Getting Started With R Programming

Conrad Williams

Workshop Goals

- Goals:
  - Install R and RStudio
  - Understand the RStudio layout
  - Understand basic programming in R
    - Variables, math, conditionals, loops, functions, etc.
    - Installing and importing packages from CRAN
  - Learn to read in, visualize, and analyze data in R
    - Examples covered are far from exhaustive, but will provide a baseline understanding
- NOT covered:
  - Choosing appropriate statistical tests for your data and interpreting results

Installing R and RStudio

- There is a short guide to installing R and RStudio in the OneDrive directory for this workshop (“01_Installing_R_and_RStudio.pdf”).
- Let’s take the time now to step through that guide and make sure everyone has RStudio up and running.

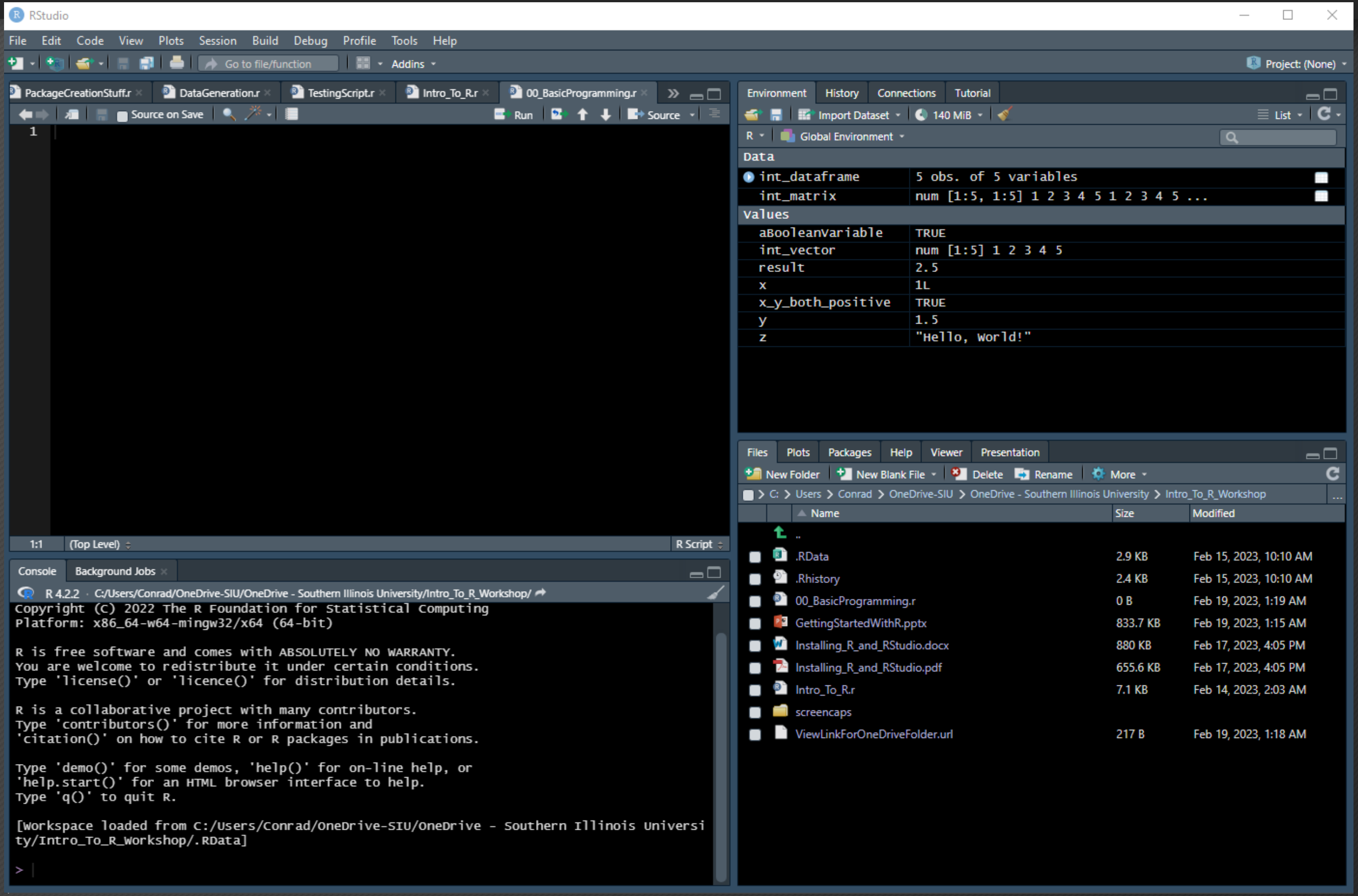
The RStudio Window: Regions

Open R Scripts

Console

Environment, History, Etc.

Files, Plots, Help, Etc.



