

Module Code: EE3616

Module Title: Graphics

Assessment 2: Multiple Choice Project -
Option 3: Data Visualisation Artefact

Name: Gethin Jones

Student No: 1422999

Course Title: Digital Design

Year of Study: 2017/18

Data Visualisation - Experimentation infographics graphic

Contents

Title	Page
1 Introduction	1
2 Design Process	2
3 Production	5
4 Final artefact	13
5 Marketing	16
6 Evaluation	17
7 Assets	18



Data Visualisation - Experimentation infographics graphic

1 Introduction

Subject

I decided to stick with the same theme as the first part of the assignment of sustainability and create a data visualisation for global sustainability statistics.

Idea & Concept

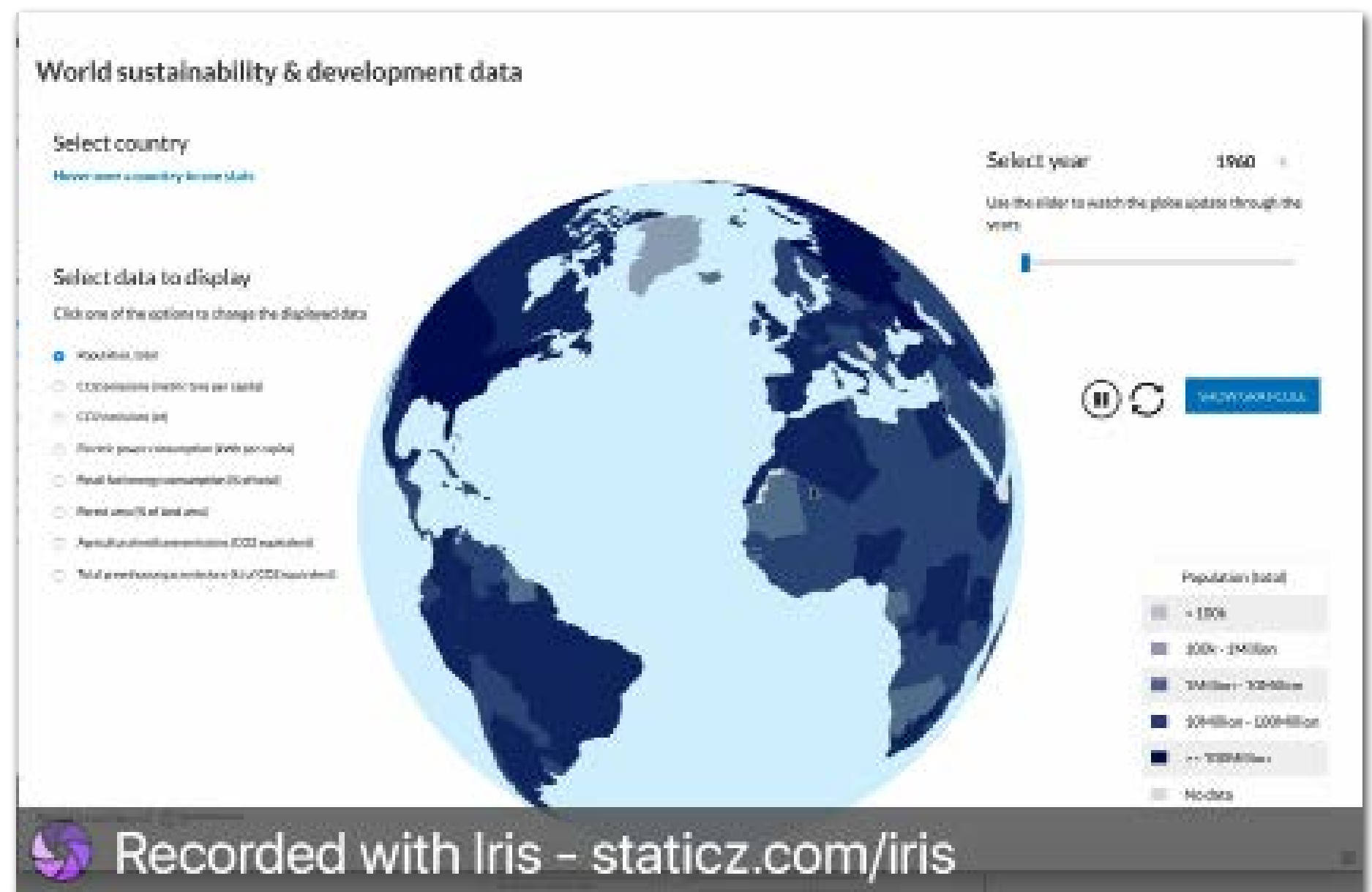
My idea was to create an interactive graphic showing world development and sustainability data mapped onto a globe. It would allow people to easily visualize large amounts of data and see how it has changed over time, by updating the visual for each year.

The data is mapped onto a globe which users can pan around and get data for each country for the selected year. The user can also slide through the past 50 years and the globe will update to show the data for that year. Each country on the globe is coloured for each data set depending on a unique set of bands showing highest to lowest to make the data really easy to digest.

