Summary Sheet

AUTHOR
Julia Gallucci

Class 7

Shiny

Layout:

Two key components of a shiny app

1. `UI or user interface` ; defines the visual elements and layout of the Shiny app. It specifies how the app will appear to the user and what input controls and output elements will be available.

2. `server function` ; handles the data processing, calculations, and interactions in the Shiny app. It defines how the app reacts to user input and generates the corresponding outputs. It specifies how the app how it will work/ the behavior of the app.

to execute **`shinyapp(ui,server)`**

Simplest way to **create a shiny app**

`File > New File > Shiny Web App > Single File > Create`

How to run app

- button in the document toolbar
- also prints a URL where your app can be found

How to stop app from running

- stop sign icon on R console *or*
- close shiny app window

Adding UI controls:

Here are some common UI controls:

• `fluidPage()` layout function sets up the basic visual structure of the page.

• `selectInput()` is an input control that lets the user interact with the app by providing it values. Creates a dropdown menu control in the UI of the app, allowng users to select a single option from a list of choices presented in the dropdown menu.

• `verbatimTextOutput() and tableOutput()` are output controls that tell shiny where to put rendered output ; used to display text and tabular data, respectively.

Adding behaviour:

By using the appropriate `render{Type}()` function in combination with the corresponding UI element, you can generate and update various types of outputs in your Shiny app based on reactive inputs and expressions.

each `render{Type}()` function is designed to produce a particular type of output ie text, tables, plots etc. Here are some common server behaviors:

• `renderText()` : This function is used to render text output. It takes an R expression or a reactive expression as an argument and updates the corresponding `textOutput()` element in the UI with the output.

- `renderTable()` : This function is used to render tabular data. It takes a data frame or a matrix as an argument and updates the corresponding `tableOutput()` element in the UI with the table.

- `renderPlot()` : This function is used to render plots. It takes a plot-generating R expression or a reactive expression as an argument and updates the corresponding `plotOutput()` element in the UI with the generated plot.

- `renderImage()` : This function is used to render images. It takes the path or URL to an image file as an argument and updates the corresponding `imageOutput()` element in the UI with the displayed image.

These are just a few examples of the `render{Type}()` functions available in Shiny.

Reducing Duplication with reactive expressions:

technique to deal with duplicate code: *reactive expressions*

you can create a reactive expression by wrapping a block of code in `reactive({…})` and assigning a variable to it. then use a reactive expression by calling it like a function.

Resources for more info:

- https://mastering-shiny.org/index.html

- https://shiny.posit.co/r/gallery/

Ethics

Key terms:

- **voluntary consent** of a human subject is crucial, ensuring that the person has the legal capacity to decide without any form of coercion, deception, or constraint. It requires providing comprehensive information about the experiment, including its nature, duration, purpose, methodology, expected inconveniences and hazards, and potential effects on the participant's health or well-being, enabling them to make an informed and enlightened decision.

- **randomized consent**, the approach for obtaining consent is limited to those who have been assigned to the non-standard treatment group. This approach aims to avoid putting individuals in stressful situations where they are asked to provide consent for a treatment they may not receive. By focusing on those assigned to the non-standard treatment, the design respects the principle of avoiding unnecessary burden and potential disappointment for participants who may not ultimately receive the treatment being considered.

- **adaptive randomization**, treatment assignments are made sequentially, and the outcome of each treatment influences the probabilities of selecting the next treatment. For example, if a subject is assigned treatment A and survives, the probability of selecting treatment A for the next subject is increased. Conversely, if the subject dies, the probability of selecting treatment B is increased. This adaptive approach allows for adjustments in treatment allocation based on observed outcomes, aiming to optimize the allocation of treatments throughout the study. The study concludes after a predetermined number of participants have been enrolled.

- **equipoise** referrs to the state of genuine uncertainty regarding the comparative effectiveness of different treatment options. It suggests that when designing a randomized controlled trial (RCT) or making treatment decisions, researchers and clinicians should be genuine uncertainty about which treatment is superior, ensuring that participants are not exposed to unnecessary risks or denied potentially beneficial treatments.

Inequity

- **diversity** encompasses a range of characteristics that influence our viewpoints, encompassing aspects such as gender, ethnicity, race, abilities, sexual orientation, age, education, values, beliefs, hobbies, family status, religion, and profession. It highlights the variety of traits that contribute to individual perspectives and experiences.

