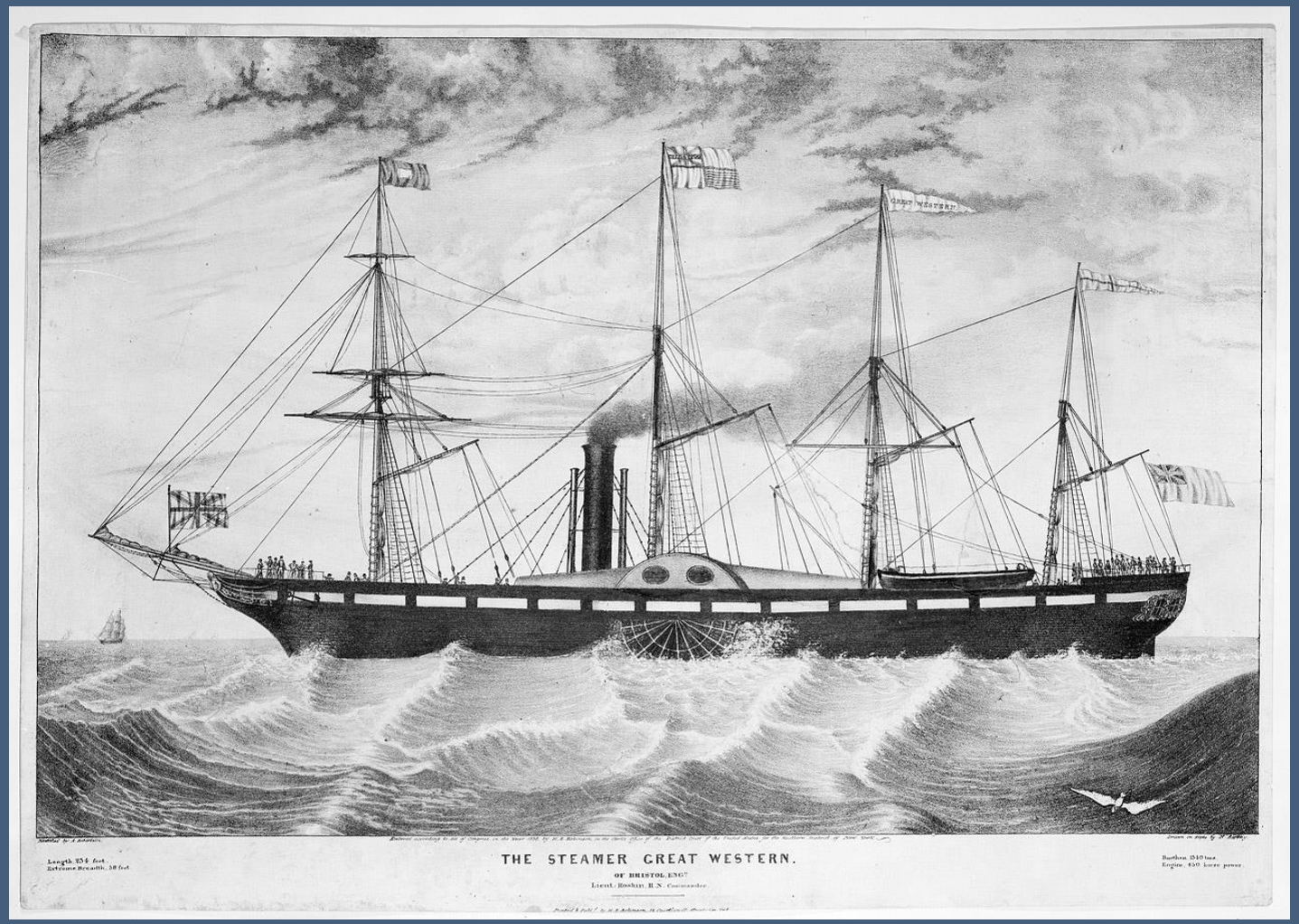


Visualizing Brunel's Steamship Social Network

Gareth Jones

Research Software Engineer
ACRC
University of Bristol

Overview

- Who was Brunel?
- The data
- Building a data model
- A quick look at d3
- JavaScript and React (not too much, I promise)
- Iterations of the graph

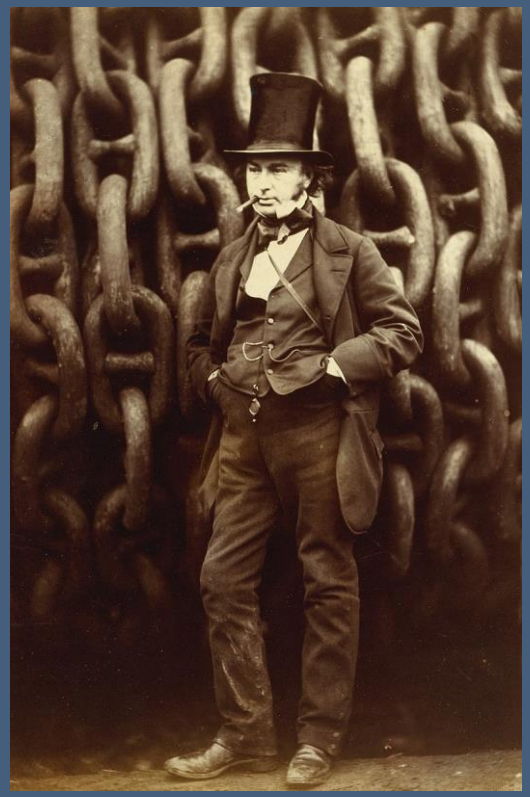
Who was Isambard Kingdom Brunel?

