Making interactive webapps in R using Shiny

Stuart Lacy

17/02/2021

Introduction

- Who is this tutorial aimed at?
- What will this tutorial cover?
- What is Shiny?
- Why use Shiny?
- How do we use Shiny (with examples)

What is Shiny

- Webapp framework built in R
- Show examples of websites:
  - COZI
  - Genetic subtype predictor
  - Multi-state model
  - A more professional option!

Why use Shiny?

- Want a website and already comfortable with R!
- Don't have to use any Javascript, HTML, CSS, or consider web hosting
- Very easy to get website up and running quickly with minimal boilerplate
- Can host on University infrastructure `shiny.york.ac.uk`
- However, if want to design a more professional looking website will need to use Javascript & CSS, at which point Shiny doesn't shine...

What sorts of websites can I create with it?

- Text, plots and tables using the exact same syntax
- Interactive widgets:
  - Plots with controls
  - User input: sliders, buttons, checkboxs
  - Maps
  - Animations
  - Upload files
- All of this just with using R only, with Javascript can do a lot more

Uses

- Have a place to visualise live data online
- Provide tools to accompany published models, i.e. showing how model works and allow users to run their own data through it
- Exploring large datasets interactively
- Sharing work with project collaborators to save them asking lots of questions
- If need some widgets, like Maps

What does code look like?

- Go through examples from:
  - Static page. Just plot, text, table. Show code layout and how server and UI link together
  - Interactivity: Use plotly and DT on the same outputs
  - User controls: Add in slider, text box etc... Discuss reactivity
  - Dashboard / UI: How to make app look better
  - Extending with JS/CSS: Worth including or just mention it's possible?

Hosting:

- Shinyapps.io
- shiny.york.ac.uk

Conclusions

- Shown how easy Shiny is to use
- Compare code structure to JS apps showing how can get hard to follow for bigger projects?