



# R in Production

*It will work just fine...*

Colin Fay - ThinkR



# \$ whoami

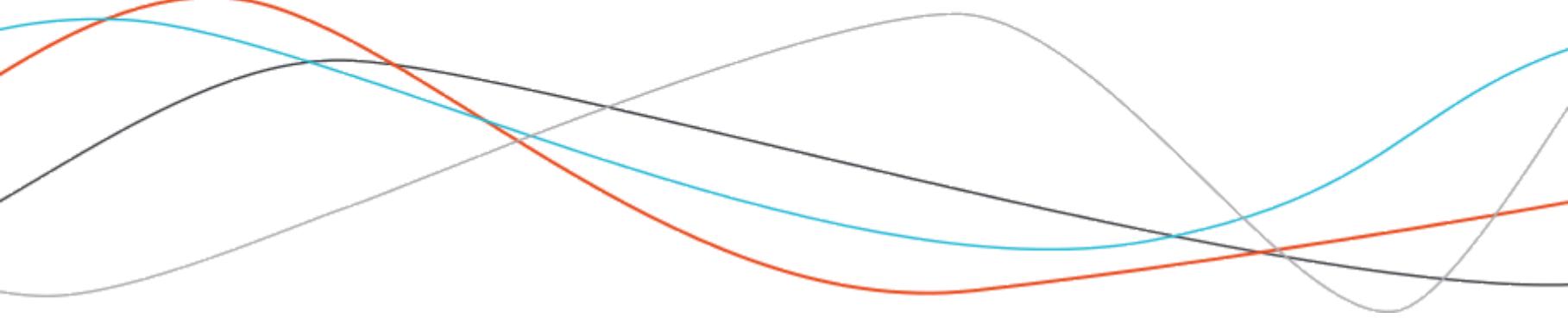
Colin FAY

Data Scientist & R-Hacker at ThinkR, a french company focused on Data Science & R.  
Hyperactive open source developer.

- <http://thinkr.fr>
- <http://rtask.thinkr.fr>
- [http://twitter.com/\\_colinfay](http://twitter.com/_colinfay)
- <http://github.com/colinfay>



# ThinkR





# ThinkR

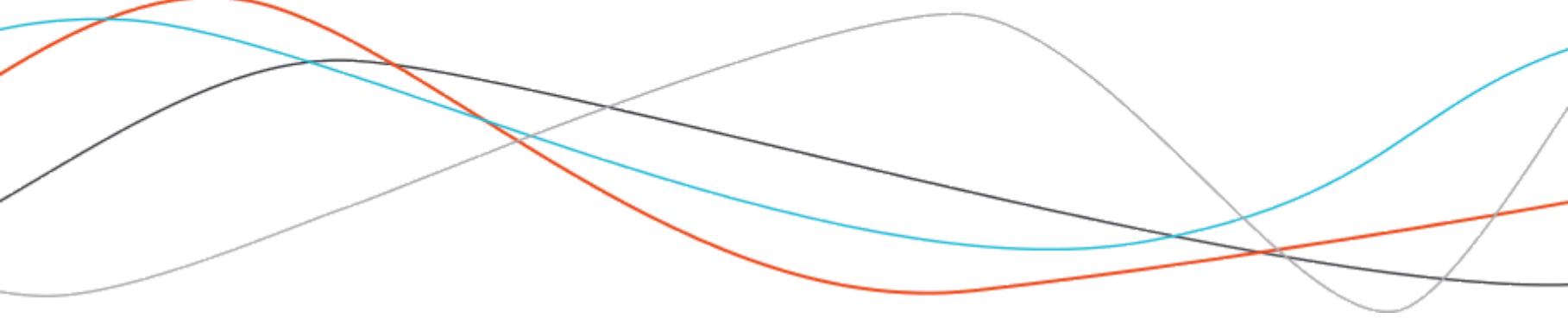
Data Science engineering, focused on R.

- Training
- Software Engineering
- R in production
- Consulting





# #RinProd





# R in Production

Them: "R is not meant for production"



# R in Production

Them: "R is not meant for production"

Me:





# R in Production

- Facebook
- Google
- Twitter
- Microsoft
- Uber
- Airbnb
- IBM
- Ford
- Capgemini
- Deloitte Consulting
- Gartner
- KPMG

## In France ?

EDF, BNP Paribas, SNCF, Sanofi, RTE, Servier, Orange, Axa, INSEE, Ipsos, Banque de France, CNRS...

