MONASH University

MONASH
INFORMATION
TECHNOLOGY

FIT3179 Data Visualisation

Week 09: Visualisation tools

GROUP
OF EIGHT
AUSTRALIA

MONASH University

Visualisation tools overview

Visualisation tools

In this unit we're mostly designing using *Tableau*, but it isn't the only choice available.

Typically as a visualiser you want to pick the most appropriate tool for the job.

- Since you know *Tableau*, it will probably be your first choice
- If you want to do visualisation as a career, there are other industry-standard approaches

Visualisation tools for every step

In creating a visualisation, the available tools can be roughly grouped into three main categories

- Data collection, checking and reformatting
- Data analysis and design of the basic visualisation
- Final tweaking of the design

Each of these stages have different tools. Additionally, different types of visualisations have different tools that are best for the job.

Main Tools

	Free Tools	Paid Tools
Data Preparation	Python, PHP, Mr Data Converter, Mr People	
Analysis & Visualisation	Tableau Public, IBM Many Eyes, Raw, Dygraphs, Wolfram Alpha, jqPlot	Excel, Tableau, Spotfire, Qlikview, MS BI Stack
Interactive Visualisation Programming	D3.js, Vega and Vega-Lite (https://vega.github.io) R & Shiny, Google Charts, Chart.js, Zinggraph, InstantAtlas	
Vector graphics editor	Inkscape	Illustrator, Powerpoint
GIS, web maps	QGIS (desktop GIS), Leaflet (web map)	ArcGIS and ArcMap by Esri

This is not an exhaustive list! There are lots of different programming libraries (mostly JavaScript based) and software.
You can find an incomplete and outdated list at: https://en.wikipedia.org/wiki/Category:Data_visualization_software

Comparison of JavaScript charting libraries: https://en.wikipedia.org/wiki/Comparison_of_JavaScript_charting_libraries

Visualisation tools

This week we look at alternative visualisation tools.

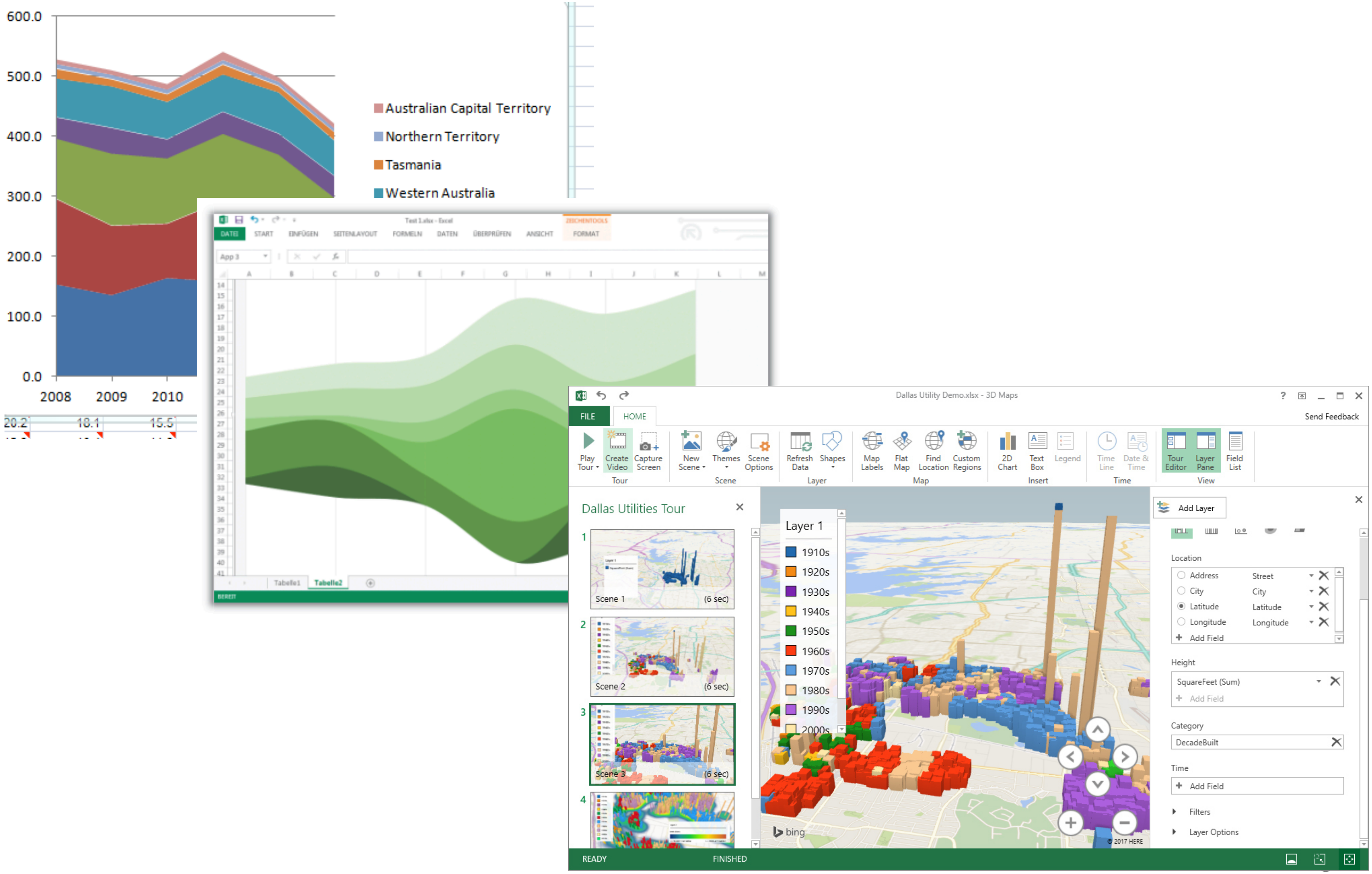
We'll start at the most basic and work our way to the more complicated options.

