RStudio first aid

Eleutherodactylus Research Course · Fall 2026

Tom Devitt

Why this deck

Many of you told me you were never properly introduced to RStudio, and that you spend more time fighting it than doing the research.

That is a fair complaint. This deck is the introduction you should have had. Work through it on your own, at your notebook, in about 50 minutes.

Contents

1. The four panes
2. Console, chunk, or Render?, then **Exercise 1**
3. Where R looks for files, then **Exercise 2**
4. Restart, or Quit Session?
5. Reading an error, and packages, then **Exercise 3**
6. Playing audio, rendering from scratch, and asking for help

The four panes

RStudio is one window split into four *panes*, each with a job. A **pane** is just a section of the window.

Pane (usual spot)	What it is for	You will use it to
Source (top left)	Shows the files you edit, such as your team's `.qmd` notebook	Write and run code
Console (bottom left)	Where R actually runs. You type a line and R answers	Try one line, and read error messages
Environment (top right)	Lists the objects R is holding in memory right now	Check that `selections` or `w` exists
Files / Plots / Packages / Help / Viewer (bottom right, tabs)	Files in your home folder, figures, installed packages, help pages, and web pages R opens	See a spectrogram, look up `?spectro`, play audio

If a pane is missing, use **View > Panes > Show All Panes**.

Console, chunk, or Render?

A notebook (`.qmd`) mixes writing with **chunks**: grey boxes of R code, each opened by a line reading `{r}` between triple backticks. You can run code three ways.

Way	How	Good for	The catch
Console	Type a line and press Enter	Trying one idea	Nothing is saved in the notebook
Run a chunk	Click the green arrow on the chunk	Building your notebook one piece at a time	A chunk can use objects made by earlier chunks you forgot to run
Render	Click **Render** at the top of the Source pane	The finished page, run from the first line to the last	If anything fails, you get no page

Rendering means running every chunk in order, in a clean session, and turning the result into a web page. Your notebook is only finished if it renders.

Exercise 1: one line three ways

Open your pod RStudio (**edupod.cns.utexas.edu**, UT EID in lowercase, VPN on if you are off campus).

1. In the **Console**, type `x <- 5 * 3` and press Enter. Look at the **Environment** pane.
2. Type `x` and press Enter.
3. Create a new notebook: **File > New File > Quarto Document > Create**. Add a chunk (**Code > Insert Chunk**), put `y <- x + 1` and `y` on separate lines inside it, and click the green arrow.
4. Click **Render** (save it as `test.qmd` when asked).

Expected result.

- Step 1: Environment shows `x` with the value `15`.
- Step 3: `[1] 16` appears under the chunk.
- Step 4: **an error** saying `object 'x' not found`. The Console made `x`, but the notebook never did, because Render starts clean.

Where does R look for files?

The **working directory** is the folder R is currently "standing in". When your code says `"selections_ab.csv"`, R looks for that name in the working directory, and nowhere else.

```
1 getwd()          # where am I standing?
2 list.files()     # what is in this folder?
```

A **path** is the address of a file. Two kinds:

- **Absolute**: the whole address from the top, for example `/stor/work/FRI_Bio206_TD_Spring2026/` (the shared course folder on the pod)
- **Relative**: starts from the working directory, for example `calibration/obs1.csv`

When you render a notebook, R stands in **the folder that holds the notebook**. That is why a file that works from the Console (standing in your home folder) can go missing at render time.

Copy first, then work

You do not edit files in the shared course folder. You work on **your own copy** in your home folder (`~/`).

```
1 file.copy("/stor/work/FRI_Bio206_TD_Spring2026/team-measurement-template.qmd", "~/")
```

`file.copy()` returns `TRUE` if it worked. It returns `FALSE` if a file with that name is already in `~/`, and it will **not** overwrite your work. That is a safety feature. If you want a fresh copy, rename your old one first.

Keep the notebook, your `selections_*.csv` files, and the output in **the same folder**. Then relative paths just work.

Exercise 2: find the file

In the Console:

```
getwd()
list.files("/stor/work/FRI_Bio206_TD_Spring2026")
file.copy("/stor/work/FRI_Bio206_TD_Spring2026/team-measurement-template.qmd", "~/")
list.files("~/")
```

Expected result.

- `getwd()` prints the path to your home folder.
- The first `list.files()` shows the course materials, including `recordings`.
- `file.copy()` prints `[1] TRUE` the first time, `[1] FALSE` if you already have the file.
- The last `list.files()` includes `team-measurement-template.qmd`.

Now try `read.csv("not-a-real-file.csv")`. You should get *cannot open file … No such file or directory*. That message means R looked where you told it, and the file was not there.

Restart R, or Quit Session?