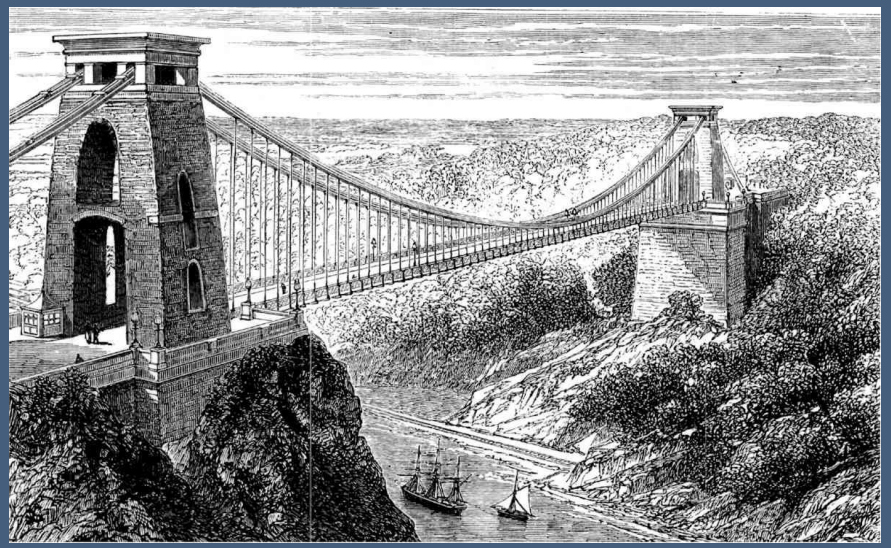
Born in 1806

Trained in France as an engineer

Key figure in the industrial revolution

Involved in the construction of the Great Western Railway, dockyards, bridges and steamships





19th October 2020

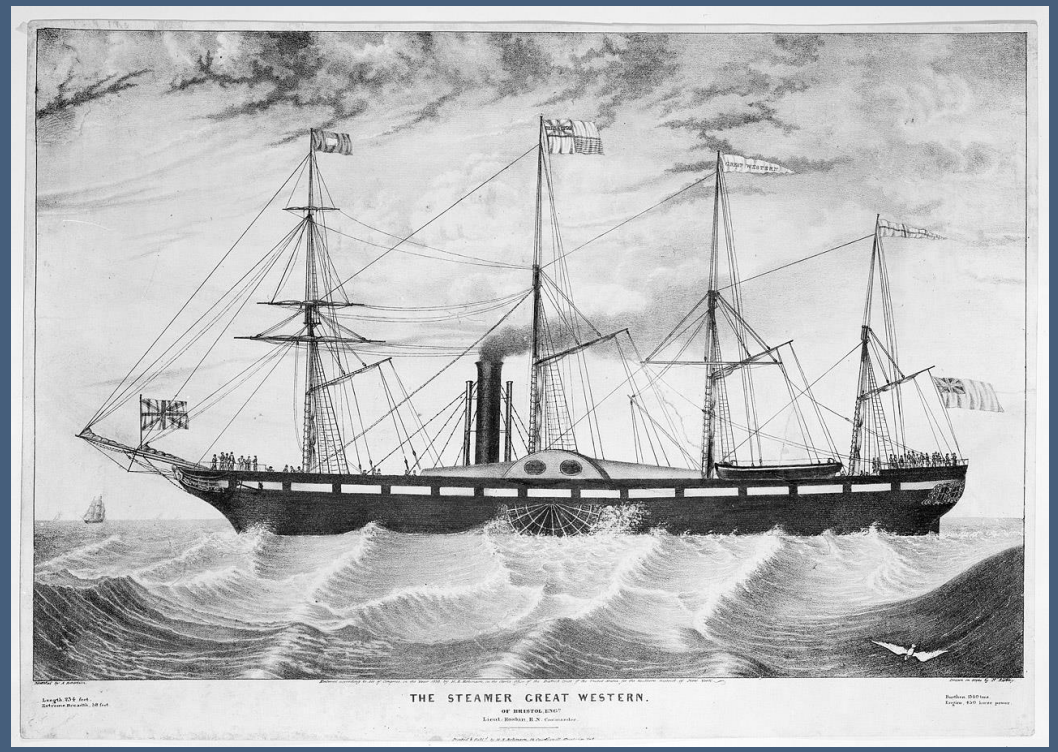
Three Great Steamships

SS Great Western
SS Great Britain
SS Great Eastern

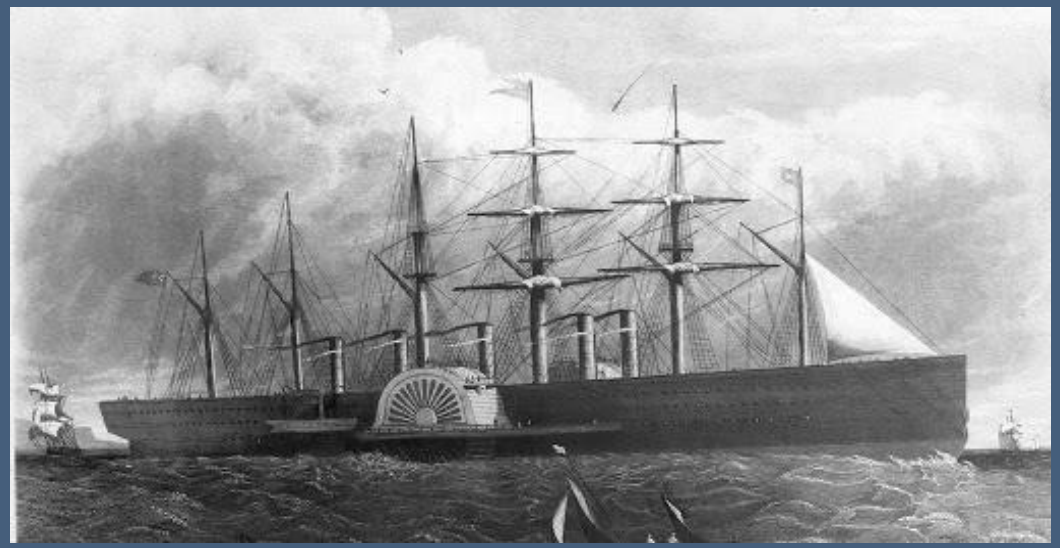
First purpose built Atlantic crossing steamships