Excel

We can create visualisations directly in Excel.

- Generally simple visualisation types
- Not many items/options to choose from
- Good enough for simple visualisations though!

Excel demonstration

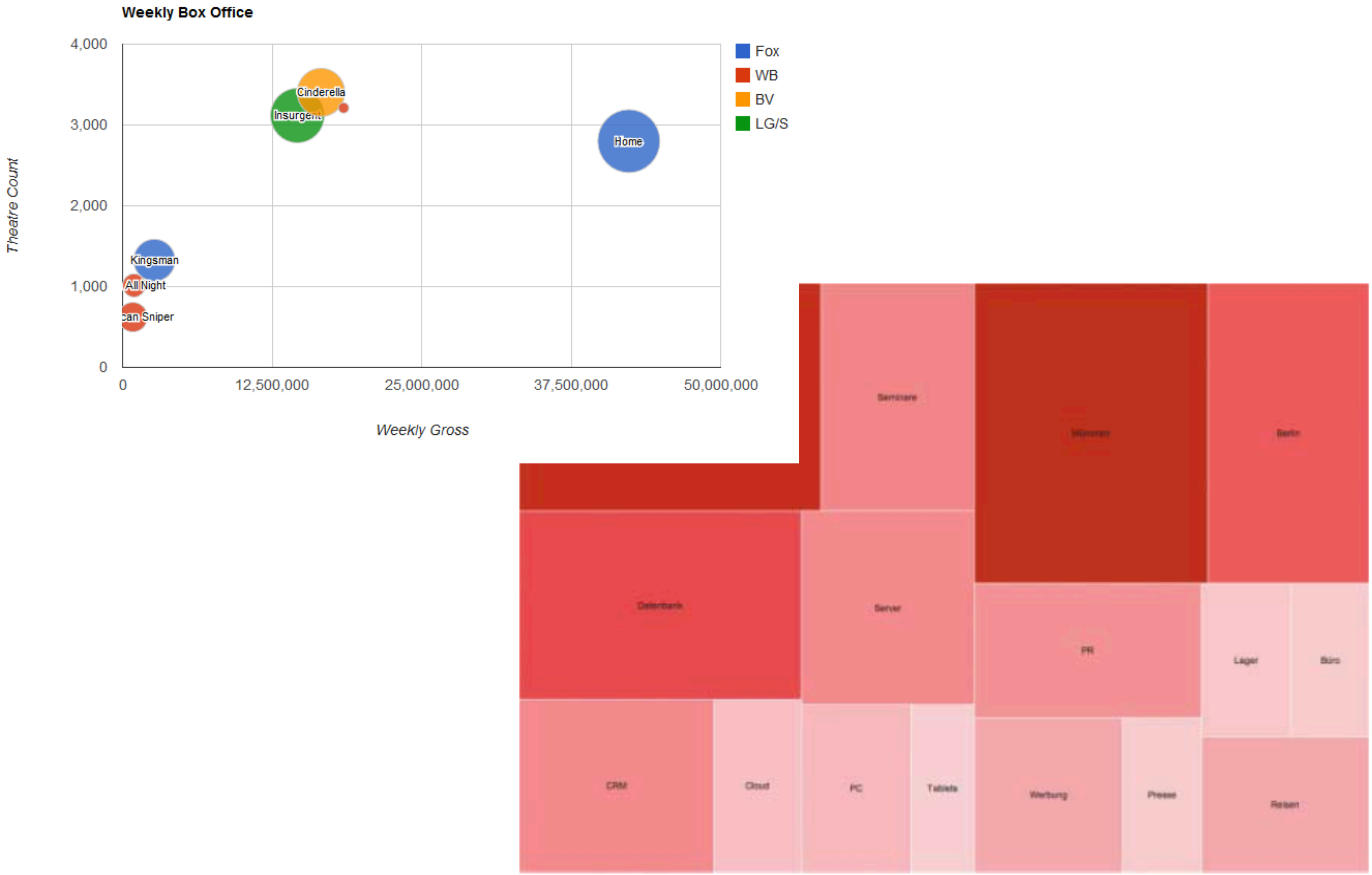


Google Charts

Similar to Excel, we can make visualisation charts using data in a Google spreadsheet

- Generally very simple visualisation types as well
- The 'basic' chart is almost the same as Excel
- BUT, we can also use *JavaScript* and Google chart libraries to create more interactive and better versions on separate webpages
  - This gives us the benefit of having a visualisation where the data is driven dynamically by the spreadsheet
  - If we change the spreadsheet, the visualisation will change (when reloaded)

