



learnr Interactives

or: How I Learned to Stop Worrying and Turn my Students into Pixelated
Farmers

Prof. Liza Bolton

February 4, 2022

Caveat

I'm coming to you as a jack of all some trades and master of none. Nothing I'm about to talk about is necessarily the BEST way to do anything, but I hope seeing what I can manage inspires you to realize just how possible anything that catches your eye here is for you!

Accessibility note

You can find my slides and images descriptions [here](#).

Statdew Valley: my favourite use



Open the activity on the JupyterHub

(From the activity) Welcome to Statdew Valley!

You're grandmother has asked you to take over her old farm plot in *Statdew Valley*. Armed with statistical knowledge, you set out to begin your new life!

There are two of datasets for you to analyze to better understand the operation of your farm.

Note: The final 'hint' for code chunks will be the solution.

*Yes, this is totally an homage to **Stardew Valley**. Have I played? No, but à la Girlfriend Reviews on YouTube, this is based on "what it's like to live with someone who plays Stardew Valley."*

What is in this activity?

- [Optional] image manipulation with R (to create a 16 (ish) bit player avatar from a picture)
 - You can use R for so much more than just running regressions!
- Areas to write code (with hints and solutions)
- Multi-choice questions (ungraded) with commentary so learners can check if they are on the right track, but also to bring attention to important features
- Egregious bee puns and B(ee)TS references





You can't fight in here. This is the War Room!

But shouldn't the learners learn to code ~properly~ in R?

But shouldn't the learners learn to code ~properly~ in R?

Of course!

But shouldn't the learners learn to code ~properly~ in R?

Of course! It isn't very impressive for our students to go to their future employers/grad schools and say, sure I can code in R, but only if you set up a nice interactive template for me. I don't know how to load data, format outputs or conduct a full analysis.

But shouldn't the learners learn to code ~properly~ in R?

Of course! It isn't very impressive for our students to go to their future employers/grad schools and say, sure I can code in R, but only if you set up a nice interactive template for me. I don't know how to load data, format outputs or conduct a full analysis.

A big priority in all the courses I am involved with is ensuring students can write up useful, reproducible reports with RMarkdown. BUT these types of interactive activities clear out the other considerations and learners focus in on concepts and highlight specific skills, functions, etc.

So, no (unnecessary) coding in here. This is the Code Tutorial!

Course context (STA303)

Syllabus

Students engage with **asynchronous activities**, BEFORE synchronous activities.

Goals of **synchronous classes**: develop group problem solving skills, practise talking about stats, build a sense of class community and focus attention on tricky concepts and assessments.

Some learners find it hard to motivate themselves with asynch content, so making it more interactive can help.

Our uses

- Asynchronous tutorials/activities
 - Students can self pace
 - Prompted to check their understanding actively, not passively
- Interactive activities for team-based learning
 - I set up an activity with learnr on the JupyterHub
 - **Team Up!**

Why this makes your life easier and you should totally try it

- **learnr**: Make fun interactive activities, using the coding skills you already have (or are working on developing).
- **JupyterHub**: No fussing with students needing to download files and navigate the (sometimes) terrifying permissions, files structures and other pains of their local machine. **You provide a link and the activity loads in their browser.**
- **All**: Easy to improve, remix and reuse semester to semester. Unleash your creativity.

(Possible) barriers

- **Git & GitHub:** You don't have to be a Git power user to make this work for you. GitHub desktop and uploading straight to website are perfectly reasonable workflows.
- **HTML & CSS:** There are certain things that can be easier or more easily customized if you have a basic comfort with HTML and CSS, but these are pretty easy to pick up as you go from the resources and other people's templates,
- **Not an R user:** Become an R user! 🤗

Things that can go wrong

Tech

- Rmd files directly in the student's home directory
 - This was a whole thing
- Mysterious connection issues and loading lags

Students

- This is not a PDF. This means that for some students this is outside their comfort zone or preferences for note taking.
 - Suggestion: Talk about how you're intending for them to use these activities and learn from them. Connect students to **learning strategists** who can coach them on effective note taking.

