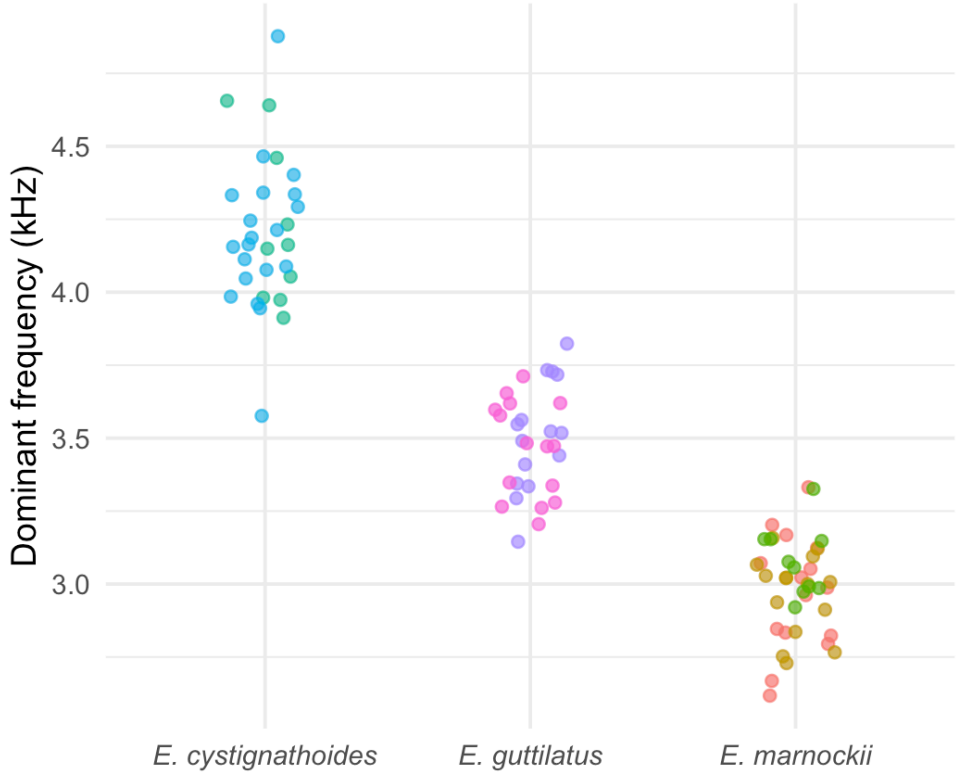
Reproducible workflows: one notebook, data to results

Eleutherodactylus Research Studio · Fall 2026

Tom Devitt

2026-10-05

Where this ends up



> We measured 100 calls from 7 males of 3 species. Dominant frequency averaged 3.5 kHz across males.

species	males	calls
E. marnockii	3	40
E. cystignathoides	2	30
E. guttilatus	2	30

Nobody typed these numbers. The sentence, the table, and the figure are all computed from the data when the document renders. Add a male, render again, and all three update. By **Milestone 2 (Oct 21)** your team notebook does this with your data.

Your turn: a notebook that works on my machine

Work in **pairs**. In the pod's RStudio Console:

```
share <- "/stor/work/FRI_Bio206_TD_Spring2026"
file.copy(file.path(share, "09-render-from-scratch.qmd"), "~/")
```

Open it and click **Render**.

- It fails. Read the error, find the line, fix it, render again
- Once it renders, **read the output closely**. One problem produces no error at all
- Write each fix in the *Fixes* list at the bottom

15 minutes. Then we compare notes.

Bug 1 · Paths that travel

`setwd()` worked for the author because `abc123` was the author. On anyone else's account, the folder doesn't exist.

```
share   <- "/stor/work/FRI_Bio206_TD_Spring2026"   # the one absolute path
rec_dir <- file.path(share, "recordings", "team3-nitidus-group")
sel_files <- list.files(".", pattern = "^selections_.*\\.csv$")
```

- Set the share path **once**, in the setup chunk
- Everything else is **relative to the notebook's folder** (`"."`). Render always starts there
- Never use `setwd()`, `~/Desktop/...`, or a path containing your EID. The steward's render has to work from the steward's folder

Bug 2 · The render is the truth

The author made `by_male` by running the lower chunk first, in the Console. Then the chunks got moved and nobody re-rendered.