Google Charts

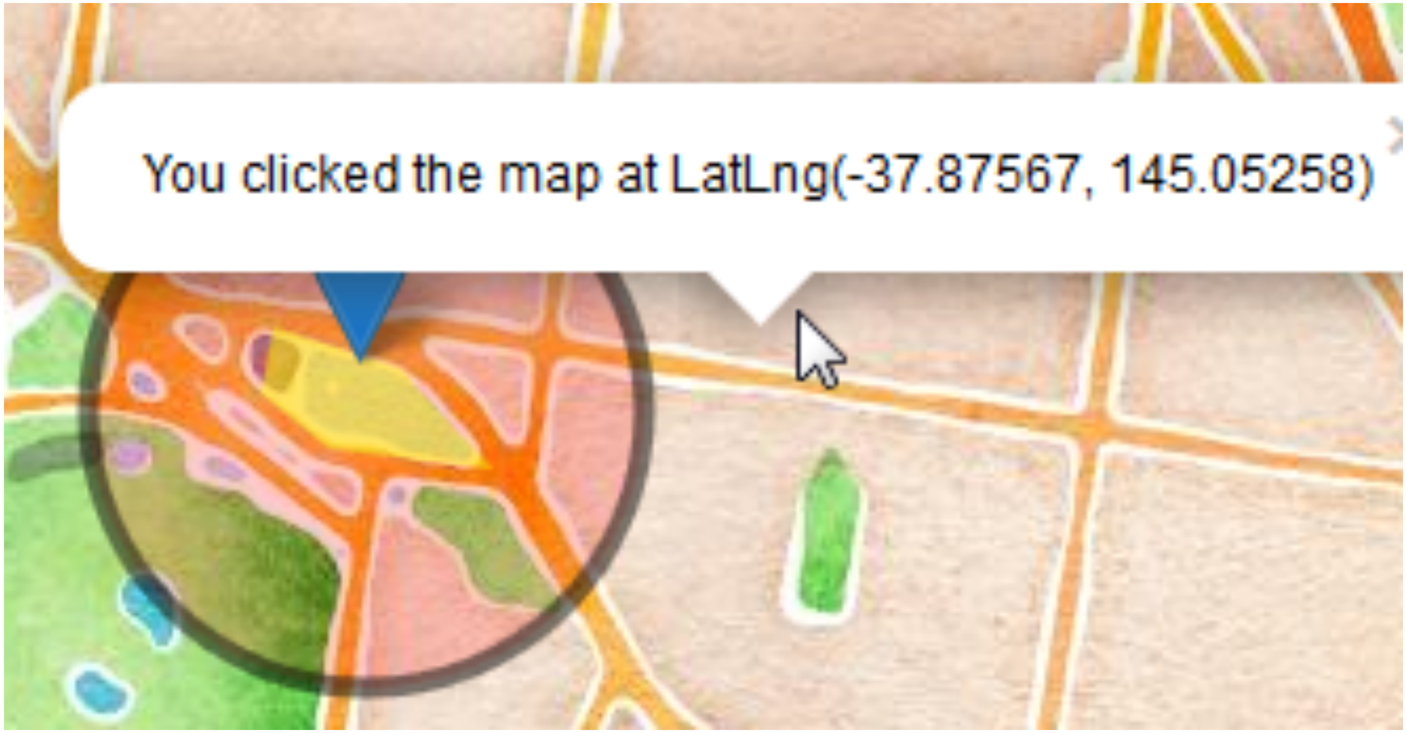


Leaflet: Maps

Similar to programming with *JavaScript* and Google charts, *Leaflet* is a JavaScript library we can use to program HTML pages to show visualisations.

- Only for **maps**
- Is a fairly common library to use (smaller and more flexible than the Google maps API)
- http://leafletjs.com
- Side note: https://www.openstreetmap.org and iD editor

Leaflet demonstration



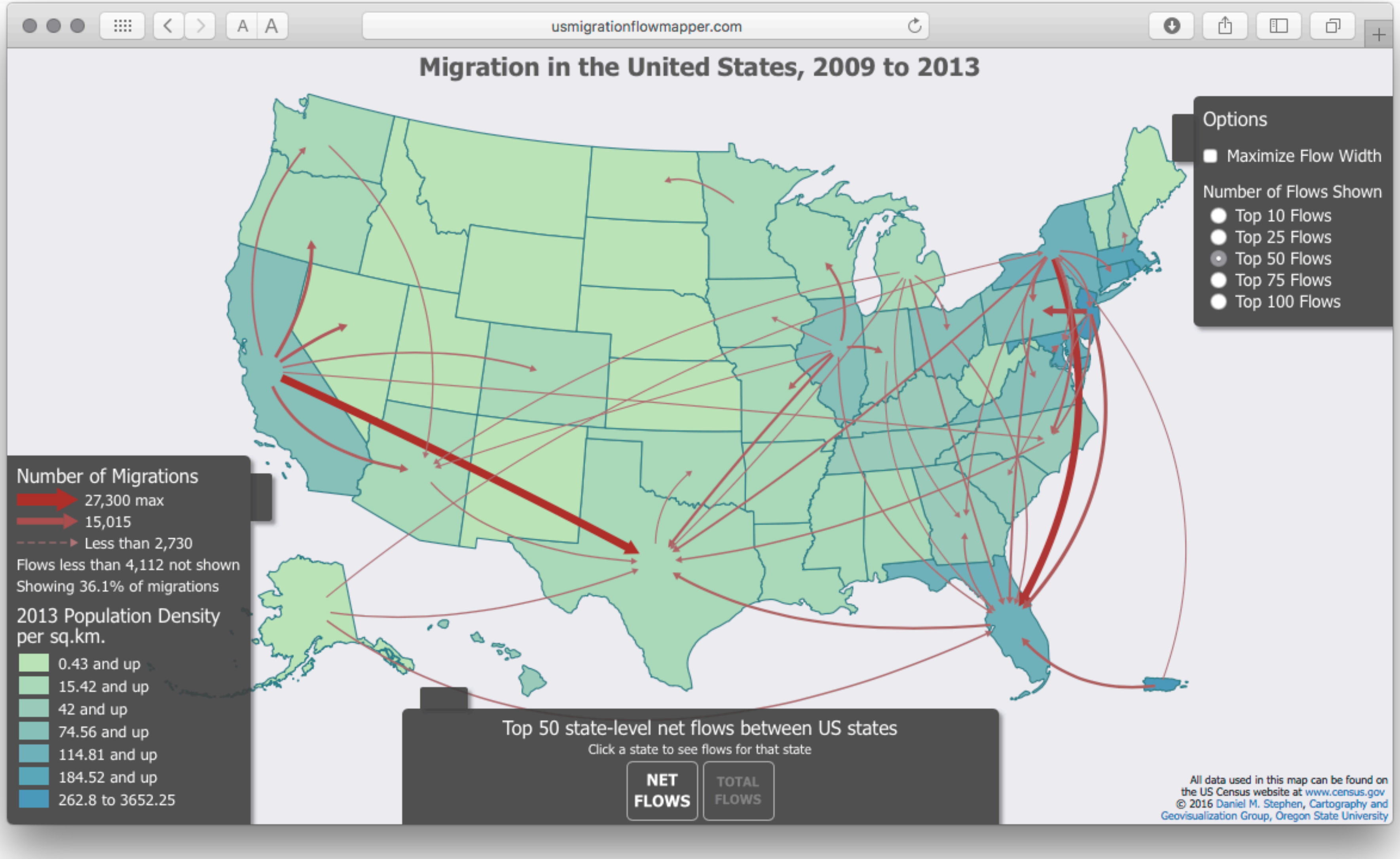
You clicked the map at LatLng(-37.87567, 145.05258)