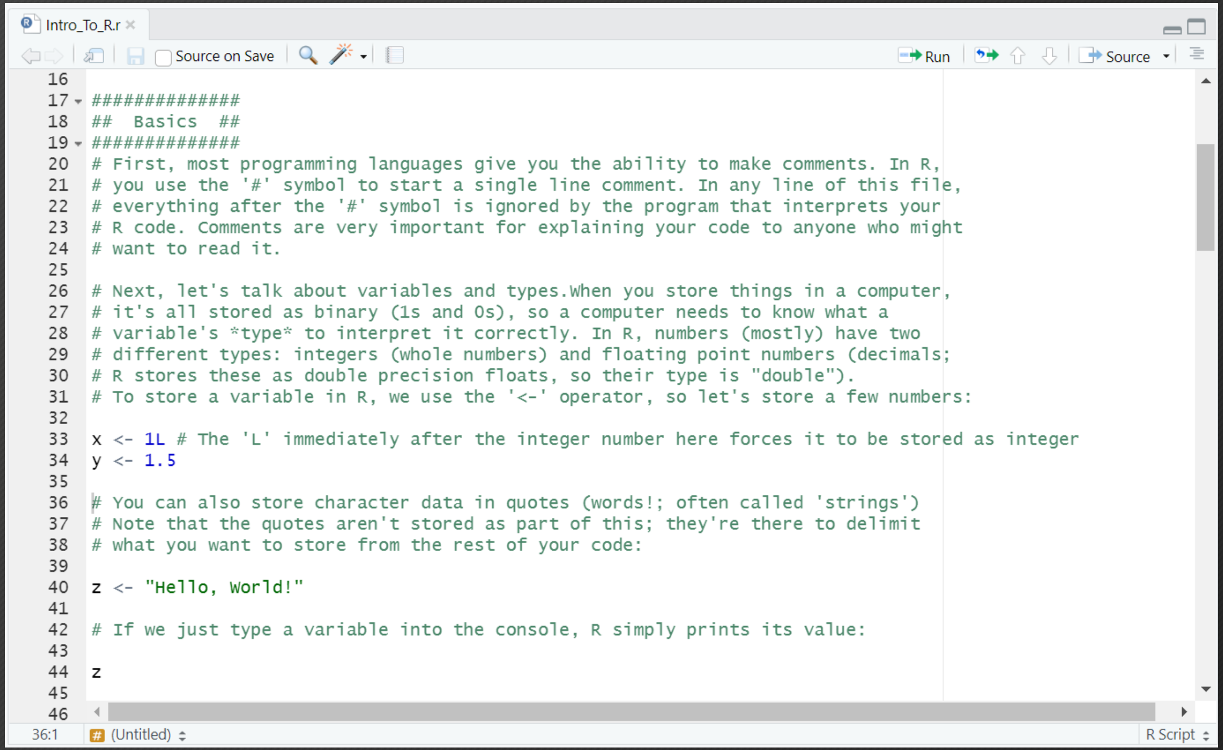
```
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

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.

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.

[Workspace loaded from C:/Users/Conrad/OneDrive-SIU/OneDrive - Southern Illinois Universi
ty/Intro_To_R_Workshop/.RData]

>
```

Data	
int_dataframe	5 obs. of 5 variables
int_matrix	num [1:5, 1:5] 1 2 3 4 5 1 2 3 4 5 ...
Values	
aBooleanVariable	TRUE
int_vector	num [1:5] 1 2 3 4 5
result	2.5
x	1L
x_y_both_positive	TRUE
y	1.5
z	"Hello, World!"

Name	Size	Modified
..		
.RData	2.9 KB	Feb 15, 2023, 10:10 AM
.Rhistory	2.4 KB	Feb 15, 2023, 10:10 AM
00_BasicProgramming.r	0 B	Feb 19, 2023, 1:19 AM
GettingStartedWithR.pptx	833.7 KB	Feb 19, 2023, 1:15 AM
Installing_R_and_RStudio.docx	880 KB	Feb 17, 2023, 4:05 PM
Installing_R_and_RStudio.pdf	655.6 KB	Feb 17, 2023, 4:05 PM
Intro_To_R.r	7.1 KB	Feb 14, 2023, 2:03 AM
screencaps		
ViewLinkForOneDriveFolder.url	217 B	Feb 19, 2023, 1:18 AM