Symptom	Try	Why
Code behaves strangely after many edits	**Session > Restart R**	Clears the objects in memory and reloads R. Your files are untouched
A notebook works only when you run chunks in a certain order	Restart R, then **Render**	Shows what a fresh run does
Functions that were fine are "not found", even right after `library(tidyverse)`	**Quit Session** (the power icon, top right), then log back in	A suspended session can be left broken, and Restart R does not fix it
The page will not load, or RStudio is frozen	Quit Session, then reload the browser tab	Same

Quit Session ends your session on the pod and starts a fresh one when you log back in. It does not delete your files. Do this *before* you spend an hour debugging packages.

Reading an error message

```
Error in seq.default(...) : wrong sign in 'by' argument
```

An error has parts:

- `Error in` names the function that stopped. Here it is `seq.default`, which may be buried inside a function you called (`meanspec()`).
- **After the colon** is what went wrong, in R's words. Read it literally.
- In a notebook, the render output also gives the **chunk name and line numbers**. Start there.

Three rules:

1. **Read the first error only.** Later errors are usually fallout from it.
2. **Copy it exactly.** Do not retype or summarize it.
3. **Say what you expected.** "I expected 12 rows and got 0" is worth more than the error alone.

A **warning** is not an error. R kept going, but check that the result makes sense.

Packages and "could not find function"

A **package** is a bundle of functions someone wrote. `library(seewave)` loads one for this session. You need to load it every time you start R.

You see	It usually means	Try
`could not find function "spectro"`	The package is not loaded	Run the `library()` lines from the Setup chunk
`there is no package called 'warbleR'`	It is not installed	Ask a mentor. On the pod, installing can fail for reasons you can't fix
`could not find function "play_wave"`	It is a helper *we* wrote, defined in a notebook chunk	Run the chunk that defines it (Setup)
The same error after you ran `library()`	The suspended session is broken	**Quit Session**

On the pod, RStudio uses **R 4.5.3**. If someone sends you code that worked on their laptop, versions may differ.

Exercise 3: break it, then read it

In the Console, run these one at a time:

```
1 library(tidyverse)
2 tibble(a = 1:3, b = c("x", "y", "z"))
3 tibble(a = 1:3, b = c("x", "y"))
4 spectro(1)
```

Expected result.

- Line 2 prints a 3-row table.
- Line 3 fails: `Tibble columns must have compatible sizes`. The message tells you the sizes: `Size 3: Existing data` and `Size 2: Column b`. You gave one column 3 values and the other 2.
- Line 4 fails with `could not find function "spectro"`, because `library(seewave)` has not been run. Run it and try again: you now get a *different* error, which means you moved past the first one.

Write down, for each error, the function that stopped and the one sentence R gave you.

Playing audio in RStudio

Clicking a `.wav` in the **Files** pane does not play it: RStudio tries to open it as text and says *File is binary rather than text*. That is expected.

Our notebooks use the **Viewer** pane instead:

```
w <- read_any("some-folder/some-file.wav", from = 0, to = 5)
play_wave(w)                          # a play button in the Viewer pane
play_spectro(w, from = 1, to = 2)     # spectrogram, waveform, and a moving line
```

- A **waveform** shows the air pressure over time. Each repeat of the shape is one **cycle**.
- A **spectrogram** shows how the energy is spread across frequency (the vertical axis) over time.
- The speed buttons in `play_spectro()` slow the sound, which also lowers its pitch.

Both functions are defined in the Setup chunks of the team notebooks. If you get `could not find function "play_wave"`, run Setup first.

“Render from scratch”

Milestone 2 asks for a notebook that **renders from scratch**. That means a classmate could open it in a fresh session, click Render, and get your result, with nothing left over from your own Console.

Test it yourself, in this order:

1. **Quit Session**, then log back in
2. Open the notebook
3. Click **Render**

If it stops, the first error tells you what the notebook was borrowing from your old session: an object you made in the Console, a package you loaded by hand, a file in the wrong folder. Fix that, and render again.

The 10-minute rule

1. Read the **first** error, out loud, to your team (2 min)
2. Check the usual suspects: Setup chunk run? Right folder? **Quit Session** and render again? (3 min)
3. Look the function up with `?function_name` (2 min)
4. Still stuck at 10 minutes: **ask**. Do not guess for longer

How to ask a mentor or Claude: paste the exact error, the chunk that failed, and what you expected. Say what works and what does not. Details are in the Oct 14 deck, Asking Claude for help with R.

Do not paste recordings, or names or localities of undescribed species, into Claude. Made-up values debug just as well.

Take-home

- A notebook is done when it renders. If it only works when you click chunks, it is not done
- R looks for files in the notebook's folder
- Quit Session before you debug packages
- Read the first error only, and copy it exactly
- At 10 minutes, ask

Need a PDF from a notebook on the pod? Use `format: typst`, never `format: pdf`. The pod's TeX install is missing fonts, so `pdf` will fail.

Prepared with Claude Code (Claude Sonnet 5.5, Anthropic): slides and first draft.