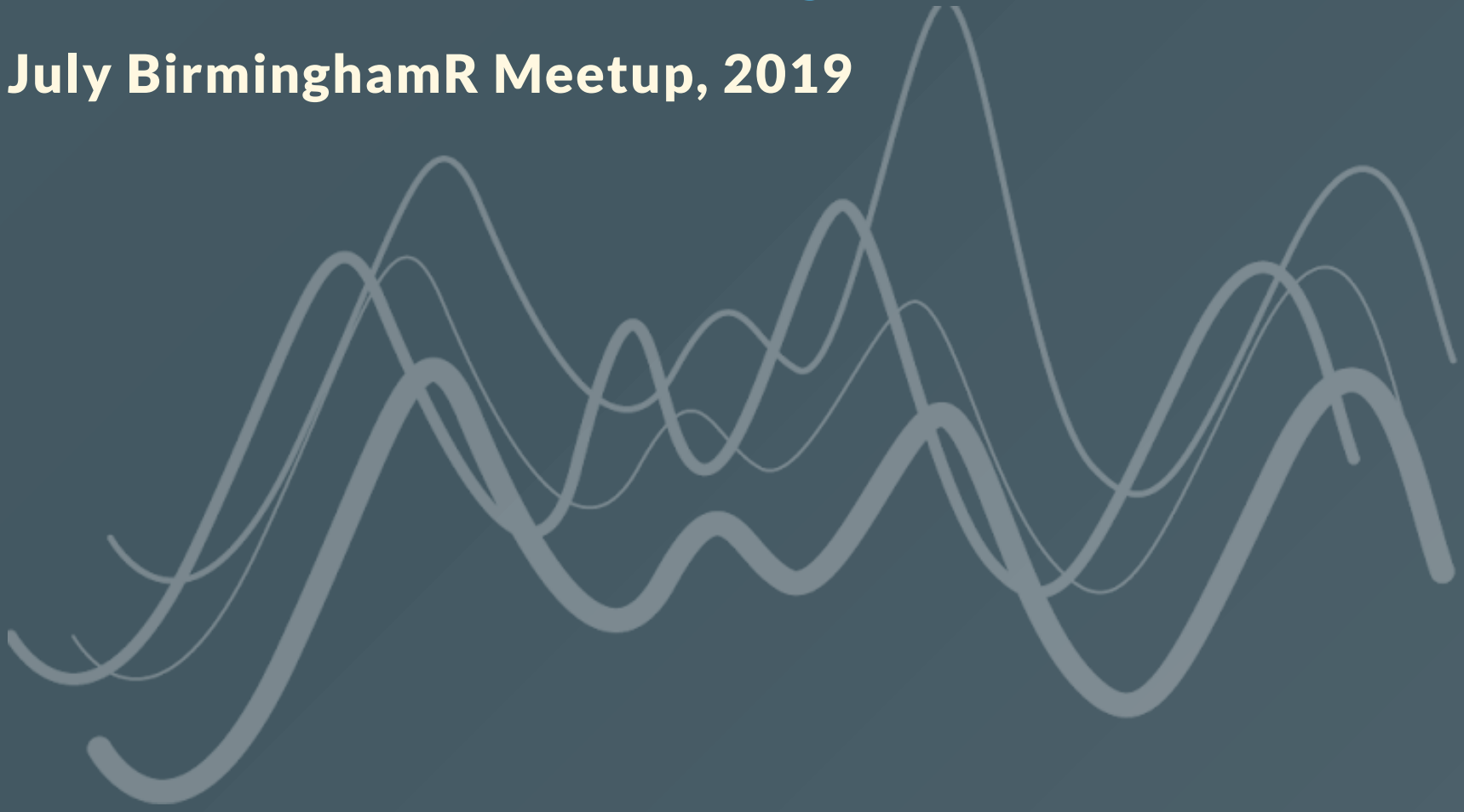
Model Deployment in R

Sebastian Mellor, Jumping Rivers

July BirminghamR Meetup, 2019



Who are we?

