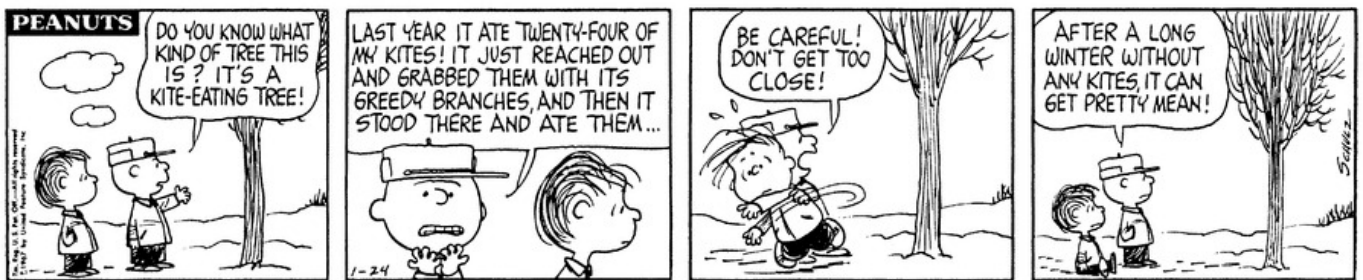


Problem Set 01: Intro to R and RStudio

Your Name Goes Here

Last modified on July 31, 2023 15:24:09 Eastern Daylight Time



The goal this week is to introduce R and RStudio which will be used throughout the course both to review the statistical concepts discussed in the course and to analyze real data and come to informed conclusions. To clarify which is which: R is the name of the programming language itself, and RStudio is an integrated development editor (IDE).

Today, we begin with the fundamental building blocks of R and RStudio: the interface, creating and saving files, and basic commands.

Opening RStudio Server

Open Appalachian's RStudio Server and sign in: [RStudio Server](#)

Your credentials are the same ones you use to log into your ASU email account.

Please **DO NOT** choose Stay signed in.

The RStudio Interface

In the RStudio Server, you should see a window that looks like the image in Figure 1.

