Shiny Workshop Setup

Welcome! This document is to help you prepare for the 2021 NCME R Shiny Workshop. We are very excited to help you learn more about `Shiny`, and we are honored you have chosen to spend some of your time with us.

This workshop assumes that you have some moderate programming experience with R. However, we recognize that the online format of NCME allows for people with a variety of skill levels to feel more comfortable in stretching their boundaries. (Chris believes this because he has recently done the same thing at other virtual conferences over this past year.) As such, we include information on getting set up with R, ***but please understand that we cannot cover any R programming basics beyond what is necessary to get a better understanding of*** `Shiny`. If you are an R novice taking this class, the materials should be easily understandable and applicable after gaining an elementary knowledge of R.

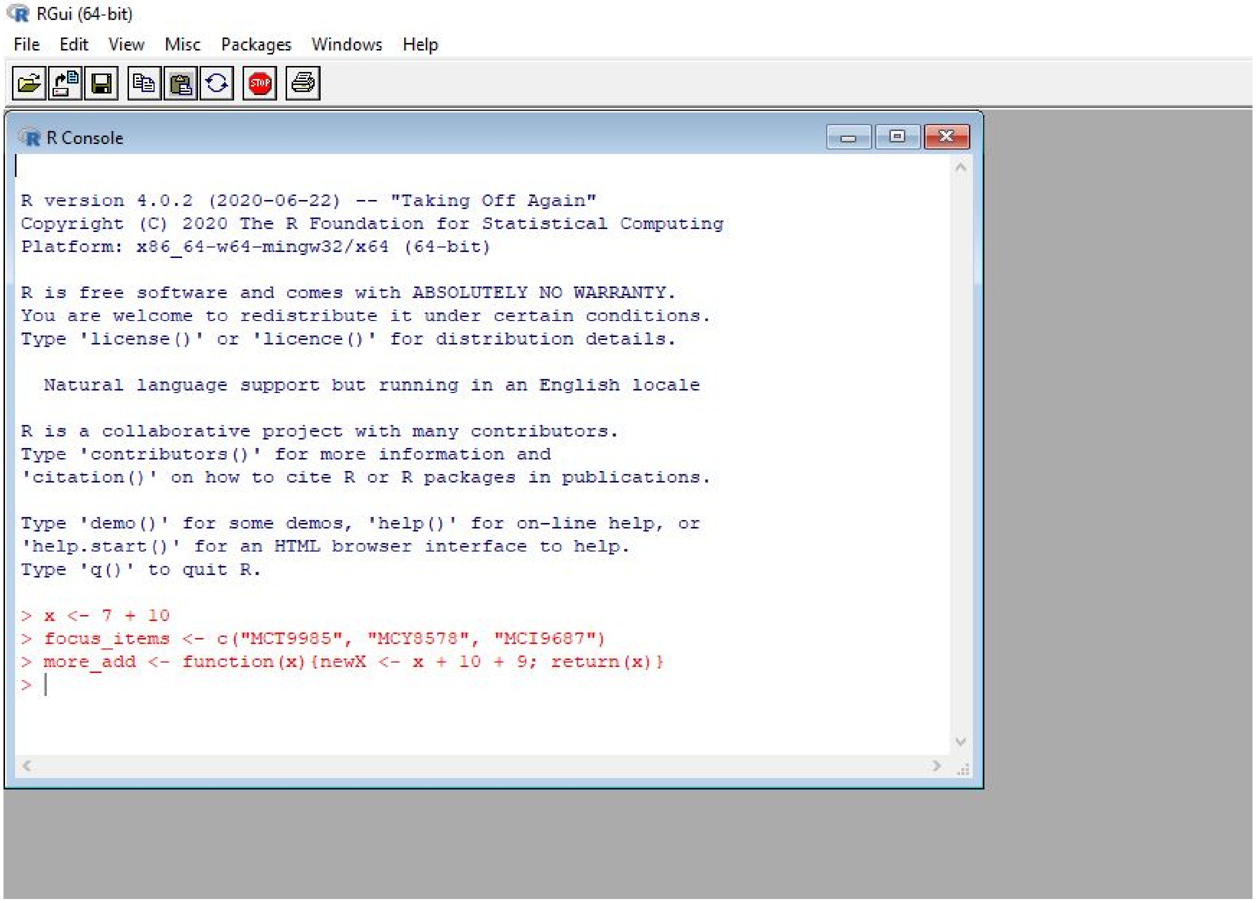
The first part of this document is to help brave novices install R and R Studio, and the second part of the document provides instructions on installing the packages and data that will be used during the workshop. In addition, the presenters are also having “office hours” from 4-6pm EST on Thursday, June 3rd to help anyone who has difficulty getting the correct software / packages downloaded. This Zoom meeting can be accessed by clicking here. Meeting ID: 971 0210 0781 and Passcode: tzzZ=1.