Scripts

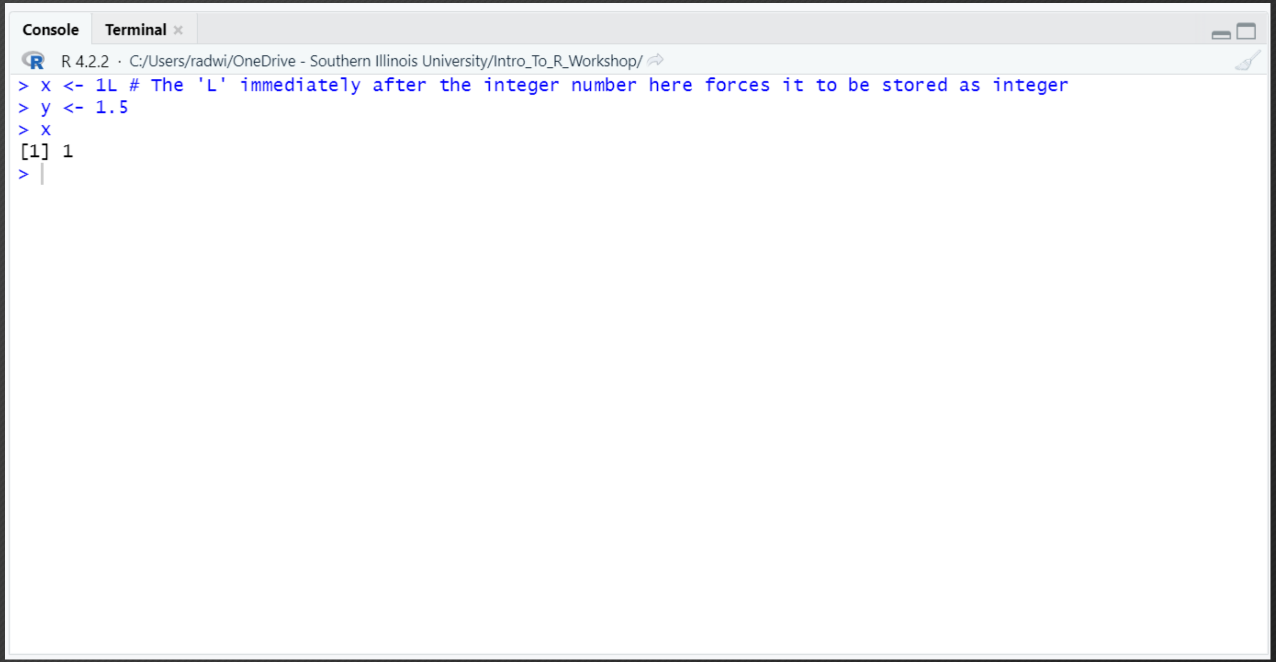
- Where R scripts open (files named like “filename.r”)
- While you can run your commands one at a time in the console, it’s often better to encapsulate a series of commands in a script that you can save
- You can run lines of code by clicking into a line and pressing “Ctrl + Enter”
  - You can also highlight multiple lines and run them



```

##############
##  Basics  ##
##############
# First, most programming languages give you the ability to make comments. In R,
# you use the '#' symbol to start a single line comment. In any line of this file,
# everything after the '#' symbol is ignored by the program that interprets your
# R code. Comments are very important for explaining your code to anyone who might
# want to read it.

# Next, let's talk about variables and types.When you store things in a computer,
# it's all stored as binary (1s and 0s), so a computer needs to know what a
# variable's *type* to interpret it correctly. In R, numbers (mostly) have two
# different types: integers (whole numbers) and floating point numbers (decimals;
# R stores these as double precision floats, so their type is "double").
# To store a variable in R, we use the '<-' operator, so let's store a few numbers:

x <- 1L # The 'L' immediately after the integer number here forces it to be stored as integer
y <- 1.5

# You can also store character data in quotes (words!; often called 'strings')
# Note that the quotes aren't stored as part of this; they're there to delimit
# what you want to store from the rest of your code:

z <- "Hello, World!"

# If we just type a variable into the console, R simply prints its value:

z


```

Console

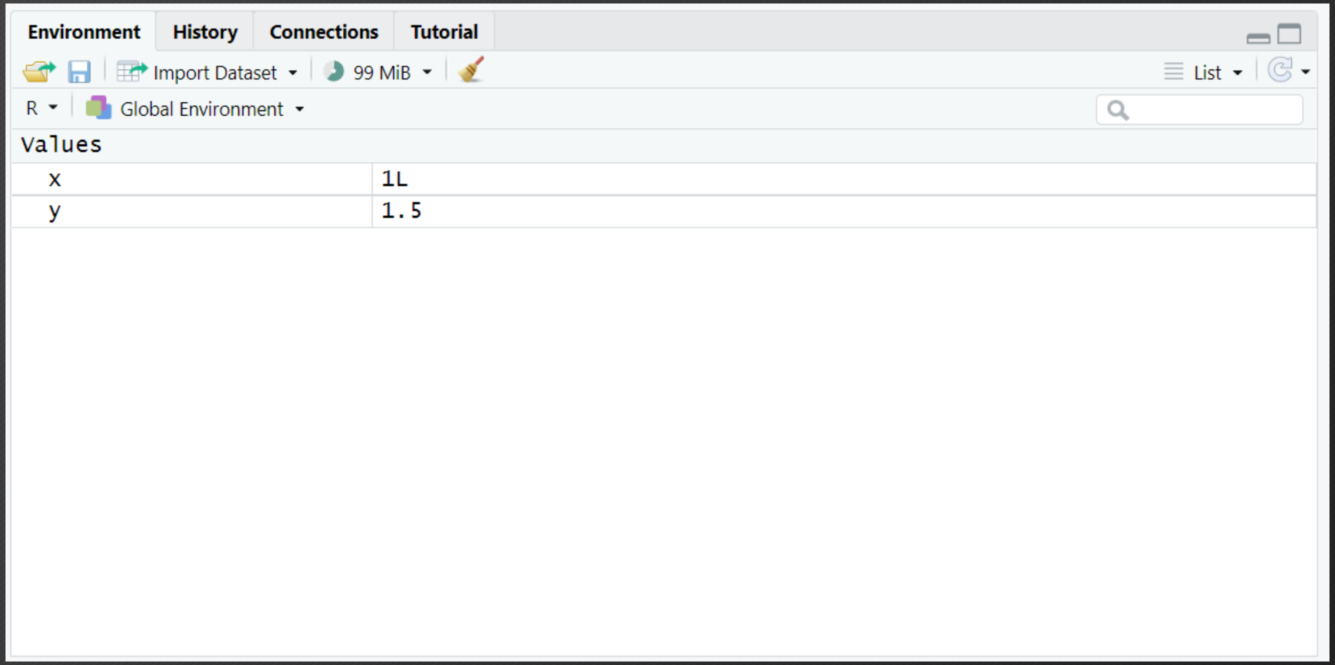
- An R prompt: where your commands are interpreted and run
- You can type commands here one line at a time
- The “Terminal” tab here opens your operating system’s terminal in R’s current working directory (we won’t use this today)


