R05: Quarto in R

Meike Niederhausen and Nicky Wakim

2024-10-14

Saving your work with Quarto

Artwork by @allison_horst

Example: creating an html file

.qmd file

Untitled2*

Render on Save | Render | Run

Source | Visual | Normal | Format | Insert | Table

```yaml
---
title: "My first Quarto file"
author: "Meike"
format: html
editor: visual
---
```

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```{r}
1 + 1
```

You can add options to executable code like this

```{r}
#| echo: false
2 * 2
```

The `echo: false` option disables the printing of code (only output is displayed).

.html output

My first Quarto file

AUTHOR
Meike

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

```
[1] 2
```

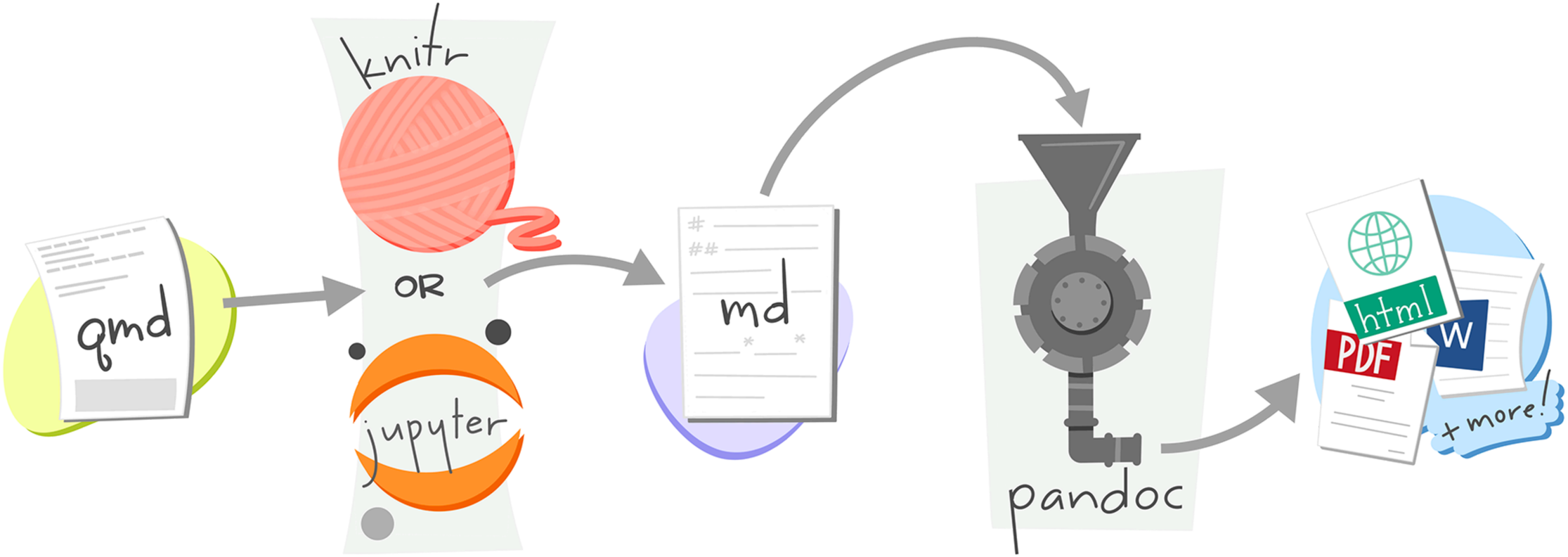
You can add options to executable code like this

```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).

Quarto = `.qmd` file = Code + text

We can take `.qmd` files containing code (R and other types) + plain text (like we might make in Word), and then to it other formats (html, pdf, Word, etc) that nicely display the code and text!



Artwork from "Hello, Quarto" keynote by Julia Lowndes and Mine Çetinkaya-Rundel, presented at RStudio Conference 2022. Illustrated by Allison Horst.

Basic Quarto example

Artwork from “Hello, Quarto” keynote by Julia Lowndes and Mine Çetinkaya-Rundel, presented at RStudio Conference 2022. Illustrated by Allison Horst.

Before we get further in `.qmd` files

- Let's make sure we all have Rstudio open
- And then open your `EPI_525_F24` project!

