1 - Introduction to R

R is both a programming language and an interactive environment for data exploration and statistics. We will first use R as an interactive environment.

Working with R is primarily text-based. The basic mode of use for R is that the user types in a command in the R language and presses enter, and then R computes and displays the result.

We will be working in RStudio. If you're working on a personal computer or laptop, you'll need to download the R language system, as well as RStudio, since RStudio depends on the R language:

> R downloads:
> - Windows: https://cran.rstudio.com/bin/windows/base/
> - macOS: https://cran.rstudio.com/bin/macosx/
> - Linux: ask me
> R-Studio downloads:
> - https://www.rstudio.com/products/rstudio/download/

Alternatively, you can login remotely to a CSSE department Linux lab machine and use RStudio remotely:

> Remote Access to CSSE Linux lab environment:
> - https://cssegit.monmouth.edu/cshelp/csseremote/-/wikis/MUCSremote

I will be working with R and RStudio primarily in the remote CSSE Linux lab environment.

The R Console

The main way of working with R is the console, where you enter commands and view results. You can start the R console by running the R command. Note that this demonstration of the bare R console is directly transferable to the R console in the RStudio environment (see below).

Getting Help

In the R console, try running

```
help.start()
```

This will start the HTML interface to on-line help (using a web browser window or tab).

Using R as a fancy calculator

In the R console, type `1+1` and press enter. R displays the result of the calculation.

```
1+1        (your input)
## [1] 2   (R console output)
```

The `+` is called an operator. R has the operators you would expect for basic arithmetic: `+ - * / ^`. It also has operators that do more obscure things.

The `*` multiplication operator has higher precedence than the `+` addition operator. We can use parentheses if necessary `( )`. Try `1+2*3` and `(1+2)*3`. When in doubt about operator precedence, use `()`.

I recommend using spaces to make code and commands easier to read. This will matter more when we start saving R programs to R program files.

We can make comparisons with the comparison operators `== < > <= >= !=`. Comparison operations always produce a *logical* or *"boolean"* value, `TRUE` or `FALSE`. Note the double equals, `==`, for equality comparisons.

```
2 * 2 == 4
## [1] TRUE
```

Generate two pseudo-random normal vectors of x- and y-coordinates:

```
x <- rnorm(50)
y <- rnorm(x)
```

Plot the points in the plane. A graphics window will appear automatically:

```
plot(x, y)
```

See which R objects are now in the R workspace:

```
ls()
```

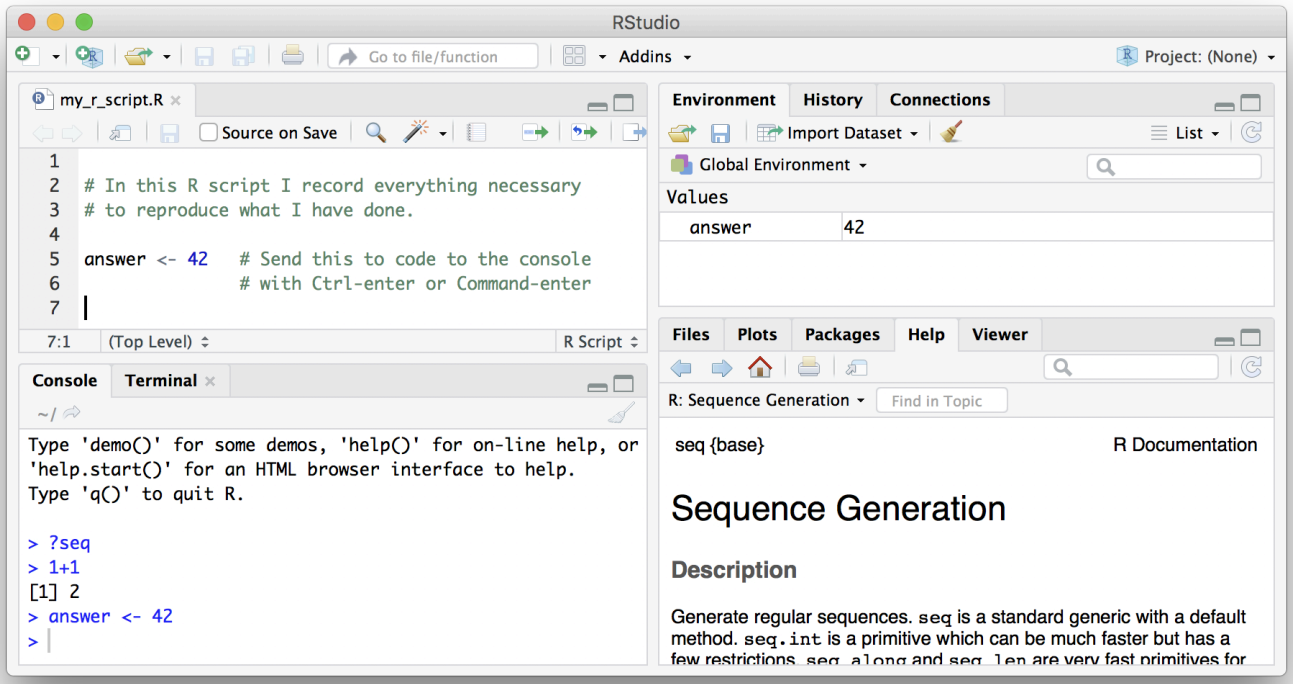
Remove objects no longer needed (Clean up):

```
rm(x, y)
```

The RStudio Integrated Development Environment

RStudio surrounds the R console with various conveniences. In addition to the console panel, RStudio provides panels containing:

- A text editor, where R commands can be recorded for future reference.
- A history of commands that have been typed on the console.
- An "environment" pane with a list of variables, which contain values that R has been told to save from previous commands.
- A file manager.
- Help on the functions available in R.
- A panel to show plots.



Try the above demo commands in the R console of RStudio.