

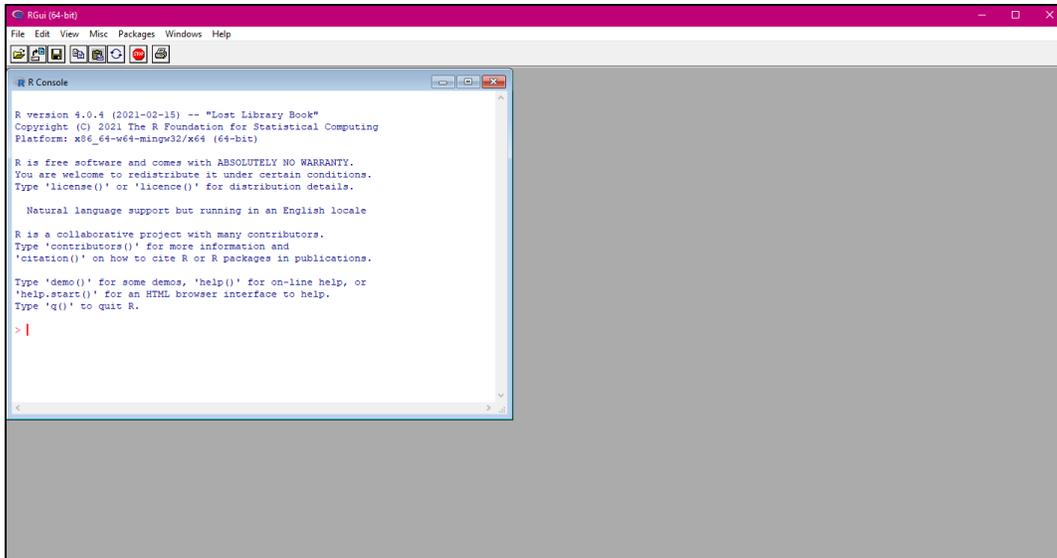
Getting Started with R

This set of instructions will discuss the basics of getting started with R including the interface, panes, and some of the menu options. Please note that this guide is assuming you have already installed RStudio. If you still need to install R and want a guide, check out the “Installing R” pdf.

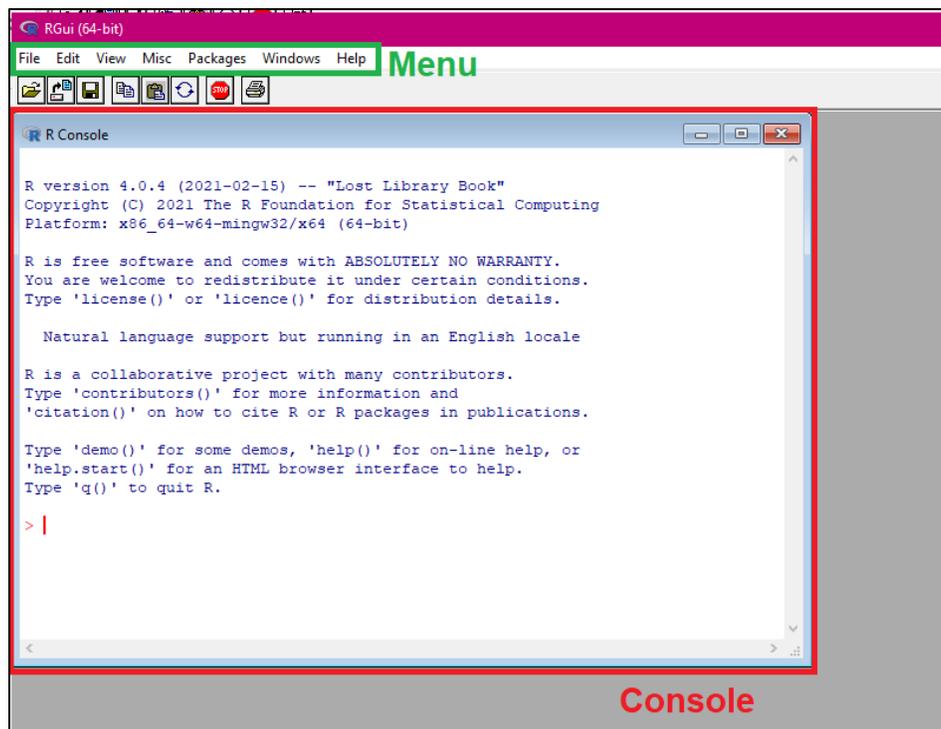
The Basics

R Interface

When you open R, this is what you should see (note: I am on a PC; you might see something slightly different on a Mac):