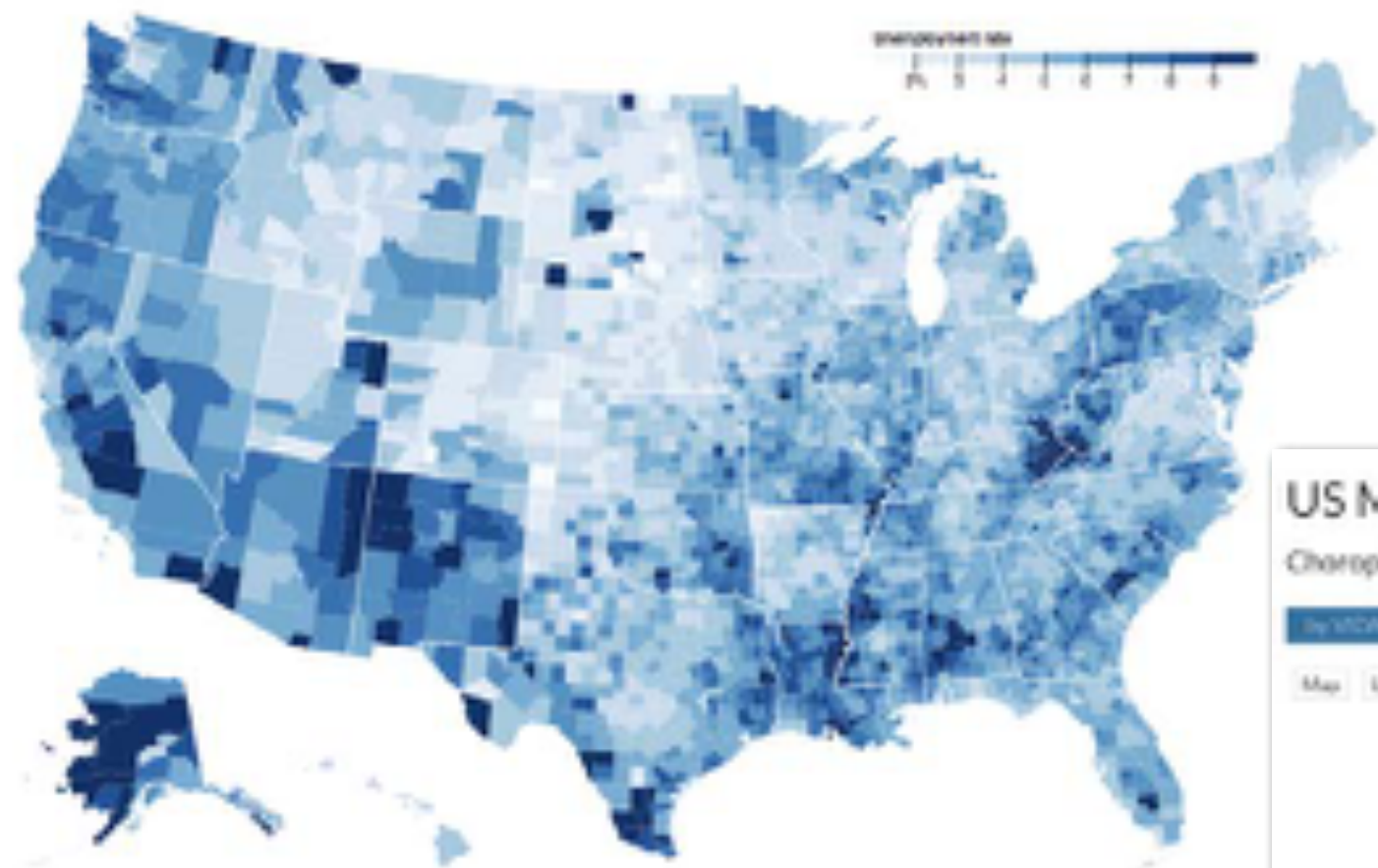
Context

The main context of the asset will be its presence online, with a dedicated website, or hosted within a related website as a feature. Users will be able to load the website and interact with the globe from any computer.

As well as hosting the site, the asset could be used as an educational tool for corporate and educational talks, in schools or fairs and events. The artefact could be left loaded on a touch screen device for people to come and interact with.



Choropleth



Source: Bureau of Labor Statistics, Census Bureau
This choropleth shows unemployment rates as of August, 2010 with a threshold scale. I employed a mix of command-line tools to transform the fixed-width text file into a CSV, including `dev2csv`.

I really liked how colour is used in these examples to visualise data. Darker areas represent higher numbers, in these examples unemployment rates and obesity in the united states. I could use a similar technique to visualise development data on a global scale.

US Map States - Choropleth Plus Bar

Choropleth with bar chart combination. Data shows obesity rate of adults in the US.