<https://github.com/ThinkR-open/companies-using-r>



But on the other hand...

Them: "I'll just push this script in prod, it will work just fine."



# But on the other hand...

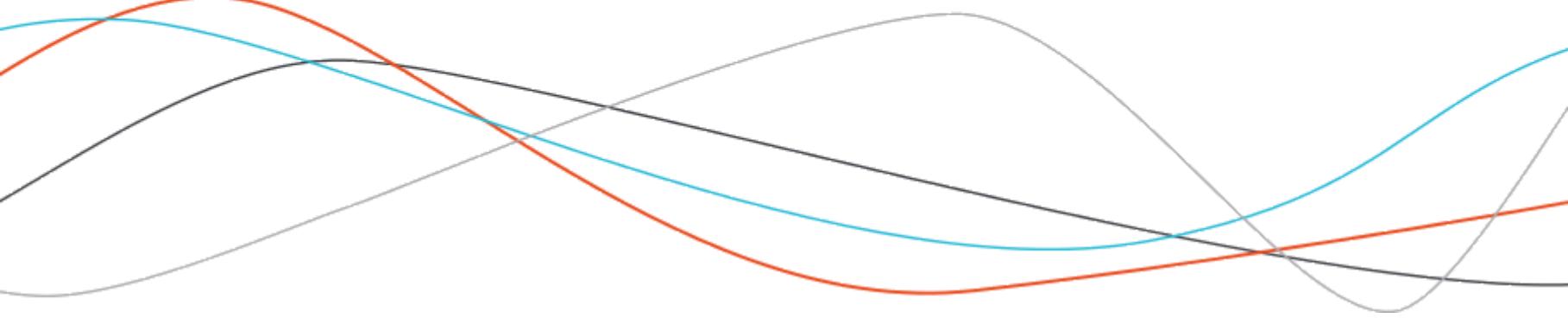
Them: "I'll just push this script in prod, it will work just fine."

Me:





# A little story





# A long time ago, in the kingdom of R in Production

Me: "Ok, let's update the app and push it into prod, should take 10 minutes"



# A long time ago, in the kingdom of R in Production

Me: "Ok, let's update the app and push it into prod, should take 10 minutes"

The prod environment:





```
root@westeros-vm:/var/log/shiny-server# cat thewall(...).log
*** caught segfault ***
[...]
address 0x5100004d, cause 'memory not mapped'
```

Traceback:

```
1: rcpp_sf_to_geojson(sf, digits, factors_as_string)
2: sf_geojson.sf(data)
3: geojsonsf::sf_geojson(data)
4: addGlifyPolygons(., data = pol_V1, color = les_couleurs, popup =
"val", opacity = 1)
5: function_list[[i]](value)
6: freduce(value, `function_list`)
7: `_fseq`(`_lhs`)
8: eval(quote(`_fseq`(`_lhs`)), env, env)
[...]
105: captureStackTraces({    while (!.globals$stopped) {
..stacktracefloor..(serviceApp())      Sys.sleep(0.001)    }})
106: ..stacktraceoff..(captureStackTraces({    while (!.globals$stopped) {
..stacktracefloor..(serviceApp())      Sys.sleep(0.001)    })))
107: runApp(Sys.getenv("SHINY_APP")), port = port, launch.browser = FALSE)
An irrecoverable exception occurred. R is aborting now ...
```



What I wanted to do:





# What I actually did:

On my machine

```
packageVersion("geojsonsf")
[1] '1.2.1'
```

On the server

```
packageVersion("geojsonsf")
[1] '1.3.0'
```

```
remove.packages("geojsonsf")
remotes::install_version("geojsonsf", "1.2.1")
```