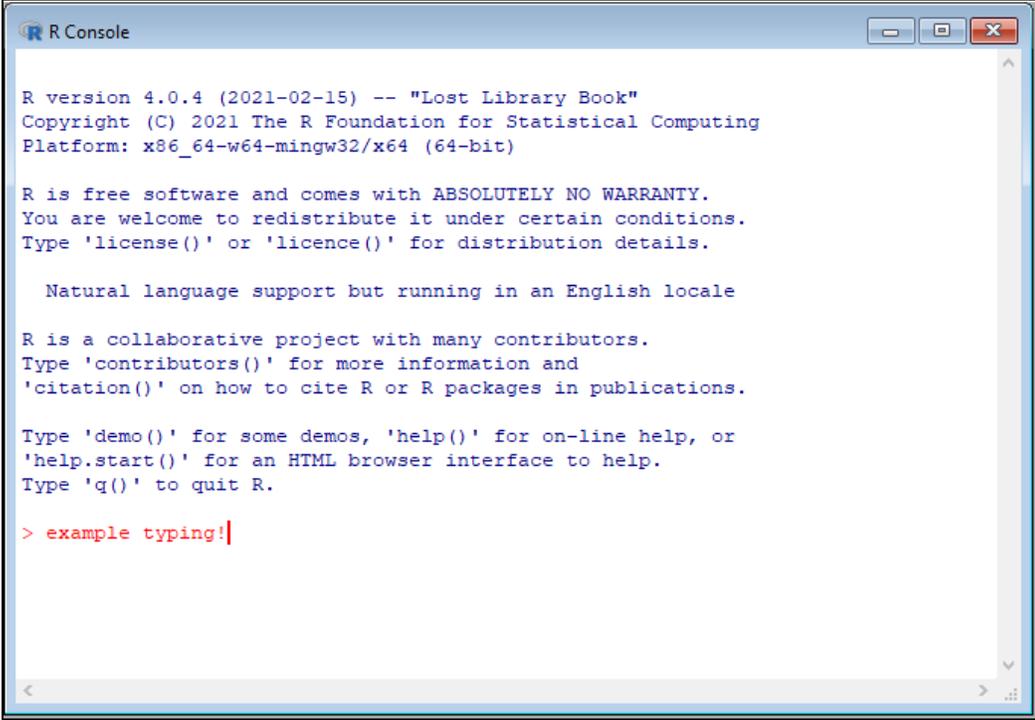


The two most important parts of what you see here are the R console and the menu at the top.



The menu options can be used to help you open and save R scripts (discussed in further detail below), change your working directory and install packages if necessary (both discussed in the “Installing and Loading R Packages Using R” document).

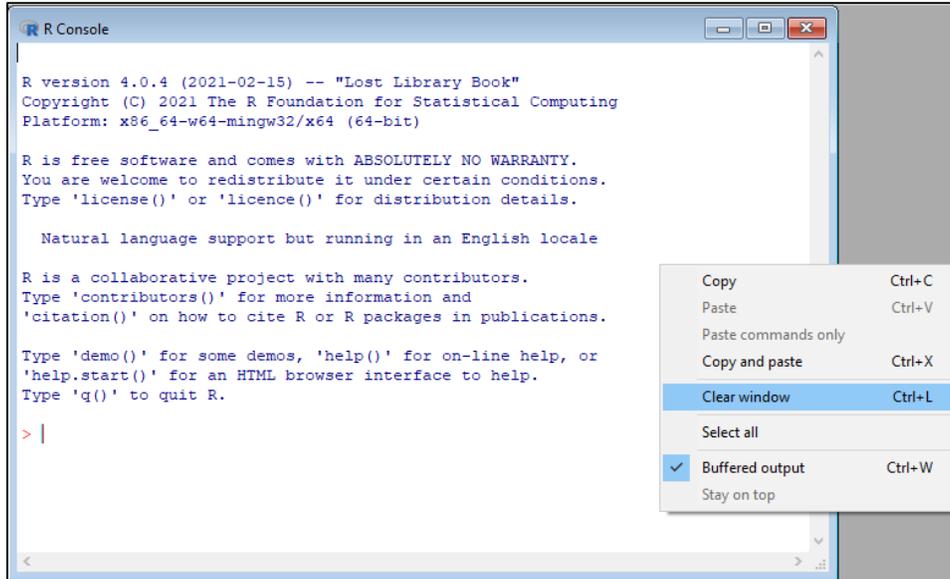
The console is where all of your R code and resulting R output will be displayed. If you click on the console window to activate it, notice that you can type and the text will show up next to the red cursor.