by NOAA Map | Obesity | 2010-2011 | 1,648,000,000

Map | US Map | Horizontal Bar Chart



Document Properties

Color 1

Color 2

Main

Washington

Average

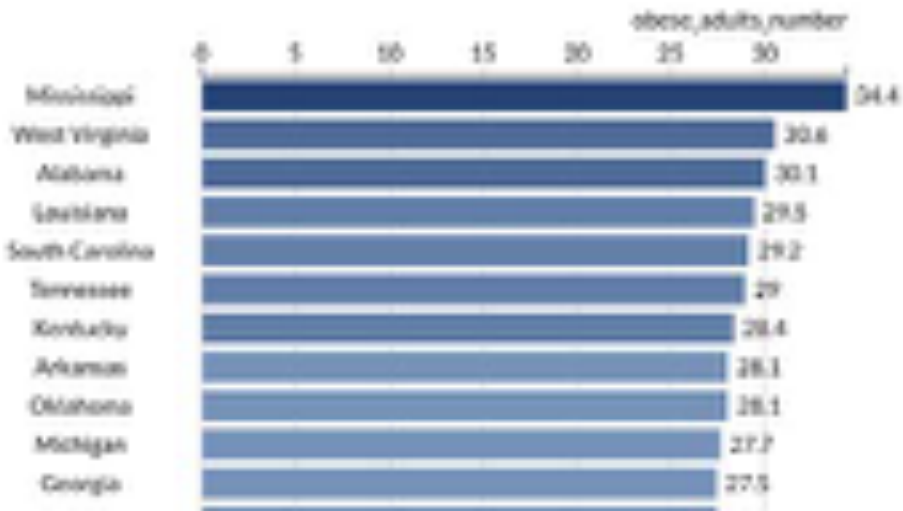
Option Aves

State Data

State And District Of Columbia

Value Data

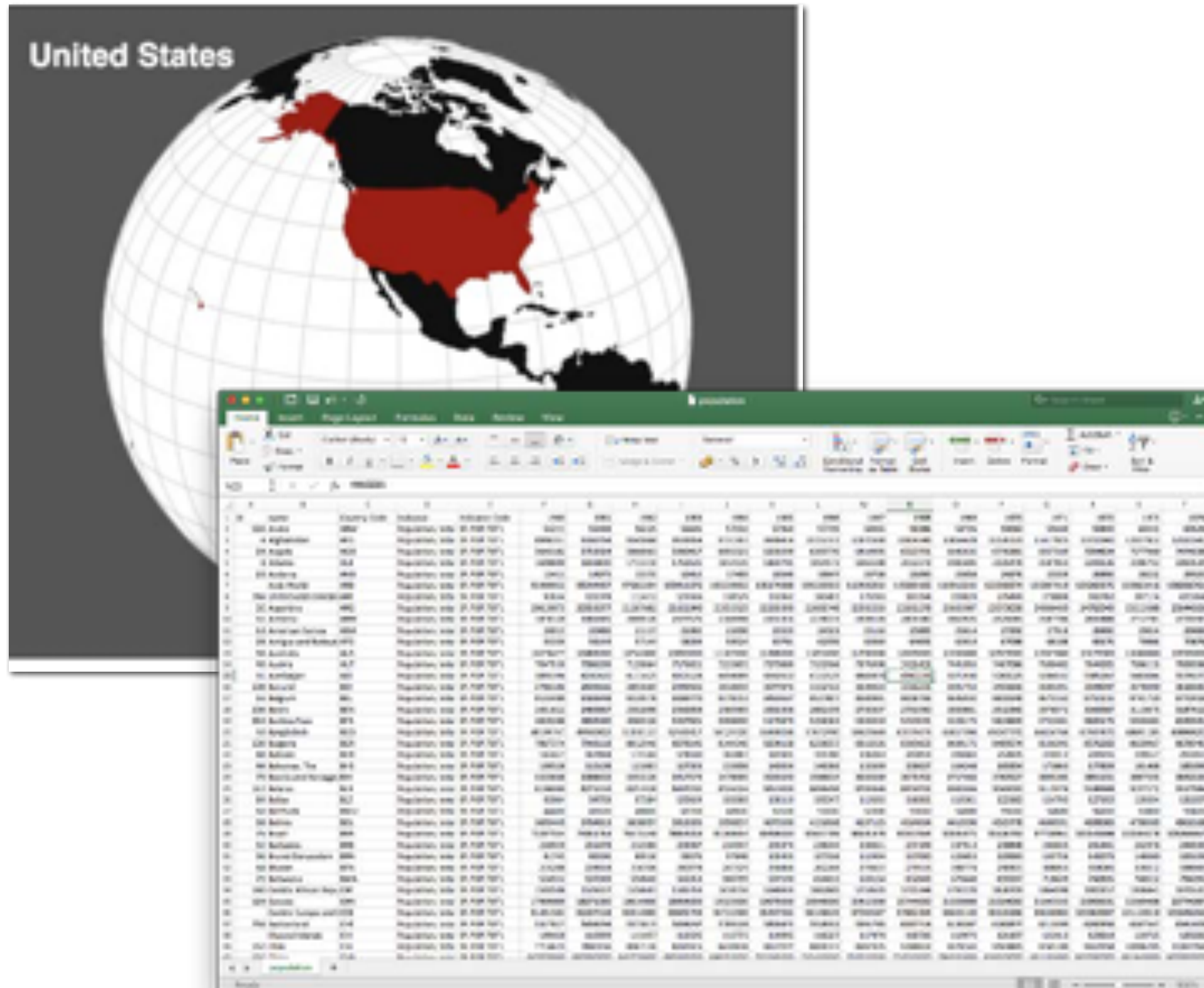
Obese Adults Number



Using a combination of the globe and the choropleth I could create an easy to digest visual showing a range of different data about each country, sorted into high and low groupings to compare statistics such as co2 and green-house gas emissions, as well as things like population.

Initial ideas/ prototyping

Having explored multiple avenues of data visualisation, I decided on implementing an interactive globe that would show data for each country, and give an overall representation of data for each country across the globe.



Graphic design execution

Existing design features

- Visualise globe in 3D
- Drag and rotate the globe
- Hover over countries to show names

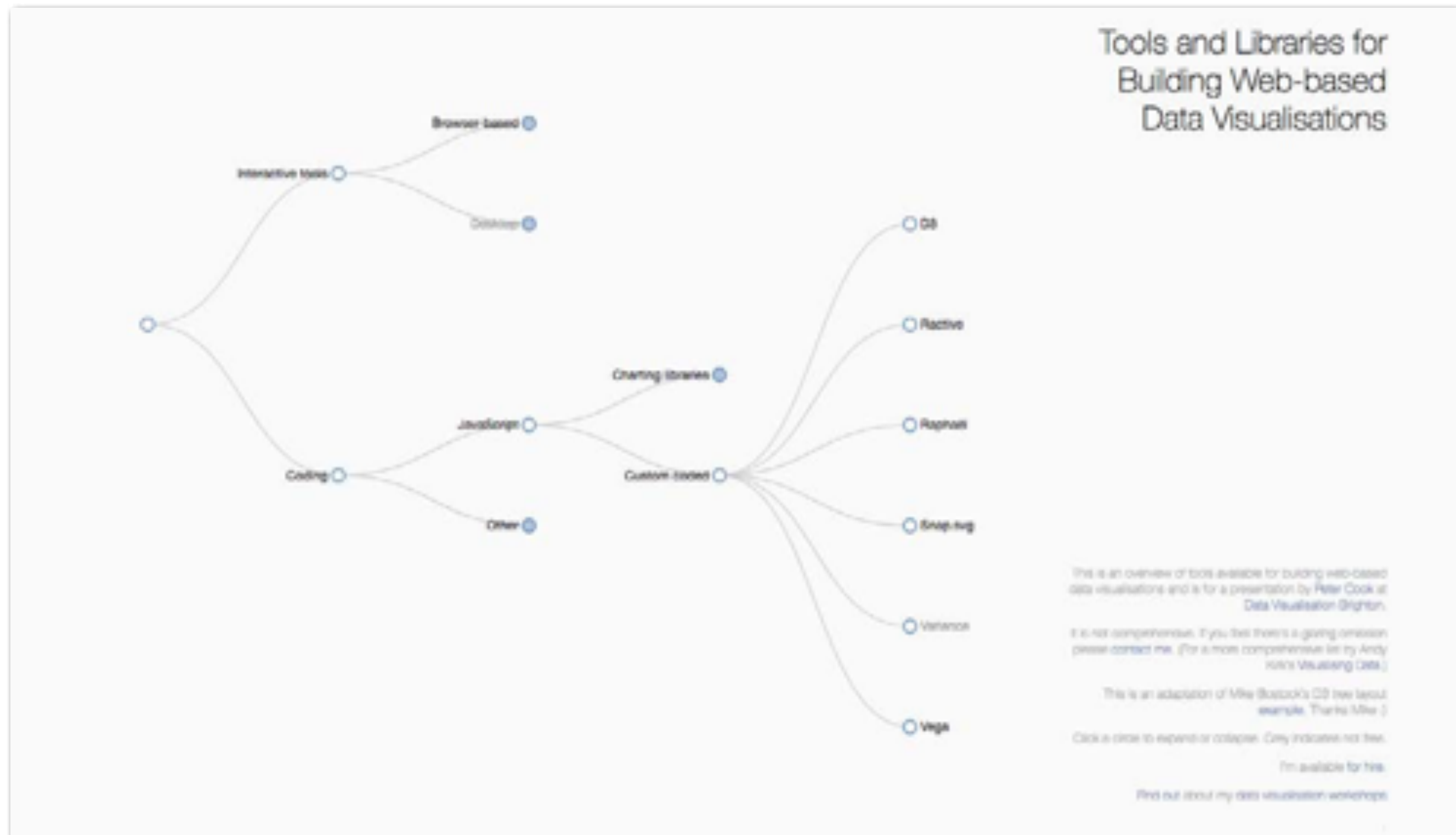
Features to add

- Add data for each country to appear next to the country name.
- A way to change what data the globe is displaying
- Colour representation of data bands
- A year slider to change the year the globe is displaying data for
- Legends for each data set to show different colour representations
- Controls to animate and reset the globe

3 Production

Choice of core technologies

I wanted to take this opportunity to learn how to use a new technology. Given this approach, I looked at a range of data visualisation tools for creating my artefact. I decided to look specifically at custom coded, javascript solutions, as this was of particular interest to me. This handy visualisation of visualisation tools provided a number of options!



Choice of core technologies



"D3.js is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation." -**D3JS.org**

I decided to use the D3 Library to support the creation of my asset due to its powerful custom data visualisation capabilities.



Using D3, a combination of SVG, HTML, CSS and JavaScript will allow me to render a globe, map data for each country, handle animation and control/update the data for each year.

Data-Driven Documents



ZURB Foundation

I decided to use the Zurb Foundation framework because of it's useful prestyled Slider element. The slider can be binded to the year display, providing an easy UI for controlling the year.



