

Lab 1: Introduction to R and RStudio

Before you start, be sure to load the packages you need. I always do this in an initial 'set up' code chunk at the top of a page. You could also use the function `library()` instead of the function `require()`:

```
require(mosaic)
require(tidyverse)
```

Learning Outcomes

At the end of today's lab you should be able to do the following:

1. Demonstrate knowledge of how to interact with R using RStudio
2. Identify and employ the basic tools of R Markdown to understand R Notebooks

Basics

The goal of this lab is to introduce you to R and RStudio. Let us clarify which is which:

1. R is the name of the programming language itself, and
2. RStudio is a convenient interface allowing you to use R easily.

I encourage you to explore beyond what the labs dictate. A willingness to experiment will make you a much better programmer. Before we get to that stage, however, you need to build some basic fluency in R. Today we begin with the fundamental building blocks of R and RStudio: the interface, R markdown, and basic commands.

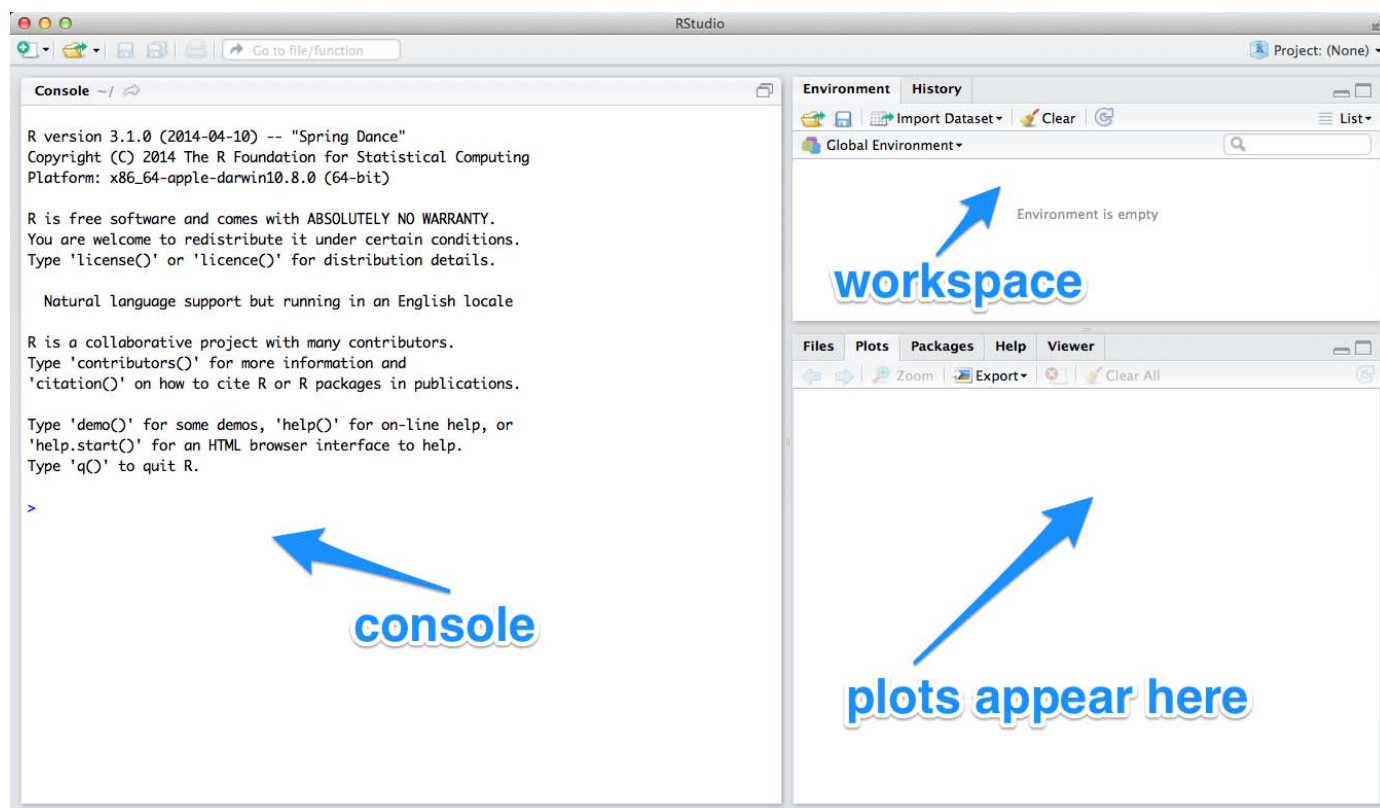


Figure 1: R interface

The panel in the upper right contains your *workspace* as well as a history of the commands that you've previously entered. Any plots that you generate will show up in the panel in the lower right corner.

The panel on the left is where the magic happens.

It's called the *console*. Every time you launch RStudio, it will have the same text at the top of the console telling you the version of R that you're running.

Below that information is the *prompt*.

As its name suggests, this prompt is really a request, a request for a *command*. Initially, interacting with R is all about typing commands and interpreting the output. These commands and their syntax have evolved over decades (literally) and now provide what many users feel is a fairly natural way to access data and organize, describe, and invoke statistical computations.

To get you started, try the following commands at the R prompt (i.e. right after `>` on the console).

1. `556 + 232`
2. `72/4 = 72 ÷ 4 = $\frac{72}{4}$`
3. `sqrt(144) = $\sqrt{144}$`
4. `log(10) = $\log_e 10 \neq \log_{10} 10$`

You can either type the command manually or copy and paste it from this document. So, as you can see, R is a good calculator. Later, we shall also see that R can be used to graph functions and create a variety of interesting graphics.