A screenshot of the R Console window. The window title is "R Console". The text inside the console reads: "R version 4.0.4 (2021-02-15) -- 'Lost Library Book'", "Copyright (C) 2021 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.", and at the bottom, a red prompt character ">" followed by the text "example typing!" and a red cursor. The window has standard Windows window controls (minimize, maximize, close) in the top right corner and a scrollbar on the right side.

We will talk about what you can type in the console in a moment. First, a few things about the console itself.

Clearing the Console

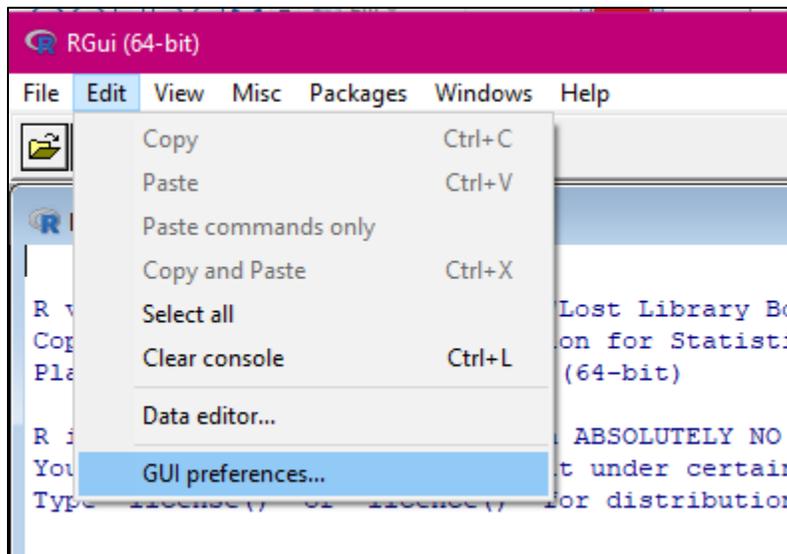
Notice that there is a bunch of blue text at the top of the console right when you open R. This text is just telling you about the version of R you're running and some details about the software. If you want to clear this text (or if you want to clear the code/output from your console at any time), either press **ctrl+l** (that's a lowercase "L") or right-click within the console and select "Clear window."



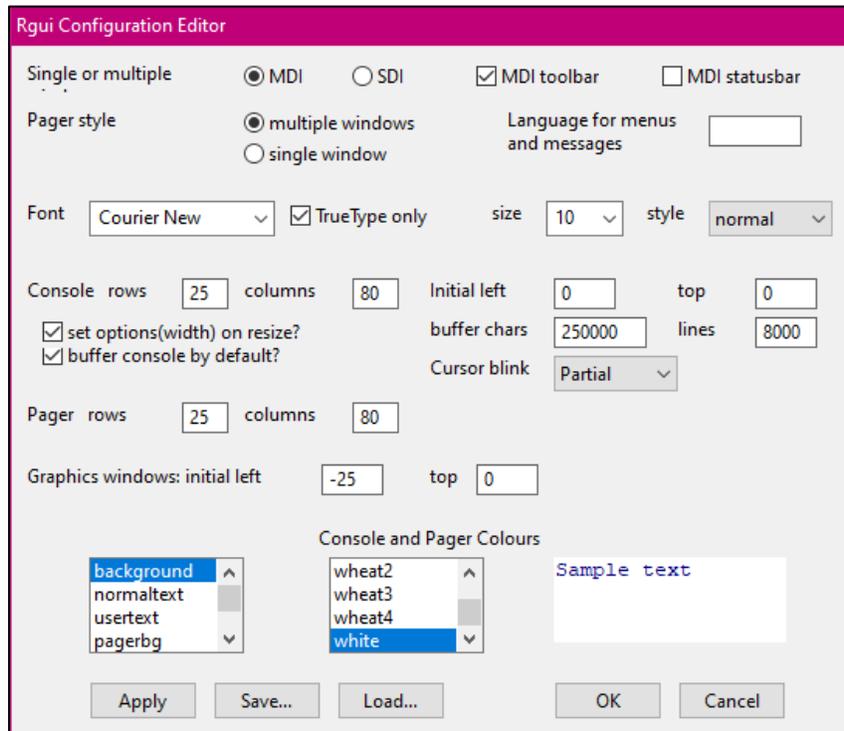
That will erase all the text from the console.