Steps for making a Quarto file

1. Create a Quarto file (`.qmd`)
2. Edit a Quarto file (`.qmd`)
3. Save the Quarto file (`.qmd`)
4. Create html file

1. Create a Quarto file (`.qmd`)

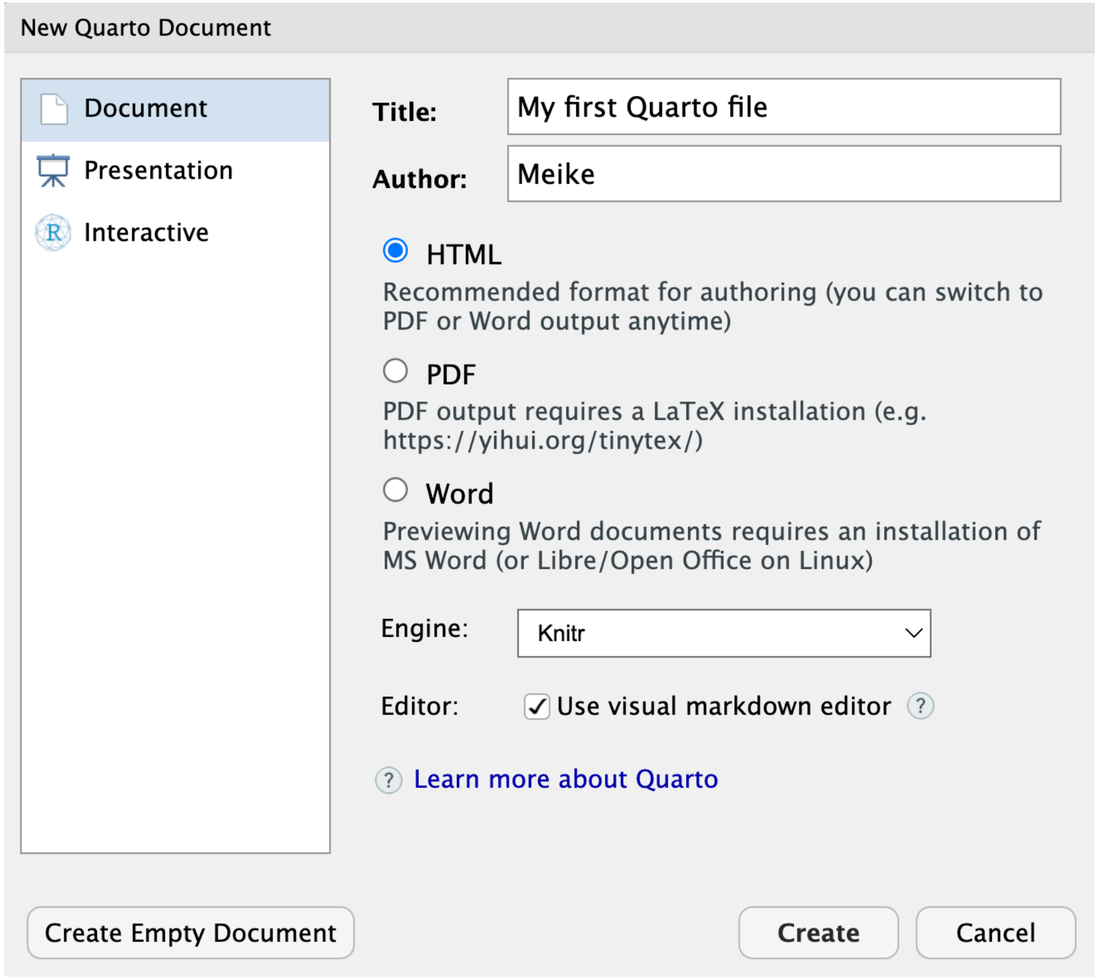
Two options:

1. click on File → New File → Quarto Document...→ OK,
2. or in upper left corner of RStudio click on  →

Pop-up window selections:

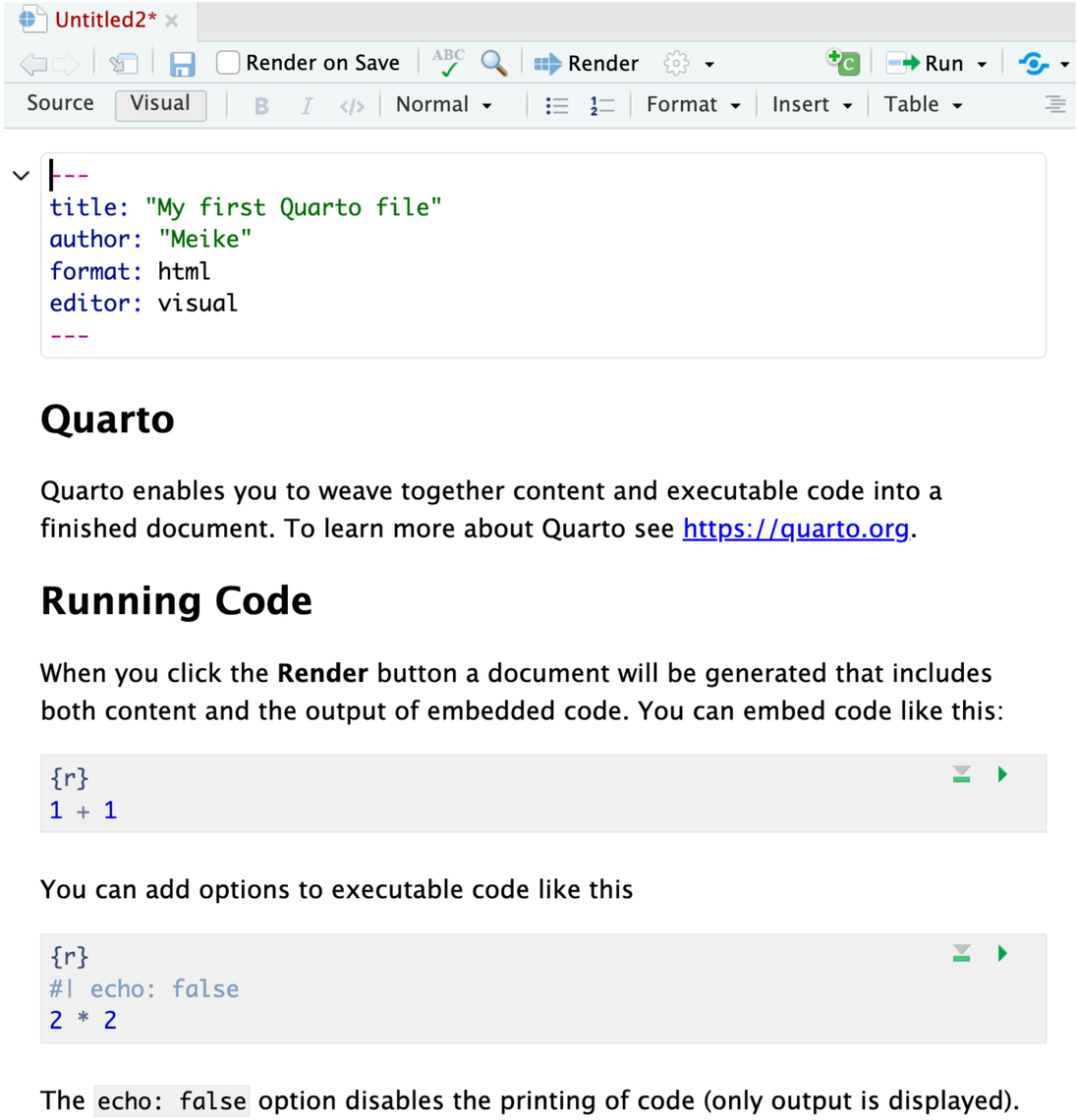
- Enter a title and your name
- Select `HTML` output format (default)
- Engine: select `Knitr`
- Editor: Select `Use visual markdown editor`
- Click `Create`



2. Edit a Quarto file (`.qmd`)

- After clicking on `Create`, you should then see the following in your editor window:

- You can try editing the text or changing the code!
  - Make sure you are only editing at the "Quarto" header and below



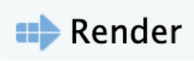
3. Save the Quarto file (`.qmd`)

- **Save the file** by
  - selecting `File -> Save`,
  - or clicking on  (towards the left above the scripting window),
  - or keyboard shortcut
    - PC: *Ctrl + s*
    - Mac: *Command + s*
- You will need to specify (Use what we learned in last lesson!!)
  - a **filename** to save the file as
    - ALWAYS use **.qmd** as the filename extension for Quarto files
  - the **folder** to save the file in
  - Hint: this will probability go under "R_activities" and with a name like "R05_Quarto-work.qmd"

4. Create html file

We create the html file by **rendering** the .qmd file.