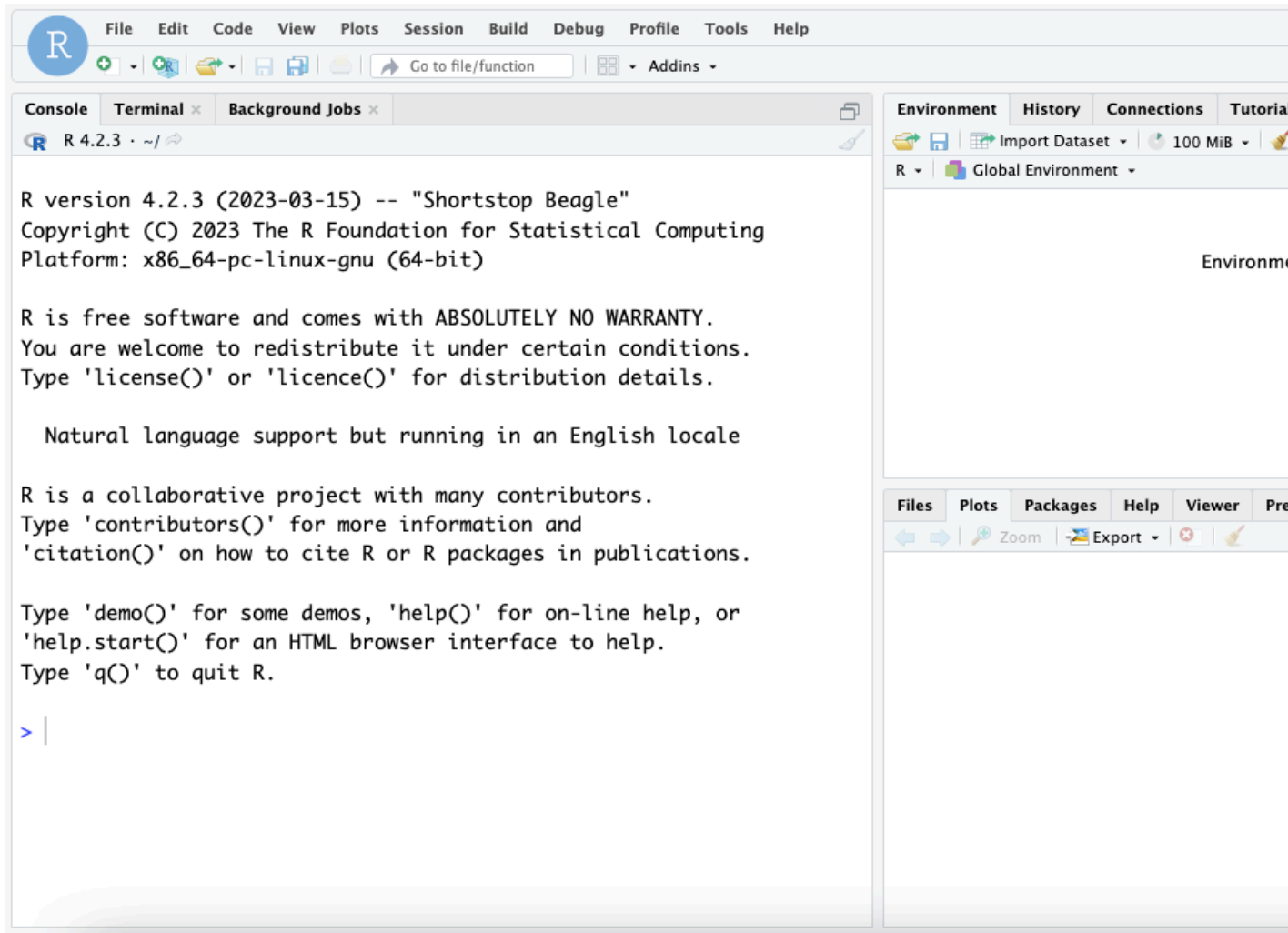


Figure 1: The RStudio interface

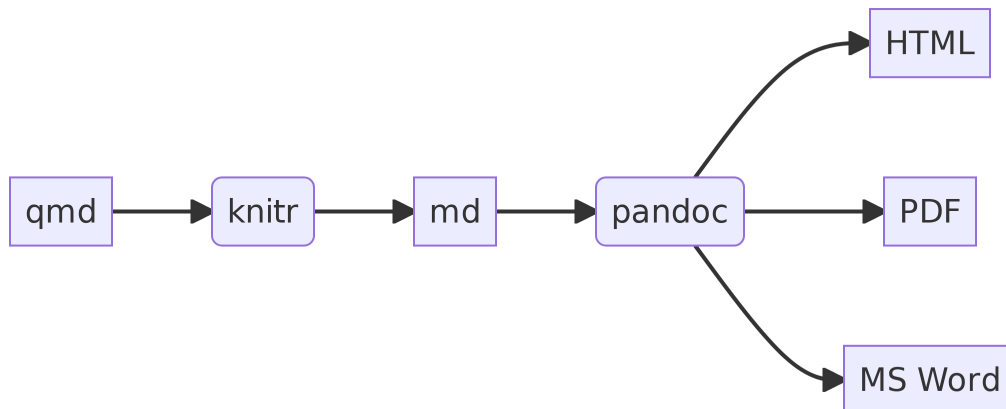
The panel on the left is where the action happens. It's called the *console*. Every time RStudio is launched, it will have the same text at the top of the console describing the version of R that is running.

The panel in the upper right contains the *workspace*. This shows the variables and objects defined during an R session and a history of the commands that are entered.

Any plots that are generated will show up in the panel in the lower right corner. This is also where you can browse your files, access help files, and upload and download files.

Using Quarto Files

At its core, Quarto works the same way as R Markdown:



Quarto combines the functionality of R Markdown, bookdown, distill, xaringan, etc. into a single consistent system. Quarto is at its core multi-language and multi-engine (supporting Knitr, Jupyter, and Observable today and potentially other engines tomorrow).

Opening a New File

Quarto documents can be used in R or python. For this course, we will use R and the RStudio IDE to work with Quarto documents. Quarto documents are useful for both running code and annotating the code with comments. The documents can be saved, so you can refer back to your code later. Quarto documents can also be used to generate other document types (HTML, PDF, MS Word, Open Office, or ePub) for presenting the results of your analyses in formats that may be required in other contexts.

To open a new Quarto document, click on the little green plus beside the circled R in the upper left hand area of the RStudio IDE as seen in Figure 1 and select **Quarto Document...**. Enter a title and author in the corresponding boxes; then, click create. See Figure 2 for an example where the document title is **Something Cool** and the author is **Joe Quarto**.

When you open a new Quarto document, there is some example code (template) in it that you might delete.