GUI Preferences

You can adjust how text is displayed in the console as well. To do so, go to **Edit → GUI Preferences...**



This will open a pop-up window where you can adjust a lot of the visual settings of the console such as the font, font color, and background color.



Changing the font/display style can be useful if you're having trouble seeing the text using the default settings. For example, I always like to set my font size to 14 and bold when using R in videos:

```
R version 4.0.4 (2021-02-15) -- "Lost Library Book"
Copyright (C) 2021 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.

> |
```

Much easier to see!

R Prompt

R is ready for us to enter a command when you see a line like this:

```
R Console

R version 4.0.4 (2021-02-15) -- "Lost Library Book"
Copyright (C) 2021 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.

> | Prompt
```

This is the prompt, or the line where you can type your command/code. To run a command, type it at the prompt and hit “enter.” Depending on the type of command, this will either return the result of the command or will return a blank prompt again (we’ll see some examples of both in a few pages).

```
> 7+6
[1] 13
> |
```

If you give R an command that is incomplete, it will ask you to complete it by displaying a plus (+) sign.

```
> 35-
+ |
```

Here, I want to subtract something from 35, but I forgot to tell R what to subtract, so it’s asking me for that information. I can type that info in that prompt line with the + and then hit “enter” again to complete the calculation.

```
> 35-
+ 12
[1] 23
> |
```

Help with Code

There may be times when you need help or documentation on a function you are using in R. To access R's "help" library, type the following into the R console and hit "enter":

```
?function
```

where function is the name of the function you want to look up. This will open the online help library in your default internet browser.

As an example, suppose a course I'm taking requires me to use the `dbinom()` function. If I want some more information on this function or forget what needs to be entered in the function, I can get some help.

```
> ?dbinom
starting httpd help server ... done
> |
```

And in Chrome, the following page has opened:

The screenshot shows a web browser window with the address bar displaying `127.0.0.1:15097/library/stats/html/Binomial.html`. The page content includes:

- Page title: Binomial {stats}
- Section title: The Binomial Distribution
- Section: Description

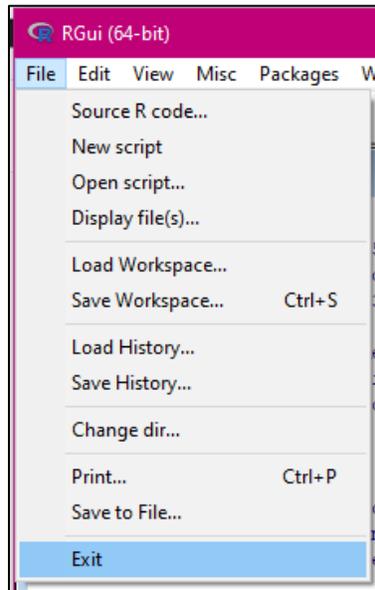
Density, distribution function, quantile function and random generation for the binomial distribution with parameters `size` and `prob`. This is conventionally interpreted as the number of 'successes' in `size` trials.
- Section: Usage


```
dbinom(x, size, prob, log = FALSE)
pbinom(q, size, prob, lower.tail = TRUE, log.p = FALSE)
qbinom(p, size, prob, lower.tail = TRUE, log.p = FALSE)
rbinom(n, size, prob)
```

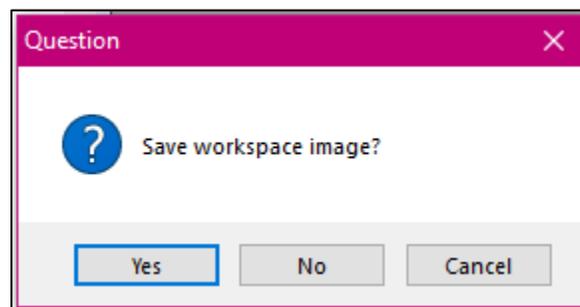
This gives me more info on what the function is, what it does, and what values I need to enter into it to use it correctly.