- DevOps at **Jumping Rivers**
- Data Science **consultancy** and **training**
  - Focused on **R**, python, ...
  - Machine Learning & Analytics
- RStudio *Full Service Certified Partners*
- Microsoft *Preferred Data & AI Training Partner*

Who do we work for?

> *I've developed an amazing model in R and I want to share it.*
>
> — An R user

What am I trying to achieve?

- My model is code, I just want to share it
  - *Publish an R package using drat?*
- My model requires private data
  - *Deploy an API with plumber?*
- My model should be easy to use
  - *Make a Shiny app?*

What is plumber?

An R package that converts your existing R code to a web API using a handful of special one-line comments.

```
#* Return the sum of two numbers
#* @param a The first number to add
#* @param b The second number to add
#* @post /sum
function(a, b){
  as.numeric(a) + as.numeric(b)
}

#* Plot a histogram
#* @png
#* @get /plot
function(){
  rand <- rnorm(100)
  hist(rand)
}
```

Using Plumber

Server:

```
library(plumber)
r <- plumb("plumber.R")
r$run(port=8000)
```

Input:

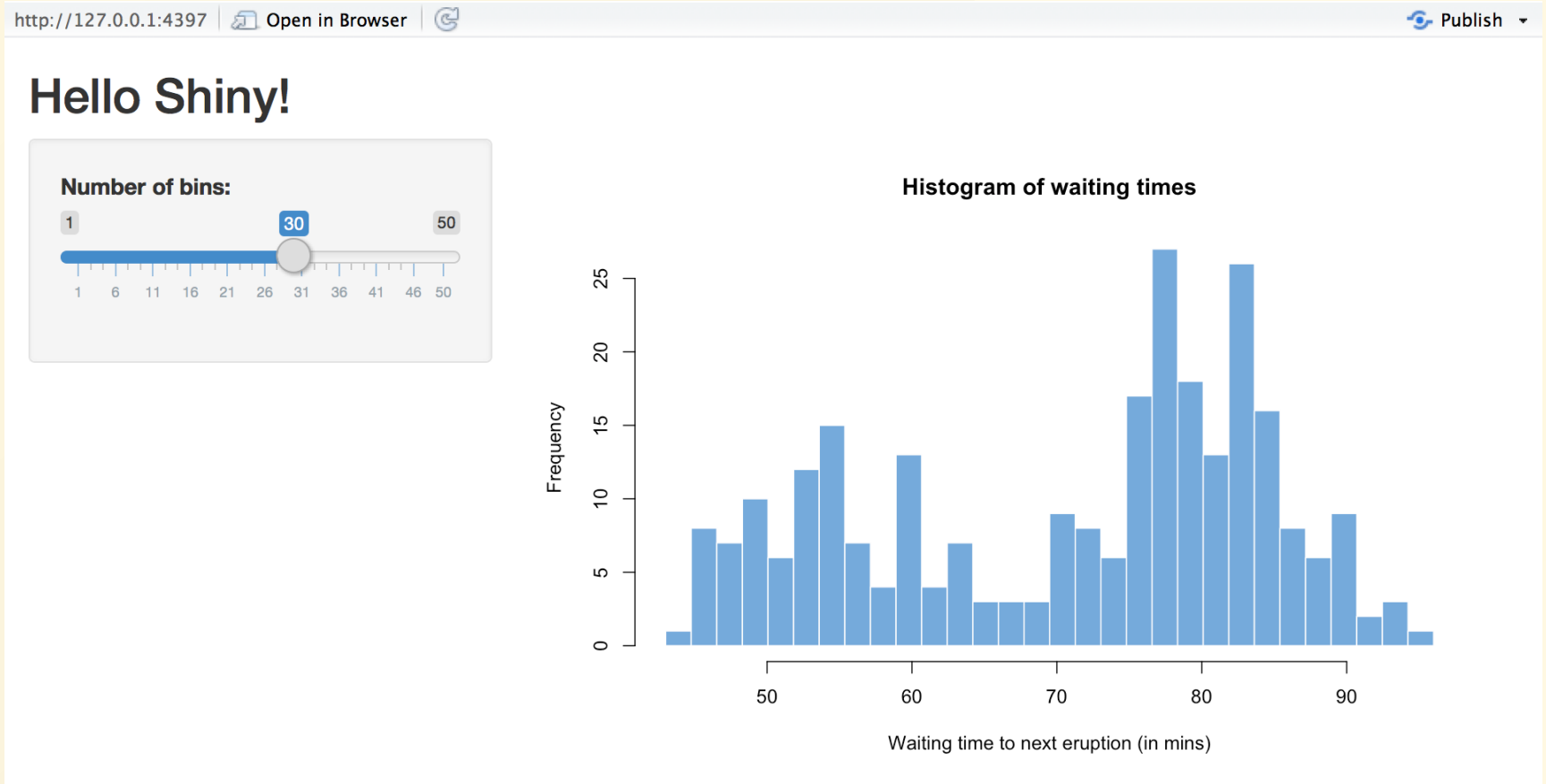
```
$ curl -X POST "http://127.0.0.1:8000/sum?a=1&b=2"
```

