FIT3179 DATA VISUALISATION

Week 7 Studio Activity (Part B): Introduction to Vega-Lite

Vega-Lite is a high-level grammar of interactive graphics. It provides a concise, declarative JSON syntax to create an expressive range of visualisations for data analysis and presentation. Based on the information on the official Vega-Lite website [link]:

> "*Vega-Lite specifications describe visualisations as encoding mappings from data to* ***properties of graphical marks*** *(e.g., points or bars). The Vega-Lite compiler* ***automatically produces visualisation components****, including axes, legends, and scales. It determines the default properties of these components based on a set of* ***carefully designed rules****. This approach allows Vega-Lite specifications to be concise for quick visualisation authoring while giving user control to override defaults and customize various parts of a visualisation. As we also designed Vega-Lite to support data analysis, Vega-Lite supports both* ***data transformations*** *(e.g., aggregation, binning, filtering, sorting) and* ***visual transformations*** *(e.g., stacking and faceting). Moreover, Vega-Lite specifications can be* ***composed*** *into layered and multi-view displays and made* ***interactive with selections****.*"

Please watch this video: a high-level overview (6 minutes) of Vega-Lite:
https://www.youtube.com/watch?v=rydth27fB3Q

1. Start from a Simple Example

In this section, we will create a simple bar chart in Vega-lite (1.1), demonstrate how you can edit, run and test Vega-lite codes (1.2), and how to embed vega-lite visualisations in a webpage (1.3).

1.1. Create a Simple Bar Chart in Vega-Lite

We will first create a bar chart (Figure 1) with a simple "city" dataset (Table 1). This data includes five rows that contain the daily costs of the five cheapest European cities. It is available here: https://raw.githubusercontent.com/FIT3179/Datasets/main/city_daily_cost.csv

Table 1. The dataset, which is available here:

id	city	daily_cost
1	Kiew	26
2	Karkow	28
3	Belgrade	29
4	Bucharest	30
5	Sofia	33

Figure 1. A simple bar chart from the city dataset.

The JSON code to generate a bar chart is straightforward, as shown below.

```
{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "title": { "text": "Cheapest European Cities" },
  "data": {
    "url": "https://raw.githubusercontent.com/FIT3179/Datasets/main/city_daily_cost.csv"
  },
  "mark": "bar",
  "encoding": {
    "x": { "field": "daily_cost", "type": "quantitative" },
    "y": { "field": "city", "type": "nominal" }
  }
}
```

```
1 {
2  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
3  "title": { "text": "Cheapest European Cities" },
4  "data": {
5     "url": “https://raw.githubusercontent.com/FIT3179/Datasets/main/city_daily_cost.csv"
6  },
7  "mark": "bar",
8  "encoding": {
9     "x": { "field": "daily_cost", "type": "quantitative" },
10     "y": { "field": "city", "type": "nominal" }
11  }
12 }
```

- Lines 1 and 12 are the round brackets that are needed for the JSON format.
- Line 2 indicates that a Vega-Lite (version 5) compiler will be used to translate the code into a visualisation.
- Line 3 defines the title of the visualisation.
- Lines 4 to 6 identify the data location via a URL. This could be a URL or a local file path. Vega-Lite supports a few types of files, including csv, tsv, json, geojson, etc. Vega-Lite can

recognise the file type based on the file extension name, or we can define the file type ourselves, which will be discussed in later weeks.

- Line 7 defines the visual mark of the visualisation, which is “bar”.
- Lines 8 to 11 define the visual channels of the visualisation, the corresponding data attributes, and the attribute types.

Questions:

1. What is a JSON file? Search it online and discuss that with your peers.

2. How is Vega-lite related to Tamara Munzner’s “What? Why? How?” framework that we learned in the course?

1.2. Edit & Run Vega-Lite Code

There are a few tools to edit, run and test the Vega-Lite code.