Render starts a fresh, empty R session and runs the chunks top to bottom. Your Console and your rendered document live in separate environments. Expect this one to bite you a few times this semester.

- "It works in my Console" doesn't count. "It renders" does
- Before you trust anything, run **Session → Restart R**, then Render
- If the pod's RStudio acts strange after a break, don't debug it: **Session → Quit Session**, start a new one, render

The point about environments is adapted from Data Science in a Box (Çetinkaya-Rundel), CC BY-SA 4.0.

Bug 3 · Typed numbers go stale

Someone added a male and re-rendered. The table updated; the sentence didn’t.

Don’t type a number you computed. Write it inline:

```
We measured `r sum(by_male$n_calls)` calls from `r nrow(by_male)` males.
```

renders as

> We measured 100 calls from 7 males.

The same rule covers everything in a results paragraph: sample sizes, means, ranges, *n* excluded.

Your turn: the test, on your team's notebook

15 minutes. The steward drives; the team reads along.

1. Delete `team<N>_measurements.csv` and the rendered `.html`
2. **Session → Restart R**
3. Render

Write down:

- Did it render? **yes / no**
- If no: the **first** error message, and which chunk it came from
- If yes: any number in your text that was typed rather than computed

Don't fix anything yet. That's the homework.

Before Wednesday

Steward (it's your copy of the notebook):

1. Fix whatever broke, **in the notebook**, never by editing a CSV
2. Step 6: give the jitter a seed, so the figure is the same every render:

```
geom_point(position = position_jitter(width = 0.15, seed = 1), alpha = 0.7)
```

3. Add a **Decisions** section and a final `sessionInfo()` chunk, then run the test again

Everyone else: check that your file is named exactly `selections_<initials>.csv`, and send the steward one Decisions line: your start/end rule, plus any file you excluded and why.

Stuck on an error for 20 minutes? Stop, and bring the message Wednesday.

Wednesday

Work session + stand-ups. Each team reports:

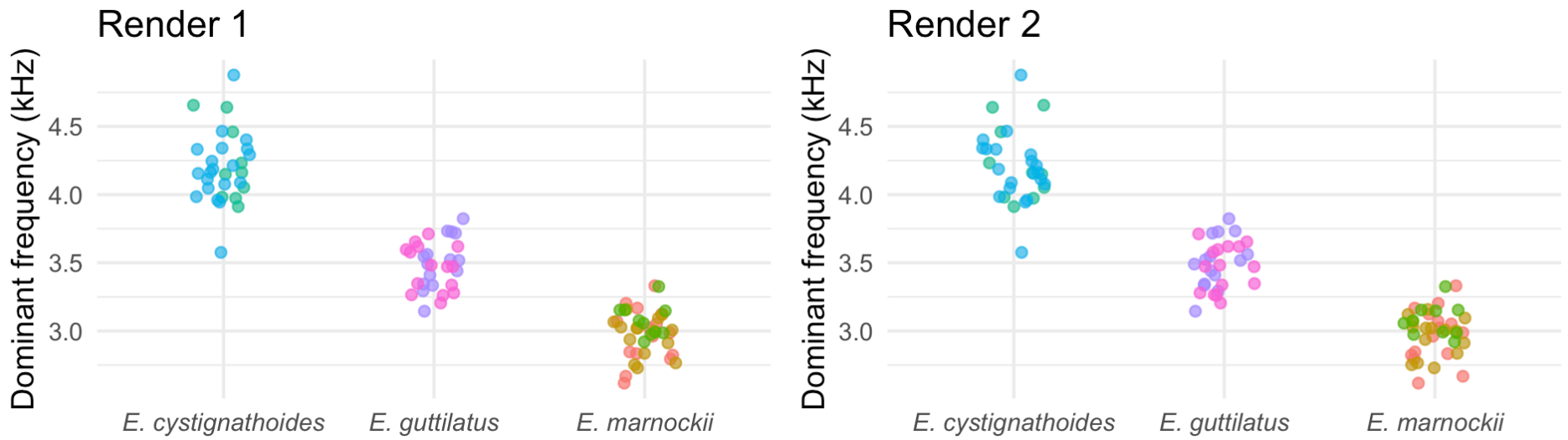
- **Does it render from scratch?** yes / no
- The error you couldn't fix, if there is one. We'll fix those first
- What's still done by hand