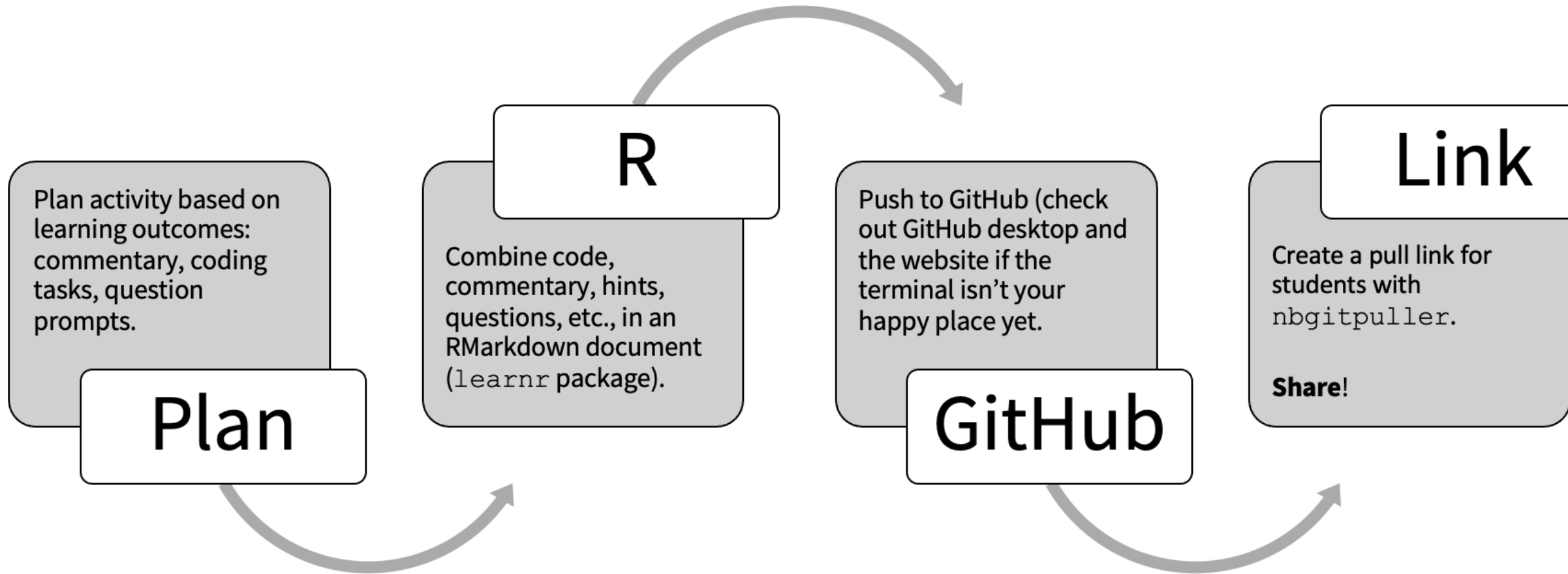
Assessments: Abundance of caution

There are a few common problems that are easy to teach students to solve, but there are occasionally connection issues that I can't get to the bottom of. With this in mind, I don't use these activities for **timed assessments**, other than the optional, graded group Team Up! activities, where usually one team member can make it work (and Team Up! activities are just *one* way to earn the possible marks).



A *hundance* of caution

The workflow



Git and GitHub for R users



Happy Git with R

"Happy Git and GitHub for the useR" was written by Jenny Bryan, the STAT 545 TAs, Jim Hester.

nbgitpuller

https://jupyter.utoronto.ca/hub/user-redirect/git-pull?repo=https%3A%2F%2Fgithub.com%2Fsta303-bolton%2Fsta303-w22-activities&urlpa

JupyterHub URL

https://jupyter.utoronto.ca

✓

The JupyterHub to send users to. [nbgitpuller](#) must be installed in this hub.

Git Repository URL

https://github.com/sta303-bolton/sta303-w22-activities

✓

branch

master

✓

Use [main](#) instead of [master](#) for new [GitHub](#) repositories

File to open

index.ipynb

This file or directory from within the repo will open when user clicks the link.

Application to Open

☐ Classic Jupyter Notebook

☐ RetroLab

☐ JupyterLab

☐ RStudio

☐ Shiny

☒ Custom URL

shiny/sta303-w22-activities/inst/tutorials/sta303_m3_statdew/sta303_m3_statdew-valley.Rmd

✓

nbgitpuller (key text)

- JupyterHub URL: `https://jupyter.utoronto.ca`
- Branch: `master` (check this on your GitHub! Use `main` instead of `master` for new GitHub repositories)
- Git Repository URL: `https://github.com/sta303-bolton/sta303-w22-activities` should be the top-level of the repo on github, not a subfolder
- Custom URL: `shiny/sta303-w22-activities/inst/tutorials/sta303_m3_statdew/sta303_m3_statdew-valley.Rmd`
structure should be: `shiny/repo-name/path-to-rmd/rmd-file.rmd`

Tips

- Plan to spend some time in a first class (or likely two) showing students the main troubleshooting options

Here are my top things to check:

- Close all other interactives
- Check no there are no naked Rmd in your Home Directory
- Delete and pull again
- Go in the back way

MarkUs

So now your students know all this great code...how do you set and mark meaningful assessments of their skills and knowledge?

MarkUs

So now your students know all this great code...how do you set and mark meaningful assessments of their skills and knowledge?

MarkUs!

MarkUs

So now your students know all this great code...how do you set and mark meaningful assessments of their skills and knowledge?

MarkUs!

I'm still very new to MarkUs, STA303's first assessments are just beginning marking on it *today* (check out the instructions for their [mini-portfolio here](#)).

I am so excited about the potential and how well it matches my course's needs. Watch this space!

Links

- [learnr documentation](#)
- [JupyterHub support \(UofT\)](#)
- [JupyterHub guide with gifs I made for my students](#)
- [nbgitpuller](#) for making pull links
- [Happy Git with R](#)
- [Statdew Valley interactive](#)
- [StatDew Valley on GitHub](#)

 We'll meet again, don't know where, don't know when


But email me in the meantime: liza.bolton@utoronto.ca