D3.js

D3.js is a JavaScript library. D3 (Data Driven Documents) is very popular amongst visualisation experts because it is flexible and allows huge control over the final result.

- Low level, not simple, but very powerful.
- Can bind data to the Document Object Model (DOM) and apply transforms to the document based on the data
  - Translation: Allows us to generate or change HTML web pages based on a dataset
- Uses CSS, SVG and JavaScript.
- https://d3js.org
- https://github.com/d3/d3/wiki/Gallery

D3.js Example



Migration in the United States, 2009 to 2013

Options

□ Maximize Flow Width

Number of Flows Shown

○ Top 10 Flows
○ Top 25 Flows
● Top 50 Flows
○ Top 75 Flows
○ Top 100 Flows

Number of Migrations

27,300 max
15,015
Less than 2,730
Flows less than 4,112 not shown
Showing 36.1% of migrations

2013 Population Density per sq.km.

■	0.43 and up
■	15.42 and up
■	42 and up
■	74.56 and up
■	114.81 and up
■	184.52 and up
■	262.8 to 3652.25

Top 50 state-level net flows between US states

Click a state to see flows for that state

All data used in this map can be found on the US Census website at www.census.gov
© 2016 Daniel M. Stephen, Cartography and Geovisualization Group, Oregon State University

Source: *http://usmigrationflowmapper.com*

D3.js Demonstration

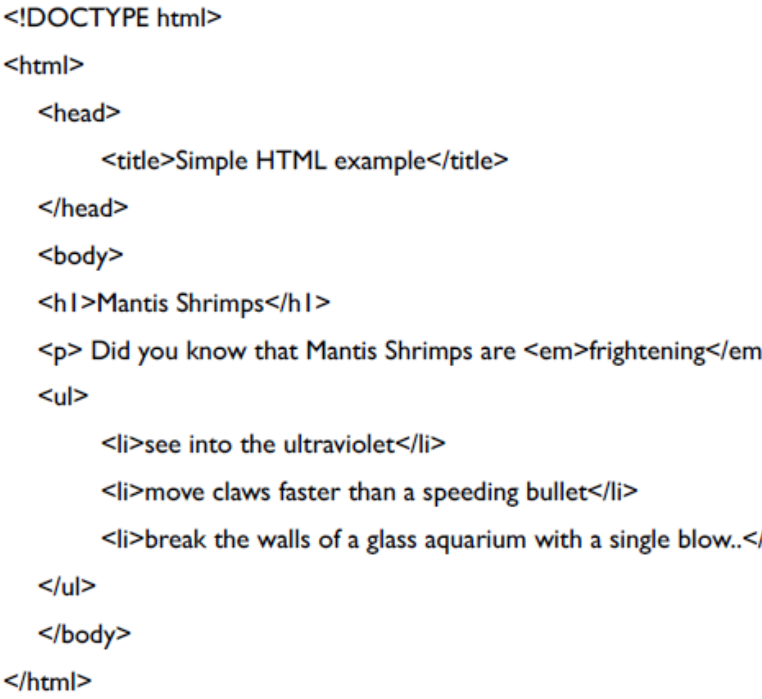
https://d3js.org

A quick primer: HTML

You're probably familiar with some of those terms. Just in case you need a refresher, let's just quickly go over them.