Exiting R

You can close/exit R by either going to **File** → **Exit** or by clicking the “x” in the corner of the RGui window (not the R Console).



Note that if you have done anything in the R Console, you will be asked in a pop-up window if you wish to save the workspace image.



Typically, you'll just want to click “no.” Further down in this document, we will talk about how you can save your code in an R script so that you can access it again after you've closed R.

Using R as a Calculator

One of the most basic things you can do with R is use it as a calculator. Many of the basic calculator functions are fairly intuitive.

Basic Calculator Functions

Operation	Function	Example
Add	<code>+</code>	<code>> 9+5</code> <code>[1] 14</code>
Subtract	<code>-</code>	<code>> 9-5</code> <code>[1] 4</code>
Multiply	<code>*</code>	<code>> 9*5</code> <code>[1] 45</code>
Divide	<code>/</code>	<code>> 9/5</code> <code>[1] 1.8</code>
Exponent	<code>^</code>	<code>> 10^2</code> <code>[1] 100</code>
Square root	<code>sqrt</code>	<code>> sqrt(100)</code> <code>[1] 10</code>
Exponential function (e)	<code>exp</code>	<code>> exp(1)</code> <code>[1] 2.718282</code>
Log base 10	<code>log10</code>	<code>> log10(10)</code> <code>[1] 1</code>
Log base e (natural log)	<code>log</code>	<code>> log(1)</code> <code>[1] 0</code>
Pi (π)	<code>pi</code>	<code>> pi</code> <code>[1] 3.141593</code>
Factorial	<code>factorial</code>	<code>> factorial(5)</code> <code>[1] 120</code>

Other Basic Functions

Operation	Function	Example
Sequence (of integers)	<code>:</code>	<code>> 1:5</code> <code>[1] 1 2 3 4 5</code>
Concatenate (make a vector of numbers)	<code>c</code>	<code>> c(2, 4, 6, 8)</code> <code>[1] 2 4 6 8</code>
Summing multiple numbers	<code>sum</code>	<code>> sum(2, 4, 6, 8)</code> <code>[1] 20</code>

Assigning Names

In math, you often want to assign a variable name to some value in order to use it in a calculation (e.g., $x = 56$). You can do the same thing in R. There are two “assignment” statements you can use in R (note they both do the same thing):

Operation	Function	Example
Assign a name	<code><-</code>	<pre>> x <- 3 ></pre>
Assign a name	<code>=</code>	<pre>> y = 7 ></pre>

Notice that the name assignment does not return the value on the next line. In order to “recall” an assigned value, just type the name and hit “enter.”

```
> x
[1] 3
> y
[1] 7
```

You can also assign names to calculations or vectors.

```
> w <- 88/8
> w
[1] 11
> u = ((4+3.5)/1.2)^3
> u
[1] 244.1406
```

Naming Conventions

R recognizes uppercase/lowercase letters, numbers, periods, and underscores as valid characters in a name. R does not recognize spaces as valid characters in a name. You cannot begin a name with a number or with an underscore. The following show some valid variable names as well as the error you will see if you use an invalid variable name.

```

> ValidName <- 11
> ValidName
[1] 11
> This.Name.Works.Too = 99
> This.Name.Works.Too
[1] 99
> variable_name_3 <- 100
> variable_name_3
[1] 100
> 4not.gonna.work = 44
Error: unexpected symbol in "4not.gonna.work"
> also not gonna work <- 12
Error: unexpected symbol in "also not"
> _LeadingUnderscore = 50
Error: unexpected input in "_"

```

How R Displays Very Large and Very Small Numbers (E Notation)

Let's see what happens if we calculate a very large number and a very small number.

```

> 45^9
[1] 7.566806e+14
> 45^-9
[1] 1.321561e-15

```

Here we've calculated 45^9 and 45^{-9} . This will give us a very large resulting value and a very small resulting value, respectively. Notice that our output looks a little weird! This is because R uses E notation for very large and very small values.

E notation is like scientific notation were the "e" stands for "10 to the power of." For example, in regular scientific notation, 1.34×10^5 is the same as 134,000. If we wanted to write this using E notation, the "e" would replace the " $\times 10$ ": $1.34e+5$.