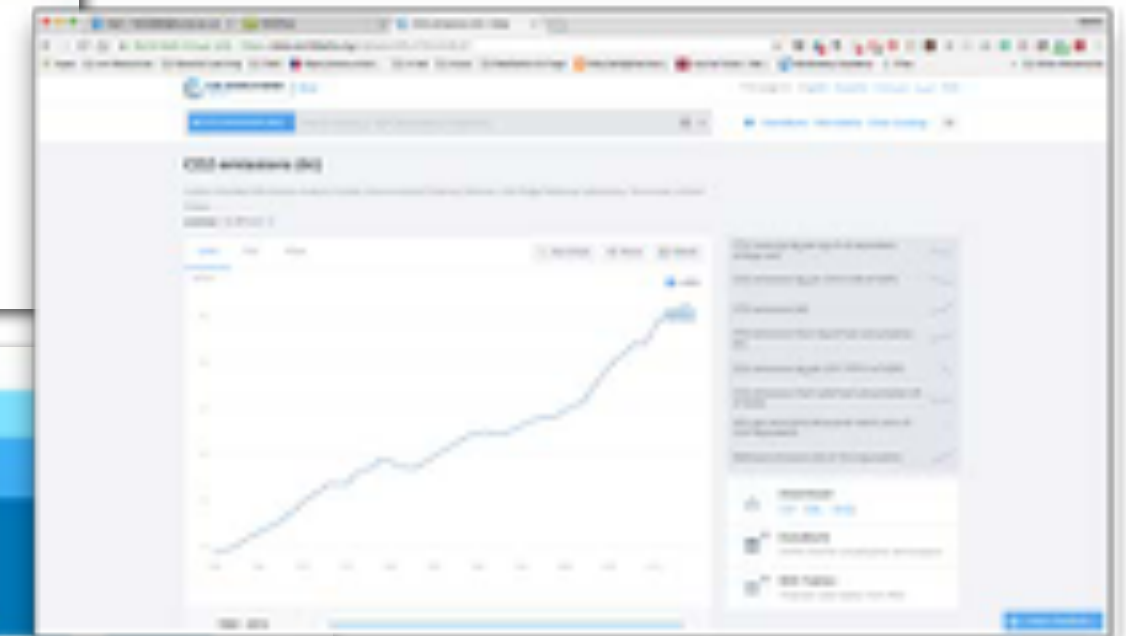
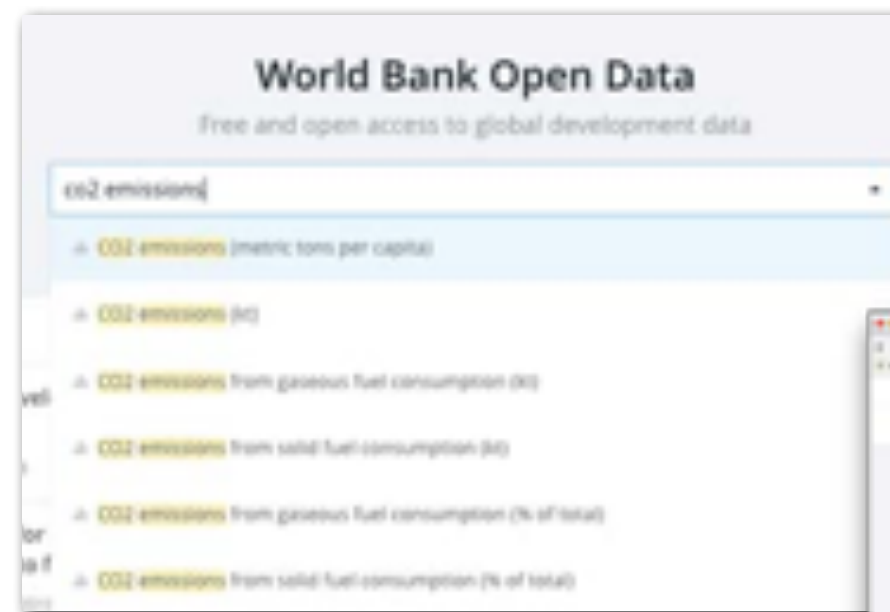
jQuery

I also included the jQuery library, to make selecting and referencing elements easier during production.

Sourcing data



I was familiar with the world data bank website from part 1 of the Assignment where I used it to find stats on co2 emissions and other sustainability issues. For this assignment I searched various compiled lists of data and produced a list of data I wanted to present. There is a lot of data to choose from so I decided to choose data connected to global warming. I then downloaded all the data to my computer in excel format to study and prepare for displaying in my visualisation.



Starting point

As a starting point for my artefact, I decided to use this globe projection by jorin.me and build my required functionality on top of it, changing to meet my needs as required. I first spend some time understanding how it worked before making adjustments.

Country Geometry

The globe loads a list of country names aswell as the topology of the countries from a JSON file. The files share an ID identifier which associates the country name with the coordinates on the globe of that country.



<https://jorin.me/d3-canvas-globe-hover/>

Rendering globe

The globe was rendered like in this example the taking countries coordinates from a json file. I could then link the globe to local data files to allow me to use the same globe topology with my own data attached to each country.

I also made sure to understand all the functions in the code so they could be rewritten to my requirements. I figured out how to change the colour of each country as well as control the animation and rotation controls.

I could also update the render function to give each country a fill based on the data it was displaying. Adding my own interface ontop of the existing globe and changing the colours and styles would give me my own custom made visualisation of a unique set of data.