HTML: **H**yper**T**ext **M**arkup **L**anguage

The coding behind webpages

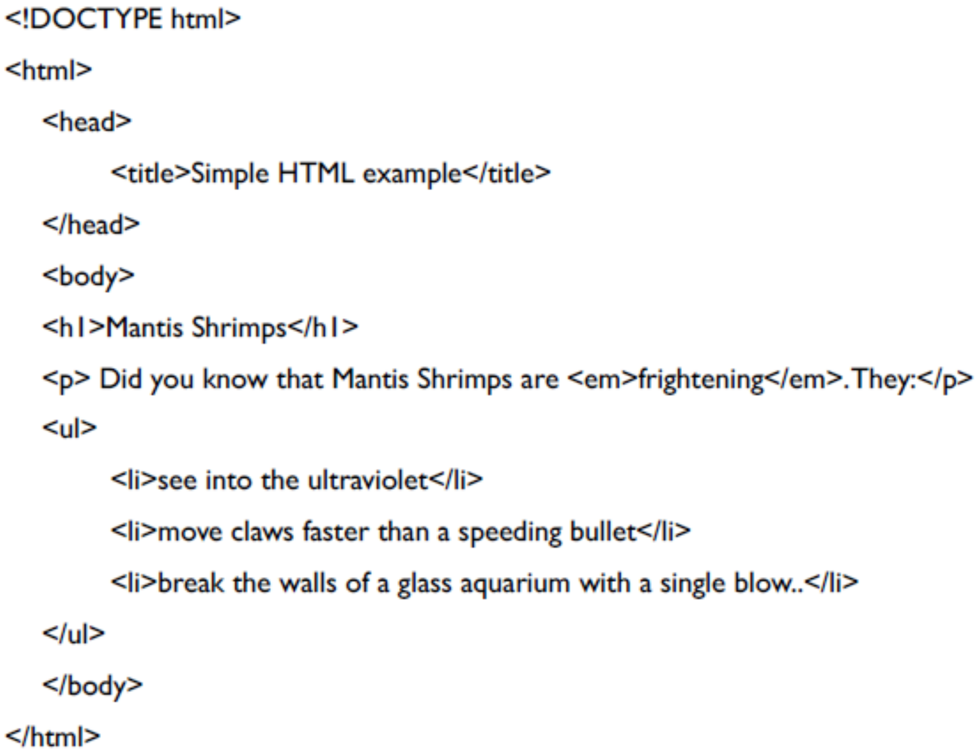


```
<!DOCTYPE html>
<html>
  <head>
      <title>Simple HTML example</title>
  </head>
  <body>
  <h1>Mantis Shrimps</h1>
  <p> Did you know that Mantis Shrimps are <em>frightening</em
  <ul>
      <li>see into the ultraviolet</li>
      <li>move claws faster than a speeding bullet</li>
      <li>break the walls of a glass aquarium with a single blow..</
  </ul>
  </body>
</html>
```

A quick primer: DOM

DOM: **D**ocument **O**bject **M**odel

DOM is the *hierarchical* structure of HTML. We use this to access different parts of the HTML



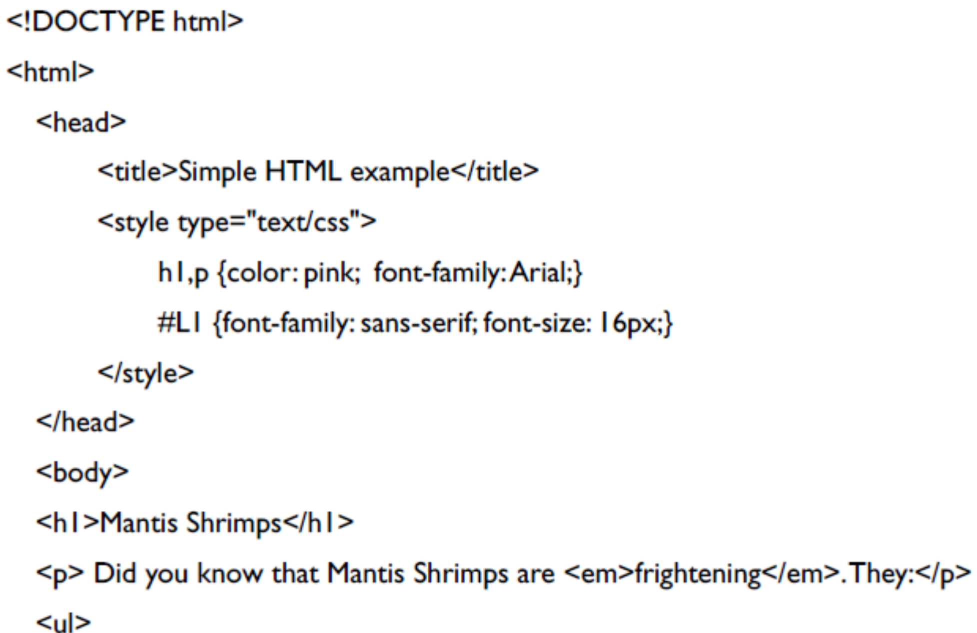
```
<!DOCTYPE html>
<html>
  <head>
      <title>Simple HTML example</title>
  </head>
  <body>
  <h1>Mantis Shrimps</h1>
  <p> Did you know that Mantis Shrimps are <em>frightening</em>.They:</p>
  <ul>
      <li>see into the ultraviolet</li>
      <li>move claws faster than a speeding bullet</li>
      <li>break the walls of a glass aquarium with a single blow..</li>
  </ul>
  </body>
</html>
```

A quick primer: CSS

CSS: **C**ascading **S**tyle **S**heets

HTML is the content, whereas CSS defines the style (size, colour, position, etc.) of the HTML elements

CSS consists of rules made of selectors and properties

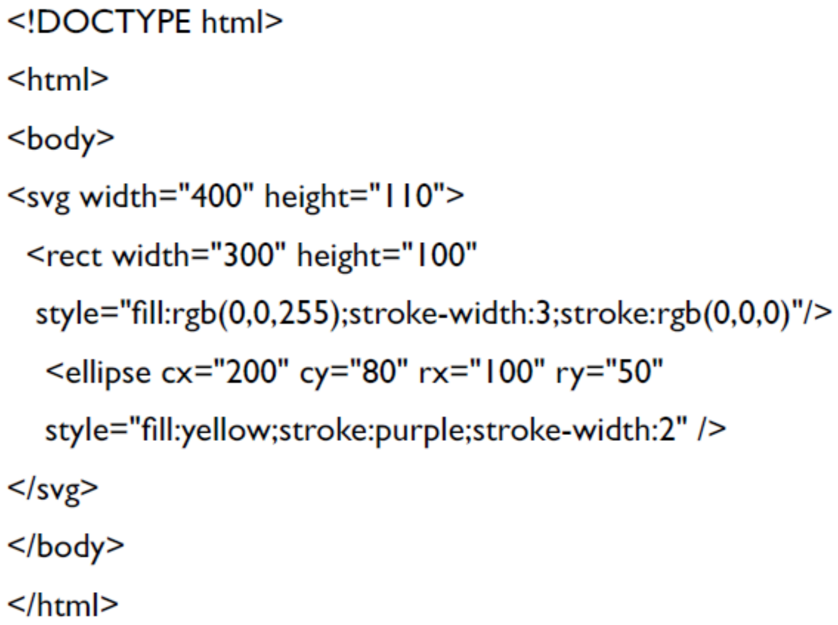


```
<!DOCTYPE html>
<html>
  <head>
      <title>Simple HTML example</title>
      <style type="text/css">
          h1,p {color: pink;  font-family: Arial;}
          #L1 {font-family: sans-serif; font-size: 16px;}
      </style>
  </head>
  <body>
  <h1>Mantis Shrimps</h1>
  <p> Did you know that Mantis Shrimps are <em>frightening</em>.They:</p>
  <ul>
```

A quick primer: SVG

SVG: **S**calable **V**ector **G**raphics

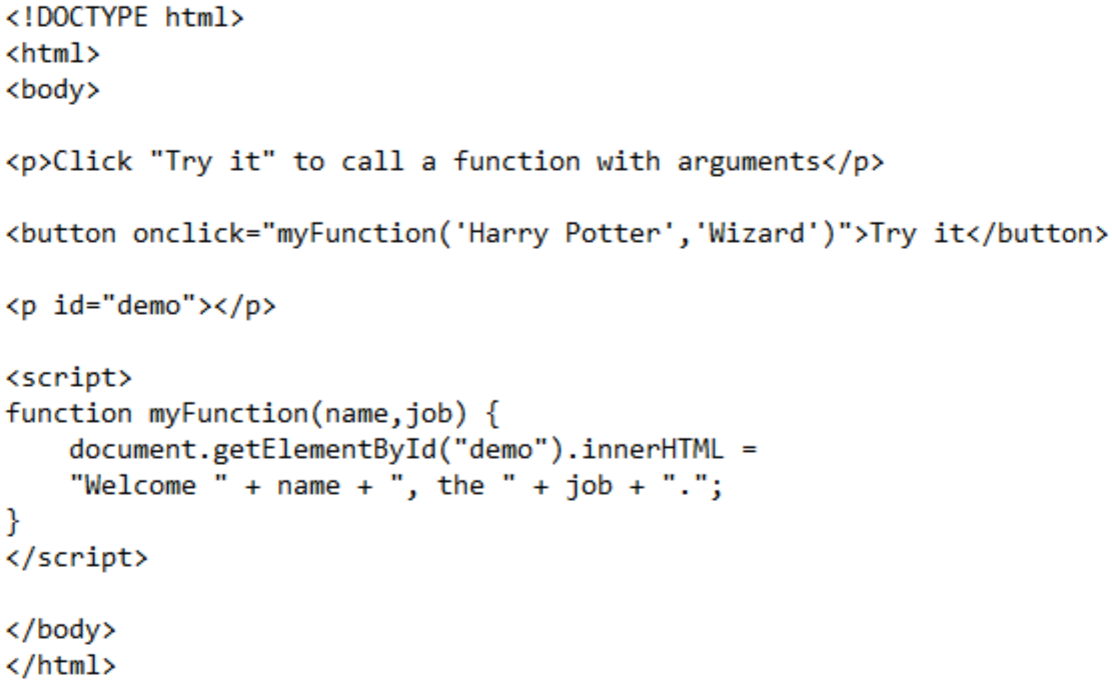
SVG is the web vector graphics format. It is designed to work with both HTML and CSS and can be used to draw graphic primitives (lines, circles, text, ellipses, rectangles, paths).



```
<!DOCTYPE html>
<html>
<body>
<svg width="400" height="110">
 <rect width="300" height="100"
  style="fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)"/>
   <ellipse cx="200" cy="80" rx="100" ry="50"
   style="fill:yellow;stroke:purple;stroke-width:2" />
</svg>
</body>
</html>
```

A quick primer: JavaScript

JavaScript is a scripting language that can make a web page dynamic *(i.e., it runs after it is downloaded by the client)* by modifying the DOM



```
<!DOCTYPE html>
<html>
<body>

<p>Click "Try it" to call a function with arguments</p>

<button onclick="myFunction('Harry Potter','Wizard')">Try it</button>

<p id="demo"></p>

<script>
function myFunction(name,job) {
    document.getElementById("demo").innerHTML =
    "Welcome " + name + ", the " + job + ".";
}
</script>

</body>
</html>
```

D3.js and SVG

Since D3 uses SVG as its primary method of drawing to the screen, we're really only restricted by SVG limitations in our visualisation options

- You can read the huge specs for SVG here https://www.w3.org/TR/SVG/
- Also a nice cheatsheet at http://www.jeromecukier.net/wp-content/uploads/2012/10/d3-cheat-sheet.pdf

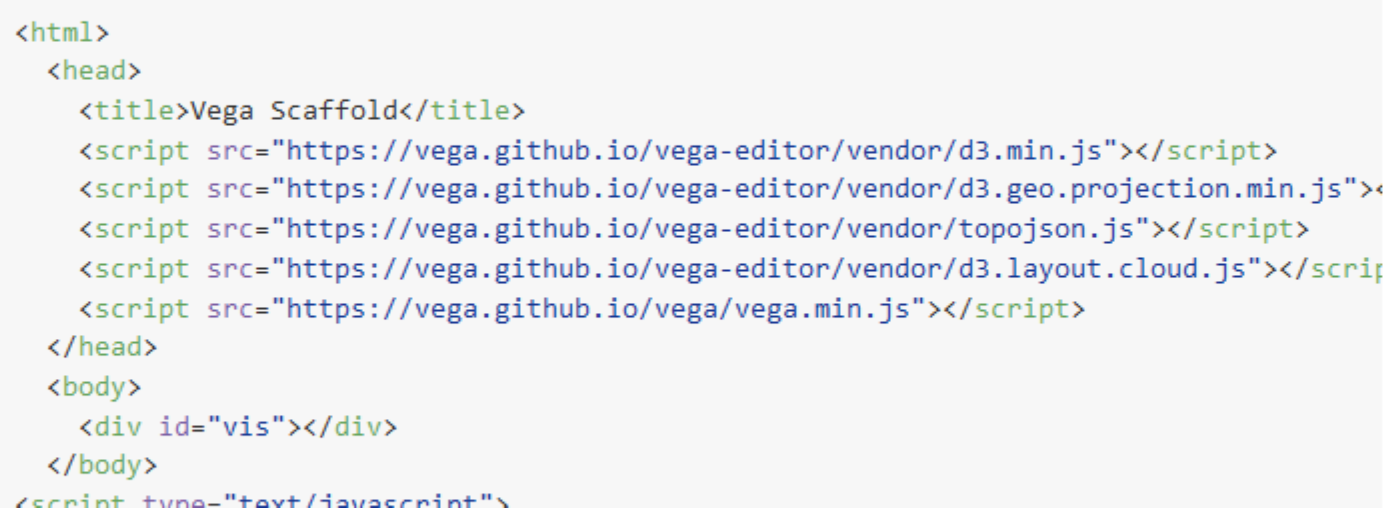
Vega

Vega is a visualisation grammar, with declarative terms for creating visualisations.

- A framework to use D3.js to create the visualisation
- Uses a JSON file to create visualisations
  - You create and share visualisations with the JSON file
  - https://vega.github.io/
- Lacks some of the flexibility and power of D3 but is easier to use.

Vega embedding into HTML

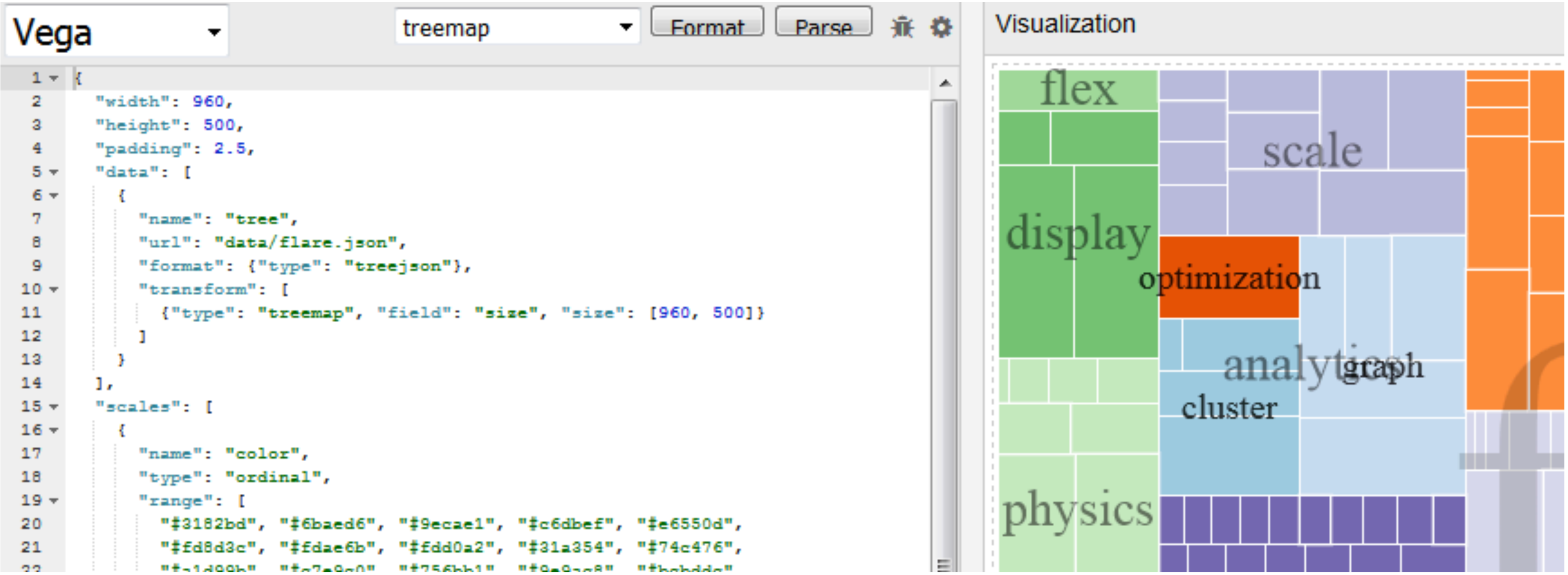
The official Vega HTML embedding document shows how much it relies on a D3 backend



```
<html>
  <head>
    <title>Vega Scaffold</title>
    <script src="https://vega.github.io/vega-editor/vendor/d3.min.js"></script>
    <script src="https://vega.github.io/vega-editor/vendor/d3.geo.projection.min.js"><
    <script src="https://vega.github.io/vega-editor/vendor/topojson.js"></script>
    <script src="https://vega.github.io/vega-editor/vendor/d3.layout.cloud.js"></scrip
    <script src="https://vega.github.io/vega/vega.min.js"></script>
  </head>
  <body>
    <div id="vis"></div>
  </body>
<script type="text/javascript">
```

https://github.com/vega/vega/wiki/Runtime

Vega Example

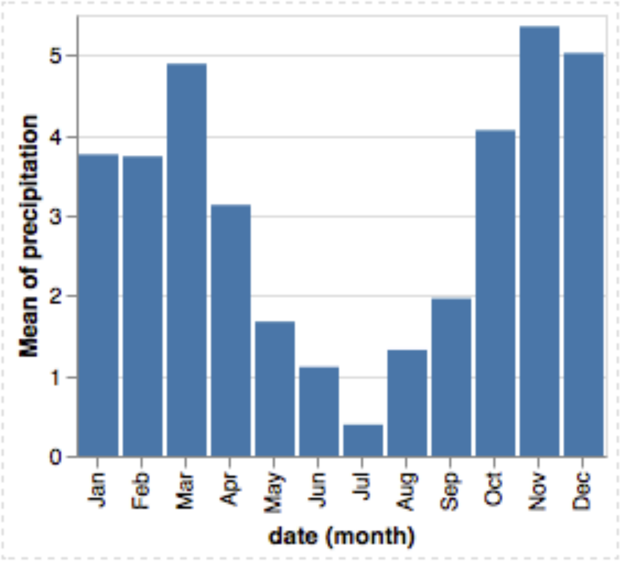


```
{
  "width": 960,
  "height": 500,
  "padding": 2.5,
  "data": [
    {
      "name": "tree",
      "url": "data/flare.json",
      "format": {"type": "treejson"},
      "transform": [
        {"type": "treemap", "field": "size", "size": [960, 500]}
      ]
    }
  ],
  "scales": [
    {
      "name": "color",
      "type": "ordinal",
      "range": [
        "#3182bd", "#6baed6", "#9ecae1", "#c6dbef", "#e6550d",
        "#fd8d3c", "#fdae6b", "#fdd0a2", "#31a354", "#74c476",
        "#a1d99b", "#c7e9c0", "#756bb1", "#9e9ac8", "#bcbddc"
```

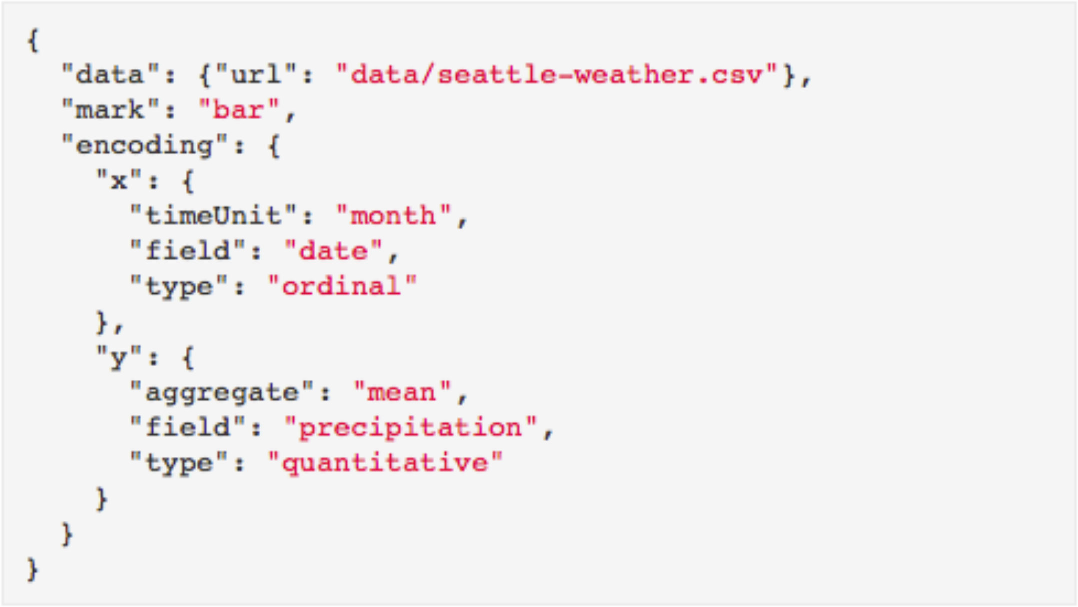
[“Example” at https://vega.github.io/editor/#/](https://vega.github.io/editor/#/)

Vega-Lite

- High-level grammar for interactive graphics
- JSON syntax
- Compiles to Vega
- https://vega.github.io/vega-lite/



Open in Vega Editor



```
{
  "data": {"url": "data/seattle-weather.csv"},
  "mark": "bar",
  "encoding": {
    "x": {
      "timeUnit": "month",
      "field": "date",
      "type": "ordinal"
    },
    "y": {
      "aggregate": "mean",
      "field": "precipitation",
      "type": "quantitative"
    }
  }
}
```

R

R Project for Statistical Computing

Many packages for creating different types of visualisations.

Example: Plotly https://plot.ly/r/