- **inclusion** refers to creating an environment where all individuals, regardless of their diverse characteristics or backgrounds, feel valued/respected. It involves actively embracing and accommodating differences, fostering a sense of belonging, and ensuring equal access to opportunities, participation, and decision-making processes. Inclusion promotes a culture of equity, diversity, and collaboration, where everyone’s perspectives and contributions are recognized and appreciated.

- **equity** refers to fairness in distributing resources and opportunities to address systemic disadvantages and promote equal access and outcomes for all.

- **stereotypes** refers to oversimplified, generalized beliefs or assumptions about a particular group of people. They are often based on limited information, preconceived notions, or societal biases, rather than individual characteristics or experiences. Stereotypes can be harmful as they can perpetuate biases, contribute to discrimination, and limit people’s opportunities based on inaccurate or unfair assumptions.

- **bias** refers to personal preferences or inclinations that can interfere with our ability to be impartial or objective. Unconscious bias occurs automatically when we make rapid judgments and assessments based on previously stored information, including stereotypes. There are several types of biases, including appearance bias (treating someone differently based on their looks), similarity bias (favoring people who are like us), affinity bias (connecting with those who share similar experiences), and confirmation bias (interpreting information in a way that confirms existing beliefs while disregarding contradictory evidence). These biases can impact our perceptions, decisions, and interactions, and it is important to be aware of them in order to promote fairness and inclusivity.

Professional skills

presenting data tables:

`kableExtra()` package is used for rendering professional-looking HTML table outputs. It provides additional customization options and styling capabilities for tables created using the knitr and kable functions in R. It extends the functionality of the basic table output and allows for more advanced formatting, such as adding captions, footnotes, merging cells, changing cell colors, applying conditional formatting, and creating stylish HTML or LaTeX tables.

Some key features of `kableExtra` include:

- **table customization:** kableExtra offers various functions to modify the appearance and formatting of tables, such as \`kable_styling()\` to apply predefined or custom CSS styles, \`footnote()\` to add footnotes to the table, and \`cell_spec()\` to format individual cells.
- **table manipulation:** kableExtra provides functions to manipulate table elements, such as \`add_header_above()\` to add headers above groups of columns, \`add_header_spanner()\` to create spanning headers, and \`collapse_rows()\` to merge rows based on specific conditions.

formatting reports:

The `bookdown` package is a versatile tool for authoring and formatting various types of documents, including reports, and interactive online publications. It builds upon the Markdown syntax and extends it with additional features specifically designed for creating professional and customizable documents.

Some key features of the `bookdown` include:

- **Customization of header** using the YAML metadata fields.
    - To set the **header** of your bookdown document, you can include the following YAML metadata fields:

      ```
      `title`: Specify the title of your book or document.

      `author`: Provide the author's name(s).

      `date`: Optionally, include the date of publication or creation.

      `site`: Set to `bookdown::bookdown_site` to indicate that this is a bookdown
      project.
      ```

    `` `output`: Specify the desired output format, such as `bookdown::pdf_book` for PDF output ``

- Customization of **set-up** chunk. The ``` `knitr::opts_chunk$set()` ``` function in bookdown is used to set global options for code chunks in your document. For example, the following options for code chunks:

    - `` `echo` `` : This is the option to hide the source code of the code chunks in the output. Only the results or output produced by the code will be displayed.

    - `` `message` `` : This is the option to hide any messages generated by the code chunks. Messages are typically informational or diagnostic messages printed during the code execution.

    - `` `warning` `` : This is the option to hide any warning messages generated by the code chunks. Warnings are issued when potential issues or problems are encountered during the code execution.

- **Cross-referencing and citations**: You can easily create and manage cross-references to figures, tables, sections, and equations within the document. It also supports generating citations and bibliographies using popular reference management systems like BibTeX or Zotero. `\@ref(fig: ) <- to reference figure, or [@mycitedwork] <-to reference citations in text`

- To set up **end-of report** , you can specify the end of your report by adding a section or a chapter that serves as the concluding part. This section is typically placed towards the end of your bookdown document, following the main content and any additional chapters or sections.

    -
      ```
      # Conclusion <- create chapter title called "Conclusions"

      <write your concluding remarks>

      \newpage

      # References <- create a chapter title called "References"

      <References will be automatically attached to the end of the report by bookdown>
      ```