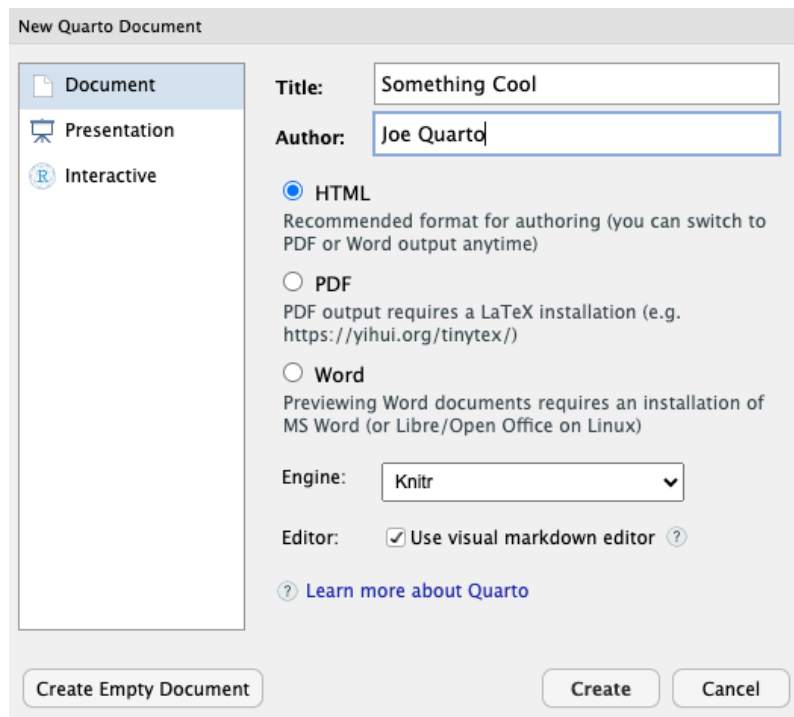


Figure 2: Starting a new Quarto document

Saving a File

Lab work will be saved as Quarto files that will be committed and pushed to the class repository. Therefore, it is important to learn how to save these files. To save the Quarto template:

- Click File > Save As...
- Name the file: `PS01_YourLastName_YourFirstName` (mine would be `PS01_Arnholt_Alان.qmd`).

Warning

The `PS01_YourLastName_YourFirstName.qmd` file is just for your own practice. The file you will save, commit, and push to the class repository is the file you are reading (`PS01_source.qmd`). You will make changes to this file starting in Section (Practice on Your Own).

- Click save. The `PS01_YourLastName_YourFirstName.qmd` is now saved in the MD-PS01-SC directory **on the server**.

Make Changes to a File

Let's make some changes to the Quarto document you just created using Figure 3 as a guide.

- First, change the title of the document to **Using the RStudio IDE with Quarto**. Be sure to surround the title with quotation marks.
- Second, add your name to the author field, making sure to include your name inside quotation marks.
- Third, click on **Source** in the upper left of the `*.qmd` source's panel so that line numbers appear on the far left.
- Fourth, delete lines 12 through 27.
- Finally, insert what is called a "code chunk." To do this, click on the green +C (insert a new code chunk) button near the top right of the `*.qmd` source's panel. R code is entered on the blank line inside the R code chunk.

Your final result should look similar to Figure 3.

