5	quartz

Each column contains only one variable, and each row contains only one observation. The "artifactID", "inventory", "bagNumber", "trench", "layer", and "rawMaterial" columns would be factor data, and the "length", "width", and "thickness" columns would be numeric data. Note that the "artifactID" and "trench" variables are numbers, but also factors.

Basic vocabulary for working with data

Operators:

- Addition: +
- Subtraction: –
- Multiplication: *
- Division: /

- Exponents: ^
- Greater than: >
- Less than: <
- Greater than/equal to: >=
- Less than/equal to: <=
- Exactly equal to: ==
- Is not equal to: !=

To read variables in data tables, use the "$" symbol. For example, to view raw materials from the above data table, use "`mydata$rawMaterial`".

Exploratory data analysis and data cleaning

- Use `str(data)` to return variable formats. If your data are not in the right format, use these commands to convert data:
  - `as.factor(data$variable)`
  - `as.numeric(data$variable)`
- Use `summary(data)` for simple descriptive statistics
- Use `hist(data$variable)` for a histogram of your data (make sure it's continuous!)

You will also need to subset data to make it useable. Subsetting involves dividing your data into different sections based on different criteria. For example, if I am interested in comparing archaeological assemblages from two sites, I would need to tell R which data come from which sites. This is easily accomplished with the 'subset' function.

Example:

If I am interested in dividing "mydata" by site, I would execute the following command:

```
htc <- subset(mydata, site == "HangTrong")
hmc <- subset(mydata, site == "HangMoi")
```

Note that a) I use the double equals sign (==), and that I defined each subset with a short, but immediately recognizable name.

Installing and using packages

Packages are a big part of what makes R so appealing to scientists. Packages are sort of mini-programs that are tailored to fit certain purposes. For example, some packages include functions to make highly customizable graphs, calibrate radiocarbon dates, or create 3D maps from digital elevation matrices. To install packages, use the "`install.packages("mypackage", dependencies = TRUE)`" function. This page has a list of packages that you might find useful for your analysis: https://github.com/benmarwick/ctv-archaeology.

After you successfully install a package, you need to use the command "`library(mypackage)`" to use it. If you have questions about certain functions of the package, you can navigate to the "Packages" pane, find the package you're using, click on it, and find a list of functions. Otherwise, you can use "`?function`" to get help for that particular function.

Basic statistical tests: the t-test and the chi-square test

Conducting statistical tests in R is easy. R automatically installs a "stats" package which includes a set of commonly used statistical tests. Here, I review how to conduct two simple statistical tests.

Student's t-test

- If I am interested in testing the null hypothesis that there is no significant difference in length (continuous variable) in artifacts from Hang Trong and Hang Moi, I would use a student's t-test. To execute this function, use the command `t.test(htc$length, hmc$length)`.

Chi-square test

- If I am interested in testing the null hypothesis that there is no significant difference in raw material distribution (discrete variable) in artifacts from Hang Trong and Hang Moi, I would use a chi-square test. To execute this function, I would need to create a contingency table (`rmtable <- table(mydata$site, mydata$rawMaterial)`), and then run a chi-square test on the table using `chisq.test(rmtable)`.

Please note that there are assumptions that are associated with each test. For example, the t-test loosely assumes a normal distribution (but is somewhat robust to skewed data), and the chi-square test assumes that the expected cell counts are greater than or equal to 5 in 80% of the cells or more. This tutorial teaches you the implementation of statistical tests in R but does not go into depth on the statistics themselves. It is important to decide which test to use, and to justify the use of different tests in your own research.

Getting comfortable in R with swirl

Swirl is a package that teaches you how to use R in R. It has many different mini-courses, and I highly recommend taking a look at it if you want a slightly more in-depth understanding of how R works. You can install swirl with the command "`install.packages("swirl")`", and access it using "`library(swirl)`". Learn more about swirl here: https://swirlstats.com/.