So for the above numbers...

$$7.566806e+14 = 7.566806 \times 10^{14}$$

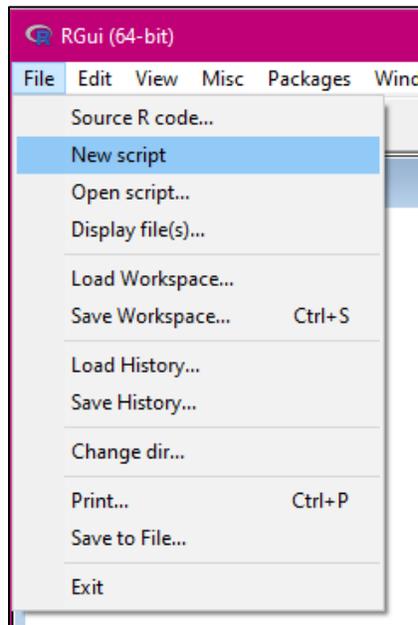
$$7.566806e-15 = 7.566806 \times 10^{-15}$$

R Scripts

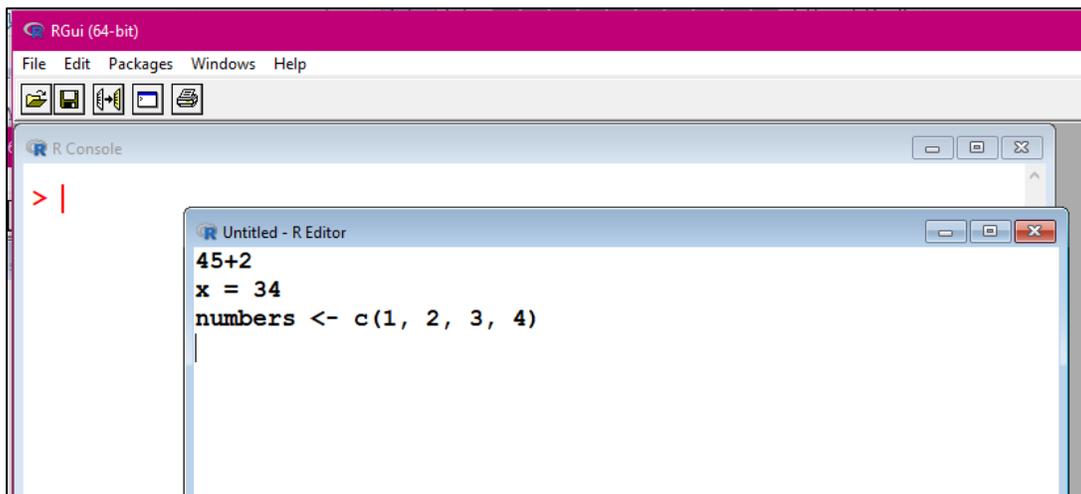
In many cases (especially once you start using R beyond just using it as a calculator), you'll want to be able to easily edit and save any code that you use in the program. One of the easiest ways to do so is to use an R script.

An R script is just a plain text file that has R code in it. But it allows you to type your code line by line without having to immediately run it in the R console. That way you can have a file full of your code and then run only the lines you want.

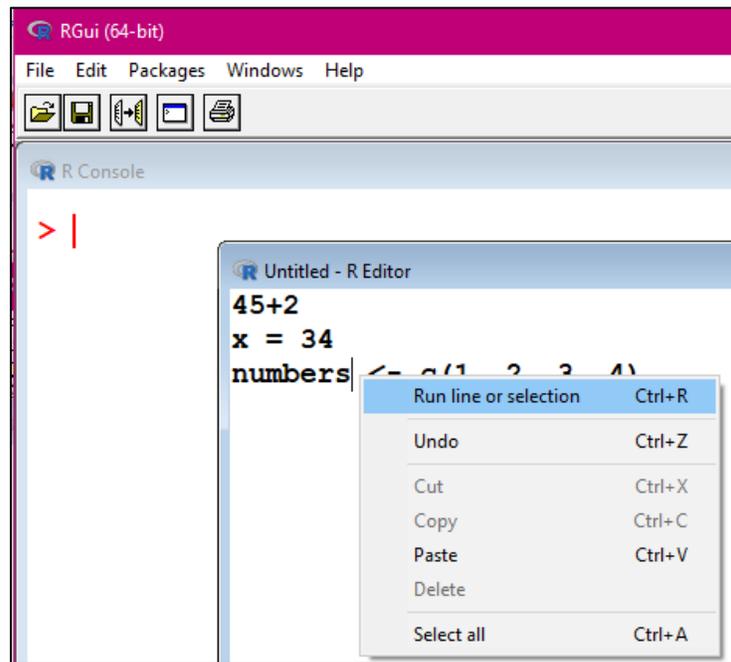
To create an R script, go to **File** → **New script**.



