Introduction to R and R Markdown

Your name here

2/13/25

Programming in Quarto markdown

This document that we are working in is called an **Quarto markdown** document. It allows us to seamlessly move between `R` code and regular text. The name of the file is easily found at the top this panel. All Quarto markdown files end in _____.

YAML

At the top of a markdown topic, you'll see code between two sets of three dashed lines. This is known as a **YAML header**, and it contains the "informational" content of a document (e.g. title, author, date). Change these accordingly for each assignment!

Go ahead and change your name in the author argument of the YAML.

Quarto Markdown basics

How do we tell the document which parts correspond to code, and which parts correspond to text?

In the following, you see three back ticks followed by a left curly brace, the letter r, and right curly brace. A few lines down, you will see three more back ticks. The background in between these lines is gray, with some symbols on the right. These backs ticks must be aligned on the same tabulation.

This defines _____. All of our `R` code should go into one.

In the code chunk above, let's evaluate $\sqrt{4}$ (the square root of 4). To do this, type the following code: `sqrt(4)`. Now evaluating the code in a code chunk is different from evaluating code in the Console. There are two ways to evaluate code in a chunk:

1.

2.

Saving progress

At this point it is a good idea to save our progress. Like most document editors, we need to explicitly save our work. You know your work is not saved when _____.

To save our work in Quarto Markdown document, we can do one of the following:

1. Click on the floppy disk at the top of the panel
2. Go to File, then Save
3. Hit Cmd+S
4. _____ the document

Coding in R

`R` is more than just a calculator! It provides lots of functionality for performing tasks related specifically to statistics.

Object types

Everything in `R` is an object. Here, we provide a non-exhaustive list of common objects (i.e. structures) you will encounter.

- A _____ object contains only a single number
- A _____ or _____ object is a set of characters within one pair of quotation marks
- A _____ object is any collection of numbers *or* any collection of strings. We can create vectors using the command `c()`:
- A _____ object is either `TRUE` or `FALSE`. It is also referred to as a boolean.
- **Data frames** are representations of datasets in `R` where the rows correspond to observations and columns correspond to variables that describe the observations (more on this later).

Functions

When we calculated $\sqrt{4}$, we used the code `sqrt()`. This is an example of a **function**. Functions allow us to automate common tasks in a general way. Functions (just like in math) take in one or more inputs. These inputs are known as _____ or _____. They will almost always return an output. We know a command in `R` is a function because it has _____.

It is possible for us to create our own customized functions (you definitely will if you take STAT 218). However, in STAT 201, we will work with pre-provided functions. All pre-provided functions in `R` are accompanied by a **Help file**. To access the Help file, simply type `?` followed by the name of the function in the Console. Try opening the Help file for the `sqrt()` function.

Storing objects as variables

Suppose I want to save the result of a big calculation to use moving forward. We _____ or _____ the value of $1 + 2 + 3 + 4 + 5$ into the _____ called `x` using the keys `<-`.

As you should note: when we assign an object a value, its value is *not* automatically shown as output. In order to display the output, you must _____.

Errors, warnings, and messages

These are always shown in red text in the Console. Whenever you see them, don't panic! With practice, you will be able to decipher the messages and de-bug your code with ease.

- **Errors:** prevent code from executing and documents from rendering It will be prefaced with "Error in…". Trying typing in the following code in your Console: `1 + a`. What happens?
- **Warnings**: your code will run with some caveats. It will be prefaced with "Warning:". We will see examples of this later on.
- **Messages**: messages in red that do not begin with "Error" or "Warning" are simply friendly messages that might provide you more information about the execution of your code.

Packages

Packages in `R` extend the functionality by providing additional functions and data. You can view them as analogous to apps you download from the App Store or Google Play on a cell phone. To use an app on a phone, you have to:

1. Download the app
2. Explicitly open the app

To use a package, we need to:

1. _____ the package
2. _____ the package

Unless you update R Studio, you will only need to install a package once. However, you will need to explicitly load in packages every time you work in a new Quarto Markdown document.

There are thousands of available packages to work with. Two of the most common packages we will use are the `openintro` package and the `tidyverse` package (though, the `tidyverse` package is actually a giant package that is comprised of several other packages).

Package installation

There are two ways to install a package:

1. Option 1:
    i. Click on the "Packages" tab in the Files pane of RStudio
    ii. Click on "Install"
    iii. Type the name of the package under "Packages (separate multiple with space or comma):".
    iv. Click "Install"
2. Option 2:
    1. Type `install.packages("package name")` into the Console. Note that the quotation marks are necessary.
    2. Press Return/Enter

We will install the `openintro` package together.

Then, try installing the `tidyverse` package on your own!

Package loading

If we want to use a package, we use the `library()` command:

Rendering

Rendering the document will turn your Quarto markdown into its final output form. To render, simply click on the blue arrow button up top that reads "Render". Rendering can take anywhere from a few seconds to a few minutes depending on the amount of code and narrative you have. It will also save your output!

Open your STAT 201 folder. What do you notice happens once we've rendered?

Submit to Canvas

When you are finished with your work and ready to submit, you should always render one last time. You should then submit the most recent rendered output (i.e. PDF) to the corresponding Canvas assignment. Sometimes I will also ask you to submit the accompanying .qmd file.