Creating Shiny Tools

Kristin Bietsch, PhD

Avenir Health

Janurary 2021

Introduction

Shiny is an extension of R where you can "quickly" make online tools. If you are already comfortable coding in R, the learning curve for Shiny is not steep, and is much easier than learning a different programming language. Shiny apps can be very simple or very complex. The goal of this tutorial is to introduce you to what Shiny can do and how it can be integrated into your work.

Benefits of Shiny

- Easy to update
  - No one is using a 5 year old version on their desktop
- Users do not need Excel
- No web programming skills required
- Code can be posted online

Two Common Shiny Apps

I would classify the Track20 Shiny apps I have built over the last few years into two broad groups:

- Choose a country, see results
  - User chooses from a dropdown or checkbox menu and sees results for selected geographies
  - Examples:
    * Equity Model
    * Nigeria PPFP
- Models with editable default data
  - User choosed a geography from a dropdown menu, default data is loaded into the model, user can edit default data, model output is displayed
  - Examples:
    * MaxCPR Model
    * DMPA-SC Model

Before Using Shiny

- Download R and R Studio
  - Install packages "rsconnect" and "shiny"
- Sign up for a free account on ShinyApps.io
- Sign up for a GitHub account if you want to make your code available online

General Steps to Making a Shiny App

- Write out your model in R
  - Example of how you would filter results
  - Include code for any graphics
  - Prepare any default data
    * I like to make a CSV with one row per country

- Sketch out what you want your app to look like
  - Gather any images you want to include
- Create a folder called "App-1"- this is a suggestion, you can change it
  - In that folder, create 2 folders: "data" and "www"
  - Put your default data in the "data" folder and any images in the "www" folder
  - I also like to put my "non-shiny" code in this folder so I can keep everything together

Writing R Code

When writing a shiny app, I start by writing the model in RStudio. I like to do all the munipulation in one section, then I filter an example region and create the graphic and/or table display I would show if that were the selected region. Pre-writing everything without Shiny helps me make sure all my code is working and to know what packages and data I will load into my app. It is also nice because I apply my model to all countries/regions with data, so if someone wants to know about all countries I can export those results.

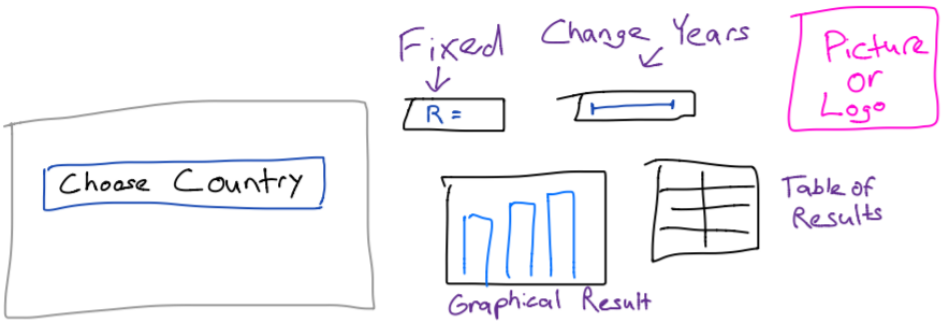
For your first Shiny app, we are going to project populatios for global regions forward from 2020. In the example folder, you will find a CSV file with default data for 6 regions, the annual growth rate of each region, and the population in 2020.

Example Code: Part 1 Writing your Model

File "Model in R not Shiny 011821.R" opens the CSV and mutates the data to create populations through 2030 with the given growth rate. After creating the dataframe "population" I create an example dataframe where I filter the data for one region. This new dataframe, "choosendata" is what I will display as a table and what I will use to make my result graphic.

I then draw a picture of what I want my shiny app to look like and which

parts will change.



My app will include:

- A photo or logo to brand the app
- A side panel where the user will choose a region
- A display of the growth rate (R)
- A slide bar where the user can choose the last year of the projection
- A bar graph showing results
- A table showing results

Example Code: Part 2 Build an Empty Shiny App

In this section, we will build an empty app which we will eventually fill in.

- Add a header with your name, date, project, etc
- Save file in "App 1" folder as "app.R"
- Load needed packages (rsconnect and shiny) plus any of the packages you used for your model
- Load your default data from the data subdirectory
- Create the UI and the Server areas
  - Note: the hardest part of building a shiny app is balancing your parentheses and making sure commas are in the right place!

- The last line of code (shinyApp(ui = ui, server = server)) creates your Shiny app
  - You should then notice in the top right corner that "Run" now reads "Run App"

Example Code: Part 3 Lay out the UI

Next we want to lay out our user interface (UI). We are going to add a title, our picture/logo, side panel, main panel, and all of our headers

- Some of the code is the same as used in HTML
- You are running your app "locally" for the time being. I like to run the app after ever step until I am comfortable with the code... it is much easier to debug now

Steps for laying out your UI

- Add your title
- Create a section called "sidebarLayout" which includes both your "sidebarPanel" and your "mainPanel"
- Create multiple columns within your rows for both displaying the input data and results

Example Code: Part 4 Add Editable Components

Now you will add in components to our UI that will change!

- First we want people to select a region, so create a dropdown menu listing all regions using "selectInput"
  - Give it a name which you will use to refer to it in the server section
  - Write a label which will be displayed above the box
  - List the choices
- Create a slidebar so the user can choose the last year of the projection using "sliderInput"

Example Code: Part 5 Build Out the Server

This next section will add our model into the server and filter our results using the information supplied by the user in the UI. You will notice it doesn't change anything about the display of the app... we will link the UI and server together in the next section.

Example Code: Part 6 Connect Server Outputs with the UI

Now create your server objects that are displayed in your UI. First create them in the server section, then go up to the UI and insert them.

- First, display the growth rate using "renderText"
- Second, make your graphic- remember your ggplot code from your model, using "renderPlot"
  - You will have to change the name of the dataset, but everything else can remain the same
  - Then place the graphic in the UI using "plotOutput"
- Third, add a table of results using "renderTable"

Bonus! Example Code: Part 7 Changing Default Data to Editable Data

The app you have now produced matches our initial drawing. But if you want, we can make the growth rate changeable by the user. We still want it to show the UN number initially, and for this to change when the region changes, but you can make it editable in case a user wants to see what would happen if the growth rate increased or decreased.

- Start in the UI by changing the displayed growth rate from a "textoutput" to a "numericInput"
- Set the "value" to anything you want, it will change once the region is loaded. I choose to put it as the growth rate of the first region, so the user doesn't see it update (try entering "1" and you will see it change)

- Then move to the server section, where we will enter an "observeEvent" section, which will include "updateNumericInput"
- We are telling the code to observe which region is choosen in the drop-down menu, then choose the R from the region in the dataframe "data" that matches the choosen region
- Next we need to update our code so the editable R is now part of our model
  - We refer to the new R as "input$R"
- We can remove "output$Growth" from our code

Deploying Shiny

When you are happy with your app, you can move it from your desktop to the web

- Save your file and run the local version
- In the local version, click "Publish"
- Connect to publishing account (ShinyApps.io)
- Go to ShinyApps account
  - Show token- copy token
  - Paste in pop up
- Choose which of your accounts you want to use
- Name the app (this will be in its URL)

Conclusions

You have now made your first shiny app! I am very excited for you all to learn Shiny and teach me new features.

Acknowledgement

Thanks John for suggesting this workshop and Shiza for helping when I get mad at Shiny.