Two options:

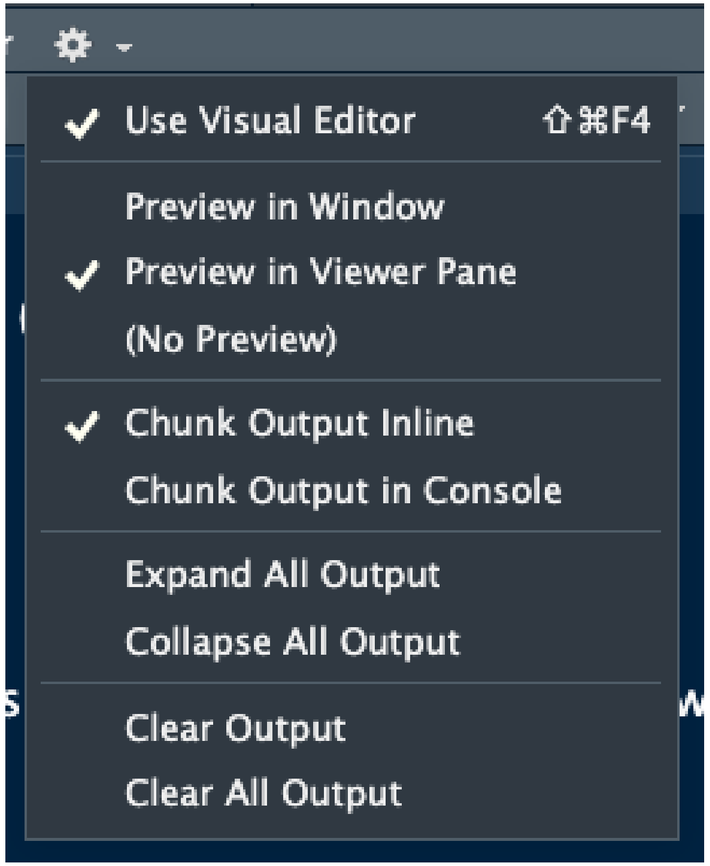
1. click on the Render icon  at the top of the editor window,
2. or use keyboard shortcuts
   - Mac: *Command+Shift+K*
   - PC: *Ctrl+Shift+K*

- A new window will open with the html output.
- You will now see both .qmd and .html files in the folder where you saved the .qmd file.

> **Note**
>
> - The template .qmd file that RStudio creates will render to an html file by default.
> - The output format can be changed to create a Word doc, pdf, slides, etc.

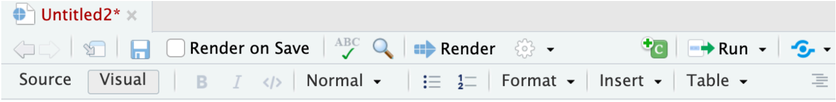
Tip: changing the render view

- You can change where your `.html` file pops up
- I have it set to open in the "Viewer Pane" in the bottom right



.qmd vs. its .html output

.qmd file




```
---
title: "My first Quarto file"
author: "Meike"
format: html
editor: visual
---
```

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
{r}
1 + 1
```

You can add options to executable code like this

```
{r}
#| echo: false
2 * 2
```

The `echo: false` option disables the printing of code (only output is displayed).

.html output

My first Quarto file

AUTHOR
Meike

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

