**Video Script: Section 7 Video 1 – creating and using tabs**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Description | Action on screen | Narration |
| 1 | Introduction  (Outcome and why it is desirable)  This should give the viewer an idea of the outcome of the task at the beginning of the videos and set the stage and expectations of the viewer. | Refer to PPT | A webpage can be composed of semantically different elements that can make it complicated and crowded.  In this video, we are going to see how to use tabs for visually separating different parts of the main panel. |
| 2 | Context(Problem/Solution)  Present the viewer with a real-world solution and how the situation would pose as a challenge. It always helps to draw the viewer's attention using a use-case. Metadata template can be used here. |  | If your data is complicated and you have a significant number of results to show on the page, it’s a good idea to group them in separate tabs. |
| 3 | Guidance (How to do it and how it works): |  | Tabs in Shiny are very easy to set up. Let’s have a look at one example. |
| 4 |  | Open RStudio and run in the command line:  Point to Sectin 07 in the console panel in RStudio  library(“shiny”)  runApp(“activity\_07\_01”) | Open Rstudio,:  Make sure you are in the folder ‘section 07’.  And run in the command line  library(shiny)  runApp(activity0701) |
| 5 |  | In a browser:  A description... | We see two tabs, each with their own title and content: Distribution and Averages. |
| 6 |  | Select a property, switch back and forth between the tabs. | Selecting a diamond property updates the content of both tabs.  Here in distribution  and here in averages |
| 7 |  |  | Let’s look at the source and see how it’s implemented. |
| 8 |  | Back to RStudio, open UI.R and server.R in the editor.  Start with UI.R  Highlight where appropriate. | In UI.R, we have defined a tabset using tabsetPanel, which will contain each tab.  Each tab is created with tabPanel.  tabPanel expects the title of the tab as a first argument, then one or more of renderers like plotOutput or verbatimTextOutput etc.. |
|  |  | In the console, type:  ls("package:shiny", pattern = "Output")  result:  [1] "dataTableOutput" "htmlOutput" "imageOutput" "plotOutput" "tableOutput" "textOutput" "uiOutput"  [8] "verbatimTextOutput" | you can use the command ls to find all the possible output types:  ls("package:shiny", pattern = "Output") |
| 9 |  | Switch to server.R | server.R doesn’t need any reference to the tabs and the code looks exactly the same, as if there were no tabs: all the communication is done through changes in output’s properties. |
| 10 |  |  |  |
| 11 |  |  |  |
| 12 |  |  |  |
| 13 |  |  |  |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 | Conclusion:The video concludes by showing the viewer that the goal has been achieved, and reminding them why they should be happy about that. A PowerPoint summary slide with the key points emphasized would make it easier for the viewer to remember what was covered in the video | Back to PPT | In this video, we've seen how to use tabs to separate different portions of our interactive webpage. This is very useful when building complex dashboards with multiple graphs and textual reports.  In the next video, we’ll learn how scoping works in Shiny. |