Console
Terminal
R 4.2.2 · C:/Users/radwi/OneDrive - Southern Illinois University/Intro_To_R_Workshop/

```
> x <- 1L # The 'L' immediately after the integer number here forces it to be stored as integer
> y <- 1.5
> x
[1] 1
> 
```

Environment, History, etc.

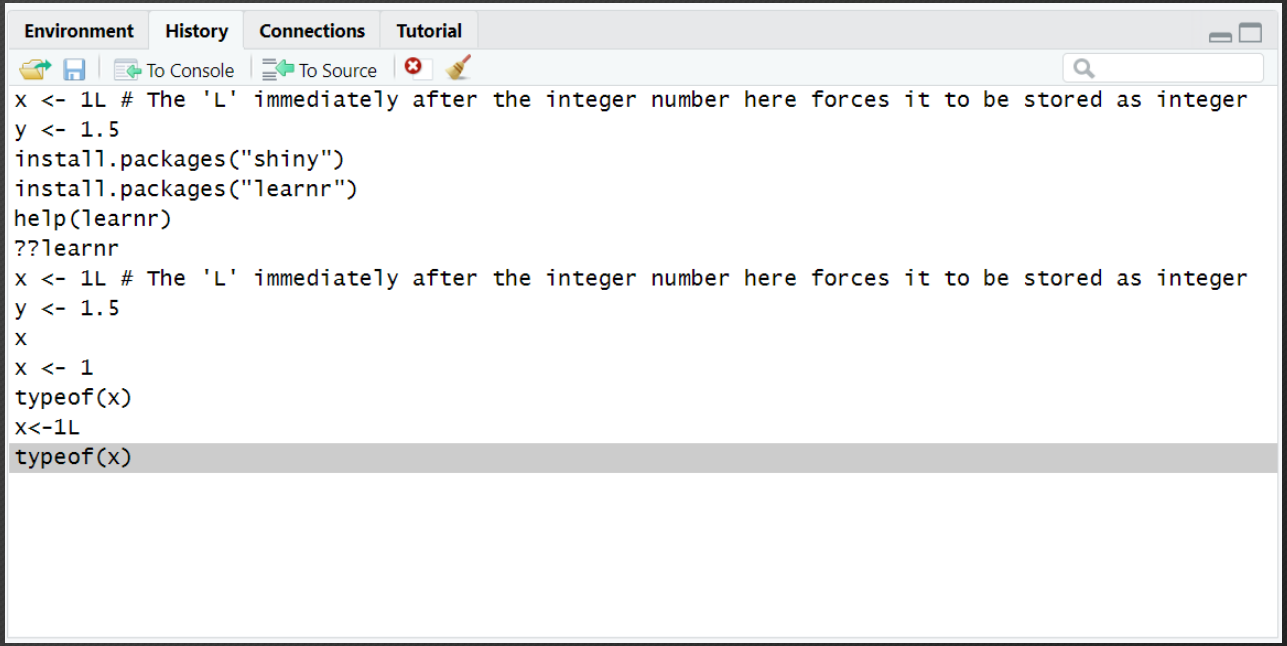
- The Environment tab shows values that you've stored in memory
  - Includes variables and functions
  - Shows how much memory is currently allocated
  - Allows you to save your current environment and load saved environments (referred to as Workspaces)
  - You can also import datasets using the GUI here!



```
Values
  x                 1L
  y                 1.5
```

Environment, History, etc.

