Crosstalk

Shiny-like without Shiny

EARL CONFERENCE

Matt Dray | Dept for Education | #EARL2018

TL;DR*

Need to create a small interactive app without a server or Shiny?

Try the `crosstalk` package.

*Too long; didn't read

Am I talking about a real policy?

Nope, sorry.

Schools in this example were chosen randomly.

The ask

Visualise school locations to help make a policy decision

The ask

Visualise school locations to help make a policy decision

Shareable

The ask

Visualise school locations to help make a policy decision

Shareable

Cheap

The ask

Visualise school locations to help make a policy decision

Shareable

Cheap

Quick

The ask

Visualise school locations to help make a policy decision

Shareable

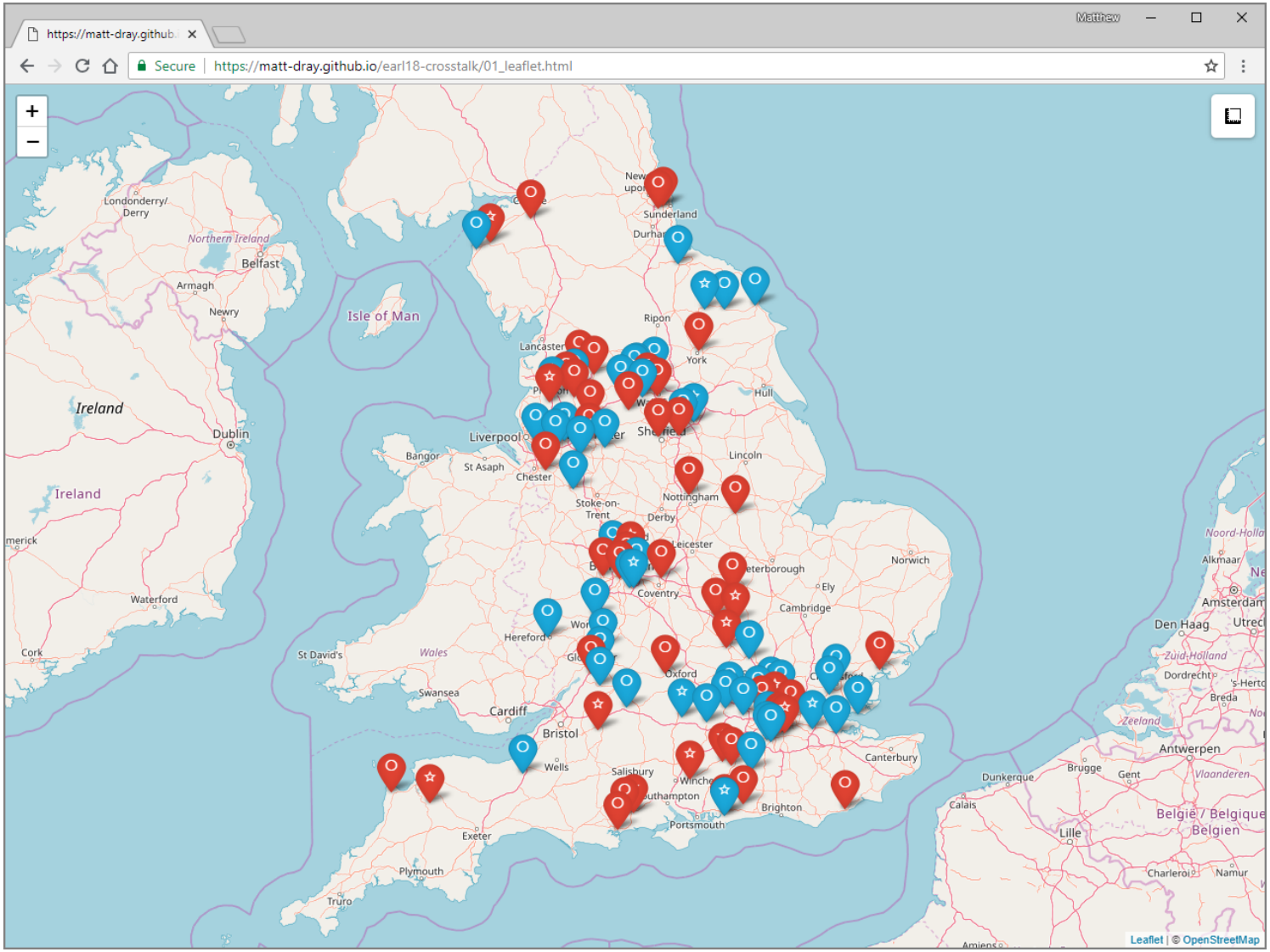
Cheap

Quick

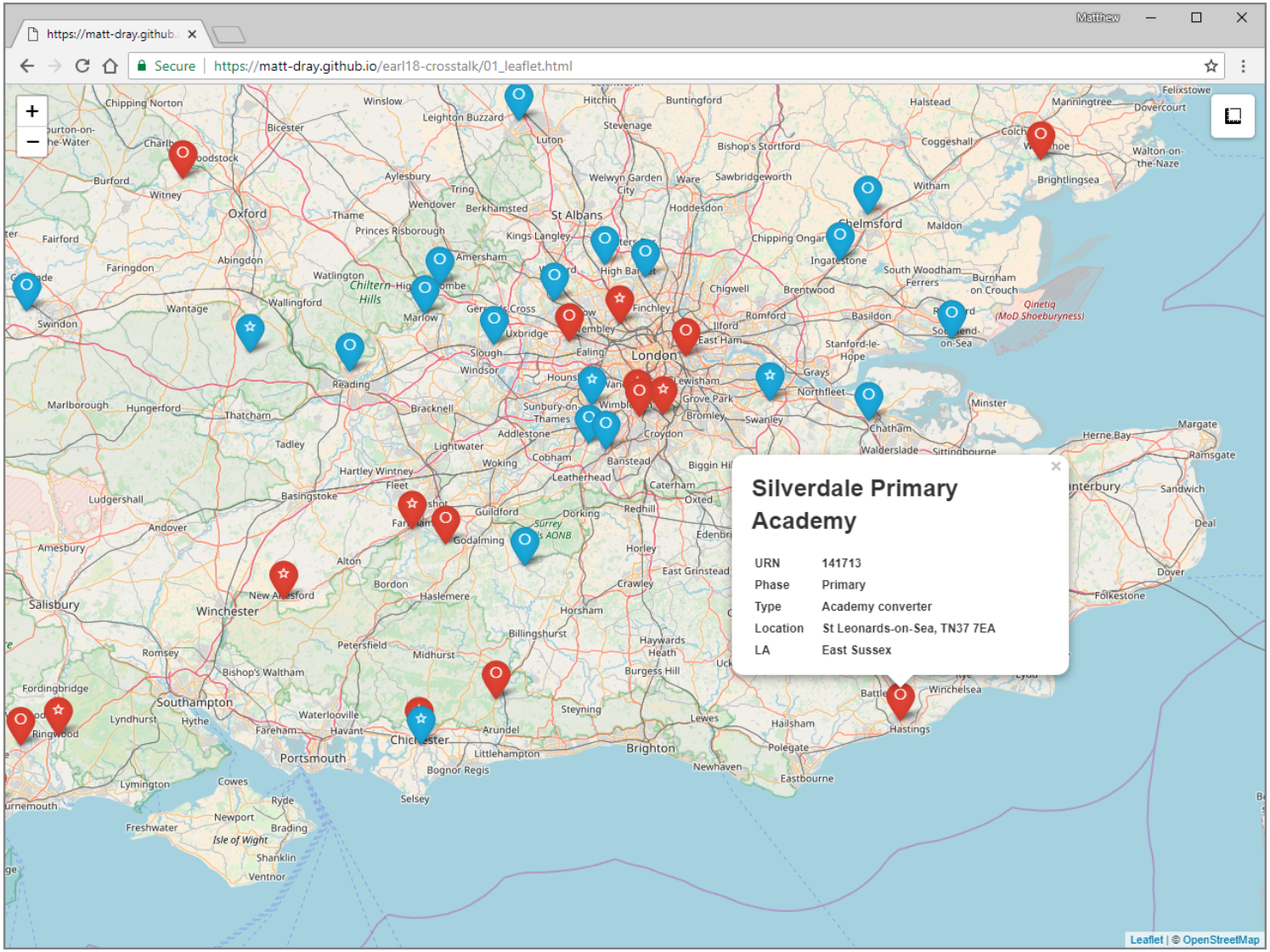
Reusable

Solution 1