Method 1. Using the online official Vega-Lite Editor

Open a Vega-Lite editor via https://vega.github.io/editor/, copy the Vega-Lite code to the coding area (Figure 2), and the result visualisation will be shown on the top-right part of the screen. There are some other views on the bottom-right of the screen, including *Logs*, *Data Viewer* and *Signal Viewer*. Click on each of them to get a sense of what information is contained there.

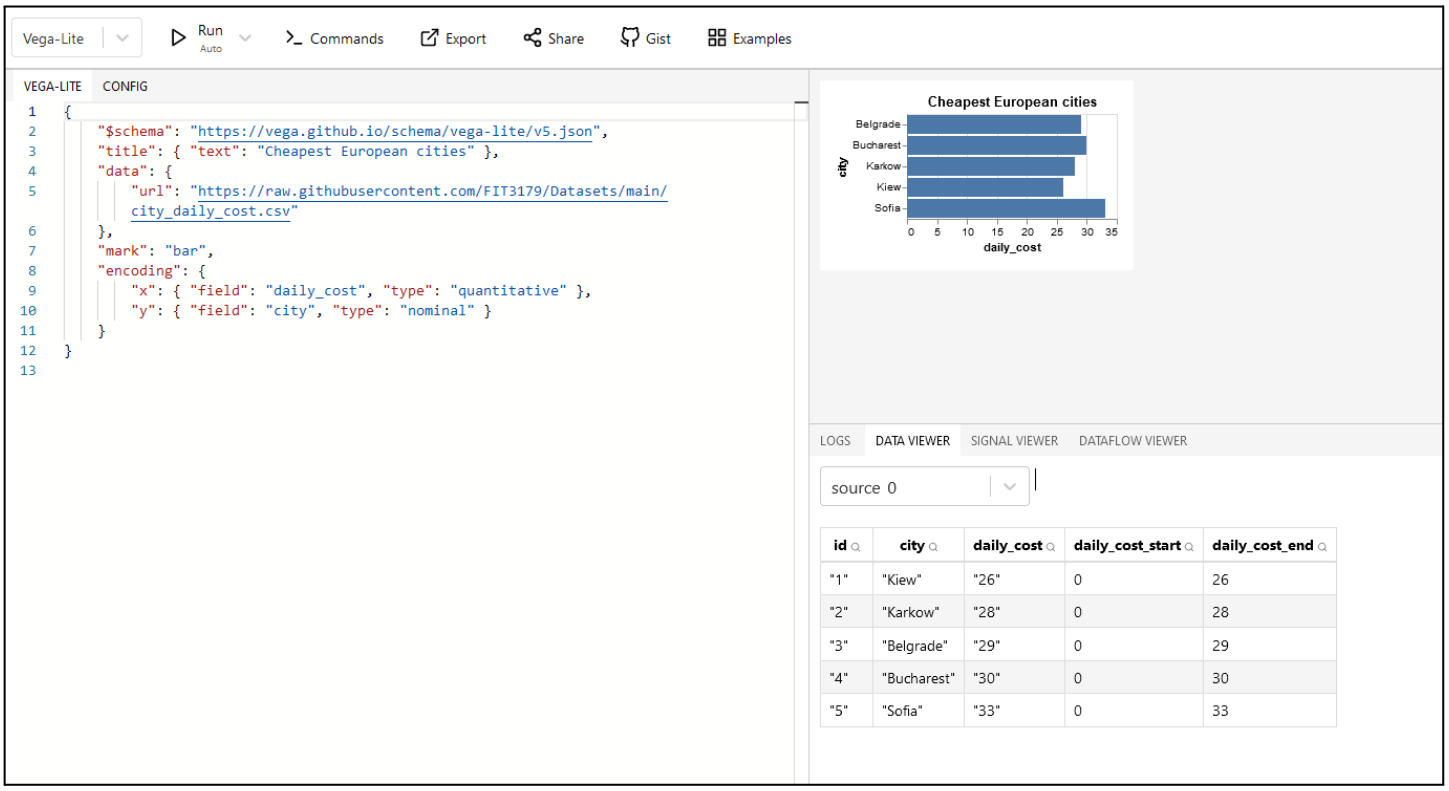

Figure 2. Running Vega-Lite code in Vega Editor.