Increasingly large and complex undertakings

IKR at the centre of a network of engineers, designers and shareholders



SS Great Western



SS Great Eastern

The data

Dr James Boyd – Brunel Institute

Hundreds of letters of correspondence

Analysed, scanned and digitized by volunteers

Importance of individuals in the construction of each ship can be assessed

The data

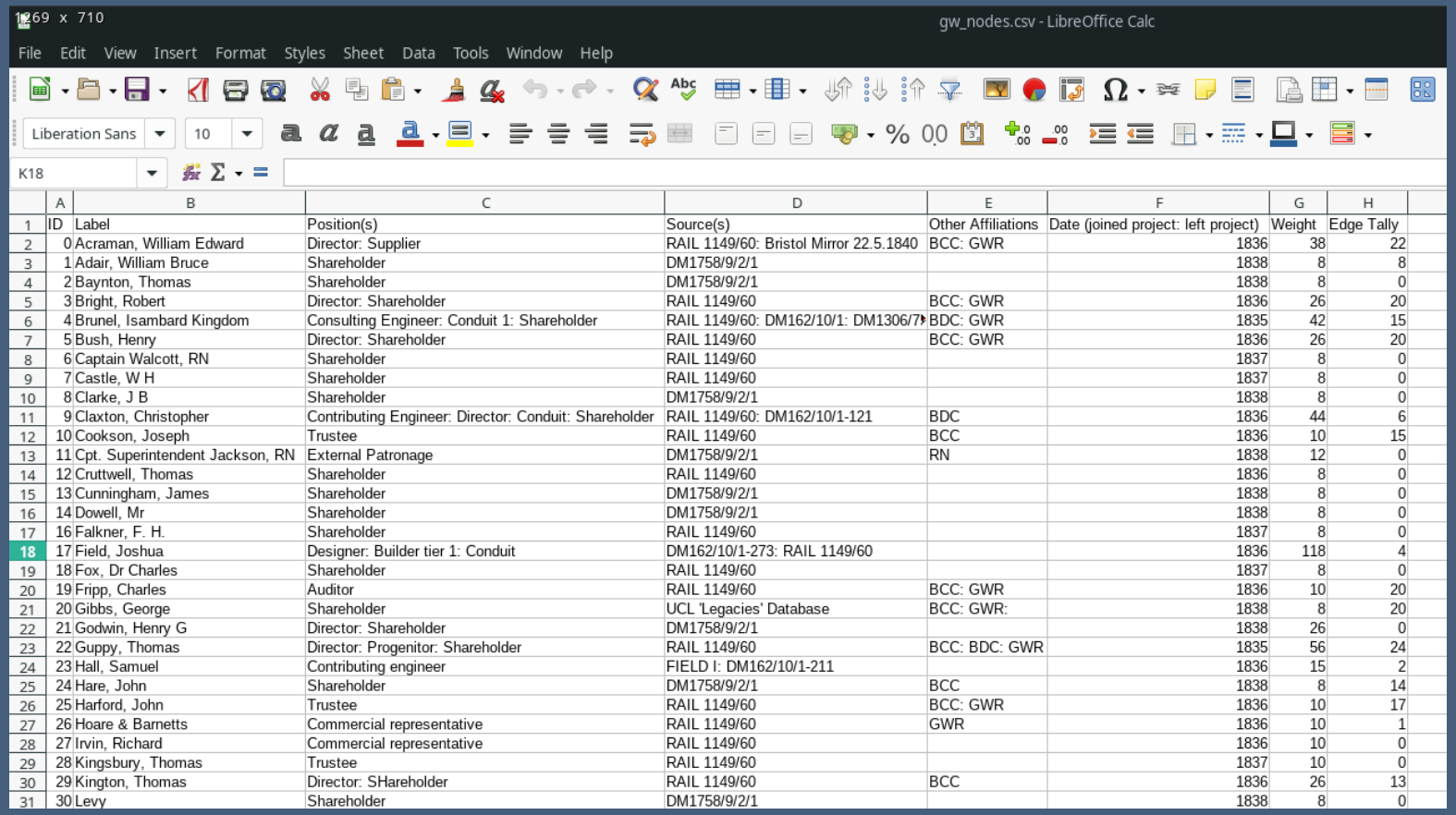
How to build a data model from something like this?