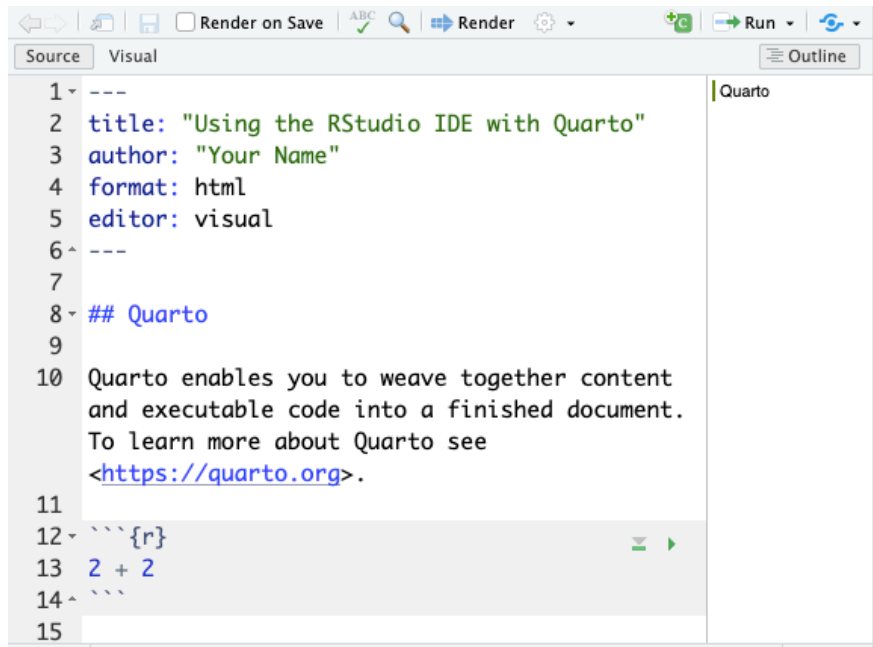


Figure 3: Quarto source file

Rendering a File

Click the \Rightarrow Render button at the top left side of the screen to “render” the file, or in other words, produce a formatted document. An *.html file will be generated. The *.html file is automatically saved in the same folder as your Quarto file.

Note that there are now a Quarto file (PS01_YourLastName_YourFirstName.qmd) and an html file (PS01_lastname_firstname.html) in the MD-PS01-SC folder in addition to the files you downloaded.

Inspect the *.html file to see how what you typed was formatted. There are lots of tricks for controlling the formatting of the rendered html file. For instance:

- putting ## and a space in front of text makes it into a large header. For example, see how ## This is a header in your Quarto *.qmd file translates in the resulting *.html output.
- putting ### and a space in front of text makes it a smaller header.

Entering and Running Commands

The code chunks are where you put R code in Quarto file. So far, your “rendered” file (your formatted document file) doesn’t show anything because we did not put any content in the code chunks yet.

Using your first code chunk, type the following command to create a new variable called `x` with the value of 6.

```
x <- 6
```

The arrow `<-` is called an **assignment operator** and tells R to save an object called `x` that has the value of 6. This is similar to saving a value in a graphing calculator.

Tip

Note that whatever you want to save must always be to the left of the assignment operator (`<-`).

To **run** this command in your console, you have a few options:

- Click on the green triangle in the first line of the code chunk that points to the right.
- Highlight the code and hit **Control-Enter** on a PC or **Command-Return** on a Mac.

Think of “running” code in your console as telling R “do this.”

Note that you now have a new object in your workspace, called `x`.

Data Types—a Brief Intro

So far, you have made a numeric variable `x`. There many other types of data objects you can make in R. In this section, we create different types of objects.

To make a **character** object called `favorite_movie` copy, paste, and run the following code in a new code chunk. Think of characters as text as opposed to numerical values. Note that R knows this is a **character** because there are quotation marks around `Star_Wars`.

```
favorite_movie <- "Star_Wars"
```

Next, copy, paste, and run the following command in a new code chunk.



```
v <- c(2, 4, 6)
```

This makes what is called a **vector**, which we have named `v`. It is a data object that has multiple elements of the same type. This vector contains three numbers: 2, 4, and 6. The `c()` function says tells R to **concatenate** the values 2, 4, 6, into a single **vector**. Note in the Environment pane that your vector `v` contains numbers (listed as `num`).

You can do math on a vector that contains numbers. For instance, copy, paste, and run the following command in a new code chunk. This tells R to multiply each element of the vector `v` by 3.

```
v * 3
```


Practice on Your Own

Directions

Type complete sentences to answer all questions in the Quarto document. Round all numeric answers you report to four decimal places. Use inline R code to report all numeric answers (i.e. do not hard code your numeric answers). Remember to save your work as you go along. Click the floppy disk (save current document) button in the upper left hand corner of the Quarto source panel.

Problem 1

Answer the following with code in a code chunk (no text necessary). Remember that the code is just **instructions** for R. You need to run the code chunk to make R execute those instructions.

- Create an empty code chunk and click on the Run all previous code chunks.
- Create a variable called `y` with the value of 7.
- Multiply `x` by `y`, and store the answer in a variable named `z` like so: `z <- x * y`.
- Run the current code chunk, and you should now see `favorite_movie`, `x`, `v`, `y`, and `z` all in your Environment pane.

Problem 1 Answers

```
# Type your code and comments inside the code chunk
```

Problem 2

- Run the following mathematical operation in a code chunk: `6 + 3`
- Where does the answer appear? (please answer with **text**)

Problem 2 Answers

```
# Type your code and comments inside the code chunk
```

- Type your answer here and delete this sentence.

Problem 3

- Now add a code chunk, and save the results of `6 + 3` as a variable called `a`.
- Does the answer appear? (please answer with **text**)
- Where can you see the value of the object `a`? (please answer with **text**)
- Next, type `a` into the code chunk and re-run the code chunk. What happens? (please answer with **text**)

Problem 3 Answers

```
# Type your code and comments inside the code chunk
```

- Type your answer here and delete this sentence.
- Type your answer here and delete this sentence.
- Type your answer here and delete this sentence.