`leaflet` interactive map



Demo available here



Demo available here

Solution 1

interactive

intuitive

Solution 1

interactive

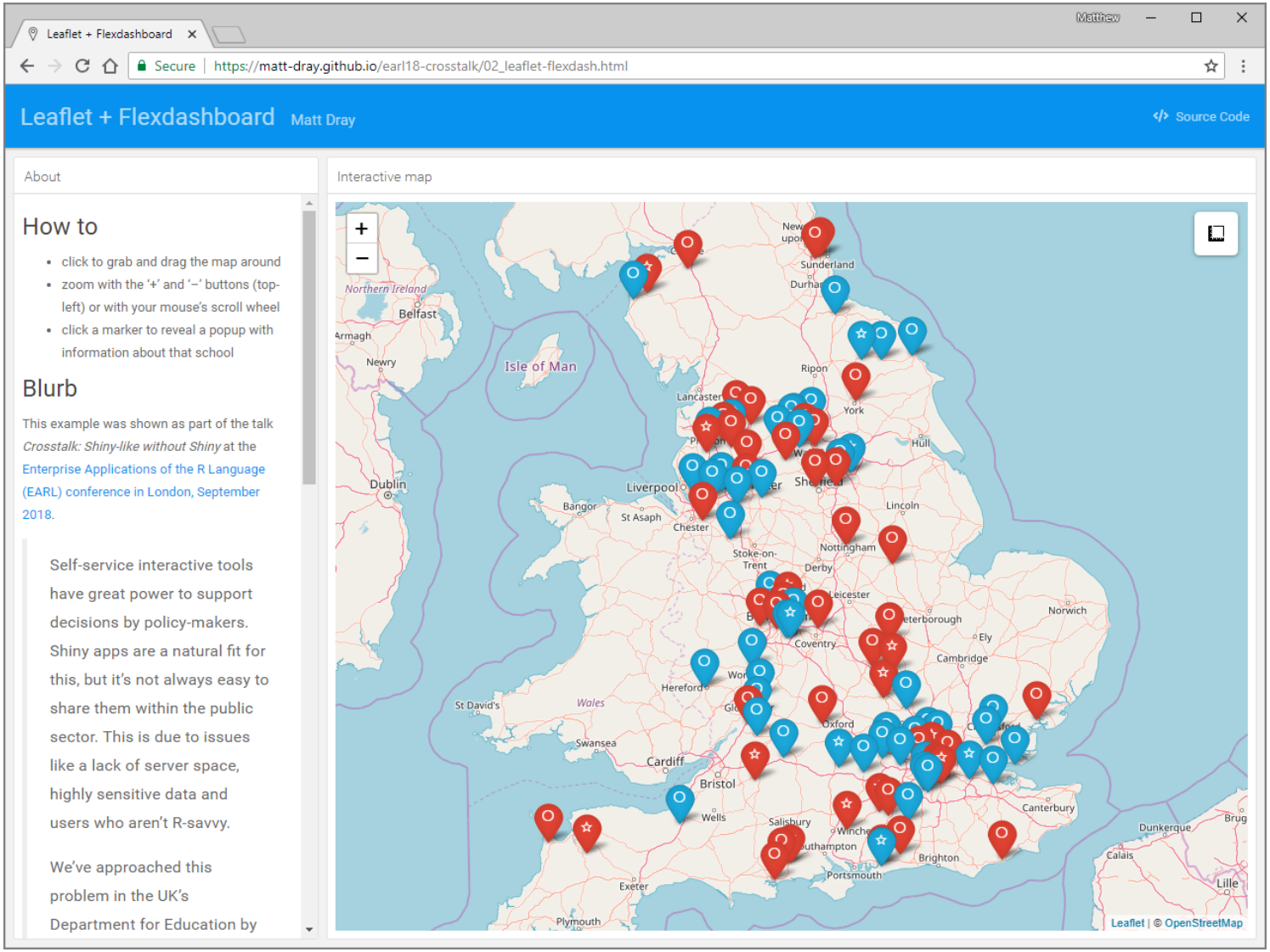
intuitive

documentation

Solution 2

`leaflet` interactive map

in a `flexdashboard`



Demo available here

Solution 2

documentation

self-service

Solution 2

documentation

self-service

tabular data

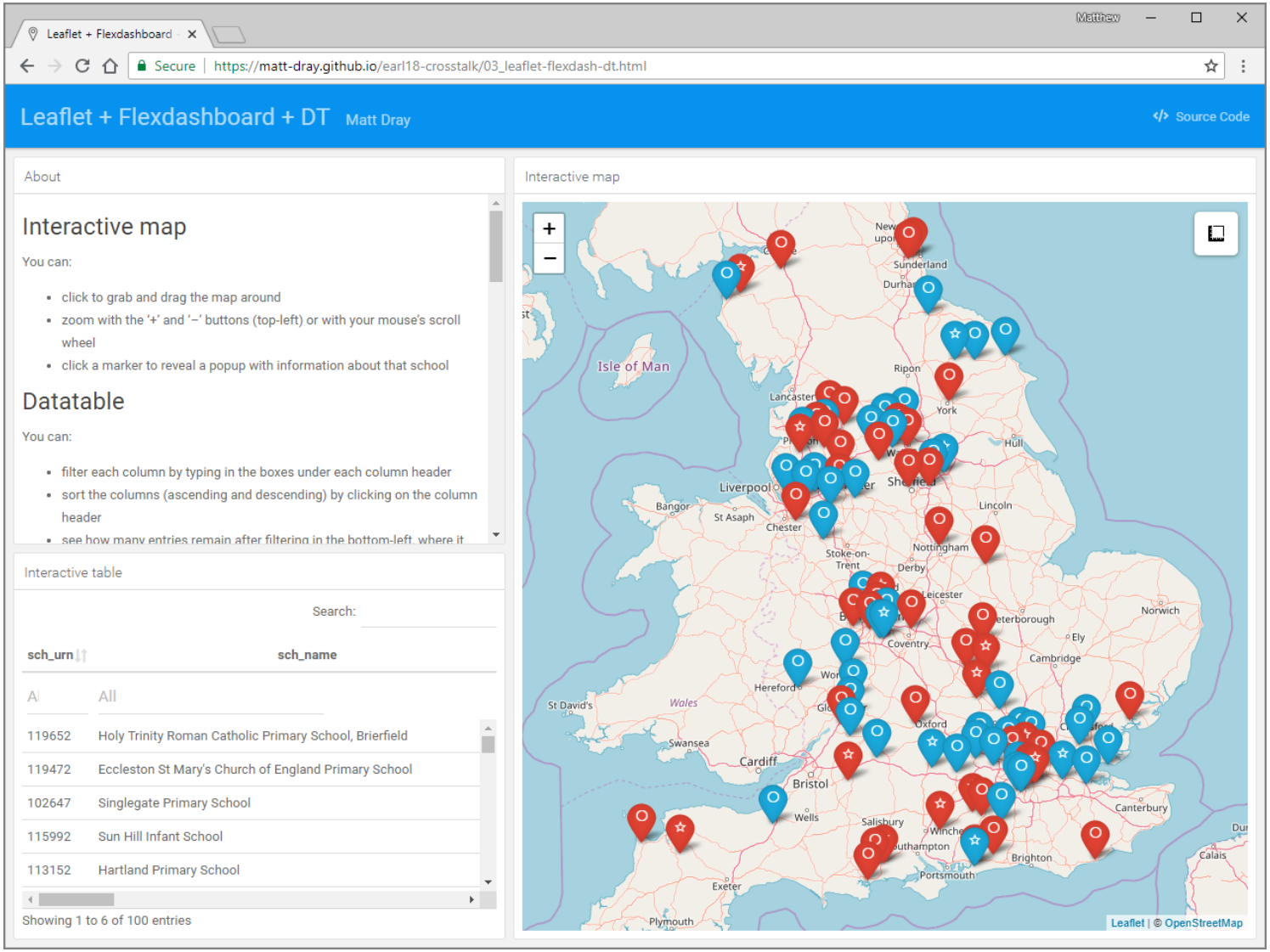
searchable