Method 2. Using “Vega Viewer” in Visual Studio Code.

Step 1. You can create a JSON file in Visual Studio Code, you can give it any name or use “daily_cost_bar_chart.vg.json”.

Note: it is not required to use the “.json” extension or the “.vg.json”, but it is a good habit to keep it this way.

Step 2. Copy the Vega-Lite code into the new file, as shown in Figure 3.

Figure 3. Vega-Lite example in Visual Studio Code

Step 3. Click “F1” on the keyboard, enter “vega” and select “Vega: Preview Vega Graph” (Figure 4). This is part of the “Vega Viewer” plugin we installed in week 4. If you cannot find this plugin, please make sure you go back to the week 4 content and install it.

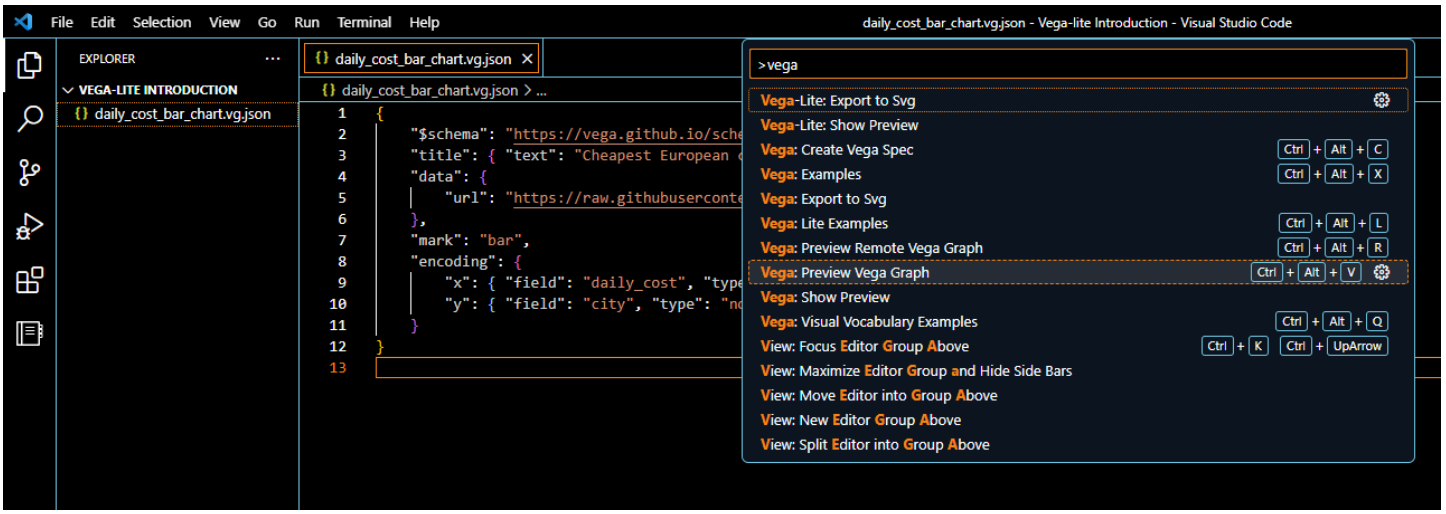

Figure 4. Searching for Vega Viewer in Visual Studio Code

The window will show the resulting visualisation on the right side of the screen (Figure 5).