0 R and R Studio Installation

0.1 Installing R

To use R, one must first install the R software. One can download the appropriate version of R (Windows / Mac / Linux) from the Cromprehensive R Archive Network. For those with previous R experience it is worth noting there were some substantial changes made when R transitioned from the 3.x to 4.x version.

Once you have downloaded R, you will naturally be interested to see what it's like to use the software. You'll be confronted with a screen similar to this (your version of R may vary from mine):



```
R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x <- 7 + 10
> focus_items <- c("MCT9985", "MCY8578", "MCI9687")
> more_add <- function(x){newX <- x + 10 + 9; return(x)}
> |
```

Figure 1: Uninformative R GUI

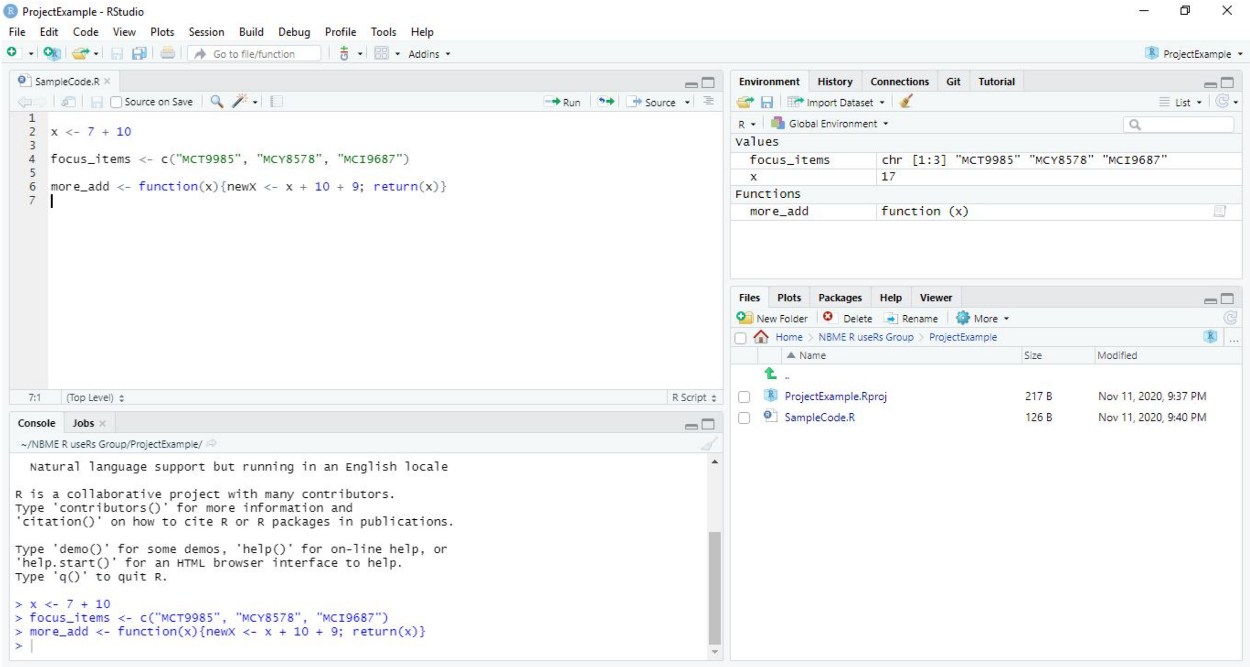
This is the basic R console, where one can enter syntax and execute a variety of R arguments. However, for those of you with SPSS or SAS experience, this interface may seem difficult. There's not much to it, and once you've typed a command and run it, it's not always clear what happened. For instance, you can see I've entered a variety of commands and there doesn't seem to be any additional information available about what has been done.

For these reasons (and more), several developers got together years ago and decided to create an interface to make the R workflow experience more user-friendly and efficient. And thus was born RStudio.

0.2 R Studio

I believe that RStudio is one of the main reasons that R has become so popular in the last decade. In addition to creating the user-friendly integrated development environment (IDE), RStudio employs some of the most prolific state-of-the-art developers and R useRs (including one of my personal heroes, Hadley Wickham). The RStudio group continues to push the boundaries of what R can do by developing unique packages that facilitate workflows and far extend R's capabilities. Much of what you will learn and use in R can probably be traced back to some of these RStudio individuals.