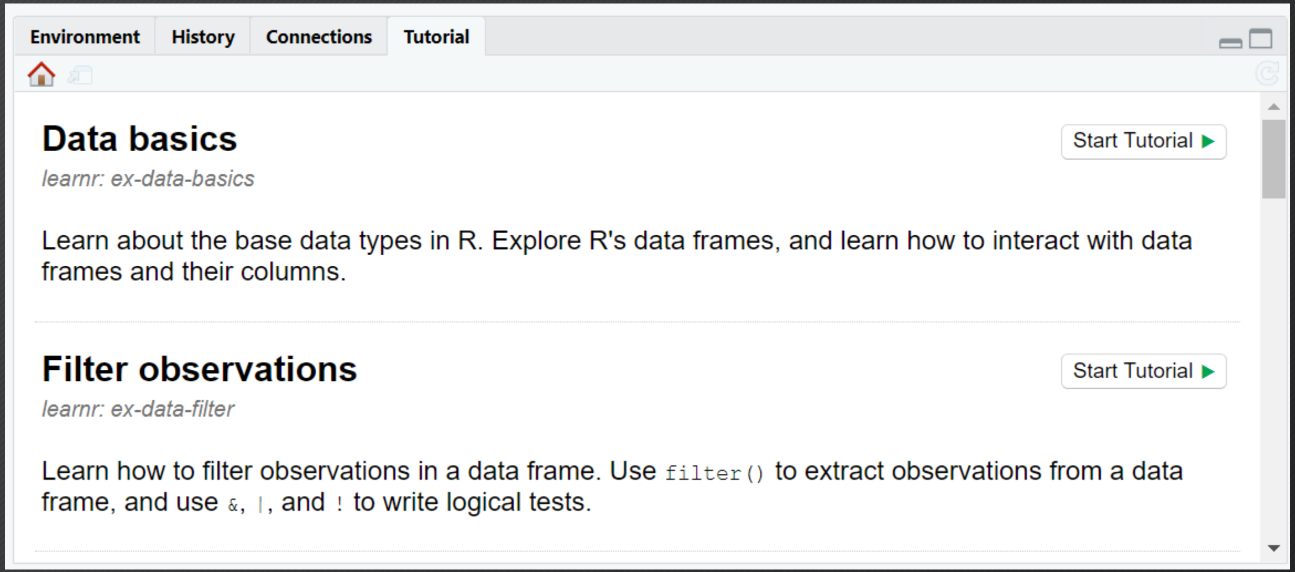
- The History tab shows recently run commands
  - You can rerun commands here or send them to an open script
  - Mostly useful if you've been typing commands in the console instead of keeping them in a script
  - Can also save and load history to and from files (*.Rhistory)



```
x <- 1L # The 'L' immediately after the integer number here forces it to be stored as integer
y <- 1.5
install.packages("shiny")
install.packages("learnr")
help(learnr)
??learnr
x <- 1L # The 'L' immediately after the integer number here forces it to be stored as integer
y <- 1.5
x
x <- 1
typeof(x)
x<-1L
typeof(x)
```

Tutorial Tab: Accessing Built-in Tutorials

- RStudio has a series of built in tutorials that can introduce you to more advanced features and packages
  - This is contained in the "Tutorial" tab of the upper-right region
  - Prior to running these, you'll need to install the "learnr" and "shiny" packages
  - Good place to keep learning after today's workshop



Data basics

learnr: ex-data-basics

Learn about the base data types in R. Explore R's data frames, and learn how to interact with data frames and their columns.