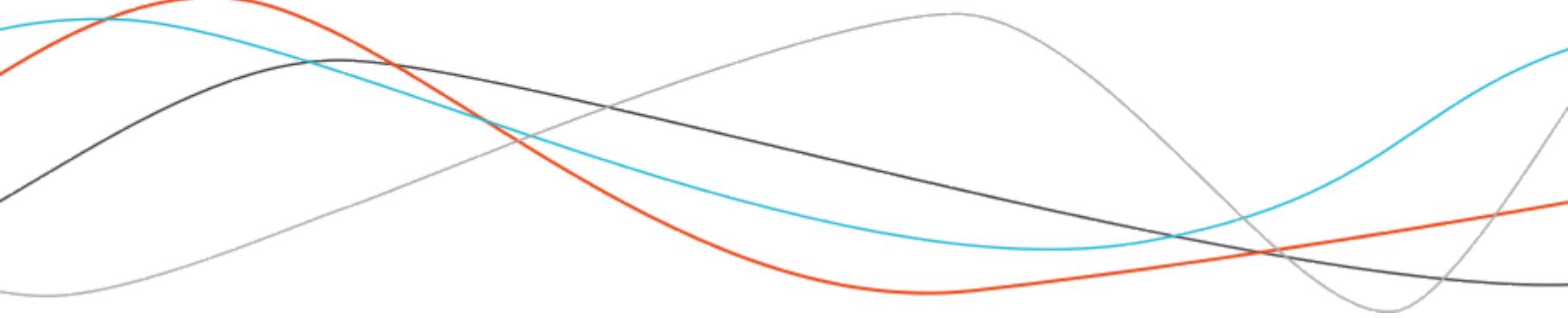
# What has happened?

## v1.3

- restructured C++ src code for easier navigation and linking
- added dependency on jsonify v0.2 and fixed tests
- Factors now default to strings



# R in Production





# In production?

- 👉 Great definition of what "in production" means:
- 🗣 "Software environments that are used and relied on by real users with real consequences if things go wrong"
- Colin Fay 🤘 (@\_ColinFay) **January 17, 2019**

=> Joe Cheng, #RStudioConf2019

Also :

*"Production is anything that is run repeatedly and that the business relies on"*

=> Mark Sellorm, #RStudioConf2019



# In production?

- Not a Proof Of Concept
- Not a prototype
- Not a testing env
- Not a sandbox
- Not "working on my machine" only



*"used and relied on"*

- Make it work
- Make it usable
- Make it safe
- Make it last
- Make it scale





*"by real users"*



## Three types of users

- IT (doesn't know anything about R)
- R developers (don't know anything about IT)
- R-products users (don't know anything about R or IT)



*"if things go wrong"*

## What could go wrong?

- The white walkers break the wall and start marching south
- R and/or the R-products are not accessible
- An update to an application breaks the application
- An update to an application breaks another application
- Deploying a product on another server leads to different results
- The product gets veeeeeeeery slow
- ...

*"with real consequences"*

=> People rely on the product to do their job correctly.



## What 'in production' implies

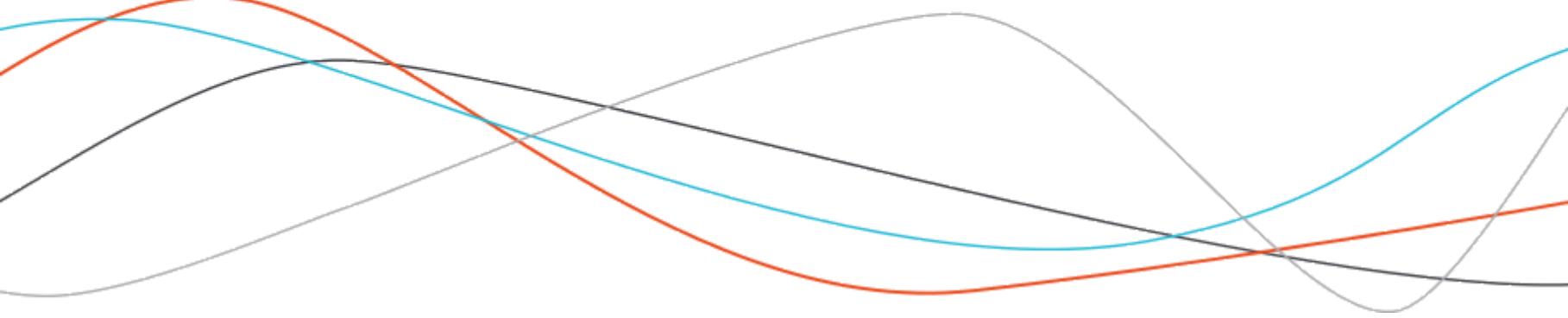
- Moving away from the comfort of your 🖥, onto a server
- Dealing with system-requirements, libraries, and 📦 versions...
- Write a reliable, fast product that can scale and which you can maintain

## What 'in production' might implies

- Talking to other languages
- Be integrated in another software environment



# What can we do?





What can we do?





## The first barrier is cultural

Every single single technical barrier to running #rstats in "production" is easy to overcome. It's the cultural barriers that slow us down. #RinProd

— Mark Sellors (@sellorm) September 17, 2018



## Cultural?

✓ The good thing about R is that anybody can start using it and get results in a couple of hours.

⚠ The bad thing about R is that anybody can start using it and get results in a couple of hours.

**It's easy to do 'quick and dirty' things in R. Production ready R products demand extra work.**

-> We need to advocate for more and more Software Engineering culture in the R world.



## Cultural?

-> Lot of people learn R as a Data Science tool, not as a programming language. R products written by users who might not be Software engineers :

- Don't know SE best practices.
- Don't realize what is needed for pushing something in production (budget and tech).
- Efficiency and scaling is not a central concern.

-> The IT might not be receptive (Conservatism, don't know R, at ease with other languages, "R is not a real language" ...)



The tools are there (so no excuses)





The tools are there (so no excuses)

# BRACE YOURSELF



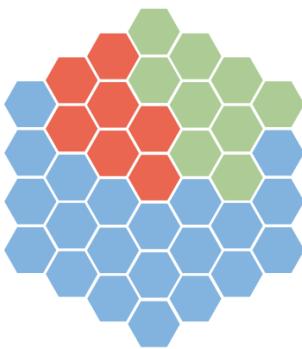


# The tools are there (so no excuses)

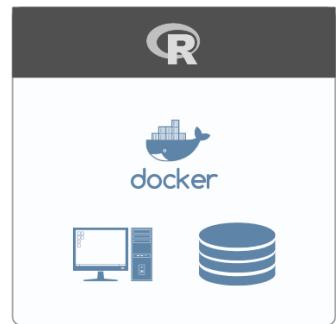
R Packages



Administrators

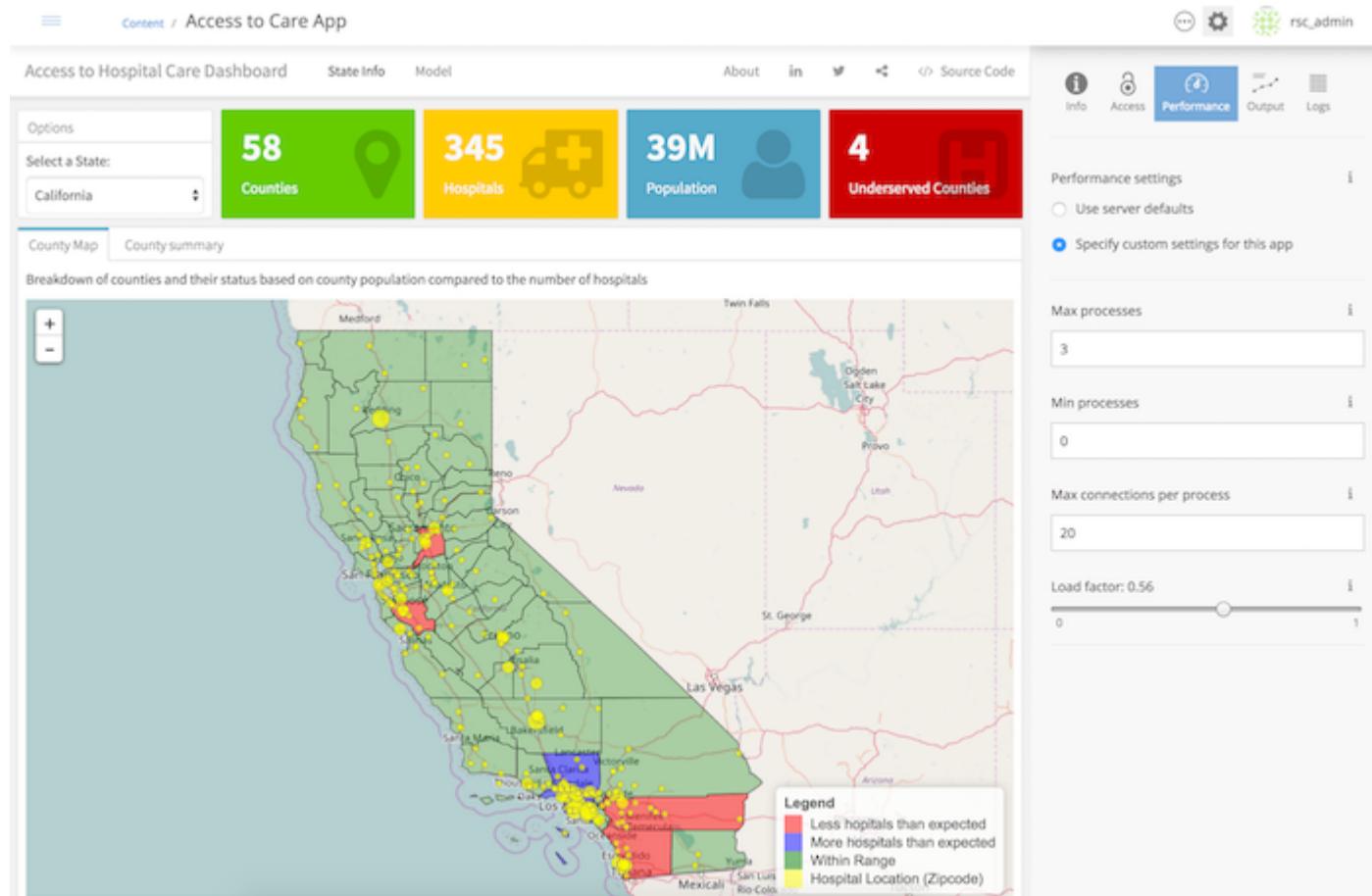


Users





# The tools are there (so no excuses)





# The tools are there (so no excuses)

## Put Shiny Web Apps Online

RStudio lets you put shiny web applications and interactive documents online in the way that works best for you.

For Shiny applications, consider Shiny Server or Shiny Server Pro, which adds enterprise grade scaling, security, and admin features to the basic open source edition.

If you prefer for us to host your Shiny applications, one of our shinyapps.io plans is sure to work for you.

When you're ready, RStudio Connect is a new publishing platform for all the work your teams create in R. Share Shiny applications, R Markdown reports, dashboards, plots, APIs, and more in one convenient place. Use push-button publishing from the RStudio IDE, scheduled execution of reports, and flexible security policies to bring the power of data science to your entire enterprise.



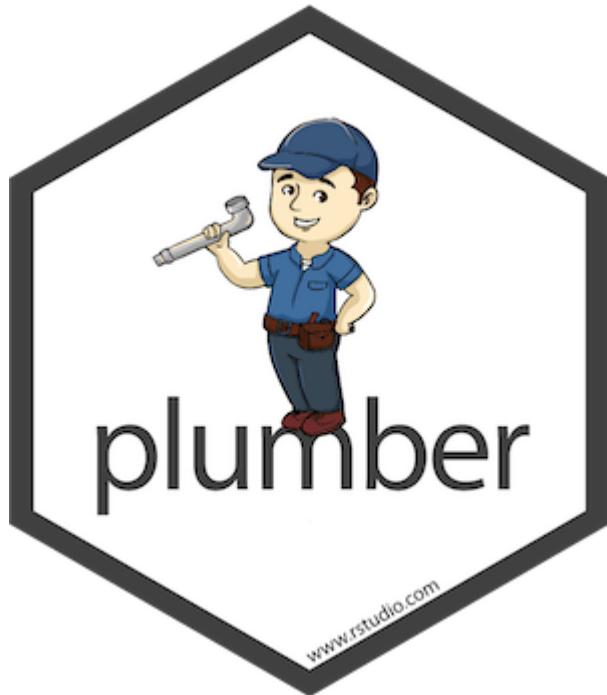
 [DOWNLOAD SHINY SERVER OPEN SOURCE](#)

 [DOWNLOAD SHINY SERVER PRO](#)