From xlsx to csv

Sometimes their internet isn't working and you get read missing data over the phone (thanks lockdown)

Processing data is part of the job

pandas is great



	A	B	C	D	E	F	G	H
1	ID	Label	Position(s)	Source(s)	Other Affiliations	Date (joined project: left project)	Weight	Edge Tally
2	0	Acraman, William Edward	Director: Supplier	RAIL 1149/60: Bristol Mirror 22.5.1840	BCC: GWR	1836	38	22
3	1	Adair, William Bruce	Shareholder	DM1758/9/2/1		1838	8	8
4	2	Baynton, Thomas	Shareholder	DM1758/9/2/1		1838	8	0
5	3	Bright, Robert	Director: Shareholder	RAIL 1149/60	BCC: GWR	1836	26	20
6	4	Brunel, Isambard Kingdom	Consulting Engineer: Conduit 1: Shareholder	RAIL 1149/60: DM162/10/1: DM1306/7	BDC: GWR	1835	42	15
7	5	Bush, Henry	Director: Shareholder	RAIL 1149/60	BCC: GWR	1836	26	20
8	6	Captain Walcott, RN	Shareholder	RAIL 1149/60		1837	8	0
9	7	Castle, W H	Shareholder	RAIL 1149/60		1837	8	0
10	8	Clarke, J B	Shareholder	DM1758/9/2/1		1838	8	0
11	9	Claxton, Christopher	Contributing Engineer: Director: Conduit: Shareholder	RAIL 1149/60: DM162/10/1-121	BDC	1836	44	6
12	10	Cookson, Joseph	Trustee	RAIL 1149/60	BCC	1836	10	15
13	11	Cpt. Superintendent Jackson, RN	External Patronage	DM1758/9/2/1	RN	1838	12	0
14	12	Cruttwell, Thomas	Shareholder	RAIL 1149/60		1836	8	0
15	13	Cunningham, James	Shareholder	DM1758/9/2/1		1838	8	0
16	14	Dowell, Mr	Shareholder	DM1758/9/2/1		1838	8	0
17	16	Falkner, F. H.	Shareholder	RAIL 1149/60		1837	8	0
18	17	Field, Joshua	Designer: Builder tier 1: Conduit	DM162/10/1-273: RAIL 1149/60		1836	118	4
19	18	Fox, Dr Charles	Shareholder	RAIL 1149/60		1837	8	0
20	19	Fripp, Charles	Auditor	RAIL 1149/60	BCC: GWR	1836	10	20
21	20	Gibbs, George	Shareholder	UCL 'Legacies' Database	BCC: GWR:	1838	8	20
22	21	Godwin, Henry G	Director: Shareholder	DM1758/9/2/1		1838	26	0
23	22	Guppy, Thomas	Director: Progenitor: Shareholder	RAIL 1149/60	BCC: BDC: GWR	1835	56	24
24	23	Hall, Samuel	Contributing engineer	FIELD I: DM162/10/1-211		1836	15	2
25	24	Hare, John	Shareholder	DM1758/9/2/1	BCC	1838	8	14
26	25	Harford, John	Trustee	RAIL 1149/60	BCC: GWR	1836	10	17
27	26	Hoare & Barnetts	Commercial representative	RAIL 1149/60	GWR	1836	10	1
28	27	Irvin, Richard	Commercial representative	RAIL 1149/60		1836	10	0
29	28	Kingsbury, Thomas	Trustee	RAIL 1149/60		1837	10	0
30	29	Kington, Thomas	Director: SHareholder	RAIL 1149/60	BCC	1836	26	13
31	30	Levy	Shareholder	DM1758/9/2/1		1838	8	0

19th October 2020

Building a data model

1. Parse the data into a Dataframe
2. Read all the people
3. Read all the businesses
4. Store their affiliations
5. Store their position(s)
6. Store the dates over which they were involved
7. Assign a unique ID to each person, business etc
8. Store this in an easily readable format (for a machine) - JSON!

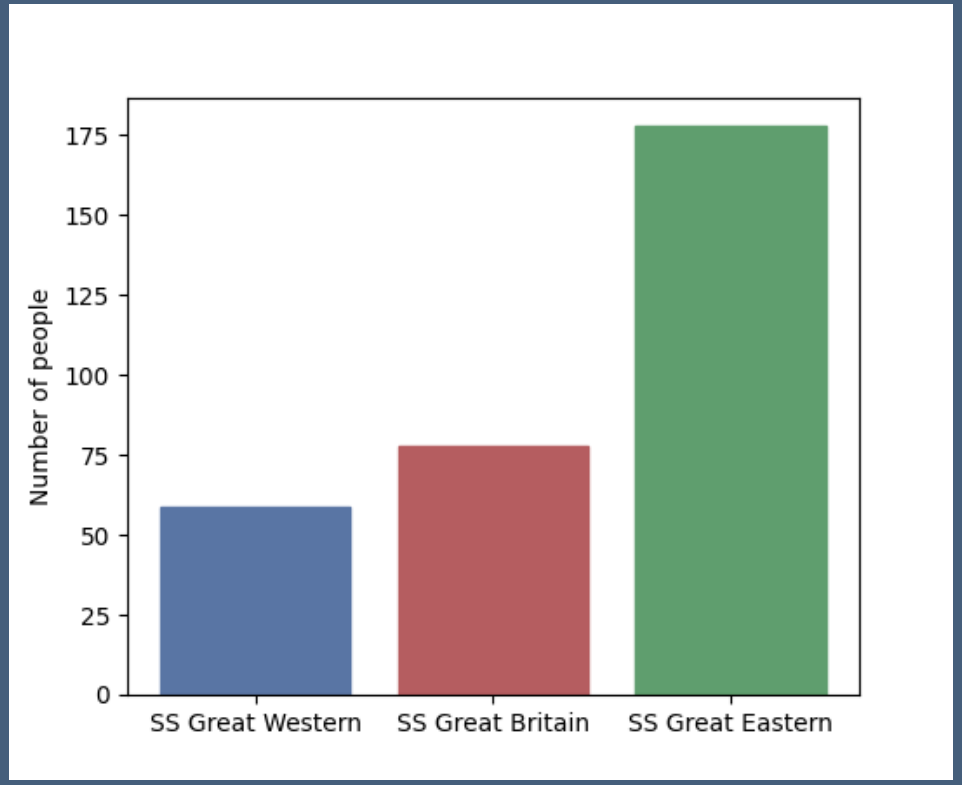
Increasing complexity

Each steamship was bigger and more complex than the last

Increasing number of people involved in their construction

How to best visualize the increased complexity of the social network?