Output:

```
[3]
```

What is Shiny?

An R package that makes it easy to build interactive web apps straight from R.



Using Shiny

```
ui <- fluidPage(
  titlePanel("Hello Shiny!"),
  sidebarLayout(
    sidebarPanel(
      sliderInput(inputId = "bins", label = "Number of bins:",
                  min = 1, max = 50, value = 30)
    ),
    mainPanel(
      plotOutput(outputId = "distPlot")
    )
  )
)
server <- function(input, output) {
  output$distPlot <- renderPlot({
    x    <- faithful$waiting
    bins <- seq(min(x), max(x), length.out = input$bins + 1)
    hist(x, breaks = bins, col = "#75AADB", border = "white",
         xlab = "Waiting time to next eruption (in mins)",
         main = "Histogram of waiting times")
    })
}
shinyApp(ui, server)
```

> *That's great, but how do I deploy it?*
>
> — The same R user

Considerations

- Where should it be accessible from?
  - Internal VPN, GDPR/EU, World Wide Web
- Does it require access to other data?
  - Temporary storage, remote database, third party API, user data
- What should it look like?
  - Integration, branding, familiar UI

More considerations...

- Access Control
  - Who, how, user management
- Extra security
  - Encryption, audit logs, updates
- Consistency
  - Version control, testing, CI/CD

Even more...

- In-house knowledge
- Maintainance and support
- On-going development
- **Reproducible research**

Let's get started!

Examples:

1. Self-host with Docker

2. Self-host with RStudio Connect

What is Docker?

Enterprise Container Platform for High-Velocity Innovation

```
FROM rocker/r-base

RUN apt-get update -qq && apt-get install -y git-core libssl-dev libcu
RUN install2.r plumber

COPY . /app

EXPOSE 8000
ENTRYPOINT ["R", "-e",
 "pr <- plumber::plumb(commandArgs()[4]); pr$run(host='0.0.0.0', port=
CMD ["/app/plumber.R"]
```

https://hub.docker.com/r/trestletech/plumber/dockerfile

How to use this

- Get a server
- Install Docker
- `vi Dockerfile`
- `docker build -t my_api .`
- `docker run -p 8000:8000 my_api /app/plumber.R`

Bonus points

- Build this image automatically with *Continuous Integration*
- Publish the image to DockerHub
- Manage the Docker service with Docker Compose
- Attach external storage with `-v`