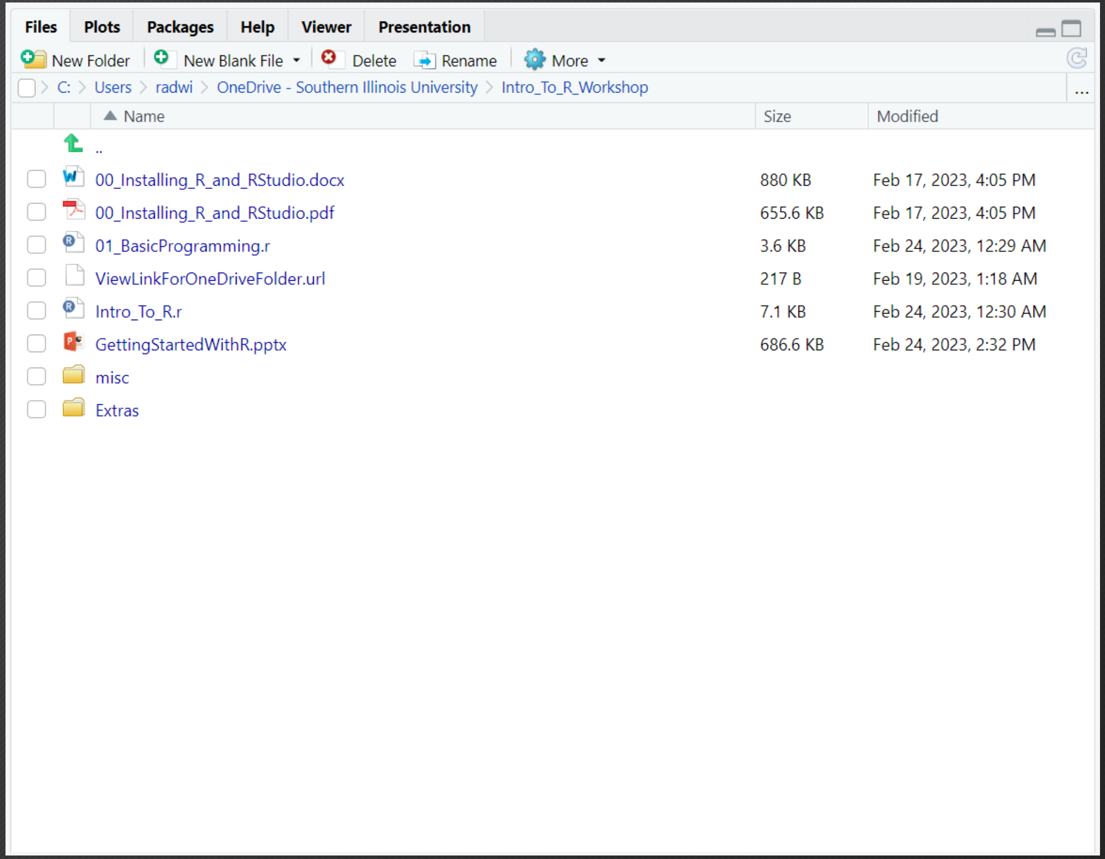
Filter observations

learnr: ex-data-filter

Learn how to filter observations in a data frame. Use `filter()` to extract observations from a data frame, and use `&`, `|`, and `!` to write logical tests.

Files, Plots, Packages, Help

- The Files tab shows files in the current working directory (more on that when we talk about reading in data later)

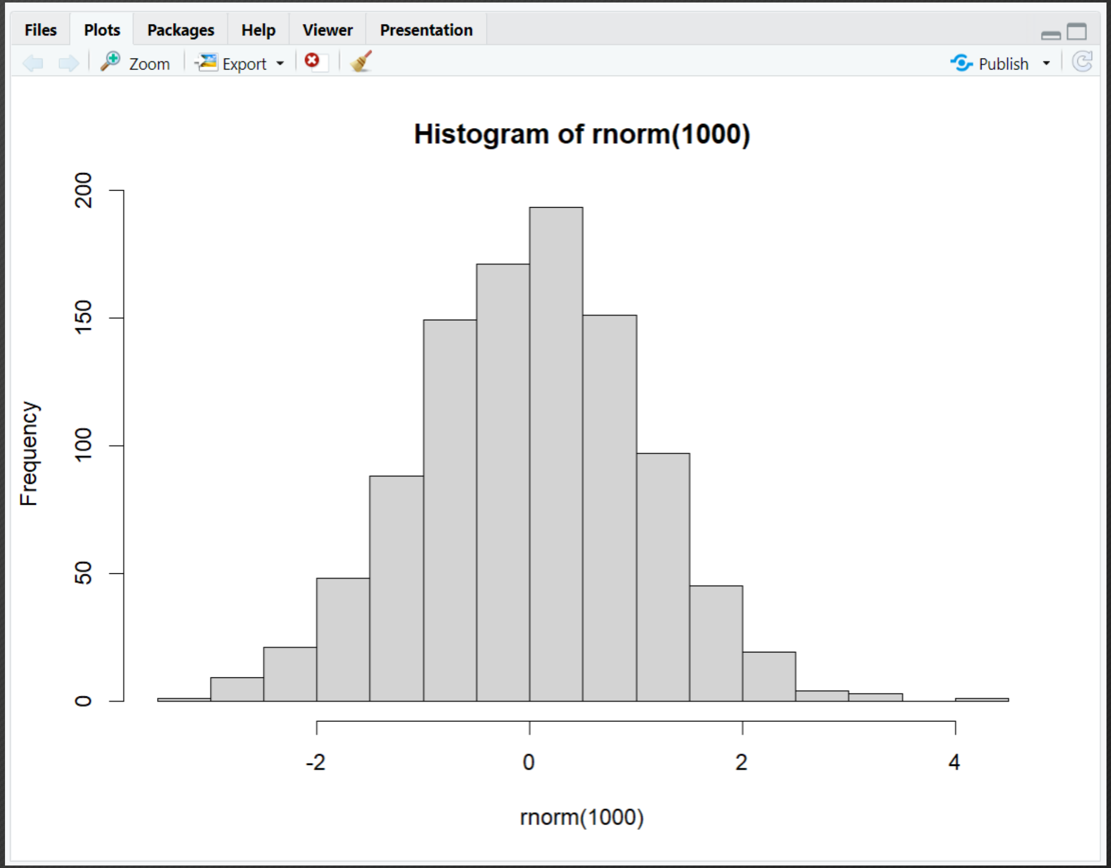


C: > Users > radwi > OneDrive - Southern Illinois University > Intro_To_R_Workshop

Name	Size	Modified
..		
00_Installing_R_and_RStudio.docx	880 KB	Feb 17, 2023, 4:05 PM
00_Installing_R_and_RStudio.pdf	655.6 KB	Feb 17, 2023, 4:05 PM
01_BasicProgramming.r	3.6 KB	Feb 24, 2023, 12:29 AM
ViewLinkForOneDriveFolder.url	217 B	Feb 19, 2023, 1:18 AM
Intro_To_R.r	7.1 KB	Feb 24, 2023, 12:30 AM
GettingStartedWithR.pptx	686.6 KB	Feb 24, 2023, 2:32 PM
misc		
Extras		