Force graphs!



Force graphs – nodes and edges

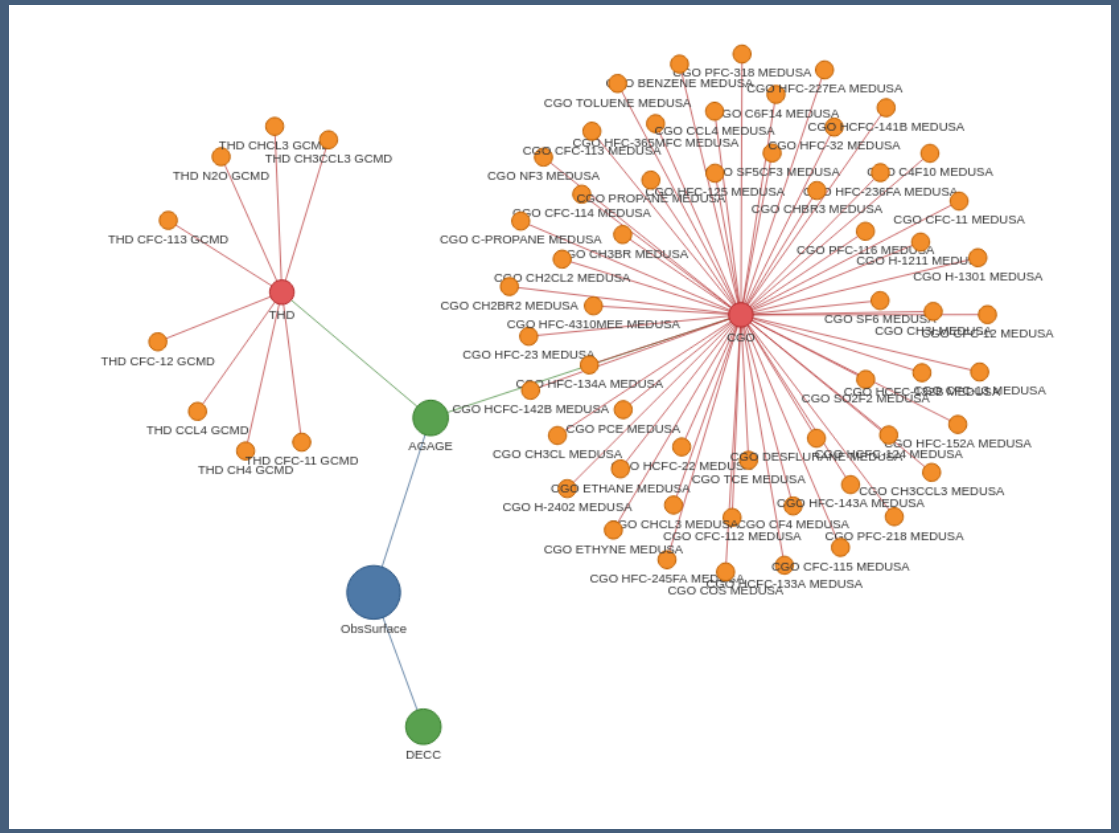
Show the relationships between data

Brain is bad at visualising things like folder structure

It can struggle when visualising relationships between more than a few people

Colour and layout lets us encode large amounts of data

Data can also be encoded in the physical interactions between nodes – gravity / repulsive forces