You can navigate to the RStudio installation page by clicking on this sentence. In the vast majority of cases, the Free version of RStudio Desktop will be more than sufficient. RStudio will automatically detect any version of R that is installed, so you can just open RStudio when starting to work.


ProjectExample - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
ProjectExample
SampleCode.R
Source on Save
Run
Source
Environment History Connections Git Tutorial
Import Dataset
List
Global Environment
Files Plots Packages Help Viewer
New Folder Delete Rename More
Home > NBME R useRs Group > ProjectExample
Console Jobs
7:1 (Top Level)
R Script

```
x <- 7 + 10

focus_items <- c("MCT9985", "MCY8578", "MCI9687")

more_add <- function(x){newX <- x + 10 + 9; return(x)}

```

```
Values
  focus_items      chr [1:3] "MCT9985" "MCY8578" "MCI9687"
  x                17
Functions
  more_add         function (x)
```

Name	Size	Modified
..		
ProjectExample.Rproj	217 B	Nov 11, 2020, 9:37 PM
SampleCode.R	126 B	Nov 11, 2020, 9:40 PM

~/NBME R useRs Group/ProjectExample/

```
  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x <- 7 + 10
> focus_items <- c("MCT9985", "MCY8578", "MCI9687")
> more_add <- function(x){newX <- x + 10 + 9; return(x)}
> 
```

Figure 2: Informative R IDE

This is the basic RStudio interface, which is only a small part of what RStudio can do. I would like to point out the top left corner, which is the syntax window where we have defined the same objects that we saw above in the regular R interface. The lower left corner is the console, which replicates what you saw on the page above. When you run syntax in the top left script window, those commands are mirrored in the console. One can also run commands directly in the console, but this leaves a less visible record of what commands have been run. This can be helpful in some specific instances, but for most purposes you'll want to use the top left window. The top right window displays the local environment and shows you how specific variables have been defined. The bottom right window displays the objects saved in the working directory.

0.2.1 Updating R

It's generally a good idea to have the newest version of R installed. The people at R continually work to fix bugs and enhance computational efficiency, and newer R packages (or updated R packages) may only be compatible with newer versions of R. The easiest way to update R is through the `installr` package. The only wrinkle is that the `updateR()` function was designed to be executed in the original R gui. Fortunately the process to install and run`updateR()` is easy; just run each of the following lines one at a time in the old R gui:

```
install.packages("installr")

library("installr")

updateR()
```

From there just follow the on-screen prompts. Running this command will automatically check for updates for all installed packages as well, making it quick and easy to have a fully up-to-date R environment. I also recommend looking over all of the functionality of the `installr` package - it really makes installing complimentary R software easy (e.g. GitHub, MiKTeX, etc.).

0.2.2 Updating R Studio

Updating R Studio is also easy. Open R Studio and go to the `Help` menu at the top, and select `Check for Updates`. It will tell you're using the latest version of R Studio or offer to help you install the new version.

1 Workshop Software

1.1 R Packages

The following syntax installs the packages required for (1) the basics covered in Day 1 of the workshop; (2) extensions of `Shiny` - Day 2 of the workshop; (3) the packages necessary to follow Hadley Wickham's "Mastering Shiny" book; and (4) other related packages that extend the functionality of `Shiny` across a variety of use cases.

Open up a new RStudio session, copy-and-paste the entire chunk of syntax below, and then execute the entire command at once (highlight the text and either click the "Run" arrow or Ctrl+Enter). Installing packages works best when you're starting a new RStudio session because you may get a warning that "One or more of the packages to be updated are currently loaded. Restaring R prior to install is highly recommended." And although you can say "Yes" to "Do you want to restart R prior to install?", this can sometimes end up in a recursive loop where you need to end up completely restarting RStudio anyway.

```
install.packages(c(
  "datasets", "DT", "gapminder", "ggforce", "ggpmisc", "gh",
  "globals", "MBESS", "mvtnorm", "openintro", "ploty", "profvis",
  "readxl", "RSQLite", "shiny", "shinycssloaders",
  "shinydashboard", "shinyjs", "shinyFeedback", "shinythemes",
  "shinyWidgets", "testthat", "thematic", "tidyverse", "vroom",
  "waiter", "xml2", "zeallot"
))
```

That's it! We hope you are as excited to meet us as we are to meet you. See you soon!