Shiny

Joseph Longworth

What is Shiny

- An R package for building interactive ‘web’ applications
- Leverages R’s statistical and graphical capabilities
- Offers a variety of user interface (UI) elements for easy interaction

Class activity 1 : Explore some shiny examples 10 min

Look at what can be achieved with shiny with some of the exalples at https://shiny.posit.co/r/gallery/

Building Blocks

Key Components:

- `ui`: Defines the user interface (layout, elements)
- `server`: Handles user interactions and updates outputs

Different styles:

- Single `App.r` file containing all code
- Multiple files `ui.r`, `server.r`, `global.r` each containing components (shorter scripts)
- Modules - Shorter targeted sections of a code that can be re-utalised and isolate variables

Class activity 2: 10 min

Go to https://shinylive.io/r/examples/#hello-shiny

Keeping in mind:

- Look at the code
- Can you identify the UI and server sections
- Using **Hello Shiny!** look to understand how `inputId = "bins"` and `input$bins` are linked and drive the activity of the site.

User Interface (UI)

Components:

- `fluidPage`: Layout container for your app
- `titlePanel`: Sets the title of your app
- `sidebarLayout`: Divides the app into a sidebar and main panel
- Input elements like `sliderInput`, `radioButton`, `checkboxInput`, etc.
- Output elements like `plotOutput`, `textoutput`, etc. to display content

Server Logic

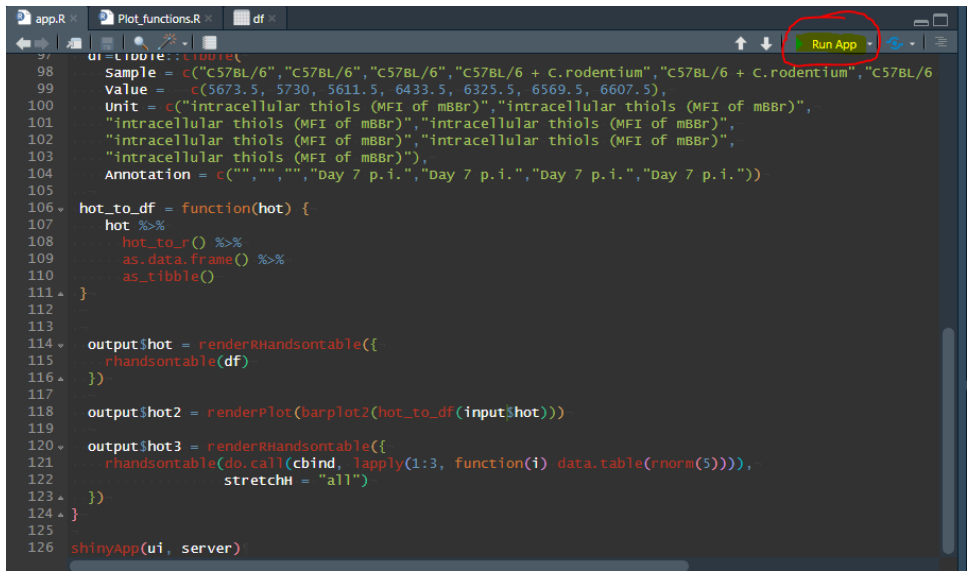
Functionality:

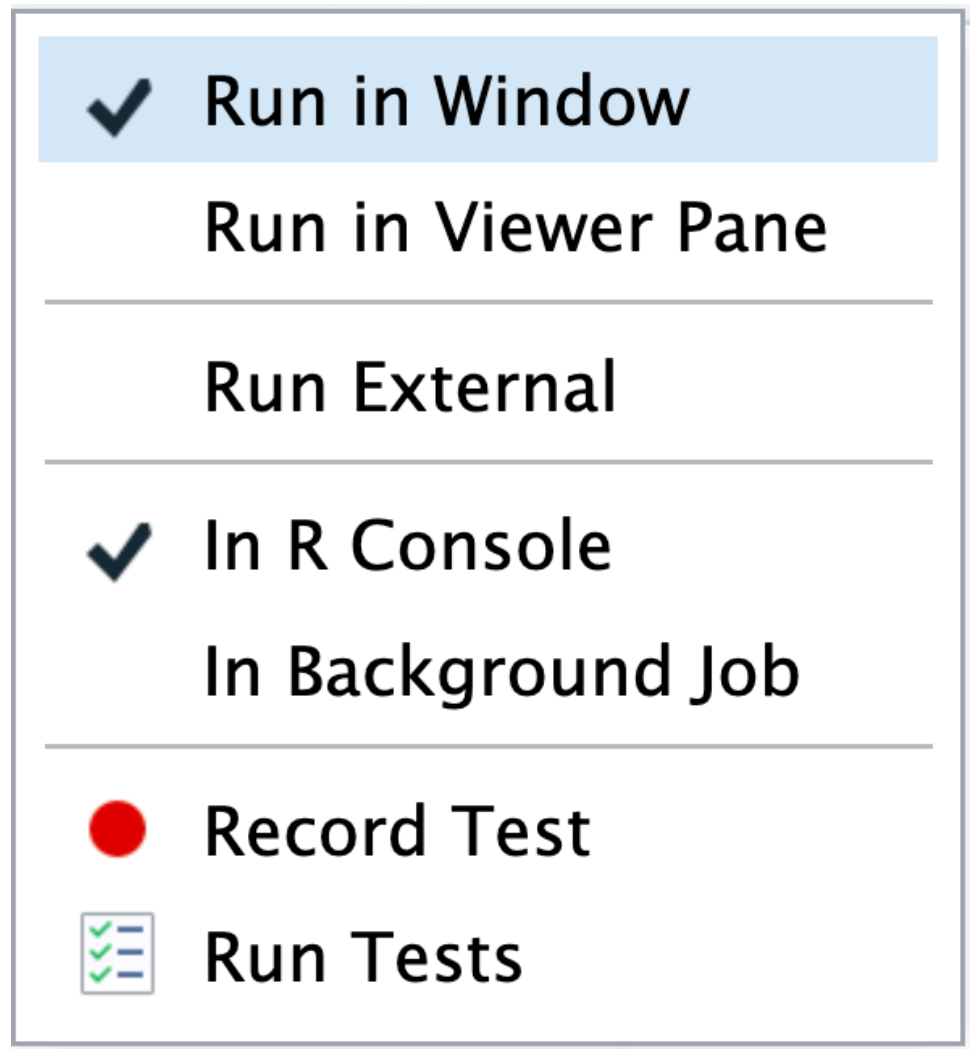
- Processes user interactions from the UI / Defaults
- Updates the application's state based on user input
- Generates content for output elements using functions like `renderPlot`, `renderText`, etc.
- Written same as general R code
- Observe events and reactive events trigger sections of server based on ui triggres

Deployment

- Locally in Rstudio
  - Files shared by **website**, **Github** etc…
- On a Shiny Server locally hosted
- On a Deployment site e.g. https://www.shinyapps.io/
- New ‘serverless’ using **WebR** / **Shinylive** hosted on static pages and processing in browser

Launching an app





Class activity 3: part 1 5/15 min

Run a shiny app directly from your own Session of R

open r studio and using the `File` menu select `new file`

select `Shiny Web App...`

follow the instructions for naming and saving location (choose wither app.r or ui.r and server.r

open the file created

run using the Run App button

test how it can be viewed in viewer or browser etc…

Class activity 3: part 2 5/15 min

Run a shiny app directly from remote

Applications are often shared on Github of GIST, if the App.r or ui.r/server.r are in the top level shiny apps can be downloaded and run using runGitHub

```
library(shiny)
```

```
runGitHub(repo = "Brenner_Graphs",username =
"josephlongworth")
```

If it does not run look at the consol some packages will need to be installed then try again

Class activity 3: part 3 5/15 min

Shiny can also be incorporated into packages which can simplify package management

run

```
library(devtools)
```

```
devtools::install_github("DII-LIH-Luxembourg/cycadas", dependencies = TRUE)
```

This will download the package from github

```
library(cycadas)
```

```
cycadas()
```

Modern Shiny

- Shiny is developing fast!!!
- Shiny also works with python
- shiny can be embedded in documents and presentations i.e. quarto
- packages and ‘other web coding’ can be incorporated styling the application as a modern website
- Pipelines are available to auto generate complex app structure Aplison

Class activity 4: 20 min

using the app local to your session. Try copying widgets form online to explore

See what you can produce.