Greenhouse gas data stored in the cloud

19th October 2020

Force graphs – nodes and edges

The d3-force library – relatively simple, performant

Many simulation types – gravity, repulsive force

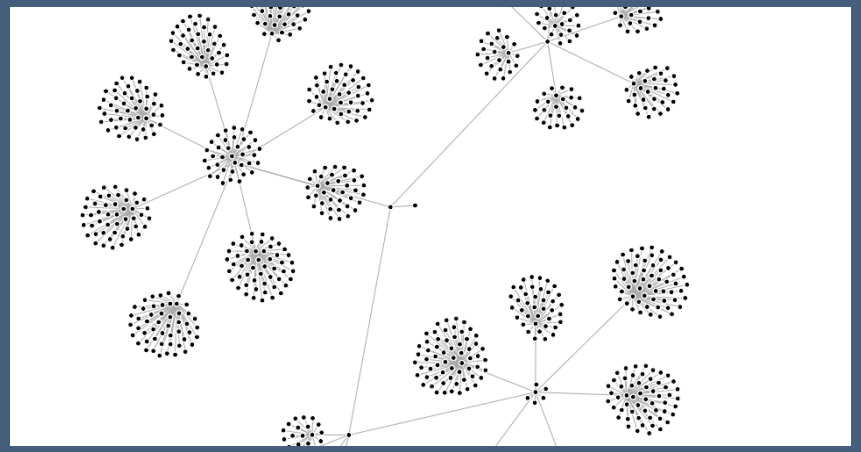
Clean, modern JavaScript

Nice way to visualise the connections between individuals in a social network

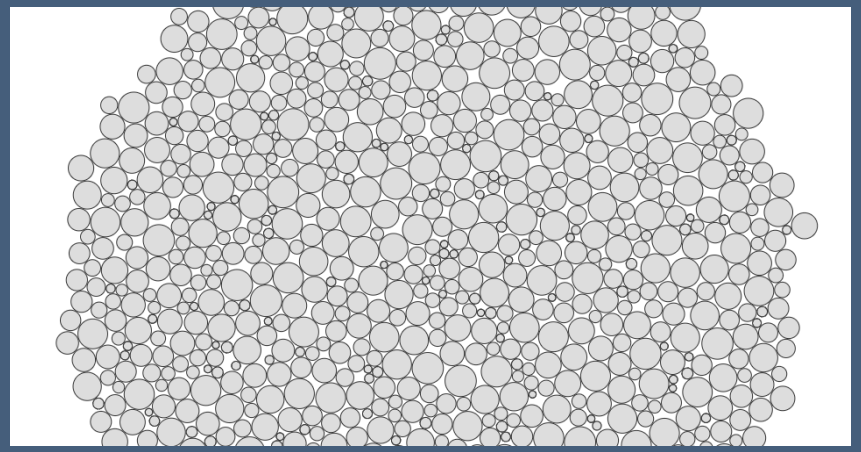
Nodes: people involved in the project

Edges: relationships between people

Force-Directed Tree



Collision Detection



Images taken from https://github.com/d3/d3-force

d3-force

Mike Bostock, the creator of d3 has many example online at

https://observablehq.com

https://observablehq.com/@d3/force-directed-tree

https://observablehq.com/@d3/collision-detection

React

Facebook's JavaScript library

Good documentation

Supports modern ES6 version of JS

Easy to get started with

```
class ShoppingList extends React.Component {
  render() {
    return (
      <div className="shopping-list">
        <h1>Shopping List for {this.props.name}</h1>
        <ul>
          <li>Instagram</li>
          <li>WhatsApp</li>
          <li>Oculus</li>
        </ul>
      </div>
    );
  }
}
```

Example React component class

A simple d3 force graph with React

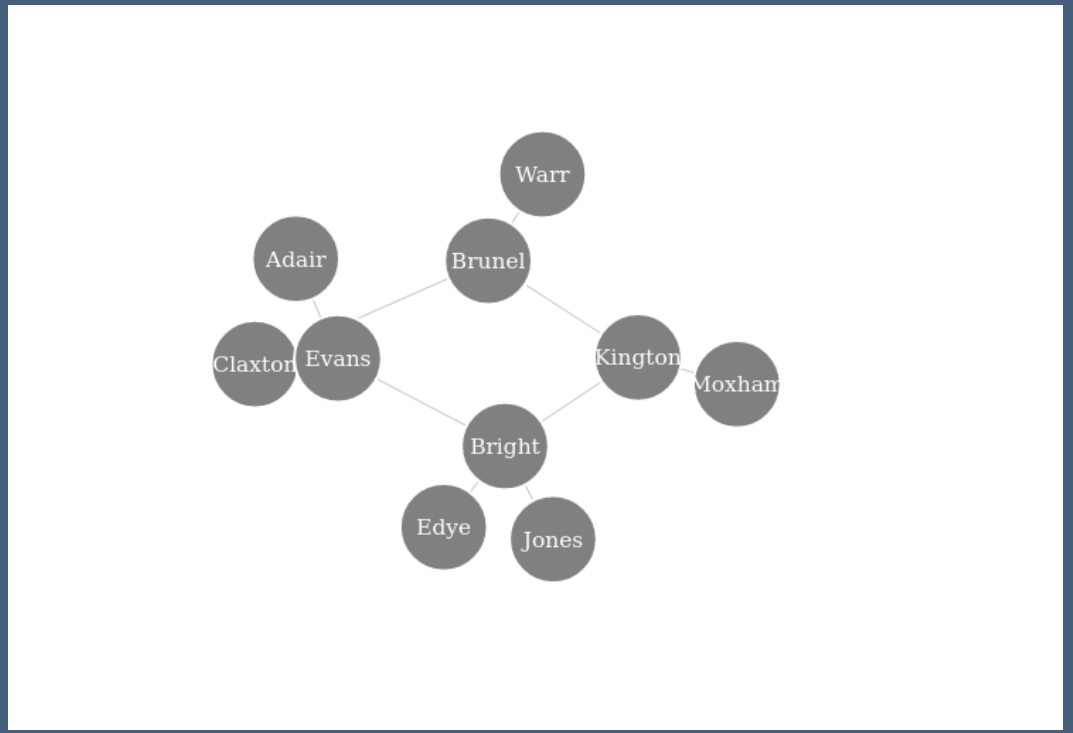
https://github.com/gareth-j/d3-react-example

