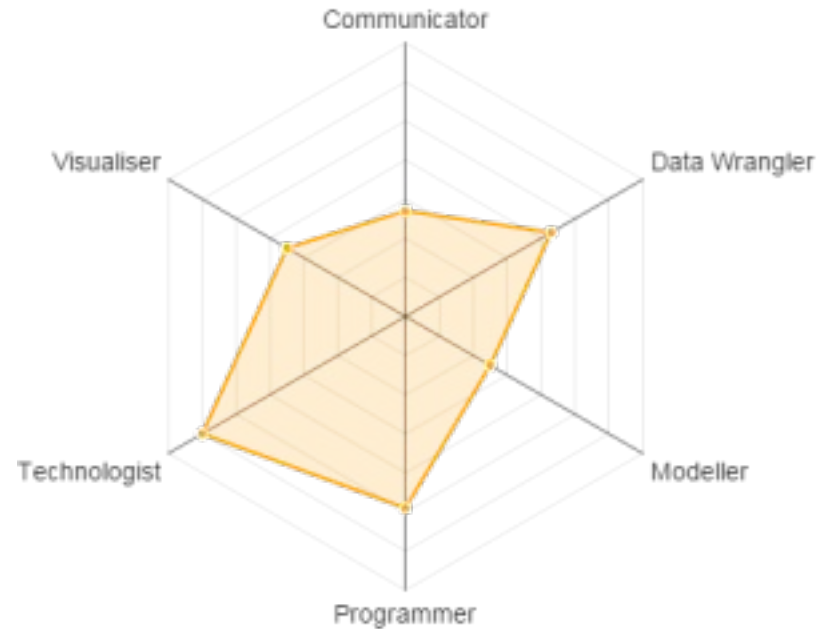




Me

- Technical Architect
- Data Engineering
- DataOps
- Analytic Maturity
- Automation
- Industrialisation



Mango Solutions

- R Specialists
- Data Science Consulting
- Data Science Training
- Software Development

Web: www.mango-solutions.com

Twitter: [@MangoTheCat](https://twitter.com/MangoTheCat)



Andy Nicholls
Richard Pugh
Aimee Gott

Sams **Teach Yourself**

R

in **24**
Hours

SAMS



EARL

EFFECTIVE APPLICATIONS OF THE R LANGUAGE

EARL is a Conference for users and developers of the open source R programming language. The primary focus of the Conference is the commercial usage of R across a range of industry sectors with the aim of sharing knowledge and applications of the language.

EARL 2016 London

13th-15th September 2016



Following the overwhelming success of the EARL 2015 Conference and the feedback received, the 2016 Conference will again be held in the Tower Hotel in London and the dates are: 13th – 15th September 2016. [Click for more information.](#)

EARL 2016 Boston

Date TBC



We were so amazed with the dedication and commitment of all our contributors and attendees at last year's conference that we've decided to forge ahead and run the conference again this year. Date and venue to be confirmed, why not sign up above to receive the latest updates as they become available.