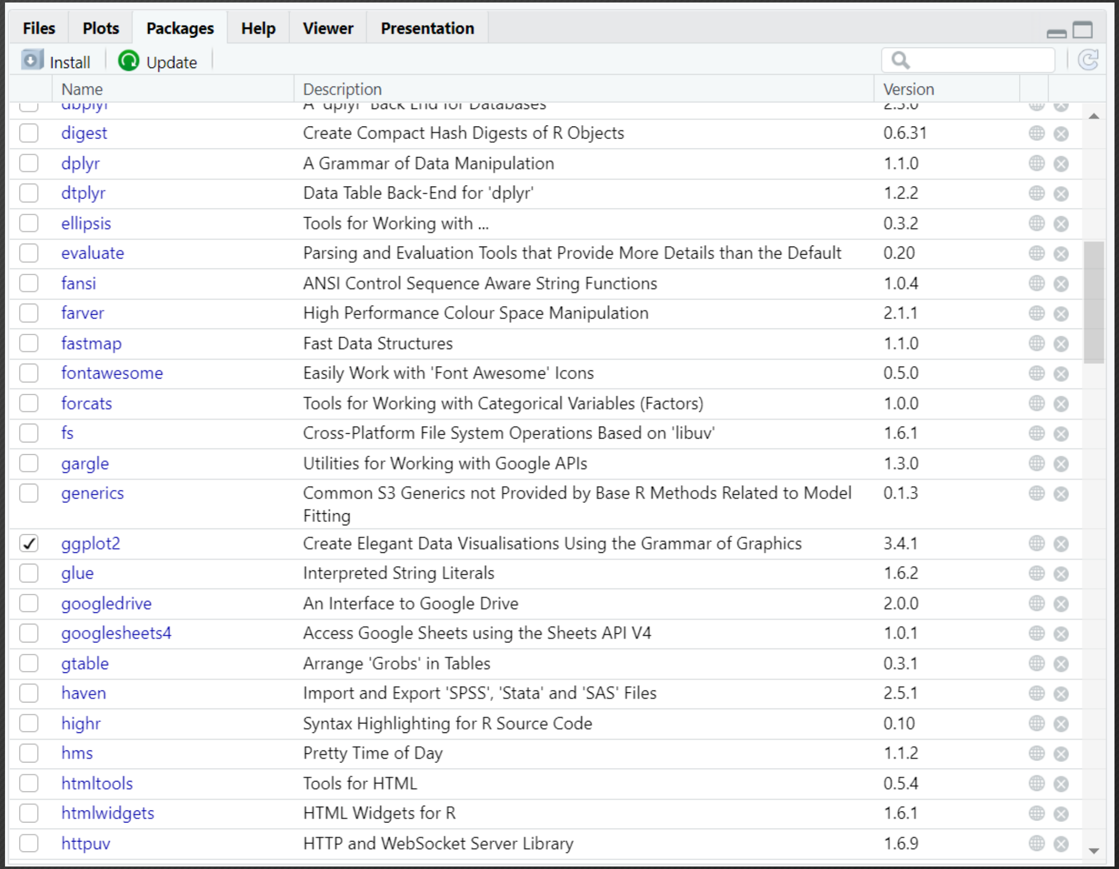
Files, Plots, Packages, Help, etc.

- The Plots tab shows graphical output from functions you run in R
    - You can also save your plots from here using the "Export" drop-down menu (covered later)
    - The plot to the right is a histogram of 1000 random sample from a normal distribution generated with the following command:
        - hist(rnorm(1000))



Files, Plots, Packages, Help

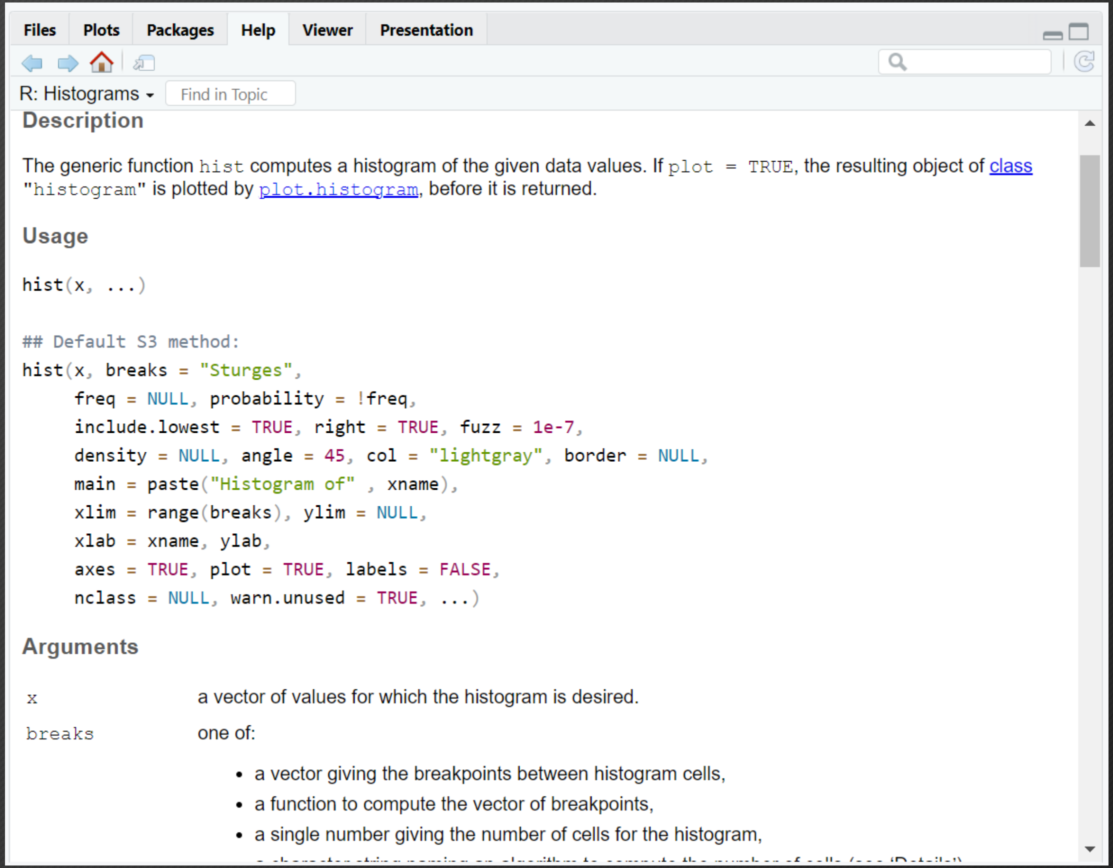
- The Packages tab shows installed ‘packages’ which are collections of functions and other objects
  - ☑ packages are loaded so we can call functions from them
  - You can also install new packages using the “Install” menu on this tab
- More on packages later!