Preparing data

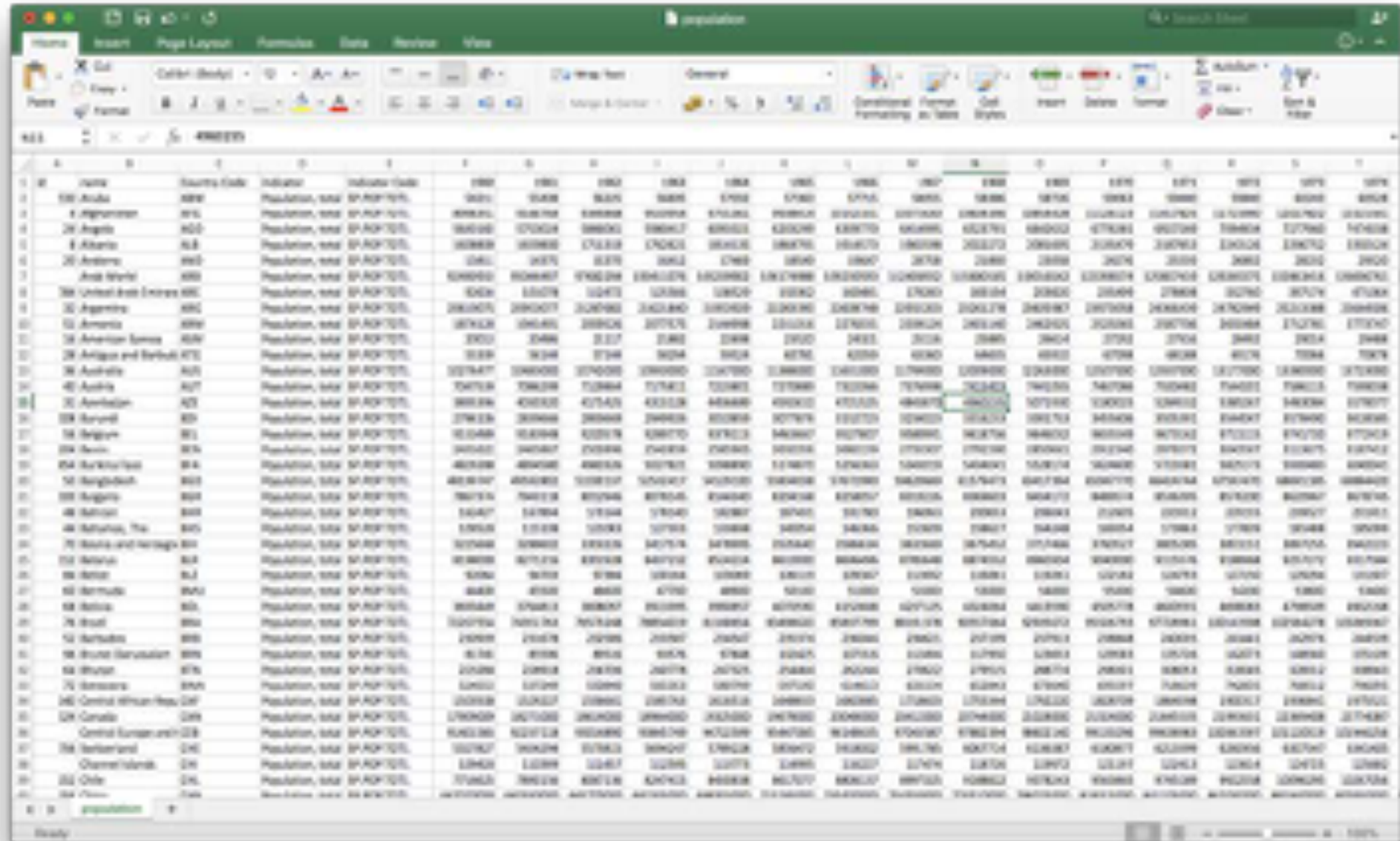


Microsoft Excel

Data was available in seperate spreadsheets for each set of data eg population, co2 emissions. In order to be able to map the data to the countries on the globe, each country was given a unique ID number, corresponding to the geometry coordinates for the shape of each country in another file used to create the globe.

This means that when a user hovers over a country, the code can first calculate which country the user is hovered on from the mouse position, then display the data for that country.

File format



To make the data readable by d3, I decided to save it as tab delimited text, which could then be interpreted by d3. This meant that each column and row value is stored in a text file with a tab between each value.

Loading data

When the page loads, the initial dataset for “population” is loaded. Each subsequent dataset is then loaded when clicked. I created a function for each set of data loads the file with the corresponding data.

```
function loadData(cb) {
  d3.json('110m.json', function(error, world) {
    if (error) throw error
    d3.tsv("co2_metric_tons_per_capita.txt", function(error, countries) {
      if (error) throw error
      cb(world, countries)
    })
  })
}
```

Displaying data on hover

The first requirement of my visualisation is that it displays the data for the selected year and country when a user hovers over a country on the globe. To achieve this, I used the functionality from the d3 example to calculate which country the user was hovered over and display the data from the prepared data file for the selected year.

To give some visual feedback, I added a stroke around the country the user is hovered over to show it is selected, as well as displaying the country name above the data.



Select data to display

Click one of the options to change the displayed data

- ☒ Population, total
- ☐ CO2 emissions (metric tons per capita)
- ☐ CO2 emissions (kt)
- ☐ Fossil fuel energy consumption (% of total)
- ☐ Electric power consumption (kWh per capita)
- ☐ Agricultural methane emissions (CO2 equivalent)
- ☐ Total greenhouse gas emissions (kt of CO2 equivalent)
- ☐ Forest area (% of land area)

Selecting year

Select year

2008

Use the slider to watch the globe update through the years

```
//set the year the stats are showing  
var year = $("#year").val();
```



Australia
Population, total
10742000

Select year

1962

Use the slider to watch the globe update through the years



110m.json

	A	B	C	D	E	F	G	H	I
1	id	name	Country Code	Indicator	Indicator Code	1960	1965	1962	1963
2	519	Aruba	ABW	Population, total	SP.POP.TOTL	54211	55438	56225	56695
3	4	Afghanistan	AFG	Population, total	SP.POP.TOTL	8996351	9166764	9345868	9533954
4	24	Angola	AGO	Population, total	SP.POP.TOTL	5643182	5753024	5868061	5980417
5	8	Albania	ALB	Population, total	SP.POP.TOTL	308800	3658800	1713319	1762621
6	20	Andorra	AND	Population, total	SP.POP.TOTL	3411	14375	15370	16412
7		Arab World	ARB	Population, total	SP.POP.TOTL	93495302	95044497	97682294	100613376
8	784	United Arab Emirates	ARE	Population, total	SP.POP.TOTL	90634	100378	112472	125566
9	32	Argentina	ARG	Population, total	SP.POP.TOTL	10619075	10953077	11283682	11623840
10	51	Armenia	ARM	Population, total	SP.POP.TOTL	5874120	5874495	6009526	6077575
11	36	American Samoa	ASM	Population, total	SP.POP.TOTL	28013	28086	21117	23882
12	28	Antigua and Barbuda	ATG	Population, total	SP.POP.TOTL	55339	56144	57144	58294
13	36	Australia	AUS	Population, total	SP.POP.TOTL	10276477	10483000	10742000	10950000
14	40	Austria	AUT	Population, total	SP.POP.TOTL	3047539	3086299	3129864	3175811
15	31	Azerbaijan	AZE	Population, total	SP.POP.TOTL	3895396	4030320	4179425	4315128
16	128	Burundi	BDI	Population, total	SP.POP.TOTL	2786108	2839668	2893649	2949926
17	56	Belgium	BEL	Population, total	SP.POP.TOTL	9153489	9183948	9220578	9289770

```
current.text(country && country.name || '')  
//if there is data for the year, show it, otherwise show no data message  
if (country[year]){  
  currentStats.text(country.indicator + country && country.indicator || '')  
  current2.text(country.indicator + country && country[year] || '')  
}  
else {  
  currentStats.text(country.indicator + country && country.indicator || '')  
  current2.text(country.indicator + country && "No data available for this year" || '')  
}
```

Grouping & shading

In order to give a visual representation of the data, I needed to group each data set into 5 bands specific to the data set. These are used to shade the countries on the map different colours depending if the data for that country is low/ high compared to the global average.



population

Group 1

Group 2

Group 3

Group 4

Group 5

No data



110m.json

Population (total)	
	< 100k
	100k - 1Million
	1Million - 10Million
	10Million - 100Million
	>= 100Million
	No data