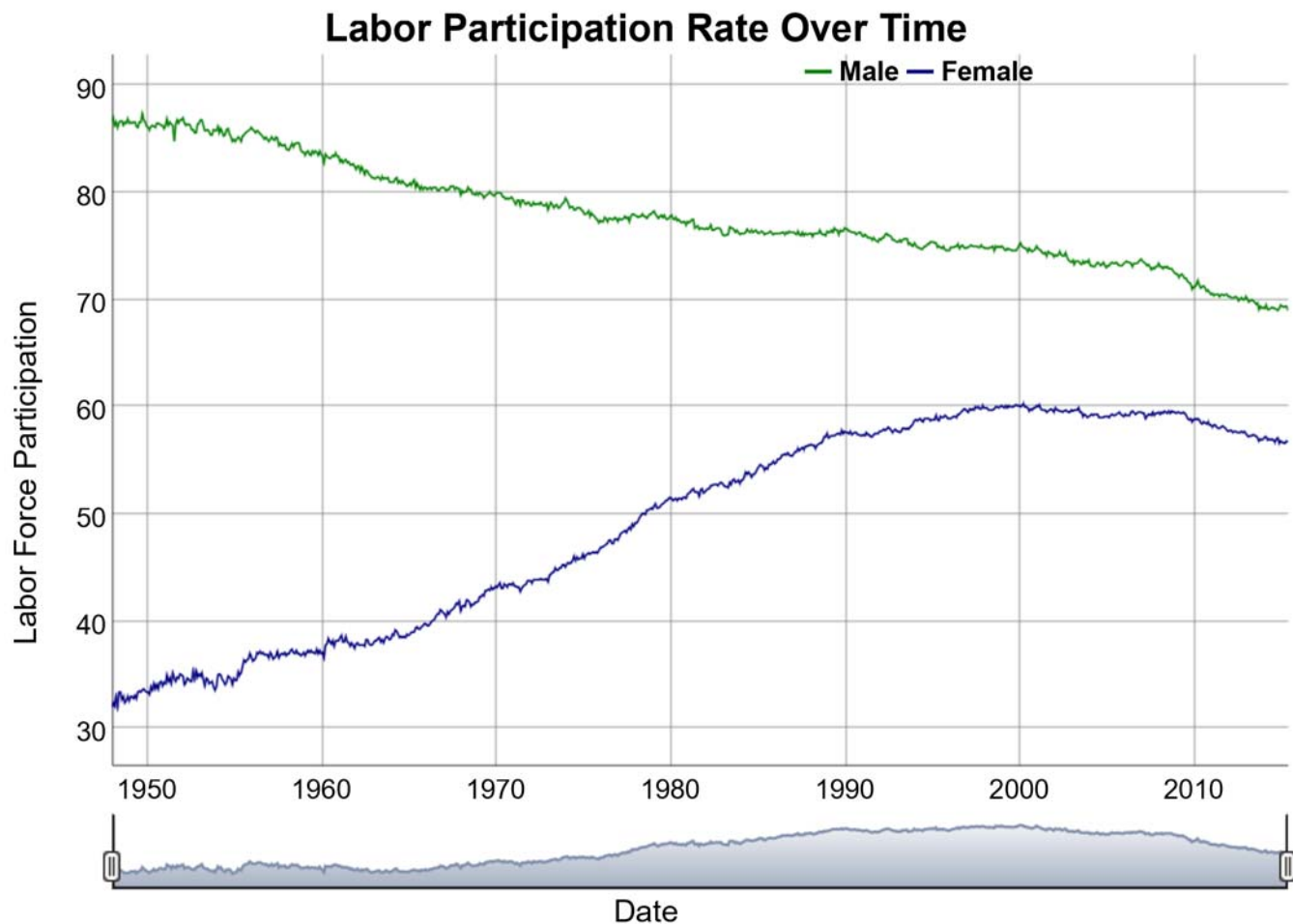
Reference point

Later, I'll talk to you about participation levels. To get a sense of what you can do with data and interactivity in R and R Markdown with HTML, take a look at this graph at FRED: research.stlouisfed.org/fred2/graph

[/?id=LNS11300001,LNS11300002,#](https://research.stlouisfed.org/fred2/graph/?id=LNS11300001,LNS11300002,#) (<https://research.stlouisfed.org/fred2/graph/?id=LNS11300001,LNS11300002,#>)

R can make a dygraph like FRED

Here you can see a lovely `dygraph` that can be generated using `htmlwidgets` (go to www.htmlwidgets.org (<http://www.htmlwidgets.org/>) to see a variety of other advanced options for what you can do when you become and R Champion). You saw this graph in my presentation, but I wanted also to show you that it can be produced in an html document that would be online.



R Markdown and R Notebooks

This document is based on an R Notebook or R Markdown document. R notebooks are meant to be like Lab Notebooks in the sciences where you “do (data) science” by:

1. using code chunks to perform an activity
2. seeing the output of the code chunk, and
3. commenting on the code and its output in prose between the code chunks.

R Notebooks are *interactive* documents. The notebooks can check the code while you write it, suggesting corrections where errors appear and displaying output when you run a code chunk. We shall see this in action several times during the workshop.

Practice what you've learned by creating a brief CV of your **fictional alter ego** in an R Markdown document.

First, open an R Markdown document:

1. Go to File -> New File -> R Notebook
2. The file will be pre-populated with some content. Ignore it for the moment.
3. You will lightly edit the top of the page between the --- lines (you'll only change the title)
4. Then, you'll mostly edit after that in the *body* of the document

Using the R Markdown quick reference (In R Studio navigate to Help -> Cheatsheets -> R Markdown Reference Guide), figure out how to ensure the following:

1. The title of the file (and the document itself) should be your name
2. You should include headings for (at least) education or employment
3. Each of the sections should include a bulleted list of jobs/degree(s)
4. Highlight the year in **bold**
5. Include an image of yourself (either a local file or url).
6. Add a horizontal rule
7. Add a block quote
8. Having done this, go to the top of the page where it says 'Preview' on the arrow next to preview click on the arrow and select 'Knit to HTML.' It will then produce an html document you can compare with other people.

I'll come around and check what people have done.

See an example here: [simonhalliday \(./simonhalliday.html\)](#).

[Back to Simon's R for Econ Workshop page \(./index.html\)](#).