Tip

It is a good idea to render your document from time to time as you make changes and updates to your work. Go ahead, and make sure your document is rendered and that your html file includes Exercise headers, text, and code. Note that rendering automatically saves your `*.qmd` file.

Problem 4

- Run following command in a new code chunk: `a^2`.
- What does the `^` operator do? (please answer with **text**)

Problem 4 Answers

```
# Type your code and comments inside the code chunk
```

- Type your answer here and delete this sentence.

Problem 5

- Type the following command into a new code chunk. `sum(a, x, y)`
- `sum` is a function. Based on the output, what do you think the `sum` function does?

(please answer with **text**)

Problem 5 Answers

```
# Type your code and comments inside the code chunk
```

- Type your answer here and delete this sentence.

Problem 6

- Click the little broom icon in the upper right hand corner of the **Environment** pane. Click yes on the window that opens. What happened? (please answer with **text**, and don't freak out)

Problem 6 Answers

- Type your answer here and delete this sentence.

Problem 7

- Go to the **Run** button at the top right of the Quarto pane, and choose **Run All** (the last option)
- What happened? (please answer with **text**)

Problem 7 Answers

- Type your answer here and delete this sentence.

Problem 8

Recall the vector `v` we created earlier. Copy, paste, and run the following in a code chunk. What does this code accomplish? (please answer with **text**)

```
v + 2
```

Problem 8 Answers

```
# Type your code and comments inside the code chunk
```

- Type your answer here and delete this sentence.

Problem 9

Copy, paste, and run the following code to make a vector called `music` that contains music genres. Recall a vector is a data object that has multiple elements of the same type. Here the data type is a **character**. Look in the environment pane. How does R tell us that this vector contains **characters** not numbers? (please answer with **text**)

```
music <- c("bluegrass", "funk", "folk")
```

Problem 9 Answers

```
# Type your code and comments inside the code chunk
```

- Type your answer here and delete this sentence.

Problem 10

Now let's practice some basic formatting. Using https://rmarkdown.rstudio.com/authoring_basics.html, figure out how to put italic, bold, and superscripted text into your lab report.

You do not need to do anything for this problem other than to observe the code in the `.qmd` document that generates the italicized, bolded, and superscripted output.

Italicize like this

Bold like this

A superscript: R^2

Extra Credit

What does Charlie Brown have to do with R?

Extra Credit Answer

Type your answer here and delete this sentence.

Turning in Your Work

You will need to make sure you commit and push all of your changes to the github education repository where you obtained the lab.

Tip

- Make sure you **render a final copy with all your changes** and work.
- Look at your final html file to make sure it contains the work you expect and is formatted properly.

Logging out of the Server

There are many statistics classes and students using the Server. To keep the server running as fast as possible, it is best to sign out when you are done. To do so, follow all the same steps for closing Quarto document:

Tip

- Save all your work.
- Click on the orange button in the far right corner of the screen to quit R
- Choose **don't save** for the **Workspace image**
- When the browser refreshes, you can click on the sign out next to your name in the top right.
- You are signed out.

```
sessionInfo()
```

```
R version 4.2.3 (2023-03-15)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Red Hat Enterprise Linux 9.2 (Plow)
```

“Goodbye always makes my
throat hurt.”

-Charlie Brown



Matrix products: default
BLAS/LAPACK: /usr/lib64/libopenblas-r0.3.21.so

locale:

[1] LC_CTYPE=en_US.UTF-8	LC_NUMERIC=C
[3] LC_TIME=en_US.UTF-8	LC_COLLATE=en_US.UTF-8
[5] LC_MONETARY=en_US.UTF-8	LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8	LC_NAME=C
[9] LC_ADDRESS=C	LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8	LC_IDENTIFICATION=C

attached base packages:

[1] stats graphics grDevices utils datasets methods base

other attached packages:

[1] knitr_1.43

loaded via a namespace (and not attached):

[1] compiler_4.2.2	fastmap_1.1.1	cli_3.6.1	tools_4.2.2
[5] htmltools_0.5.5	rstudioapi_0.15.0	yaml_2.3.7	rmarkdown_2.23
[9] jpeg_0.1-10	jsonlite_1.8.7	xfun_0.39	digest_0.6.33
[13] rlang_1.1.1	png_0.1-8	evaluate_0.21	