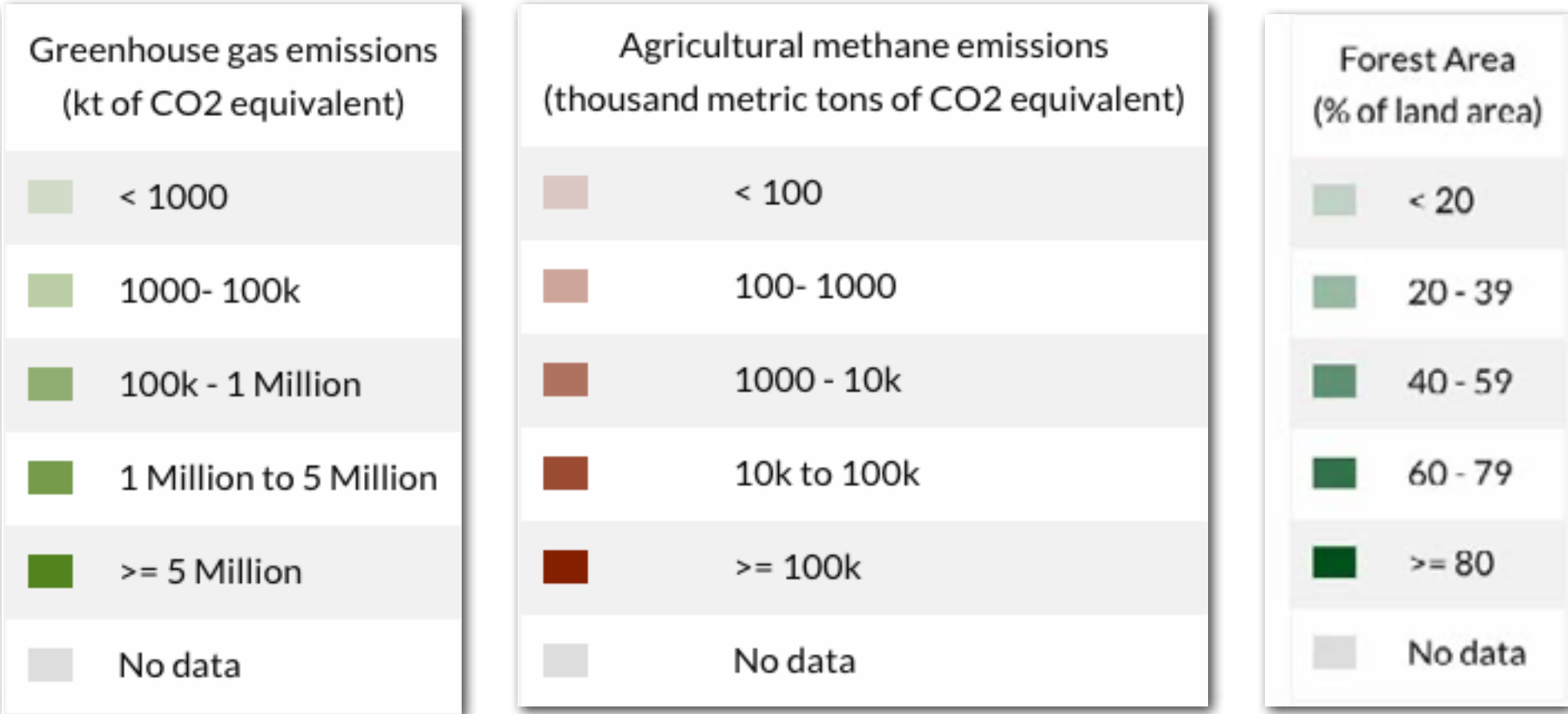
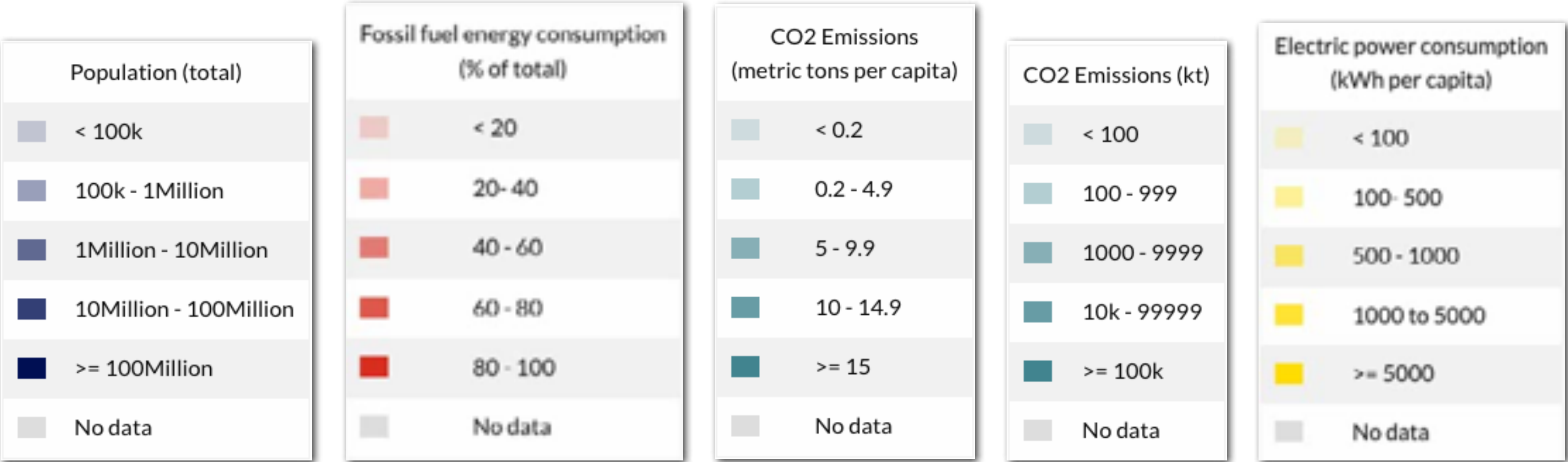
When data is loaded it is split into 6 groupings from low to high, with another for countries with no data. These groups are unique to the specific data and determine which countries to fill which colour. When the globe renders, it fills each country based on which group the country is in for that year. The back end is a little more complex than this, but this gives a general understanding of the functionality.

```
function makeGroups(){
  if (indicator === "CO2 emissions (metric tons per capital)") {
    nodata = countryList.filter(function(d) { return d[year] === "" });
    highestEmissions = countryList.filter(function(d) { return d[year] >= 35 });
    highEmissions = countryList.filter(function(d) { return d[year] < 35 && d[year] >= 10 });
    midEmissions = countryList.filter(function(d) { return d[year] < 10 && d[year] >= 5 });
    lowEmissions = countryList.filter(function(d) { return d[year] < 5 && d[year] >= 0.2 });
    lowestEmissions = countryList.filter(function(d) { return d[year] < 0.2 && d[year] > 0 });

  } else if (indicator === "CO2 emissions (kt)") {
    nodata = countryList.filter(function(d) { return d[year] === "" });
    highestEmissions = countryList.filter(function(d) { return d[year] >= 100000 });
    highEmissions = countryList.filter(function(d) { return d[year] < 100000 && d[year] >= 10000 });
    midEmissions = countryList.filter(function(d) { return d[year] < 10000 && d[year] >= 1000 });
    lowEmissions = countryList.filter(function(d) { return d[year] < 1000 && d[year] >= 100 });
    lowestEmissions = countryList.filter(function(d) { return d[year] < 100 && d[year] > 0 });
  }
}
```

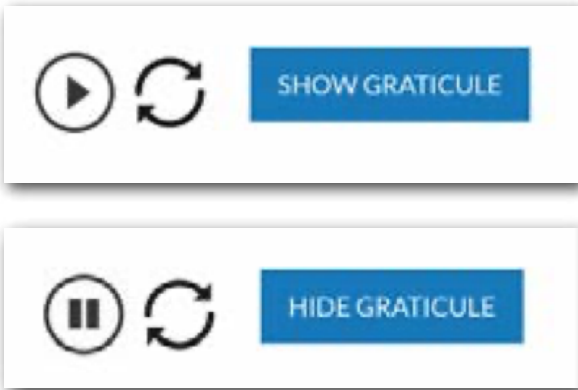
```
//for each of the countries with no data
nodata.forEach(function(d) {
  //store the id of the country
  var str = d.id;
  //update variable to padded version
  str = pad(str, 3);
  //create a new variable for countries in feature list whose id match the emission range
  nod = countries.features.find(function(c) {
    return c.id === str
  });
  //push each value to an array
  nod2.push(nod);
});
```

Each set of data was given its own set of groups, depending on the scale and range of the data. The easiest way to visualise this is with the legends I created depicting what data each shade of colour represents. Percentages are split into equal bands while other groups are calculated based on averages in the datasets. I chose different colours for each statistic and tried to choose fitting colours where appropriate and possible.