Solution 3

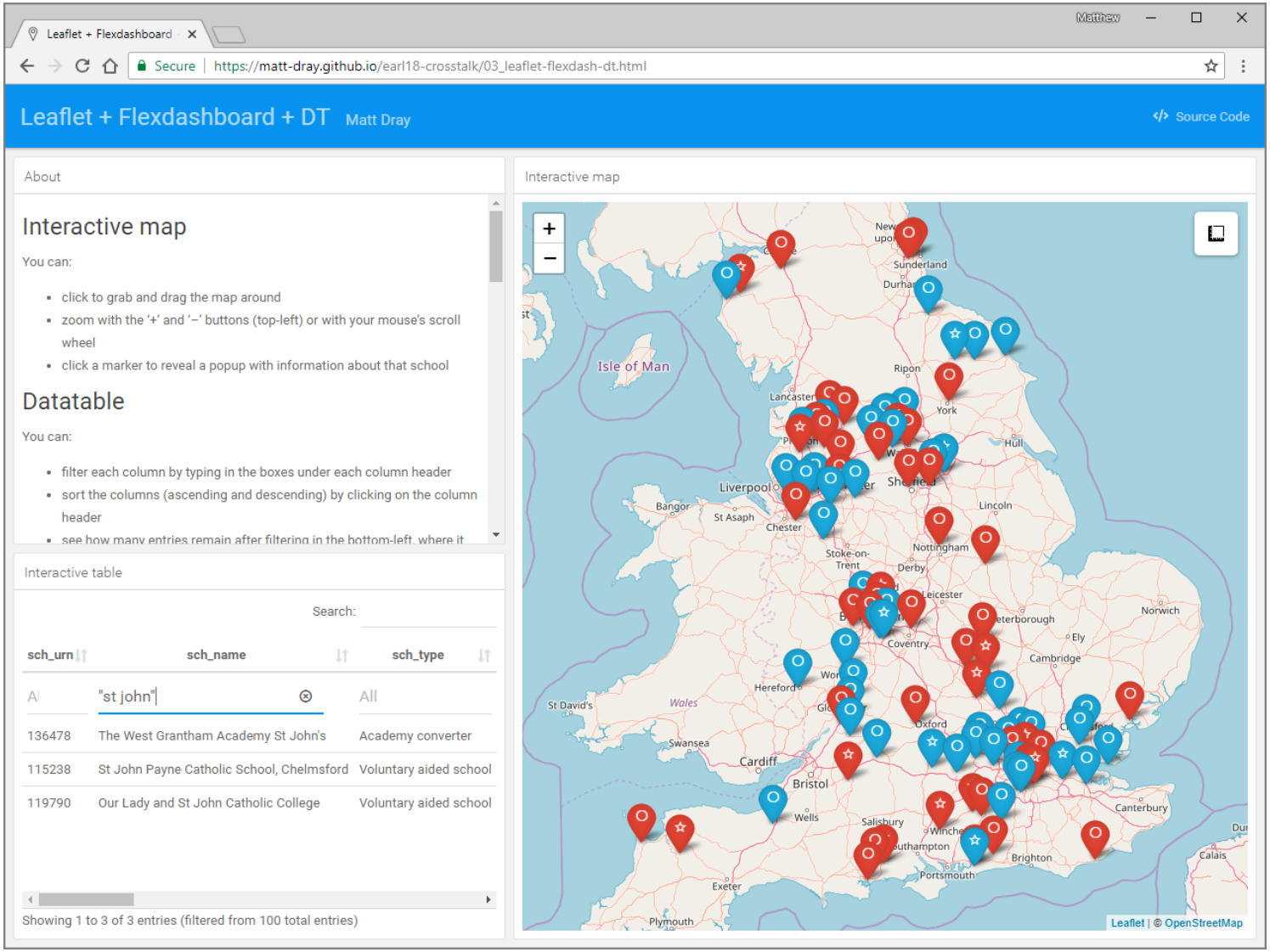
`leaflet` interactive map

in a `flexdashboard`

with a `DT` interactive table



Demo available here



Demo available here

Solution 3

tabular data

searchable

Solution 3

tabular data

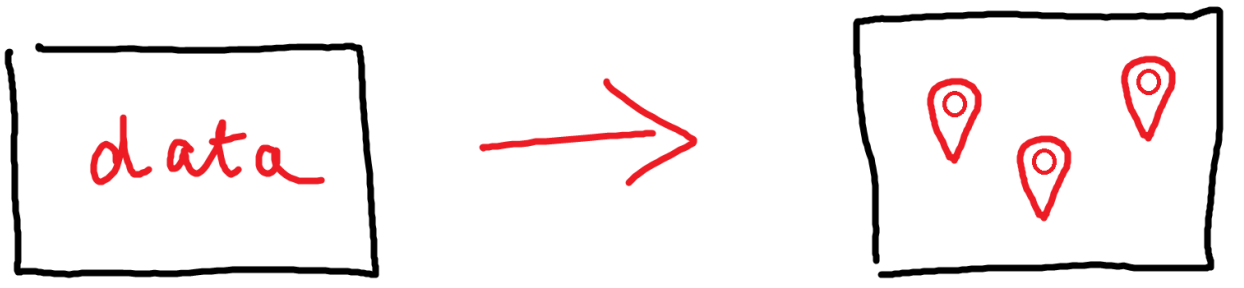
searchable

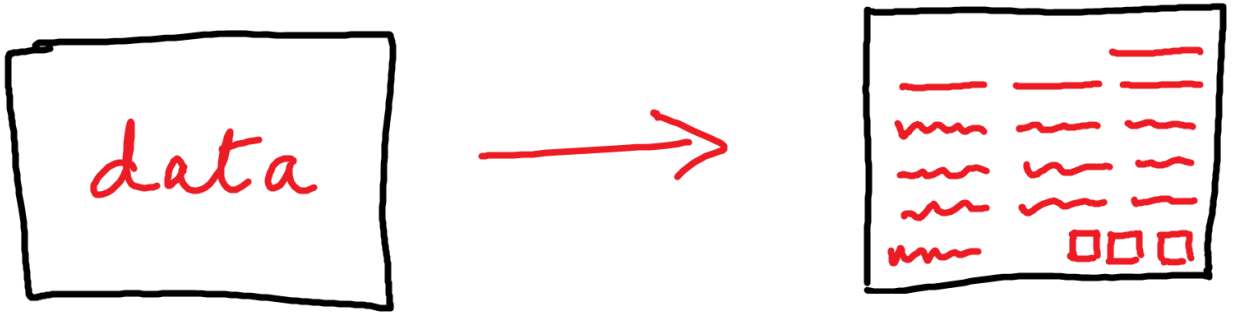
download option

linked table and map

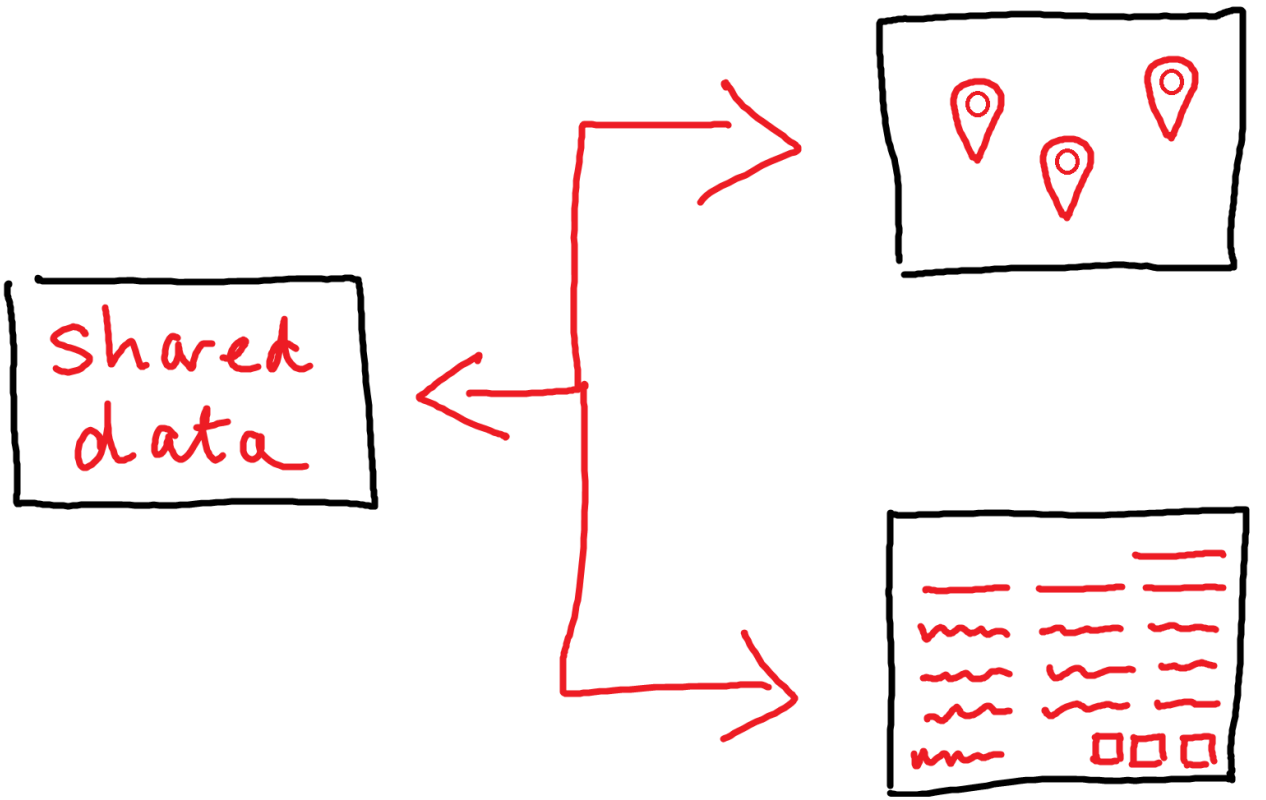
What now?

