Shiny (introductory article)

(This article was first published on **ipub » R**, and kindly contributed to R-bloggers)
No code for once. This is an independent, introductory article about RStudio's Shiny web application framework for R. This could be useful for people who don't know what Shiny is. Or for those of you who want to find out when to use Shiny.

What does Shiny do?

Shiny is a fine product. It allows you to make your data and research accessible to a non-R audience. This is useful in many situations. For example, if you are a pharma researcher, you can provide your decision makers with the summary data, research and reports they need. Or, if you are an academic researcher, you can **publish** your research online, in a reproducible manner.

Also, Shiny brings **interactivity** to R, letting you add selectors, sliders, and other input widgets to your screens.

If you have never seen a Shiny app, make sure to check out the Shiny Gallery.

What is Shiny?

Shiny is an ecosystem, which comprises:

1. shiny, an R package
2. Shiny Server, an application Server (in different flavors)

The first one, the R package, contains itself a minimalistic application server as well. So, to get started, all you need is the R package.

How do I build a Shiny app?

No worries, this post contains no code. But there are plenty of introductory examples in the web. For example, I can highly recommend the RStudio Shiny tutorial and this page: http://shiny.rstudio.com/articles/.

How difficult is Shiny?

Even if you are a seasoned R developer, some things will look unfamiliar. Shiny implements a form of event handlers, called **reactive** elements. These work great and contribute a lot to the Shiny magic. However, reactivity is not a concept common in R, where program flow is typically sequential. If you want to know more about reactivity, see here.

But even if you are a seasoned web developer, Shiny might not be a piece of cake. R syntax was not built for web development, and making the curly braces match is sometimes bloodcurdling.

Also, Shiny relies heavily on java script, internally. And even if they tried to hide it as much as possible, this often shines through (I like this pun ;-). So, for anything that goes a bit beyond the very basic examples, be prepared to learn a bit of java script.

Having said this, I promise that anyone who has mastered the 16 parameters of the ols function is intellectually fit to learn Shiny. The API is clean and consistent, the documentation is pretty good, and lots of examples will help you getting started quickly.

Alternatives to Shiny

Shiny is the right tool if:

1. you want to make a single app accessible to a wider audience
2. if you want to add interactivity to your R code (though there are other ways to do this, e.g. directly in the viewer in RStudio)

Shiny might **not** be the best choice if:

1. you are a seasoned web developer
2. if you want to build an entire web application with multiple connected screens, etc.

If either one is true, a **best-of-breed approach** might be better:

1. build the client, server, and database access in RoR (or php, or node.js / AngularJS, or meteor or whatever other web framework you fancy)
2. build the quantitative code in R, and access it in a stateless fashion through an OpenCPU or rApache API or Azure ML.

Finally, in certain situations you might fancy a **hybrid approach**:

1. Build the basic structure of your app in a dedicated web framework (e.g. RoR)
2. integrate specific Shiny apps in your web application, e.g. using iFrames or by directly reverse proxying through Apache

For more details on these options, refer to this jenunderwood blogpost.

Shiny Server Licenses

Academics and corporations have different budgets, and sometimes different needs. For this reason, RStudio, the company behind Shiny, decided to offer Shiny with different licenses:

1. an open source version
2. a commercial version, called Shiny Pro: This version is also targeted at self-hosters. It adds a few enterprise features (such as https, authentication, scalability) and support, starting at roughly 10'000 USD per year.
3. a hosted cloud solution, called shinyapps.io: With this offering, you do not need to install and manage your

own server, worry about backups, scaling out, etc. There are different subscriptions available, starting with the free subscription.

Each has its use cases. Here is my recommendation:

1. **Beginner:** if you are new to Shiny, you won't need any of the above. The bare Shiny R package, available from CRAN will get you started immediately.
2. **Casual User:** if you have very few **public** apps (less than 5), and want to share them with very few people (one at the time), who use them very rarely (less than 25 hours per month), then shinyapps.io FREE is a very convenient solution. Don't use this in a blog post (as I have done), or your application stops working after a day because you'll be over the 25 hours limit quickly.
3. **Company:** if you work in a company, and if you rely on Shiny for your business, you'll most certainly want to Shiny Server Professional. It lets you keep your apps and data private, in-house, while guaranteeing continuity through the RStudio support
4. **Researcher:** if you have many **public** apps and/or many users, then the Shiny Server Open Source is probably the right choice. It's not that hard to install it, and hosting on e.g. AWS or at your institution can be very cheap.

The paying shinyapps.io offerings probably also have their use case. But even if you don't have a lot of hosting experience, deploying Shiny Server Open Source is not very difficult. And even if it does not offer https and password protection out of the box, these can be achieved e.g. with an Apache proxy, as I've done before in a few projects.

The post Shiny appeared first on ipub.