~~Limitations~~ Challenges

- No SSL
- No authentication
- No user management at all
- No process scaling

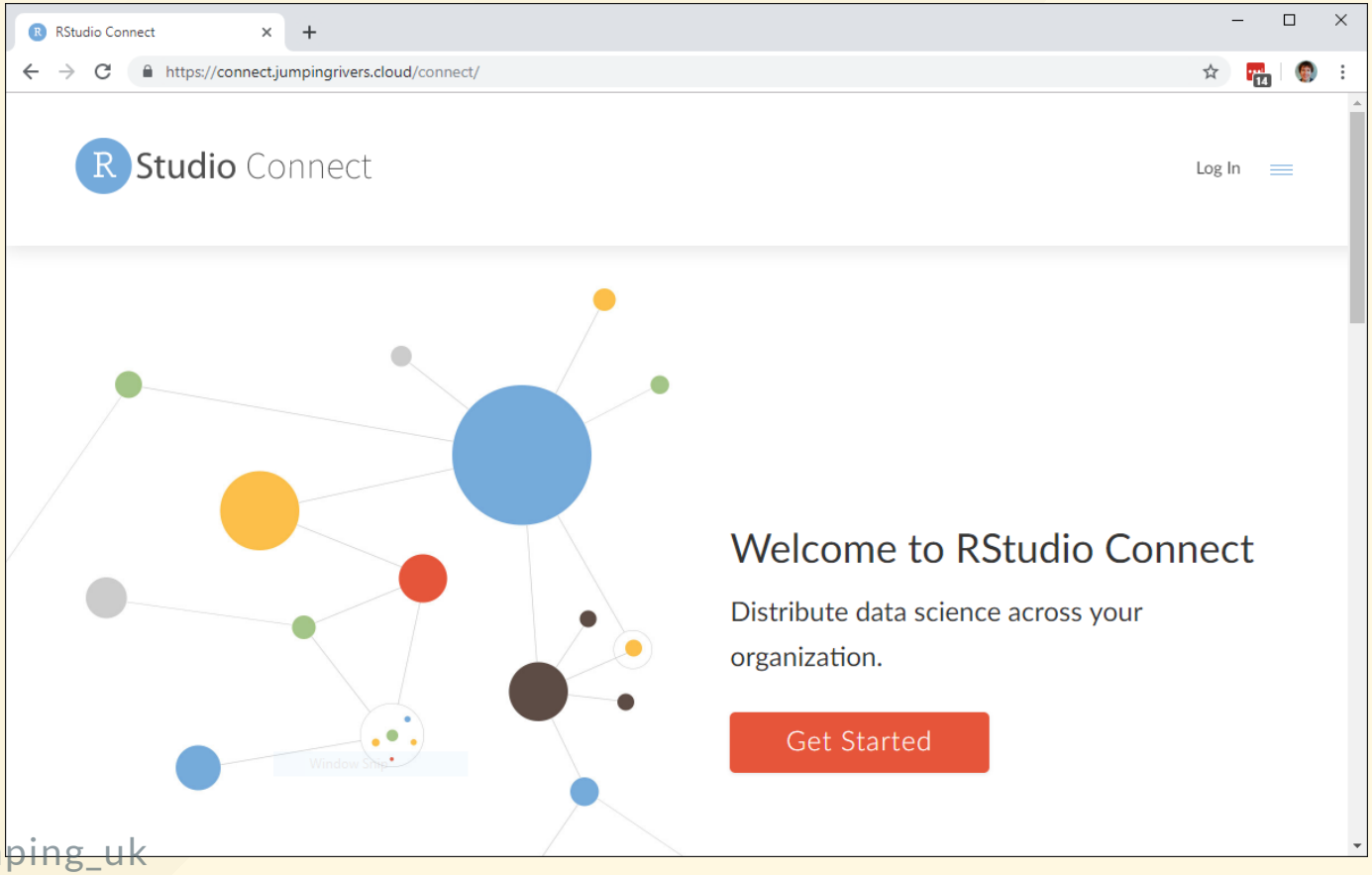
Recommendations

- Web proxy like Nginx or Traefik
- Integrate a *Single Sign-On* service

> *Who can setup the proxy and user authentication gateways?*
>
> — A client

What is RStudio Connect?

RStudio Connect is a publishing platform for the work your teams create in R and Python.



How to use this

Click the button.

Demos

Honerable Mentions

- Plumber & AnalogSea: Deploy API from R with code
- ShinyApps.io: one-click, remotely hosted (apps only)
- Shiny Server: self-hosted, app server

Summary

- DIY Docker
  - **Free**, **powerful**, **simple**, expert
- Using Shiny Server Open Source / Plumber server
  - **Free**, limited features (auth, scaling)
- RStudio Connect
  - **Easy**, **powerful**, paid (official support)
- Jumping Rivers
  - *Training*, *development*, *support*