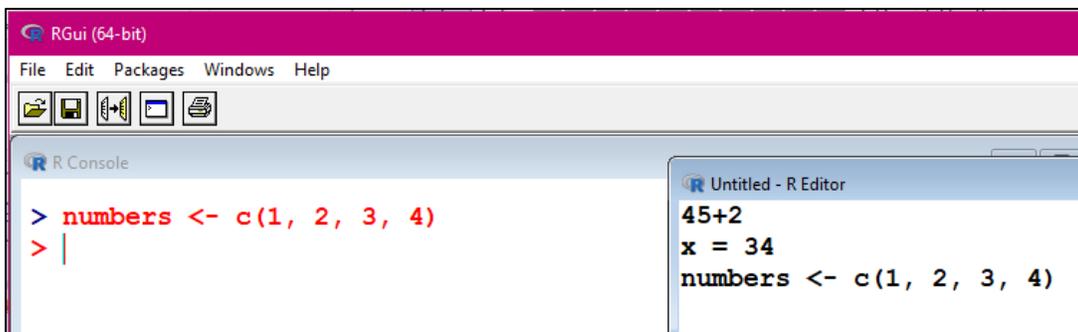
This will open a new blank pane in your RGui window. Notice that this pane is different from your R Console. You can treat this new pane as just a plain text file. You can type any code you want in it and when you hit “enter,” it does not automatically run the code like it would in the R Console.



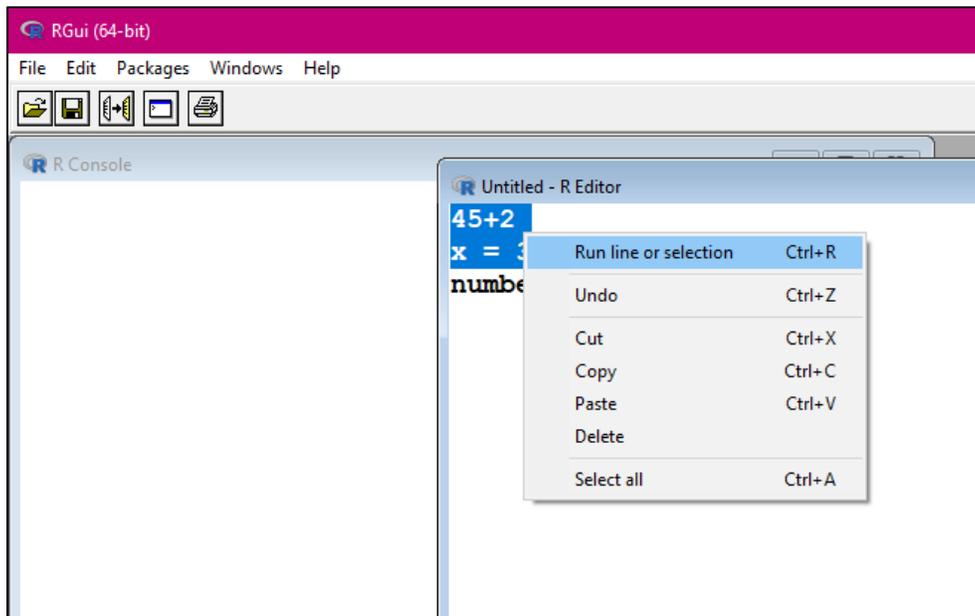
If you want to run a single line of code (that is, send it to the R console), put your cursor somewhere in the line you want to send, right-click, and select “Run line or selection” (or press ctrl+R in Windows).



This will send just that line to the R console. Then you can do anything else with it you like.