Very simple app built with React and d3

Commented code

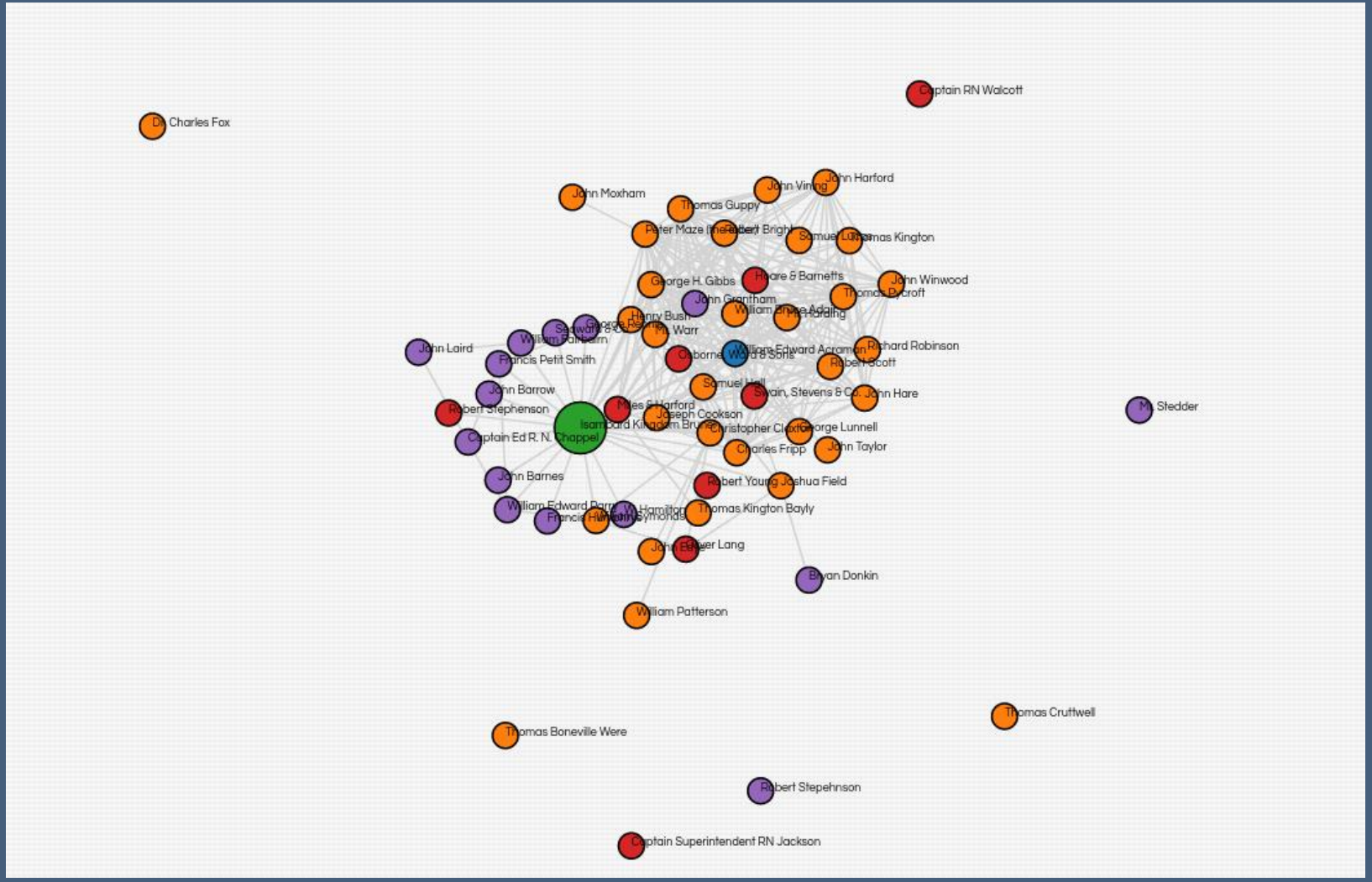
Easily modifiable

Follows best practices



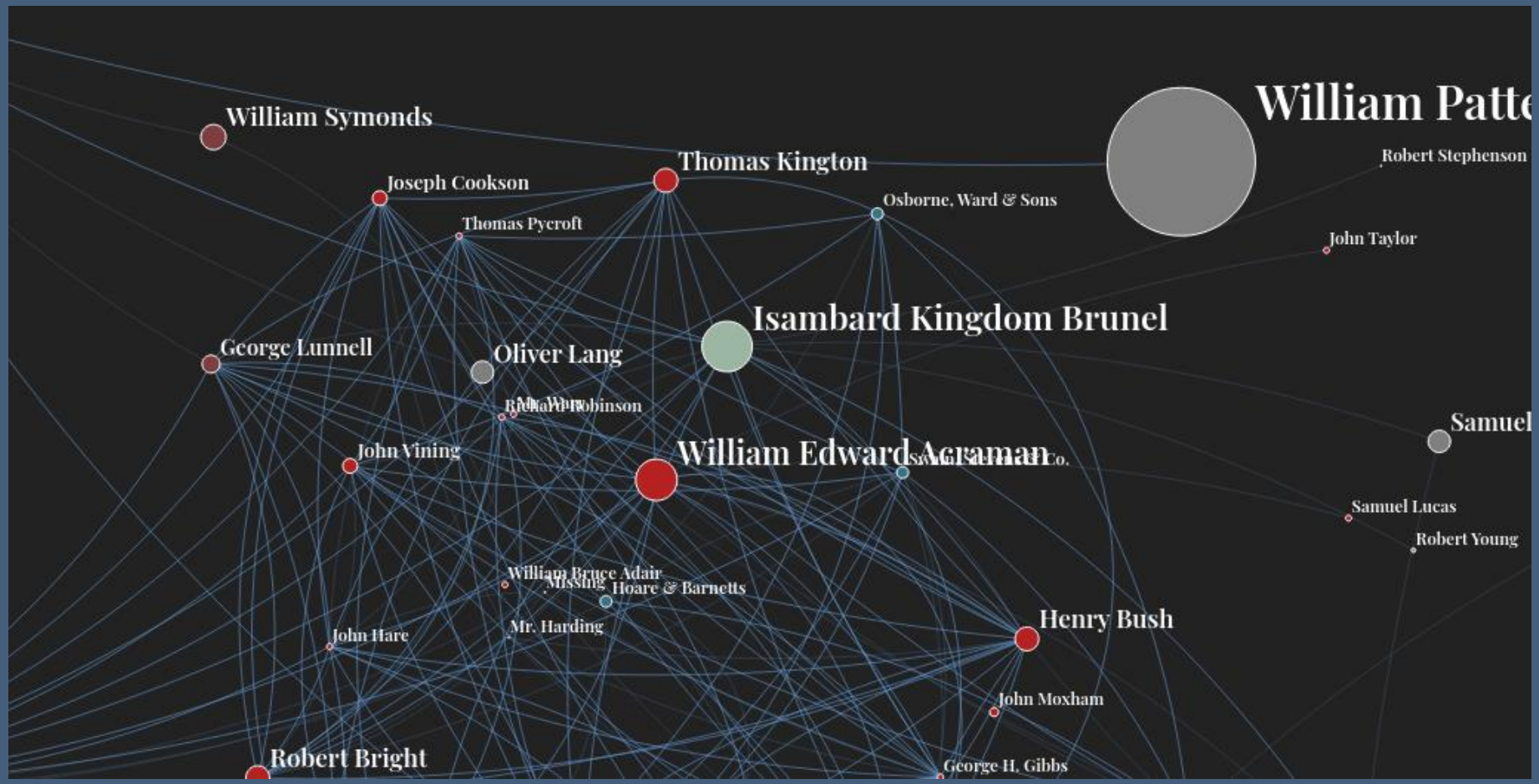
19th October 2020

Beginnings



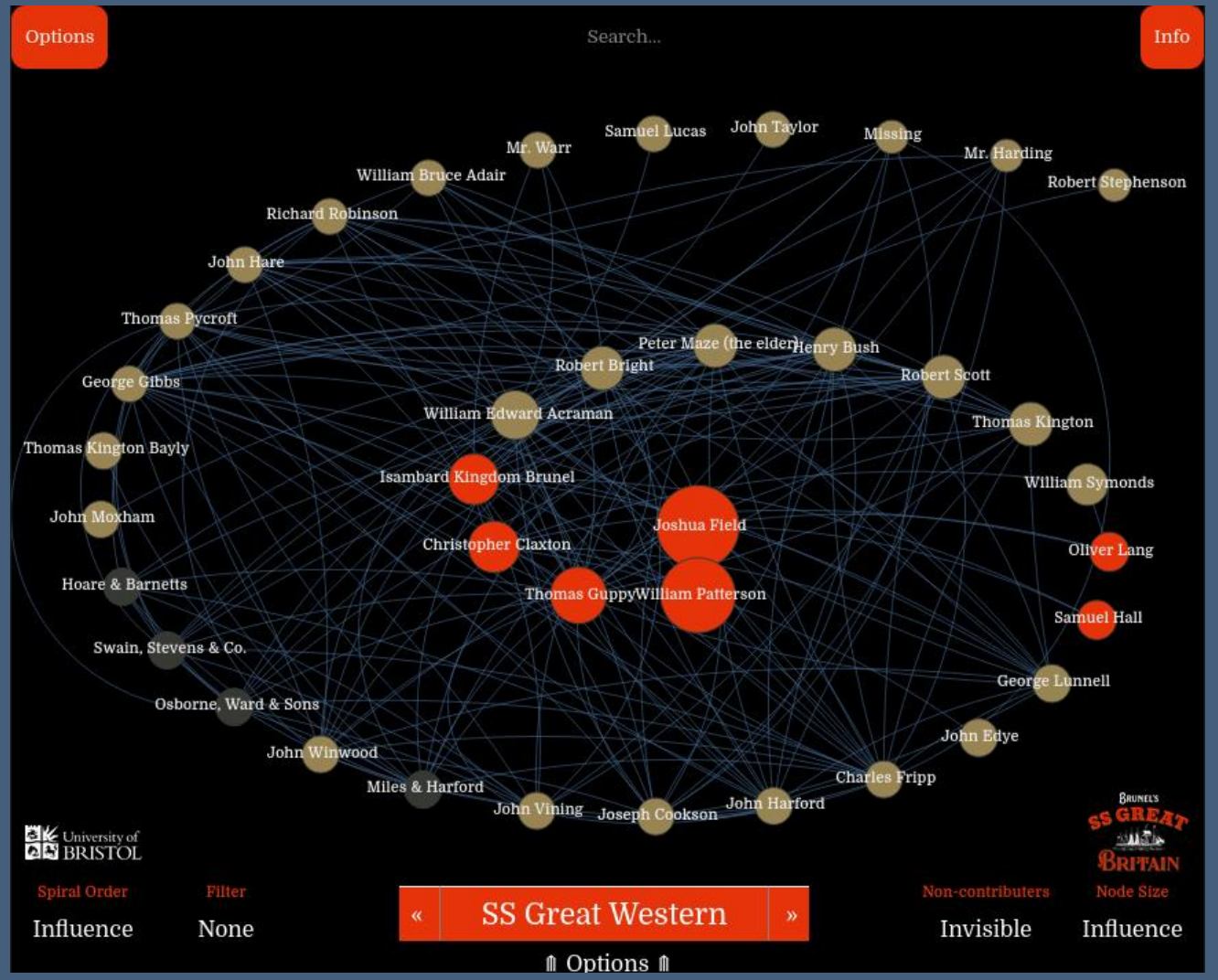
19th October 2020

Iteration 4/5



19th October 2020

Current



19th October 2020

Process overview

Spreadsheet data

data model

force graph

GitHub Pages

19th October 2020

Lessons learnt

- Don't transfer data by voice over the phone – low bitrate
- Build a data model that describes your data and allows manipulation
- Pick a well used tool for the job – newest and shiniest may not be the best
- Many eyes – easy to miss things on a project
- d3 is nice

Ask RSE

Any questions?

Help with code – code design, testing and performance questions

ask-rse@bristol.ac.uk

g.m.jones@bristol.ac.uk

19th October 2020