Globe controls

I then created controls for the globe to play/ pause the rotation, reset the globe and show/ hide the graticule. I used jQuery to toggle to buttons display and handle the click events.



4 Final artefact

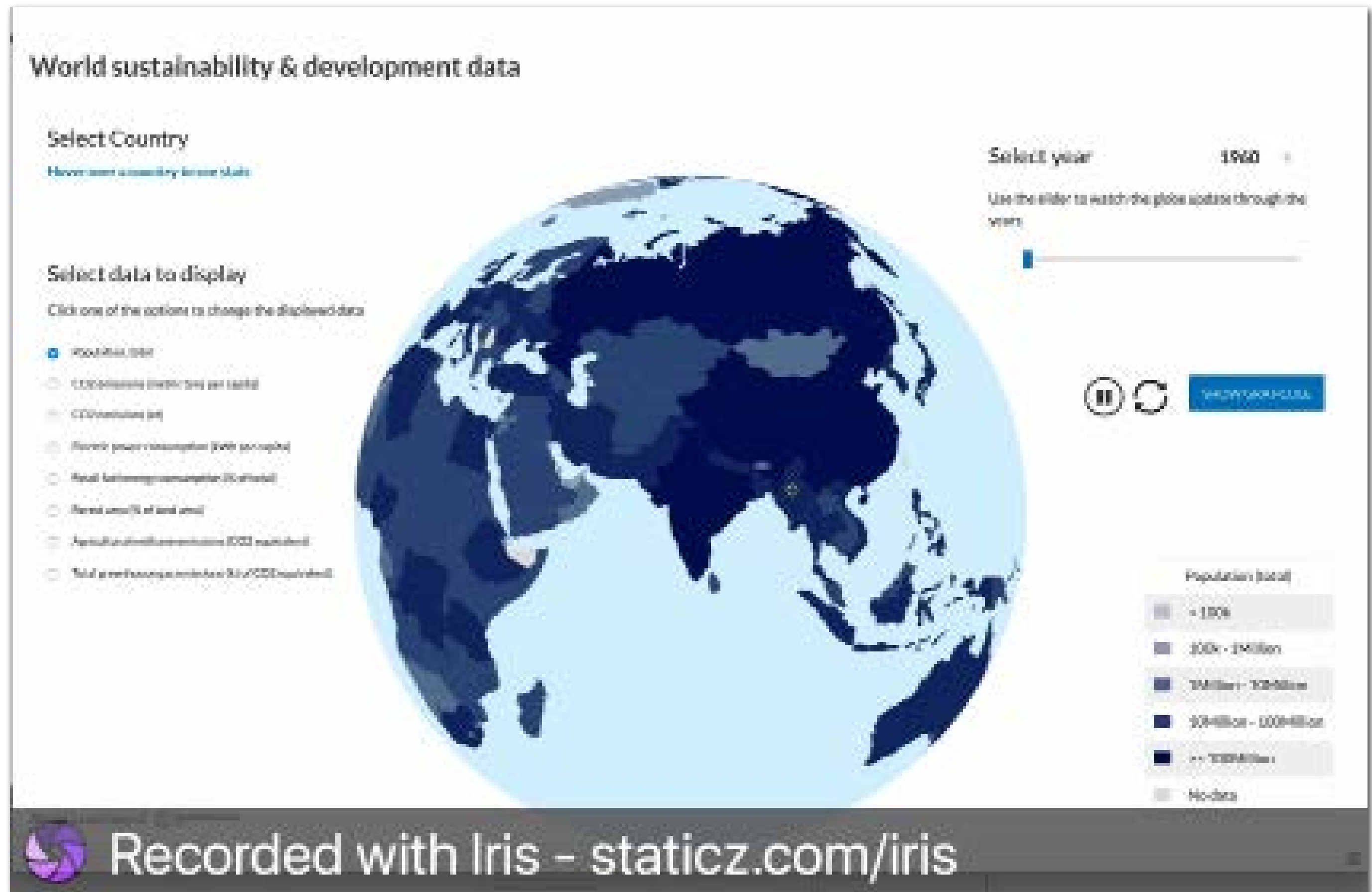
Description- The final artefact is an interactive globe visualisation which shows data for each country for each year from 1960-2014. The data is coloured in bands for each country, which change colour depending on the data you are looking at. The year can be changed with the year slider on the right to see the globe update and the actual figures for each country can be seen by hovering over a country. Finally It displays a legend for each data set and controls to rotate the globe and add a graticule.

Screenshots



Video of functionality

If this video doesn't load, please update Flash or watch the functionality from the video in the submission folder called "GLOBE_DEMO.mov".



Visualising data

Here is an example of how the application can be used to visualise data. In this example, you can see an increase in population as countries become darker over time as the population increases:

Select data to display

Click one of the options to change the displayed data

- ☒ Population, total
- ☐ CO2 emissions (metric tons per capita)
- ☐ CO2 emissions (kt)
- ☐ Fossil fuel energy consumption (% of total)
- ☐ Electric power consumption (\$/kWh per capita)
- ☐ Agricultural methane emissions (CO2 equivalent)
- ☐ Total greenhouse gas emissions (kt of CO2 equivalent)
- ☐ Forest area (% of land area)

Population (total)

- ☐ < 100k
- ☐ 100k - 1M
- ☐ 1M - 10M
- ☐ 10M - 100M
- ☐ >= 100M
- ☐ No data

Select year 1960

Use the slider to watch the globe update through the years



Select year 2014

Use the slider to watch the globe update through the years



World sustainability & development data

Chad

Population, total

3001593

World sustainability & development data

Chad

Population, total

13569438



5 Marketing

Target Audience

The target audience for this application is very large, as it would be useful for anyone to know more about climate change and see for themselves the impact humans are having on the globe. In particular, younger more tech literate users may benefit from the artefact more, as well as institutes such as universities or libraries which could include adapted versions of the artefact on their websites or on displays in their buildings.

Platforms

Dedicated web host

Downloadable asset

Responsive version- app?

Strategy

The marketing strategy for the artefact would include having it hosted on a dedicated web server and allowing access to the public. The site could then be shared on social media and hopefully become a kind of viral asset, being shared by various climate change and sustainability social media pages.