 [DOWNLOAD RSTUDIO CONNECT](#)

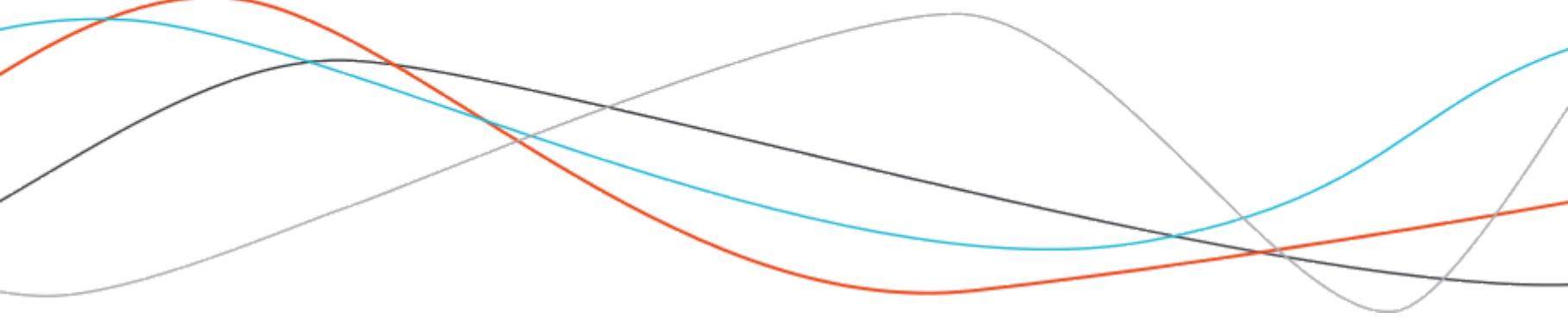


The tools are there (so no excuses)





# What can YOU do?





# Make your R products production ready

Rule n°1: don't send an RScript to your IT team and ask them to deploy it into production.

## Everything is a package

- 📦 are documented
- 📦 have tests
- 📦 list dependencies
- 📦 work everywhere



# Make your R products production ready

Rule n°2: Assume that if it works on your machine, it won't work in production.

¬\\_(ツ)\_/¬  
**IT WORKS**  
on my machine



# Make your R products production ready

Rule n°3: Be gentle with your IT team, and present your R-product as a "real" software, not just a POC.

## Things to think about

- System requirements
- CI, CD and version control
- Long term maintenance
- Security & integrity
- User-support
- ...



# Make your R products production ready

Rule n°4: Learn about IT, "hardcore" software engineering skills, DevOps...

- Docker
- bash & Linux
- Git
- Gitlab CD and CI
- Jenkins
- Travis
- ...



# YARPC (Yet Another R in Production Checklist)

An incomplete list of things to check before sending my app into prod

- [ ] Server configuration (e.g: "my app needs internet, does the server have access to the internet?" or "Can I install this system requirement for package X?")
- [ ] Does the server has the good R & package versions? If not, is this an issue?
- [ ] If we need to install or update package(s), will it break other things?
- [ ] There are tests for the product so there are no regression when we need to update it.
- [ ] We use version control.
- [ ] We use automated tests, continuous integration, and sandboxing so nothing is put into prod before having been thoroughly tested.
- [ ] There will be  $n$  users using the app, so we planned to scale for  $n + 1$  users.
- ...



Don't wanna do all of that? Call me





## Some resources

- Field Guide to the R Ecosystem: <https://fg2re.sellorm.com/>
- Supplement to Shiny in Production: <https://kellobri.github.io/shiny-prod-book/>
- An Introduction to Docker for R Users: <https://colinfay.me/docker-r-reproducibility/>
- [WIP] Building Big Shiny Apps - A Workflow <https://thinkr-open.github.io/building-shiny-apps-workflow/>



# Ready to send R to production?





# Thx! Questions?

---

Colin Fay

[colin@thinkr.fr](mailto:colin@thinkr.fr)

<https://thinkr.fr/>

[http://twitter.com/\\_colinfay](http://twitter.com/_colinfay)

<https://rtask.thinkr.fr/>

[http://twitter.com/thinkr\\_fr](http://twitter.com/thinkr_fr)

<https://colinfay.me/>

<https://github.com/ColinFay>