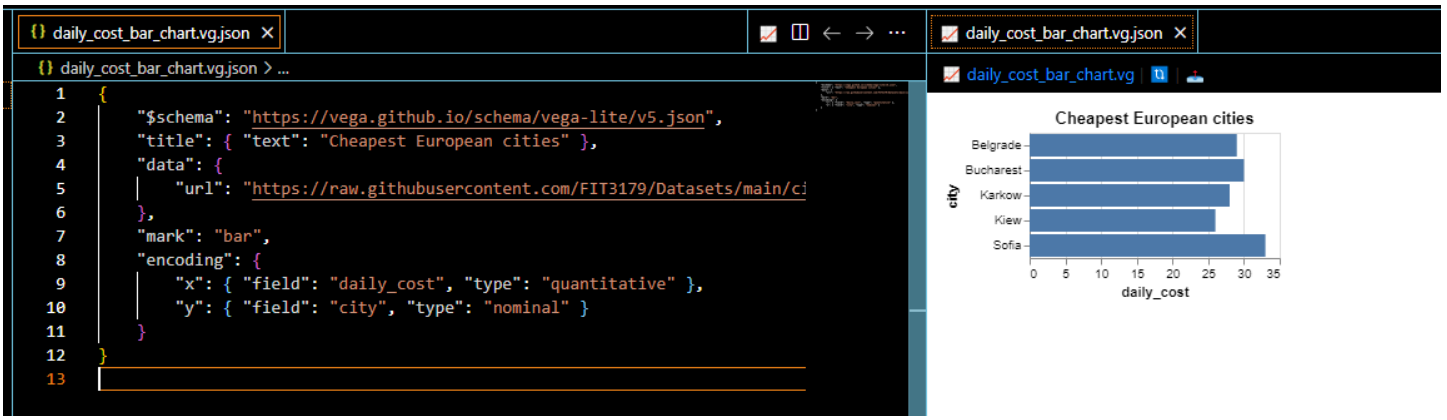

Figure 5. Preview Vega-Lite visualisations in Vega Viewer

An advantage of using Visual Studio codes and the Vega viewer plugin is that you can store the data locally and use the local file path for the data.

Open your web browser and go to the url where the data is stored. Then, save the page as file named “city_daily_cost.csv”. Move this file into the folder of your current project.

Change the url parameter to the file name, as shown in Figure 6.

Figure 6. Using a local file path

At last, we need to fix the axes titles to make them more readable by adding a custom title attribute for each axis (x and y encodings), as shown in Figure 7.

Figure 7. Customise axes titles

1.3. Include Vega-Lite Code to Show on a Webpage

Step 1. Create an HTML file. The HTML code is shown below.

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8
9      <script src="https://cdn.jsdelivr.net/npm/vega@5.20.2"></script>
10     <script src="https://cdn.jsdelivr.net/npm/vega-lite@5.1.0"></script>
11     <script src="https://cdn.jsdelivr.net/npm/vega-embed@6.17.0"></script>
12
13     <script src="vega_lite_vis.js"></script>
14
15     <title>First Vega-Lite graph </title>
16 </head>
17
18 <body>
19     <h1>First Vega-Lite graph</h1>
20     <div id="bar_chart">
21     </div>
22 </body>
23
24 </html>
```

Lines 1-8 are general HTML 5 elements. Let's pay attention to the following line of code related to Vega-Lite:

- Lines 9 to 11 are all the essential JavaScript libraries that we need to include for Vega-Lite.
- Line 13 includes an extensional JavaScript file that we will define in Step 2.
- Lines 20 to 21 define a div with an id called "bar_chart". This will be the container that we use to embed our Vega-Lite visualisation.

Step 2. Use JavaScript to load the Vega-Lite visualisation into the HTML page.

We will create a JavaScript file called “vega_lite_vis.js”. Please note that this file should have the same name as the one in Line 13 of Step 1. Then copy the code below to this new JavaScript file.

