**Video Script: Section 8 Video 2 – Building the time series plot**

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | In this video, we are going to build a shiny app showing the time series of an economic indicator. |
| 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. |  | As part of our dashboard, we need to build a user interface and the plot for the evolution of some quantities through time. |
| 3 | Guidance (How to do it and how it works): | Launch R and run in the command line:  worldBank<- read.csv("WDIDataDashboard.csv", sep = "\t", header = TRUE)  View(worldBank) | Let’s first see what the data looks like.  In Rstudio, click on import dataset and select WDIDataDashboard.csv  Select the header.  And click import.  The data is imported and shown in the editing panel |
| 4 |  | Appearing:  iso2c Country Year AG.YLD.CREL.KG EG.IMP.CONS.ZS EG.USE.COMM.FO.ZS FI.RES.TOTL.CD MS.MIL.MPRT.KD NY.GDP.PCAP.CD  1 CA Canada 1960 NA 25.01942 83.19244 2012986400 1.710e+08 2294.569  2 CA Canada 1961 984.989 18.36862 83.65490 2296085100 1.747e+09 2231.294  3 CA Canada 1962 1595.418 10.51505 84.33322 2561922010 1.071e+09 2255.230  4 CA Canada 1963 1838.864 11.42195 85.34442 2615182920 1.017e+09 2354.839  5 CA Canada 1964 1527.185 10.00631 85.42013 2893026720 7.280e+08 2529.518  6 CA Canada 1965 1730.392 11.53602 86.21466 3040845600 5.270e+08 2739.586  SH.DYN.MORT TX.VAL.MRCH.CD.WT GDPPCKD AG.PRD.CREL.MT SP.POP.DPND SP.POP.TOTL EN.URB.LCTY.UR.ZS SP.URB.TOTL.IN.ZS  1 32.6 5818300000 13499.54 NA 70.66209 17909009 14.10340 69.0610  2 31.6 6085260000 13650.65 16730306 70.95974 18271000 14.34011 69.8272  3 30.5 6212630000 14352.69 29261250 71.04911 18614000 14.39919 70.5934  4 29.4 6776990000 14817.80 34122307 70.87631 18964000 14.45979 71.3596  5 28.3 8037050000 15515.16 28504578 70.37828 19325000 14.51966 72.1258  6 27.1 8473210000 16248.04 32228771 69.52798 19678000 14.59111 72.8920  SP.DYN.TFRT.IN AG.LND.AGRI.ZS EN.ATM.CO2E.PC EG.USE.ELEC.KH.PC  1 3.811 NA 10.77085 5630.630  2 3.753 7.678553 10.62790 5605.112  3 3.681 7.692629 11.13063 5791.125  4 3.607 7.706705 11.13210 5966.041  5 3.456 7.720671 12.30537 6468.926  6 3.115 7.734857 12.81400 6909.899 | This data frame contains about 20 economic indicators for 9 countries, for the period 1960 to 2011. |
| 5 |  |  | The column names are the codes used by the world bank database. We’ll use the vector *codeToName* to translate a code into something more human-readable. |
| 6 |  | Open timeSeries/UI.R, timeSeries/server.R and global.R in the editor. | Let’s have a look at the code. Open timeSeries/UI.R, timeSeries/server.R and timeSeries/global.R in the editor. |
| 7 |  | Switch to global.R | We don’t want to present the cryptic column names to the user, so both UI.R and server.R will need to have access to codeToName (and nameToCode, its reverse) to be able to display the full name of the indicators. We therefore store them in global.R |
| 8 |  |  |  |
| 9 |  | Switch to server.R | We only need the data to be loaded once, so we put the code outside shinyServer(). |
| 10 |  |  | The ggplot contains 2 layers: a geom\_line for a continuous line and a geom\_point for each data point.  Xlab and ylab set the axes’ titles with the indicators descriptions. |
| 11 |  |  | To switch the log-scale on, we simply add scale\_y\_log10() to the ggplot object p. |
| 12 |  | In command line:  shiny::runApp(“timeSeries”)  A description... | To see it in action, run:  shiny::runApp(“timeSeries”)  The dropdown menu and the y-axis label have descriptive names.  Clicking on log scale toggles the log scale on the y axis on and off |
| 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 built the time series interface for our dashboard. We saw how to map cryptic column names to more readable descriptions, where to put global variables and how to add a logarithmic scale to a plot.  In the next video, we’ll see how to make bubble charts. |