```
[1] 2
```

You can add options to executable code like this

```
[1] 4
```

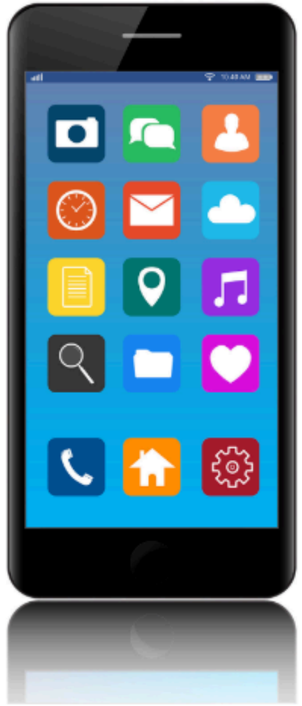
The `echo: false` option disables the printing of code (only output is displayed).

R Packages


stringr
tidyverse
www.rstudio.com
dplyr
tidyr
readr
purrr
Horst

R Packages

A good analogy for R packages is that they are like apps you can download onto a mobile phone:

R: A new phone	R Packages: Apps you can download
	

ModernDive Figure 1.4

- Packages contain additional functions and data

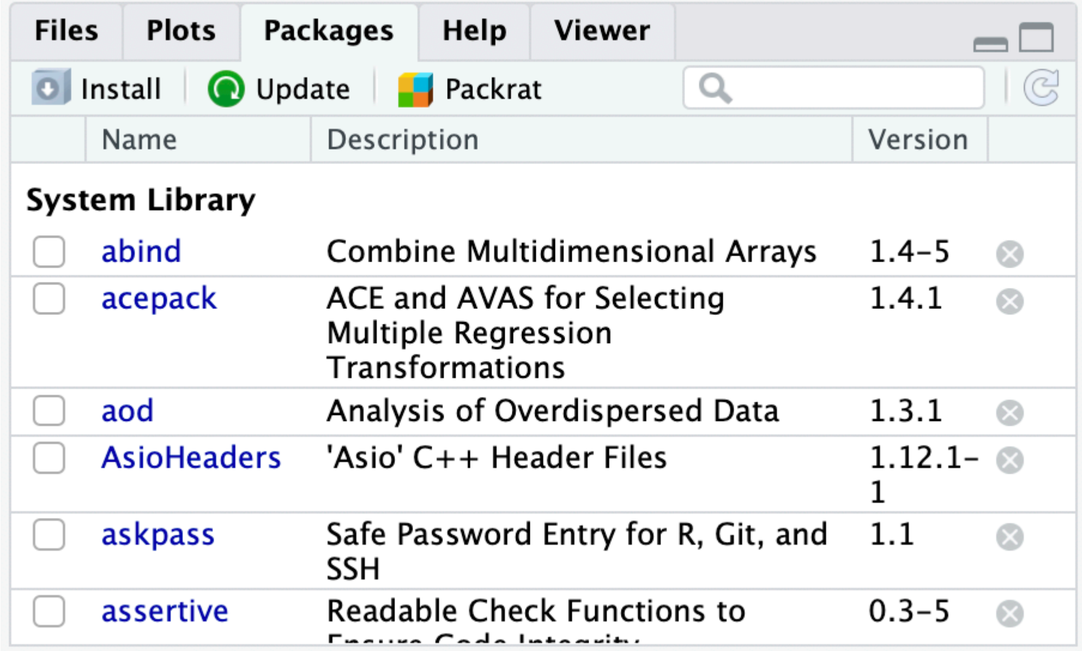
Installing packages

Two options to install packages:

1. `install.packages()` *or*
2. The "Packages" tab in Files/Plots/Packages/Help/Viewer window

```
install.packages("dplyr")   # only do this ONCE, use quotes
```

- **Only install packages once** *(unless you want to update them)*
- Installed from Comprehensive R Archive Network (CRAN) = package mothership



Video on installing packages

- Danielle Navarro's YouTube video on ***Installing and loading R packages***: https://www.youtube.com/watch?v=kpHZVyDvEhQ
  - If you want to get more information on packages

Load packages with `library()` command

- Tip: at the top of your Rmd file, create a chunk that loads all of the R packages you want to use in that file.
- Use the `library()` command to load each required package.
- Packages need to be reloaded *every* time you open Rstudio.

```
library(dplyr)    # run this every time you open Rstudio
```

- You can use a function without loading the package with `PackageName::CommandName`

```
dplyr::arrange(iris, Petal.Width)   # what does arrange do?
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1          4.9         3.1          1.5         0.1     setosa
2          4.8         3.0          1.4         0.1     setosa
3          4.3         3.0          1.1         0.1     setosa
4          5.2         4.1          1.5         0.1     setosa
5          4.9         3.6          1.4         0.1     setosa
6          5.1         3.5          1.4         0.2     setosa
7          4.9         3.0          1.4         0.2     setosa
8          4.7         3.2          1.3         0.2     setosa
9          4.6         3.1          1.5         0.2     setosa
```

Install the packages listed below

- `knitr`
  - this might actually already be installed
  - check your packages list
- `tidyverse`
  - this is actually a bundle of packages
  - *Warning: it will take a while to install!!!*
  - see more info at https://tidyverse.tidyverse.org/
- `rstatix`
  - for summary statistics of a dataset
- `janitor`
  - for cleaning and exploring data
- `ggridges`
  - for creating ridgeline plots
- `devtools`
  - used to create R packages
  - for our purposes, needed to install some packages
- `oi_biostat_data`
  - this package is on github
  - **see the next slide for directions on how to install `oi_biostat_data`**
- `here`
  - More info in slides ahead

Directions for installing package `oibiostat`

- The textbook's datasets are in the R package `oibiostat`
- Explanation of code below
  - Installation of `oibiostat` package requires first installing `devtools` package
  - The code `devtools::install_github()` tells R to use the command `install_github()` from the `devtools` package without loading the entire package and all of its commands (which `library(devtools)` would do).