In awareness to the context of the artefact, I would also ensure that the access is free, and offer downloadable versions which do not require internet access. These same versions could be put on interactive displays at corporate events, at libraries and museums and other learning institutions.



Locations

Interactive displays



Conferences/ Museums/ Libraries/ Uni's



6 Evaluation

Analytical evaluation of creative journey and learning outcomes.

Creative journey

This project has been a great opportunity to explore a topic that I found particularly inspiring during my module lectures. Initially I was a little overwhelmed by the choice of topics but by tying it in with the first half of the project I was able to focus my efforts towards an asset which accompanies and adds to an initial project.

I initially looked at existing examples of data visualisation, getting inspiration as to how I could present my own data. I also spent time analytically thinking about what data to present that would have the best visual impact and how to present this data. This initial process was challenging but opened me up to a new world of technology to explore.

The journey was then largely a learning process, trying to search for and understand methods to create the globe. I eventually created a plan and understood the functionality required so was able to set about actually creating the artefact in a fairly organised and structured way. I initially made the globe, then prepared the data to the required format I had prepared, loaded the data and built the interface to control what is displayed.

I was very open initially to what my artefact would be, so it allowed me to have a broad creative vision and get a lot of inspiration from looking at lots of examples. From initially planning a graph or chart, to creating a fully interactive asset, my creative journey was influenced heavily by what I learnt from other examples.

Learning outcomes

I have learned a lot in this project, choosing a topic I knew little about. I was eager to learn and this helped me to pick up a lot and create something above my initial expectations. Initially, I planned to create some kind of 2D posted style animation, but as I looked at interactive examples and began to understand their functionality, I decided to be more ambitious and create my own interactive asset.

I began the project with a basic understanding and knowledge of JavaScript but had never worked with data in this way. My experience of using the world data bank website in part of this assignment meant I knew I could source data and thought it would be a great introduction to learn how to use the d3 library to visualise it.

The initial part of the project was the most difficult with a steep learning curve to figure out how some existing examples worked. This gave me a lot of insight and understanding as to different methods that can be used, and some of the core functionality of the d3 library. Soon I had replicated tutorials and understood the functions well enough to create my own working additions.

I also learnt some additional applications of the foundation library that I used to create the slider. Discovering that I could bind the slider to the date field was a very cool trick that I can use in other projects as a user interface.

I am very pleased that I was able to tie in creating an interface to users to interact with in my data visualisation as this is something I am passionate about and think it makes compliments a visualisation being able to interact with it. Learning the skills to achieve this was very valuable and will be useful in my portfolio of skills for future undertakings.

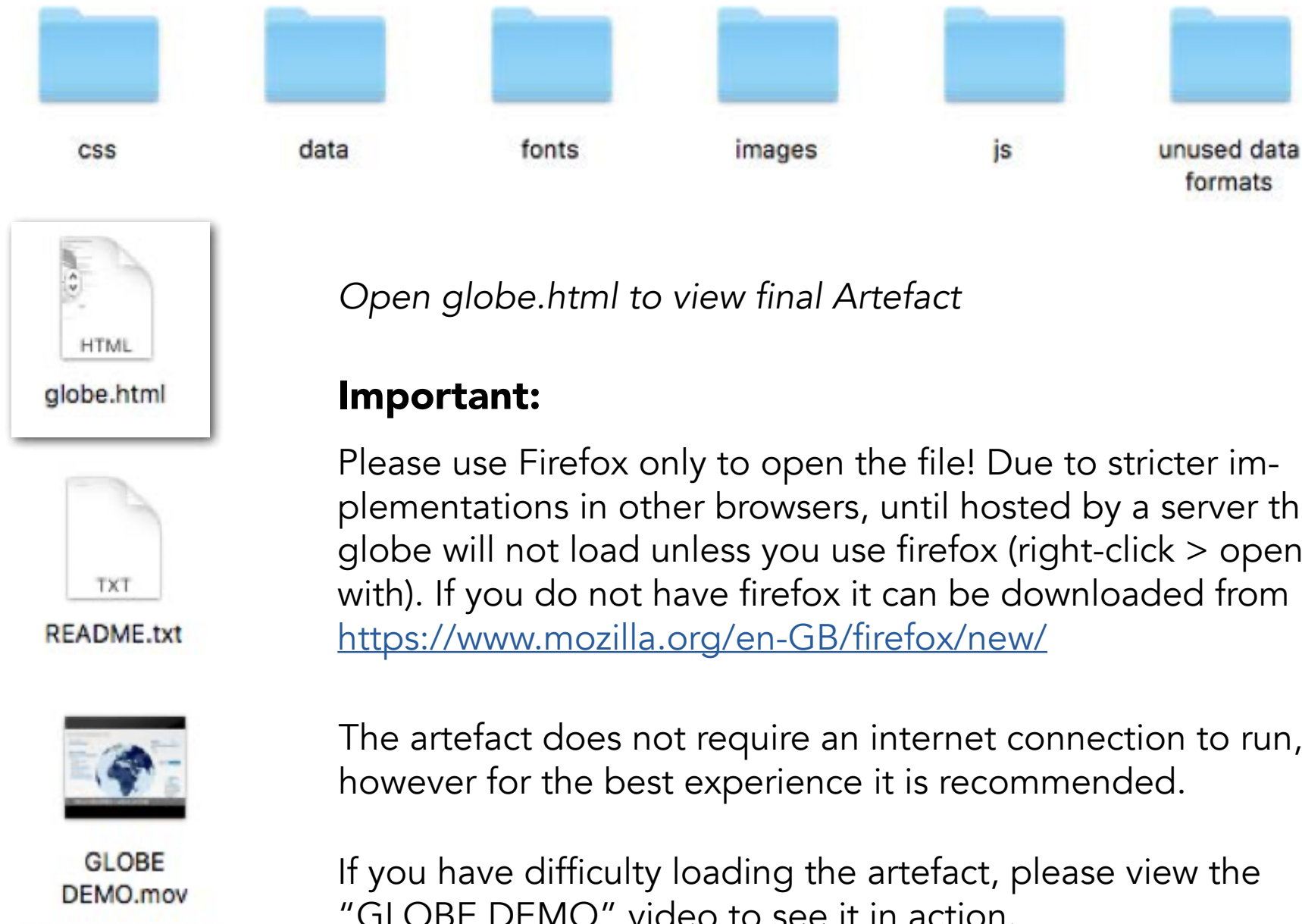
Critical review

Since climate change is a global phenomenon that has escalated over time, I decided it would be the most interesting data to visualise on a globe, to see each country's contribution and also witness the contribution over time. There was so much data available I found it difficult to choose what to include. It could have been better to include more specific data sets, or even a broader range allowing users to choose from even more options.

Having tested the final artefact, I am fairly happy with the results and feel the requirements are all met. However, it could certainly be improved. It would be nice to add the ability to keep data for a certain country on screen while you update through the years, rather than having to update it and hover back over the country.

I would also like to spend more time on the colour choices, ensuring a professional standard across all the data views.

7 Assets



Conrolling the globe:

