Treemap Visualisations

This exercise aims to be a getting started guide for building interactive Treemap visualisations using the D3 JavaScript library. While data visualisation has existed for many years and the theory is well established. Recent improvements in technology and browser performance has enabled provisioning over the web of interactive visualisations for huge datasets. Treemaps, a rich visualisation for representing numeric data, are especially suited for analysing distribution of resources in large organisations.

Introduction

This exercises is built around the web technologies that, when put together, build powerful and interactive HTML5 web pages. A core part of HTML5 is “action”, specified using the javascript language. While javascript is commonly used for manipulating elements on a web page, more advanced features include graphics and visualisation libraries. D3, one of these libraries, is built especially for multi-dimensional, interactive visualisation of large datasets.

The Dataset – Projects(2)

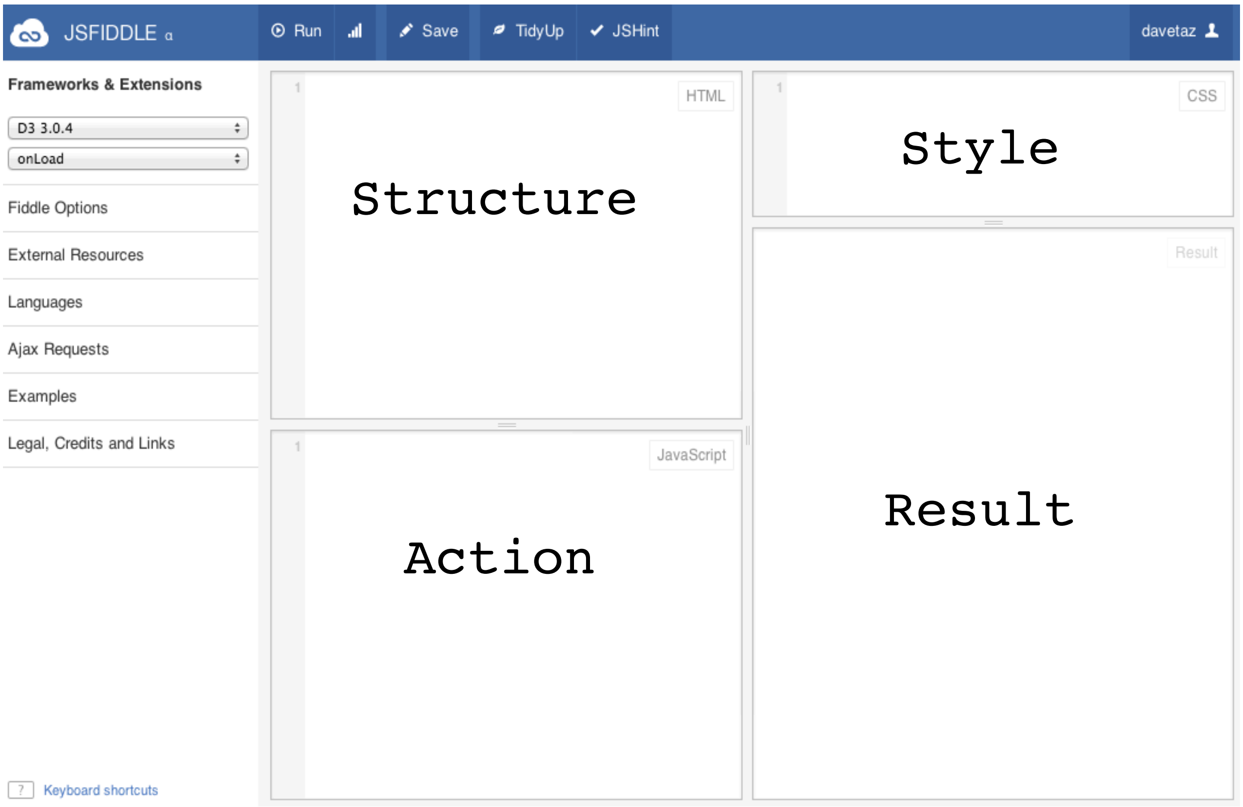
This dataset lists project data available from the US Governments IT Dashboard system at http://www.itdashboard.gov/data_feeds. For the purposes of this exercise only the top 5 most expensive projects in each agency have been selected. The data is available via the training website containing all columns but also filtered to only include the “Project Name” and “Cost” data (CSV and JSON formats).

In order to explore and clean the dataset it is recommended the “Validating and Cleaning Data” exercise precede this one.

jsFiddle.net

To effectively prototype our visualisation, we are going to use the jsFiddle prototyping tool (shown below). jsFiddle is an online sandbox which allows you freely experiment with the building blocks of the web. It has three main panels that allow you to define structure, style and actions related to your web page. The fourth panel (bottom right) shows the result of your editing, this panel is updated by pressing the run button at the top of the screen.

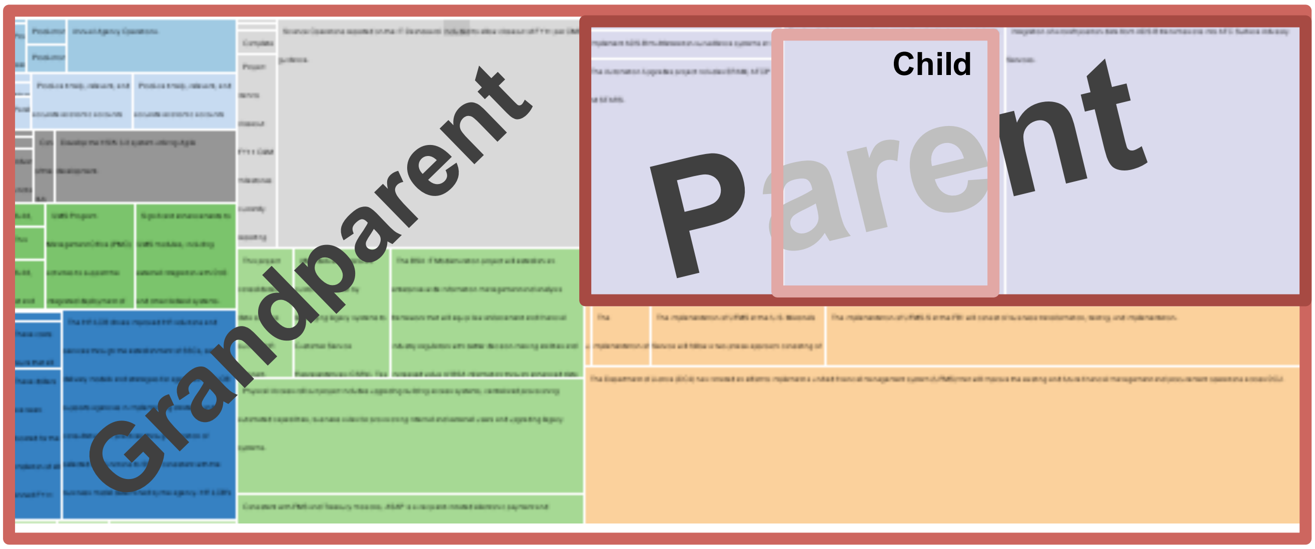
For the purposes of this exercise, please **select D3** from the “Frameworks & Extensions” panel on the left hand side as shown.



As jsFiddle does not provide a data section, throughout this exercise it will be necessary to define our data in the “Action” block.

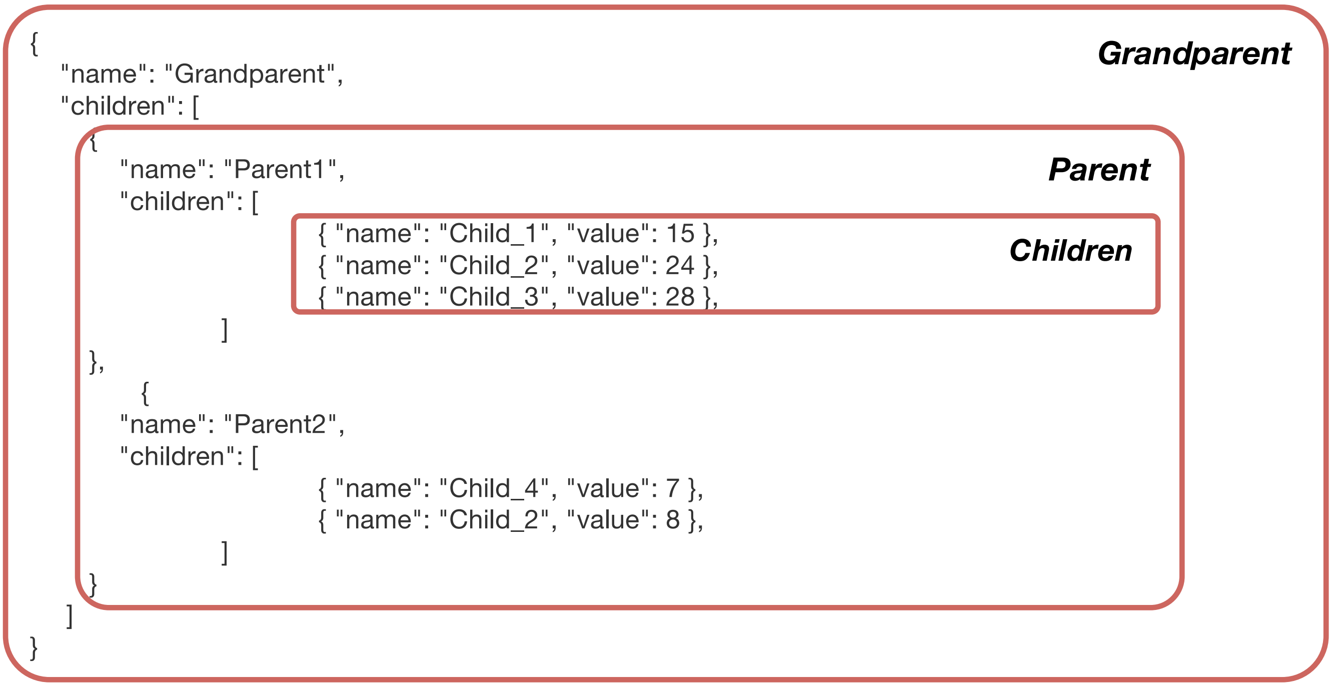
1. Tree Data for Tree Maps

In order to draw a treemap, requires data in tree form. Such data has a clear parent-child relationship between every element; the deeper you go, the more children you have and thus the greater the number of grandparents.



In order to define our “family” we use the JavaScript Object Notation (JSON) format. JSON is a tree structure natively, thus it is fairly strait forward to directly use externally available JSON data with the D3 library. Unfortunately our dataset is tabular, thus some processing will be required.

The following gives an example of how we define our “family” in the JSON format:



```
{
   "name": "Grandparent",
   "children": [
     {
       "name": "Parent1",
       "children": [
                    { "name": "Child_1", "value": 15 },
                    { "name": "Child_2", "value": 24 },
                    { "name": "Child_3", "value": 28 },
               ]
     },
        {
       "name": "Parent2",
       "children": [
                    { "name": "Child_4", "value": 7 },
                    { "name": "Child_2", "value": 8 },
               ]
     }
   ]
}
```

2. Tree Maps for Tree Data

In order to translate the projects data into the required tree data it is possible to use online tools and code libraries that automatically convert CSV to JSON. Although this process has already been done with our dataset, it is very worthwhile knowing such services exist and can be extremely useful in provide data in multiple formats from your own websites.

http://www.cparker15.com/code/utilities/csv-to-json/

Due to the scale of the data it is recommended that the dataset be simplified by removing un-necessary columns (using any tabular editor). For the purposes of demonstration and prototyping it may be necessary to further divide the provided dataset and select data from only a few agencies.

When simplifying the data, note that the example visualisations will **ONLY** accept data in the structure shown previously. This invludes the keys, which **MUST** be “**name**” and “**value**”.

3. Making your Visualisation

As D3 is a complex library, it is recommended that you find examples to work from and learn as you go. This exercise is no different and the URL below contains the boilerplate for the code to draw the treemap.

In order to work with this example and be able to save your changes, you should **fork** this jsFiddle into one of your own. Creating an account with jsFiddle will also allow you to retrieve this work later without having to remember the URL.

http://jsfiddle.net/davetaz/P7s4g/

When complete, this first example will draw you a static, non-interactive treemap of the data you provide. Once you have your data in a tree format, as shown in section 1, you need to copy it into the JavaScript action block as the data fore the *json_object* variable:

```
var json_object = {
…
…
}
```

Once you have your object, don’t forget to press the **run** button in jsFiddle, this should draw the visualisation in the result panel.

If you don’t see a visualisation the chances are that your data is malformed. To check your JSON object you can use and online validator:

http://jsonlint.com

The most common error is that of missing braces or misplaced commas.

In order to build an interactive visualisation requires a slightly different use of D3.

http://jsfiddle.net/davetaz/Xc8nT/

In this visualisation, try changing ".sticky(true)" to ".sticky(false)", how does this change the visualisation?

4. Extension Exercises

D3 has thousands of example visualisations you can use, there is also the nvd3 library which makes re-using the D3 visualisations even easier.

http://d3js.org

http://nvd3.org

Take a look through both websites for inspiration before trying to build one of your own visualisations from the examples. You should also be able to find plenty of examples in jsFiddle by simply including this is a google search term alongside D3 or NVD3 and the type of visualisation you want to create.