```
install.packages("devtools")
devtools::install_github("OI-Biostat/oi_biostat_data", force = TRUE)
```

- After running the code above, put # in front of the commands so that RStudio doesn't evaluate them when rendering.
- Now load the `oibiostat` package
  - the code below needs to be run *every time* you restart R or knit an Rmd file

```
library(oibiostat)
```

here package

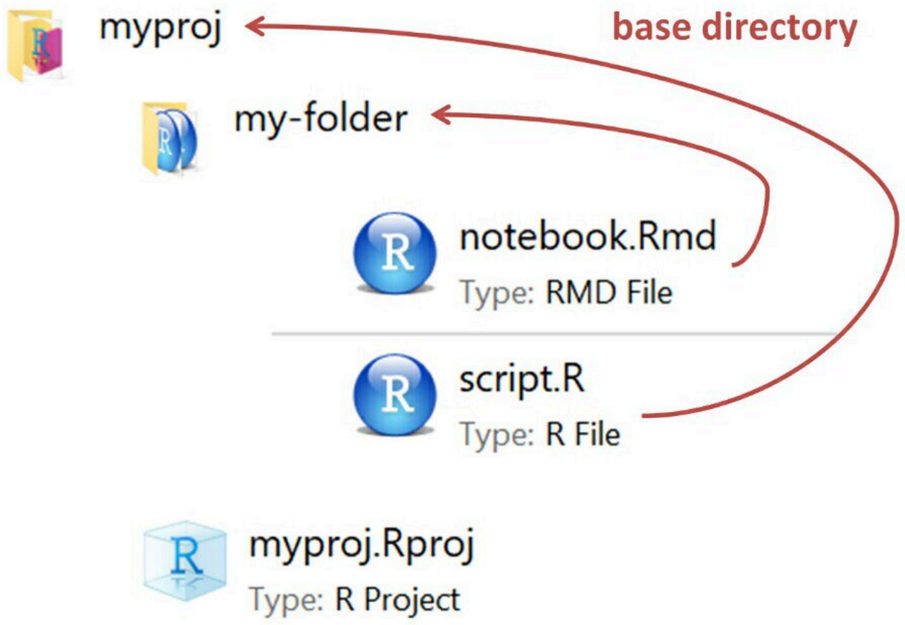
Illustration by Allison Horst

`here` package

- Good source for the `here` package
  - Just substitute `.Rmd` with `.qmd`
- Basically, a `.qmd` file and `.R` file work differently
  - We haven't worked much with `.R` files
- For `.qmd` files, the automatic directory is the folder it is in
  - But we want it to be the main project folder
- `here` can help with that

- **Very important for reproducibility!!**

...but relative paths work differently in .R files versus in .Rmd's



Using `here` package

- Within your console, type `here()` and enter
  - Try this with `getwd()` as well

```
library(here)
here()
```

```
[1] "/Users/wakim/Library/CloudStorage/OneDrive-
OregonHealth&ScienceUniversity/Teaching/Classes/F24_EPI_525/F24_EPI_525_site"
```

```
getwd()
```

```
[1] "/Users/wakim/Library/CloudStorage/OneDrive-
OregonHealth&ScienceUniversity/Teaching/Classes/F24_EPI_525/F24_EPI_525_site"
```

- `here` can be used whenever we need to access a file path in **R code**
  - Importing data
  - Saving output
  - Accessing files

Importing data

Using `here()` to load data

- The `here()` function will start at the working directory (where your `.Rproj` file is) and let you write out a file path for anything
- To load the dataset in our `.qmd` file, we will use:

```
library(readxl)
data = read_excel(here("./data/BodyTemperatures.xlsx"))
data = read_excel(here("data", "BodyTemperatures.xlsx"))
```

Watch out when using lubridate package simultaneously

Use `here::here()` if you have `lubridate` loaded within same `.qmd`. This will tell R to use the function `here()` within the `here` package instead of `lubridate`'s `here()` function. To call lubridate's function, we'd use `lubridate::here()`

Common functions to load data

Function	Data file type	Package needed
`read_excel()`	`.xls`, `.xlsx`	`readxl`
`read.csv()`	`.csv`	Built in
`load()`	`.Rdata`	Built in
`read_sas()`	`.sas7bdat`	`haven`

Using `here()` to load the data!

- I put the dataset “BodyTemperatures.xlsx” in your student files (under Data then Lessons)
- Go into those files and download the dataset into your personal class folder
  - Probably good to put it under “data”
- Within your Quarto file, in an R code chunk, load the data!