```
1 var vg_1 = "daily_cost_bar_chart.vg.json";
2 vegaEmbed("#bar_chart", vg_1).then(function(result) {
3     // Access the Vega view instance
4 (https://vega.github.io/vega/docs/api/view/) as result.view
5 }).catch(console.error);
```

- Line 1 defines a variable called “vg_1” which contains the location of the Vega-Lite JSON file. This file name should be the same as the file you created in Section 3.1 – Method 2 (Figure 3).
- Line 2 embeds the Vega-Lite visualisation stored in “vg_1” in the div container named “bar_chart” (do not forget the “#” key before the div name).

If you have more than one Vega-Lite visualisation, you can add another div container to the page, define a “vg_2” variable to embed the second (and third, etc.) Vega-Lite visualisations in the corresponding div.

Step 3. Right-click “index.html”, and select “Open with Live Server” to view the visualisation in the browser (Figure 7).

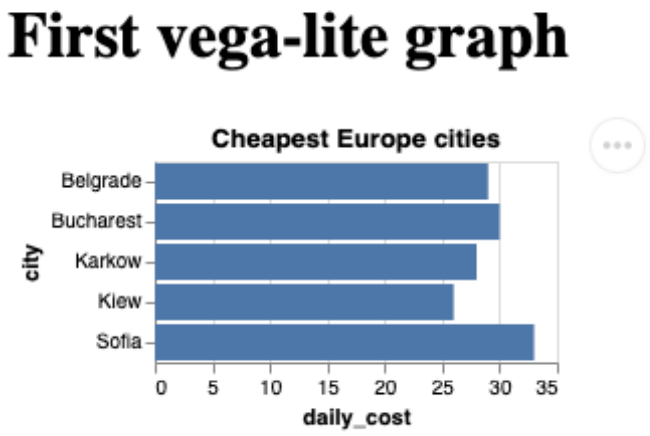

Figure 7. Webpage in browser containing the Vega-Lite codes

Exercise:

1. Have you noticed the three dots on the top-right corner of the figure? Click on it and explore the options there.

2. Do you still remember how to publish the webpage to GitHub? If not, check the week 4 content and publish this Vega-Lite example on GitHub.

2. Duplicate the “European city Dashboard” Example

At the end of week 5, we used CSS and HTML to duplicate some parts of the “Explore European Cities” dashboard (Figure 8).

Figure 8. European Cities Example in Week 5

Remember that we used a figure for the bar chart on the right side? Now that you know how to create a bar chart with Vega-Lite, can you replace this bar-chart figure with a Vega-Lite visualisation?

Please check this week’s material for the answer. The result visualisation is shown in Figure 9. Please note that we will learn how to add text labels to the bar charts in later weeks.