Next Monday: summary statistics. Have your notebook rendering through Step 5 by then.

Reference

These slides aren’t covered in class. They’re here for when you need them.

Reproducible ≠ repeatable



Same data, same message, different pixels. Wilke calls this **reproducible** (data and transformations fully specified) but not **repeatable** (identical down to the pixel). The fix is one argument:

```
1  geom_point(position = position_jitter(width = 0.15, seed = 1))
```

Your team notebook's Step 6 uses `geom_jitter()` with no seed. (Wilke, *Fundamentals of Data Visualization*, ch. 27)

So what does "reproducible" mean?

Question	Where your notebook answers it
Do the tables and figures come from the code and data?	**The test** (delete outputs, Restart R, Render)
Does the code do what you think it does?	Step 4b outlier spot-checks; looking at the spectrogram
Is it clear **why** each choice was made?	A **Decisions** section
Could the code run on more data?	A teammate's new `selections_*.csv` drops in and renders

The Decisions section gets one dated line per choice: your start/end rule, `wl = 512`, any band-pass, every excluded file and why. In November it becomes your Methods section almost word for word.

Checklist adapted from Data Science in a Box, CC BY-SA 4.0.

Raw data stay raw

Stage	Who touches it
Recordings on the share	nobody (read-only)
`selections_<initials>.csv`	each measurer: the **only** hand-made step
Team notebook (`.qmd`)	the steward
`team<N>_measurements.csv`, summaries, figures	nobody: Render writes them

If a number in the measurements CSV is wrong, **don't open the CSV and fix it.** The next render erases your fix, and nobody can see you made it. Fix it upstream, in code, with a comment:

```
selections <- selections |>
  mutate(male = case_when(
    file == "nitidus-group/nitidus/TJD_818_2.MP3" ~ "TJD_818",  # same male as TJD_818.MP3
    .default = male
  ))
```

One notebook, one steward

- **One** copy of the notebook, kept by the data steward. Four copies drift apart within a week, and then nobody knows which one is right
- Measurers send `selections_<initials>.csv`, not edited notebooks
- Use the filenames exactly. The notebook finds files by pattern (`^selections_.*\.csv$`), so `Selections-TD-final2.csv` is silently skipped
- End the notebook with `sessionInfo()`. It records the R and package versions your numbers came from

Not today, on purpose

These matter, and you'll hear about them later. They aren't what's standing between you and Milestone 2.

- **Git / GitHub:** not this semester. The pod plus one steward's notebook does the job with one less tool to learn. If you made an account in August, keep it
- **Pinning package versions** (renv): `sessionInfo()` is enough for a class project
- **Writing your own functions:** the template's `read_any()` and `measure_file()` already do it for you. Read them; you don't need to write more

Credits

- Lesson design follows the principles in Mine Çetinkaya-Rundel, *Teaching intro data science and assessing learning* (Preparing to Teach 2026): start with cake, skip the baby steps, hide the veggies
- Reproducibility checklist and the Console-vs-document environments point adapted from Data Science in a Box (CC BY-SA 4.0)
- Reproducible vs repeatable: Claus O. Wilke, *Fundamentals of Data Visualization*, ch. 27
- This deck, the course site, and your team notebook are all Quarto documents. Same Render button
- *Prepared with Claude Code (Claude Opus 5.5, Anthropic): deck drafting and revision*