Current model

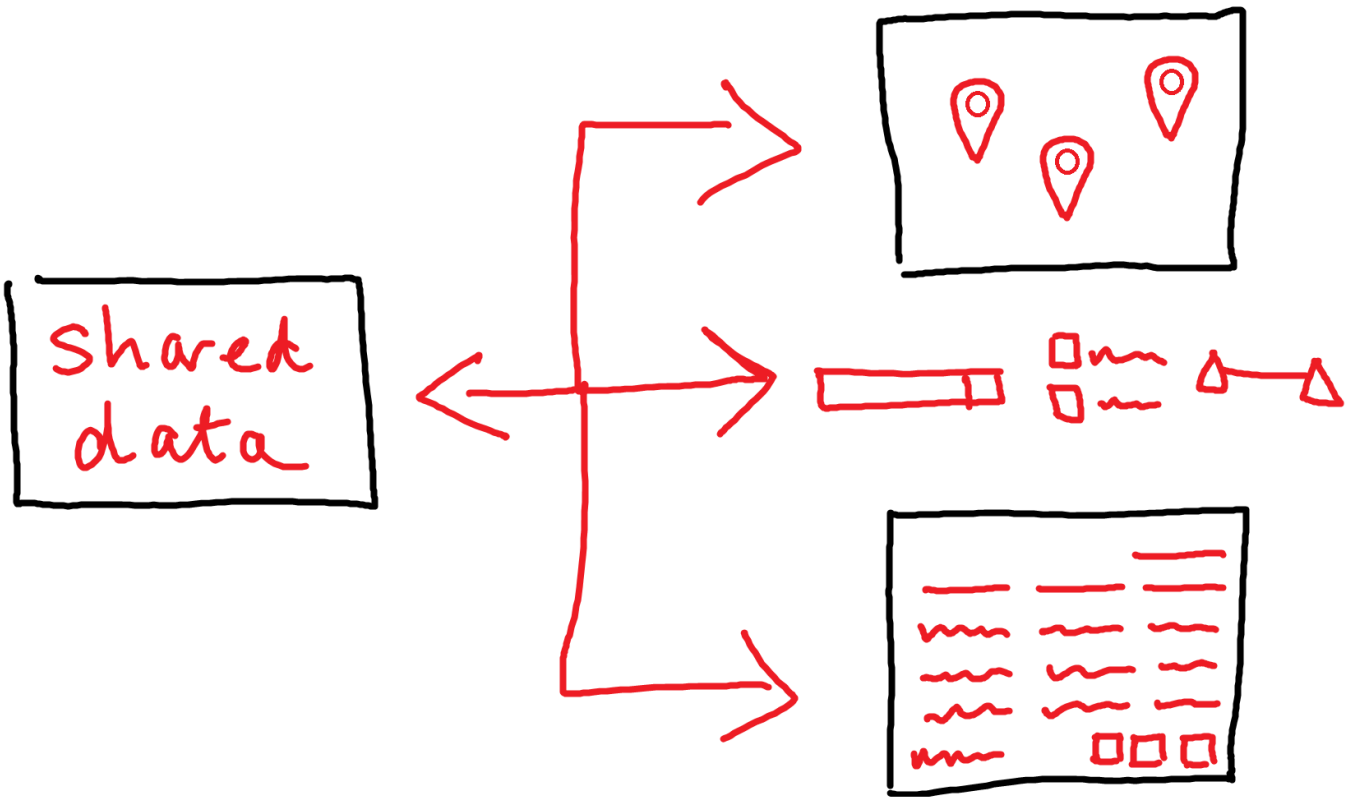




Ideal model



Ideal model



Crosstalk by Joe Cheng

You can make:

> static HTML documents that **do not require Shiny** or any kind of server support and yet they can have **interactivity between widgets**

A small code change

Before

```
data ← readRDS("data/some_data.RDS")  # get data

datatable(data)  # interactive table

leaflet(data) %>% addTiles() %>% addMarkers()  # map
```

A small code change

Before

```
data ← readRDS("data/some_data.RDS")  # get data
datatable(data)  # interactive table
leaflet(data) %>% addTiles() %>% addMarkers()  # map
```

After

```
data ← readRDS("data/some_data.RDS")
shared ← SharedData$new(data)
datatable(shared)
leaflet(shared) %>% addTiles() %>% addMarkers()
```