Figure 9. European Cities Example with Vega-Lite visualisation

3. More About Vega-Lite

The Vega-Lite website [link] contains tutorials, extensive examples, and documentation. Please complete the following tasks as homework and exercises (no submission is needed, but this is critical for Assignment 2).

Task 1. Follow the official Vega-Lite tutorials:

A. getting-started tutorial at https://vega.github.io/vega-lite/tutorials/getting_started.html

Time: 20-30 minutes.

B. exploring data: https://vega.github.io/vega-lite/tutorials/explore.html

Time: 30-40 minutes.

Task 2. Check the Vega-Lite examples on https://vega.github.io/vega-lite/examples/, choose 2 to 3 visualisations (e.g., stacked bar charts, scatter plots), and open the example in Vega Editor by clicking “View this example in the online editor”. Try to create a similar chart with your own data.

For example, if you choose the stacked bar chart example [link] shown in Figure 10, you can use the covid-19 dataset [link] and create a Vega-Lite chart similar to this page: [link]. The studio material contains an example answer to this.

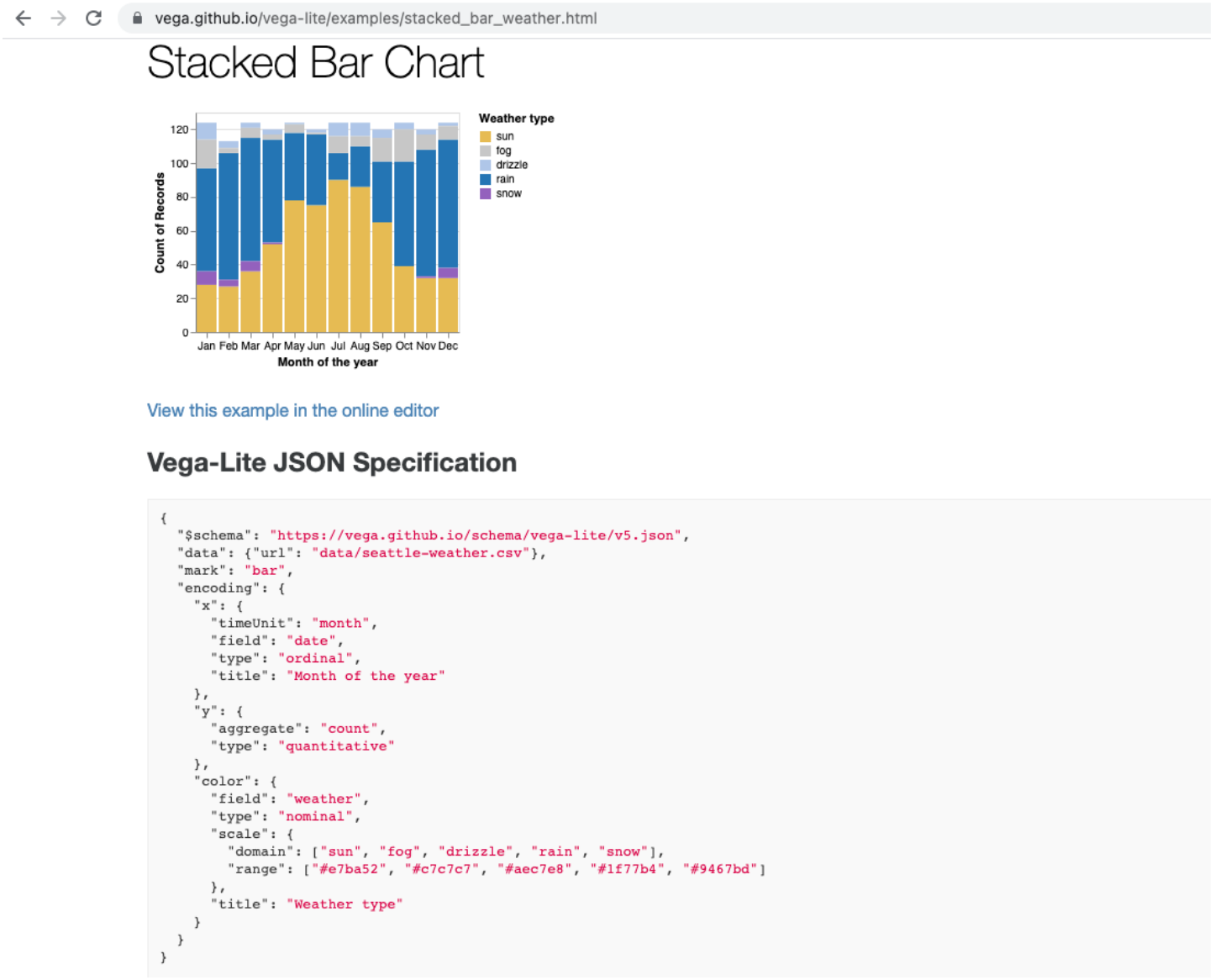

Figure 10. A stacked bar chart example in the Vega-Lite website [link].

Task 3. Choose one chart in your Assignment 1 Tableau dashboard and duplicate the chart in Vega-Lite.

Notes:

1) You can check the examples on the Vega-Lite website [link].

2) The official documentation [link] is also very handy for learning Vega-Lite. For example, if you want to change the size of the chart, you can click “Width/Height” on the documentation page [link]; or if you would like to customise your line chart, you can check the “Mark > Line” section: [link].

3) All codes in this week’s Vega-lite examples are also available on this GitHub page: [link]