Deploy R and Shiny

A practical comparison of

Shiny Server / ShinyProxy / RStudio Connect

Sebastian Mellor, Jumping Rivers

***satRday*, Newcastle upon Tyne, 2019**

Who am I?

- Developer, Operator, Data Scientist
  - Maths, Stats, Software, Infrastructure
- Jumping Rivers and R/Shiny
  - **R** at Uni, for research, now with JR
  - **R** is great for stats, surprisingly good at UI
- RStudio Partner

Google Trend Index

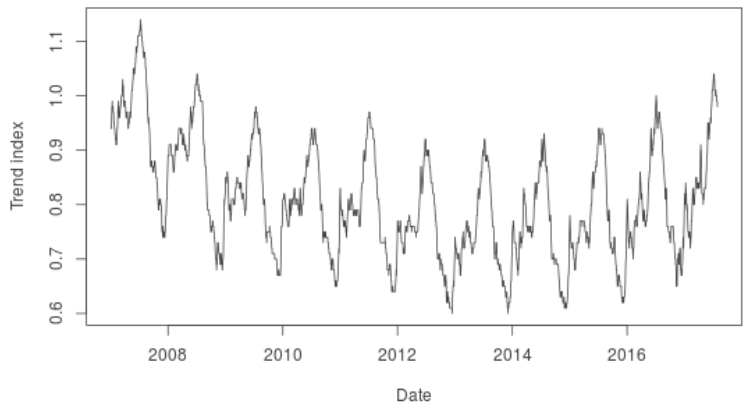
Trend index

Travel

Date range

2007-01-01 to 2017-07-31

Overlay smooth trend line



The Google Travel Index tracks queries related to airlines, hotels, beach, southwest, las vegas, flights, etc. The index is set to 1.0 on January 1, 2004 and is calculated only for US search traffic.

Source: Google Domestic Trends

Description | app.R

Shiny comes with a variety of built in input widgets. With minimal syntax it is possible to include widgets like the ones shown on the left in your apps:

```
# Select type of trend to plot
selectInput(inputId = "type", label = strong("Trend index"),
            choices = unique(trend_data$type),
            selected = "Travel")

# Select date range to be plotted
dateRangeInput("date", strong("Date range"),
               start = "2007-01-01", end = "2017-07-31",
               min = "2007-01-01", max = "2017-07-31")
```

Displaying outputs is equally hassle-free:

```
mainPanel(
  plotOutput(outputId = "lineplot", height = "300px"),
  textOutput(outputId = "desc"),
  tags$a(href = "https://www.google.com/finance/domestic_trends",
         "Source: Google Domestic Trends", target = "_blank")
)
```

Build your plots or tables as you normally would in R, and make them reactive with a call to the appropriate render function:

```
output$lineplot <- renderPlot({
  plot(x = selected_trends()$date, y = selected_trends()$close, type = "l",
       xlab = "Date", ylab = "Trend index")
})
```

Want to find out how we built the Google Trend Index app shown on the left? See the next tab for the complete source code.

https://shiny.rstudio.com/ ; https://shiny.rstudio.com/gallery/

Publish your application!

Features and requirements

Desired features

- Other people need to see this!
  - Hosting
- It should have access to all the data
  - Data storage
- It should just work
  - Integrated, branded, familiar

Required features

- Access control
  - Who can access, who can *manage* access?
- Data security
  - Where is data stored?
- Application updates
  - How is the application updated?

Additional features

- The client must understand the deployment technologies?
- The client should be able to maintain the hosting platform?
- Is there a reliable/testable deployment procedure?
- Will there be multiple versions of this application, or maybe even other applications?

Set up the server!

Potential solutions

App deployment platforms

- Managed service
  - shinyapps.io (Free...)
  - shinyapps.io (...Professional, AWS US)
- Self-hosted
  - Shiny Server (Open Source)
  - Shiny Server (Pro)
  - ShinyProxy
  - RStudio Connect

Shiny Server Open Source

Benefits

- Free*
- Flexible

Limitations

- Restricted user authentication methods
- Single-threaded with limited usage controls
- Requires **server access and knowledge** for management and application deployment

Shiny Server Open Source

Who?

- Some server/linux knowledge
- Can give access to (S)FTP for publishers
- Don't need to scale too much
- Have alternative authentication

Shiny Server Pro

Benefits

- User authentication
- Multiple processes per app

Limitations

- Requires server access for app deployment
- Not "free"

ShinyProxy

Benefits

- Free?
- Authentication with almost anything
- Scalable with Docker

Limitations

- Apps require manual config files
- Scalable with Docker?

ShinyProxy

Who?

- Time to dedicate to new technologies
- Can implement a build tool-chain
- Publisher is also administrator

RStudio Connect

Benefits

- Really simple application deployment
- Web-based user and application management
- Extra features galore
- Easy to setup

Limitations

- Price?
- Customisation?

RStudio Connect

Who?

- Wants an easy solution
- Wants plenty of features
- Wants official support
- Can purchase software licences

Cost vs. complexity

Conclusions

Summary

- Shiny Server Open Source
  - Free, **limited features** (auth, scaling)
- ShinyProxy
  - Free, flexible, **complex** (manual configs)
- RStudio Connect
  - Easy, powerful, **paid** (official support)
- **Jumping Rivers**
  - Training, development, support