⚠ Limitations ⚠

htmlwidgets must be made Crosstalk-compatible

Aggregate and summary views not yet available

You can't use 'large' datasets

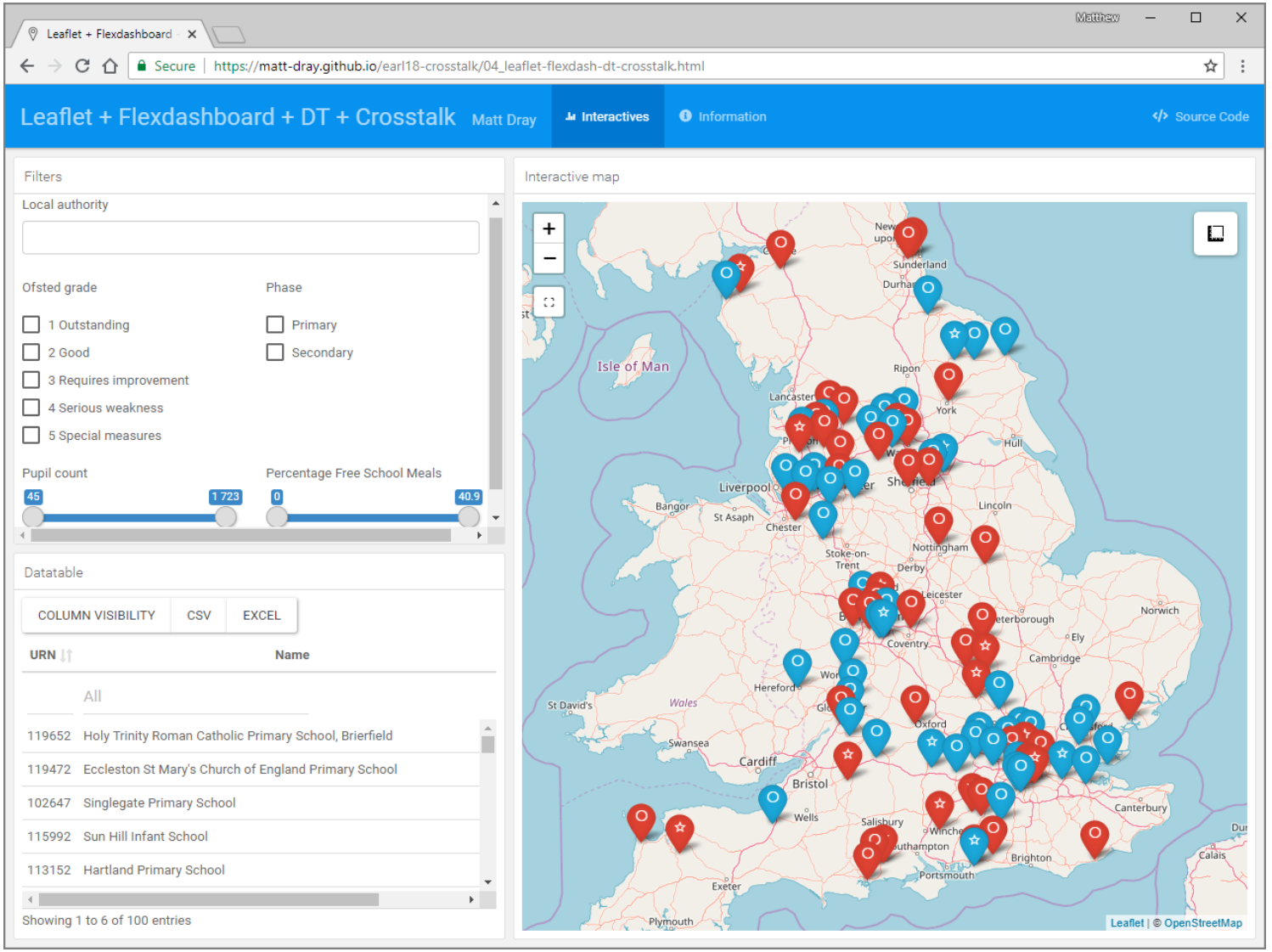
Okay, Solution 4

`leaflet` interactive map

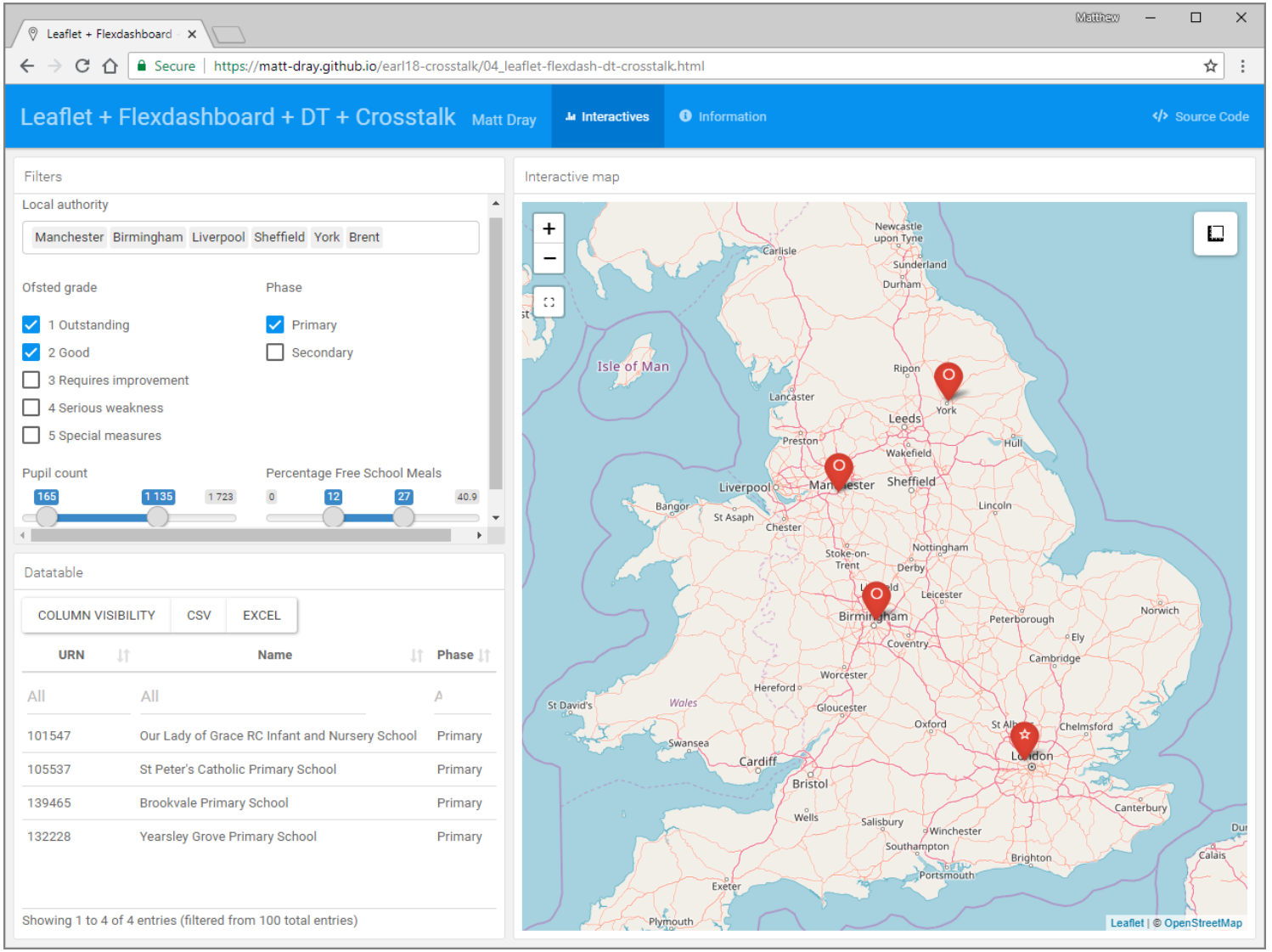
in a `flexdashboard`

with a `DT` interactive table

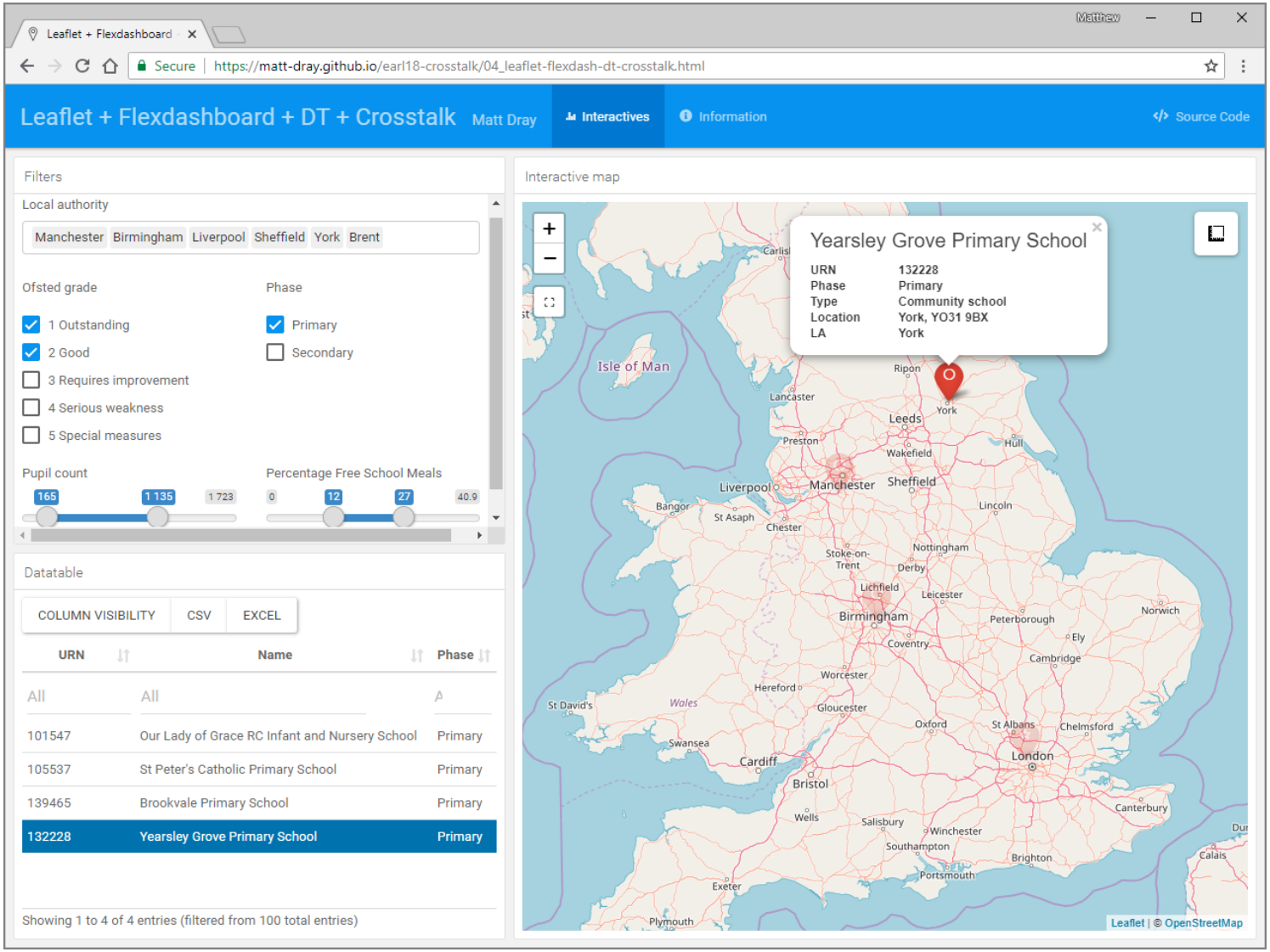
linked with `crosstalk`



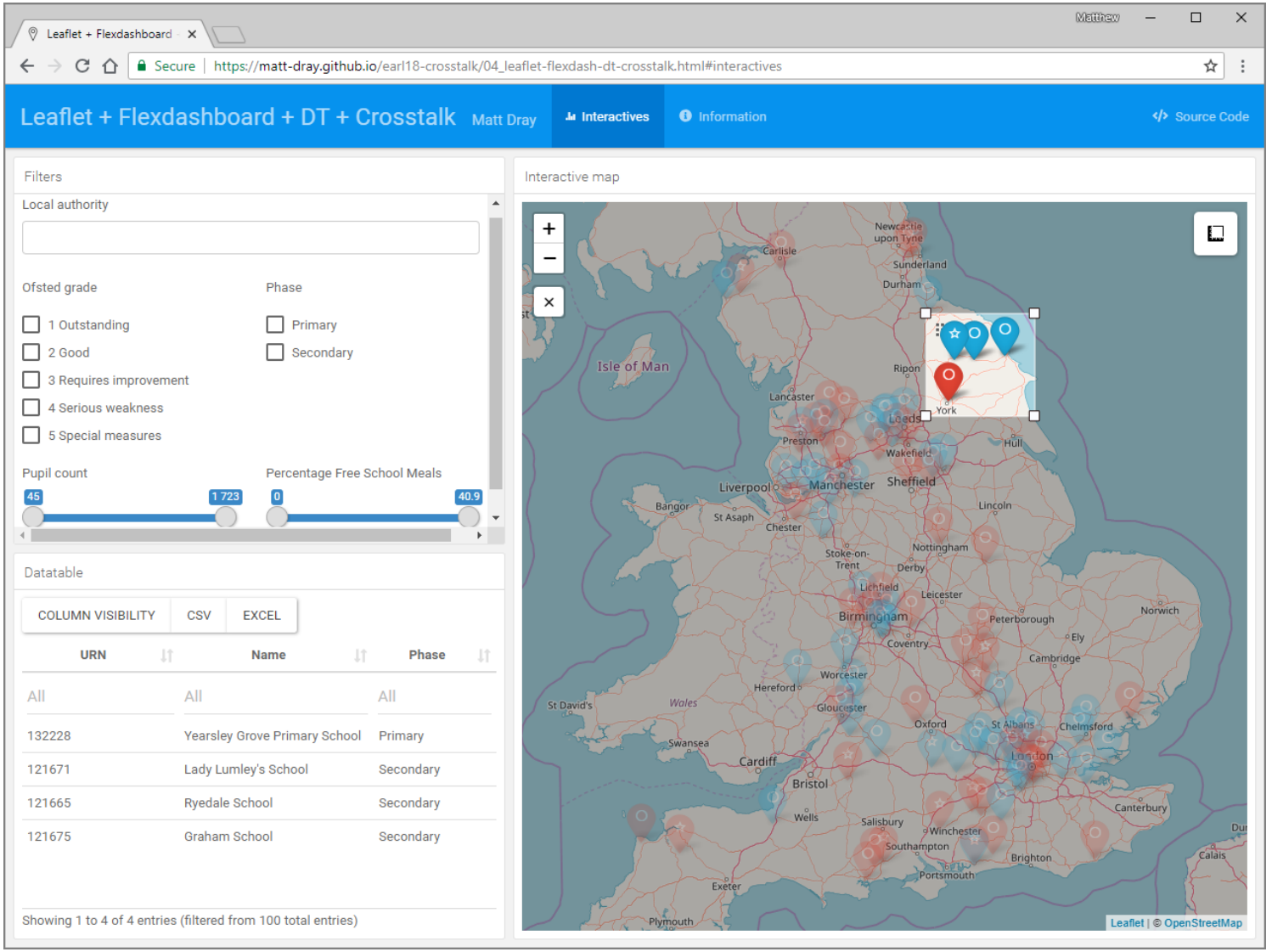
Demo available here



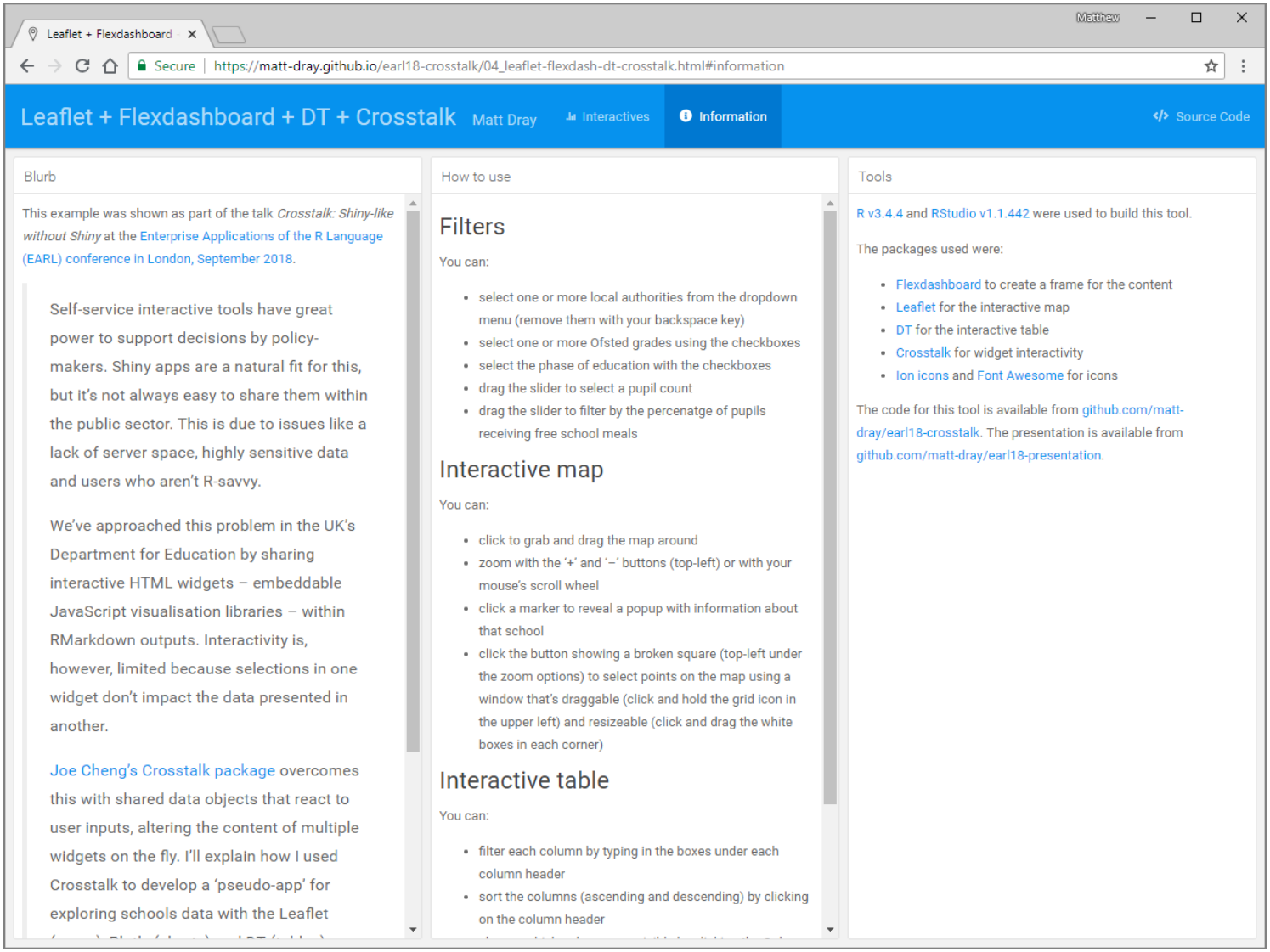
Demo available here



Demo available here



Demo available here



Demo available here

Solution 4

map and table linked

data filtering and brushing

downloads

The ask

Visualise school locations to help make a policy decision ✔

The ask

Visualise school locations to help make a policy decision ✔

Shareable ✔

The ask

Visualise school locations to help make a policy decision ✔

Shareable ✔

Cheap ✔

The ask

Visualise school locations to help make a policy decision ✔

Shareable ✔

Cheap ✔

Quick ✔

The ask

Visualise school locations to help make a policy decision ✔

Shareable ✔

Cheap ✔

Quick ✔

Reusable ✔

Impact

> the maps have been added to a submission for the minister

> policy and the external stakeholders have been extremely impressed

> due to the positive feedback we are now using the same process for [another policy]

TL;DR

Need to create a small interactive app without a server or Shiny?

Try the `crosstalk` package.

Resources

This presentation

Hosted online

GitHub repo

Made with Yihui Xie's Xaringan package

Crosstalk docs

Site and GitHub repo

CRAN manual and details

Talk at rstudio::conf 2017

Demos

Solutions 1, 2, 3 and 4

GitHub repo

Examples

Gapminder example

Demo without and with Crosstalk

Matt Dray

matthew.dray@education.gov.uk

rostrum.blog | twitter.com/mattdray | github.com/matt-dray