	Name	Description	Version
☐	dbplyr	A 'dplyr' Back End for Databases	2.3.0
☐	digest	Create Compact Hash Digests of R Objects	0.6.31
☐	dplyr	A Grammar of Data Manipulation	1.1.0
☐	dtplyr	Data Table Back-End for 'dplyr'	1.2.2
☐	ellipsis	Tools for Working with ...	0.3.2
☐	evaluate	Parsing and Evaluation Tools that Provide More Details than the Default	0.20
☐	fansi	ANSI Control Sequence Aware String Functions	1.0.4
☐	farver	High Performance Colour Space Manipulation	2.1.1
☐	fastmap	Fast Data Structures	1.1.0
☐	fontawesome	Easily Work with 'Font Awesome' Icons	0.5.0
☐	forcats	Tools for Working with Categorical Variables (Factors)	1.0.0
☐	fs	Cross-Platform File System Operations Based on 'libuv'	1.6.1
☐	gargle	Utilities for Working with Google APIs	1.3.0
☐	generics	Common S3 Generics not Provided by Base R Methods Related to Model Fitting	0.1.3
✓	ggplot2	Create Elegant Data Visualisations Using the Grammar of Graphics	3.4.1
☐	glue	Interpreted String Literals	1.6.2
☐	googledrive	An Interface to Google Drive	2.0.0
☐	googlesheets4	Access Google Sheets using the Sheets API V4	1.0.1
☐	gtable	Arrange 'Grobs' in Tables	0.3.1
☐	haven	Import and Export 'SPSS', 'Stata' and 'SAS' Files	2.5.1
☐	highr	Syntax Highlighting for R Source Code	0.10
☐	hms	Pretty Time of Day	1.1.2
☐	htmltools	Tools for HTML	0.5.4
☐	htmlwidgets	HTML Widgets for R	1.6.1
☐	httpuv	HTTP and WebSocket Server Library	1.6.9

Files, Plots, Packages, Help

- The Help tab shows the documentation for functions and packages.
  - *Extremely* useful
  - Try running the following in the console to get the documentation for the hist() function used earlier:
    - help(“hist”)



Description

The generic function `hist` computes a histogram of the given data values. If `plot = TRUE`, the resulting object of class `"histogram"` is plotted by `plot.histogram`, before it is returned.

Usage

```
hist(x, ...)

## Default S3 method:
hist(x, breaks = "Sturges",
     freq = NULL, probability = !freq,
     include.lowest = TRUE, right = TRUE, fuzz = 1e-7,
     density = NULL, angle = 45, col = "lightgray", border = NULL,
     main = paste("Histogram of" , xname),
     xlim = range(breaks), ylim = NULL,
     xlab = xname, ylab,
     axes = TRUE, plot = TRUE, labels = FALSE,
     nclass = NULL, warn.unused = TRUE, ...)
```

Arguments

`x`	a vector of values for which the histogram is desired.
`breaks`	one of:

- a vector giving the breakpoints between histogram cells,
- a function to compute the vector of breakpoints,
- a single number giving the number of cells for the histogram,

Starting Today’s Workshop Scripts

- You should now have R and RStudio installed and have become somewhat familiar with RStudio's layout.
- To start the first workshop script, locate “02_BasicProgramming.r” and double-click it.
  - This should prompt the OS to ask what program you want to open that file
  - Choose “RStudio” and allow the OS to associate the “*.r” file extension with RStudio.
- The rest of the workshop will use the scripts in this directory