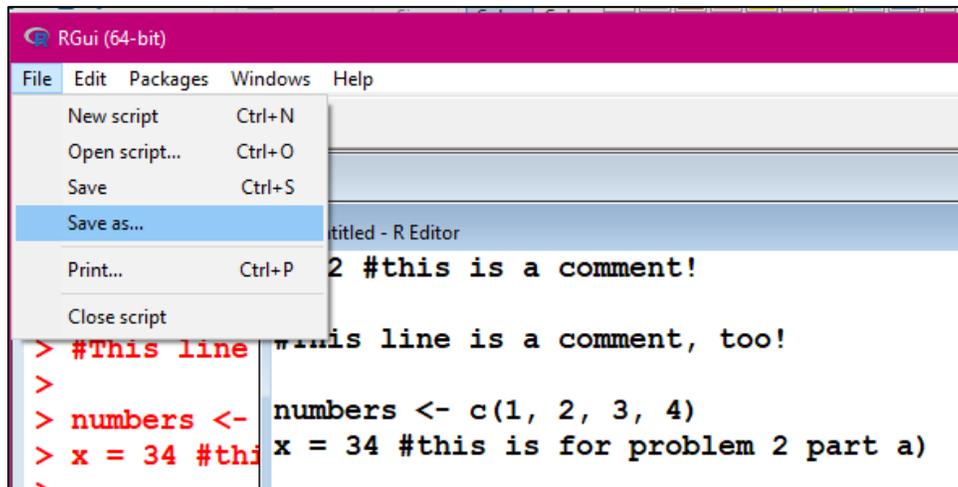


If you want to run multiple lines, select the lines you want to run, right-click, and select “Run line or selection” (or press ctrl+R in Windows).

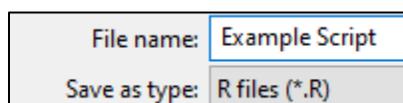


Saving an R Script

To save an R script, make sure the script window is active. Then go to File → Save As...

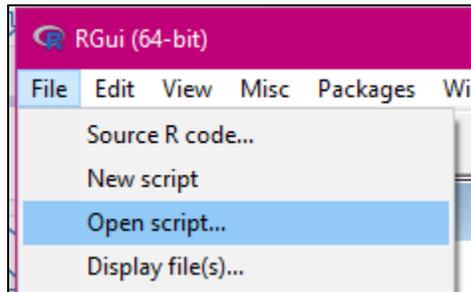


You can then save the script file anywhere you choose. Notice that it will save as an R file (with a .R extension).



Opening an R Script

To open a previously saved script, go to File → Open script... and select the script you wish to open.



Comments in Code

You can also make comments in your code. Comments can be useful if you want to explain what a calculation is for or mention what problem a calculation is for within the code itself. Comments are indicated by a line starting with a hashtag (#). You can also use a hashtag to comment out a line of code. Note that you can put a comment in its own line or in the same line of code that you want to run (as long as it's after the code itself, not before!).

```
> 45+2 #this is a comment!
[1] 47
>
> #This line is a comment, too!
>
> numbers <- c(1, 2, 3, 4)
> x = 34 #this is for problem 2 part a)
>
> #I can use a hashtag to comment out a piece of code like so:
> #77/8
> 77/9
[1] 8.555556
>
> #Notice that the calculation with the hashtag in front of it is not computed!
> |
```

Comments are a nice thing to add to your R code (especially if you're going to save it in an R script) so that future you knows what code goes with what problems/examples/etc.

Quick Reference Sheet

Here is a list of all the functions and shortcuts mentioned in this guide.

Basic Calculator Functions

Operation	Function	Example
Add	+	> 9+5 [1] 14
Subtract	-	> 9-5 [1] 4
Multiply	*	> 9*5 [1] 45
Divide	/	> 9/5 [1] 1.8
Exponent	^	> 10^2 [1] 100
Square root	sqrt	> sqrt(100) [1] 10
Exponential function (e)	exp	> exp(1) [1] 2.718282
Log base 10	log10	> log10(10) [1] 1
Log base e (natural log)	log	> log(1) [1] 0
Pi (π)	pi	> pi [1] 3.141593
Factorial	factorial	> factorial(5) [1] 120

Other Basic Functions

Operation	Function	Example
Sequence (of integers)	:	> 1:5 [1] 1 2 3 4 5
Concatenate (make a vector of numbers)	c	> c(2, 4, 6, 8) [1] 2 4 6 8
Summing multiple numbers	sum	> sum(2, 4, 6, 8) [1] 20

Assignment and Comments

Operation	Function	Example
Assign a name	<-	> x <- 3 >
Assign a name	=	> y = 7 >
Comment	#	> 90/44 #this is a comment [1] 2.045455

Keyboard Shortcuts

Operation	Keyboard Shortcut
Clear the R console	Ctrl+l
Run line(s) from R script	Ctrl+r