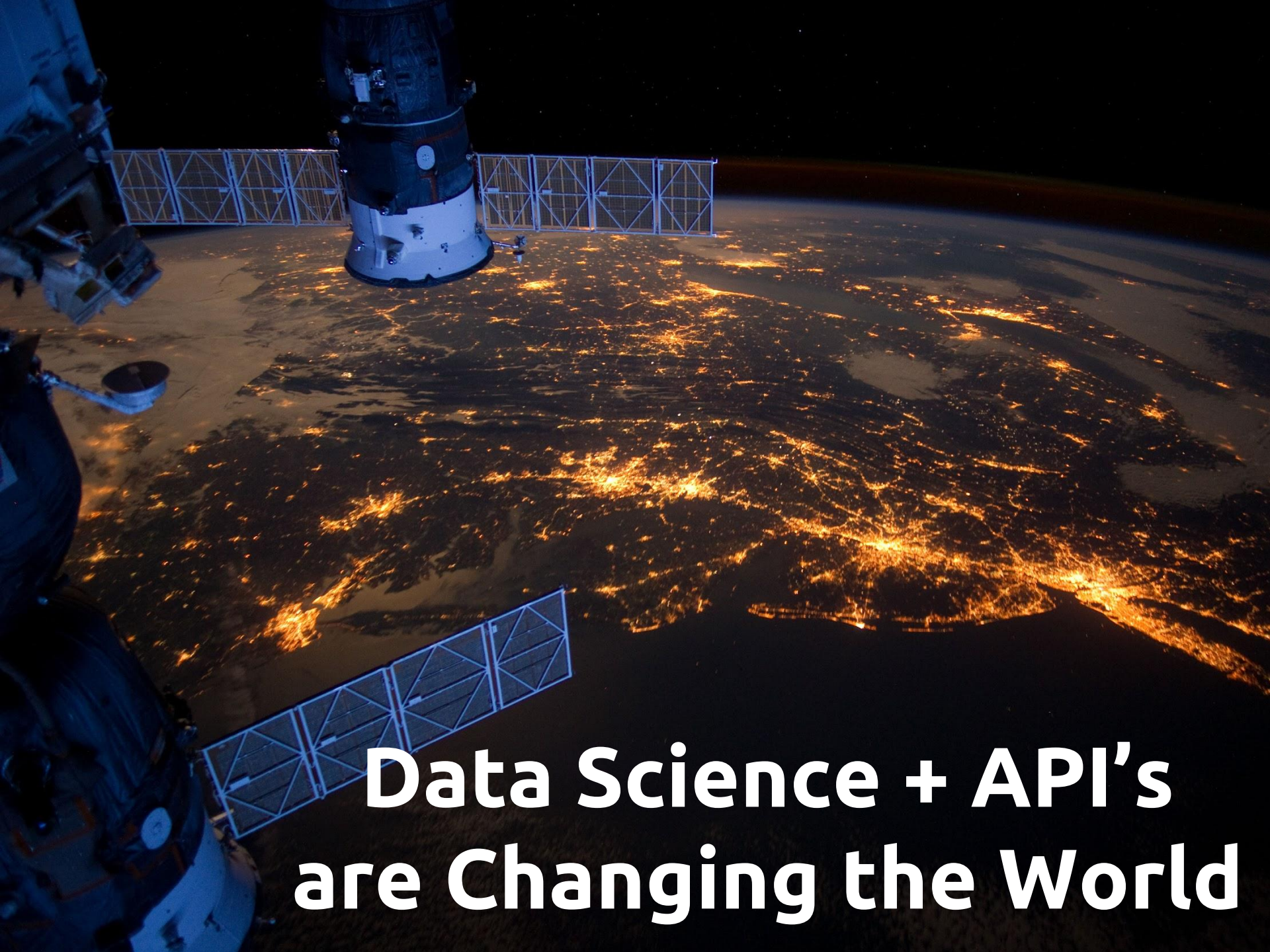
Web: earlconf.com

Twitter: [@earlconf](https://twitter.com/earlconf)



Agenda

- API's
- Introducing plumber
- How it works
- A simple case study
- Caveats
- Summary

A satellite view of Earth at night from space. The image shows the curvature of the planet with a dense network of orange and yellow city lights. The International Space Station is visible in the upper left, with its complex structure and solar panel arrays extending across the frame. The text "Data Science + API's are Changing the World" is overlaid in white at the bottom right.

**Data Science + API's
are Changing the World**

Introducing plumber



<http://plumber.trestletech.com>

Example - simple function

```
> adder <- function(a, b){  
+   a + b  
+ }  
> adder(4, 3)  
[1] 7
```

Example - add plumber

```
#* @get /adder  
> adder <- function(a, b){  
+   as.numeric(a) + as.numeric(b)  
+ }
```


Example - running with plumber

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

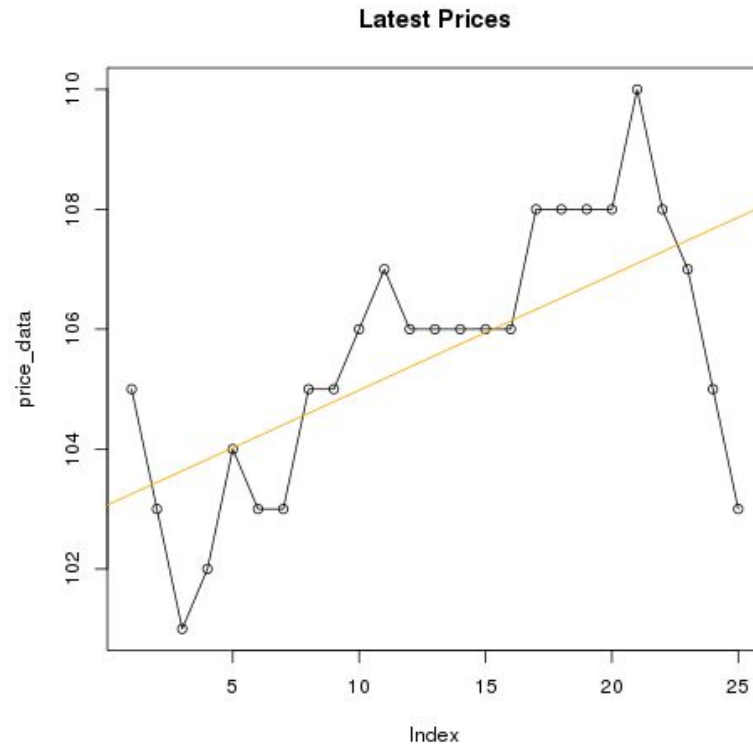
Example - Accessing the API

```
# curl http://localhost:8000/adder?a=4&b=3
```

Demo 1 - Simple API

Example - serving graphs

Tick API demo



[reset](#)

Demo 2 - Tick API

Why? What's the point?

Example - Shiny App

Simple Currency Converter

Input Amount	Input Currency
<input type="text" value="1"/>	<input type="text" value="USD"/>
Output Currency	Output Amount
<input type="text" value="USD"/>	1.00

Demo 3 - Currency Converter

Moving beyond the shiny app

- split the application up
- turn the function into a general purpose service
- refactor the shiny app to use the API

Demo 4 - Currency API



Raffle Time!

Caveats

When not to use plumber

- Shiny is a better for many users/uses
- It's predominantly for API's, not a full web stack solution

Extra things to consider...

- Security
- input sanitisation
- logging
- monitoring
- load balancing and state
- blocking

The Competition

- RServe
- OpenCPU
- DeployR
- Domino
- Others?

Conclusion

Creating an API provides a simple mechanism for exposing your functions to other users.

Plumber provides a really simple way to quickly create API's using R