I have added ten data visualisations to the app on top of the original five. Of these ten, four were outlined and demonstrated in videos on Coursera, while the other six are unique. Each extension follows the same structure. It has name and ID variables, then setup, draw and destroy methods. All but the 'Create a pie chart' extension also have a preload function. This has a callback function in it that sets a Boolean variable called loaded to true once the data has loaded. All extensions are accessible via the menu on the left-hand side. I will discuss the six unique extensions in more detail.

European Energy Data – This is a filled bar chart where the user can pick countries in the EU and see their energy sources. I have a class for the extension that uses another class for the filled bar chart.

Create a pie chart – This extension allows users to create and download a pie chart. This extension calls the pie chart class that I edited to allow the user to input data.

European Unemployment – This extension uses the MapboxGL library to display unemployment rates across the EU. The extension calls a separate map class that accesses the API and displays the map using p5.js' DOM.

The 20 most populous countries – This extension is a bar graph that shows changes over time. It calls a bar chart class that contains an array of bar classes. Each bar is a dynamic changing thing that uses its own methods.

USA box office revenues – This extension is a streamgraph, a stacked area graph with a central axis. The extension has an array of Line classes that make up the data of the graph.

Car emissions – This extension is a radar chart. It uses the RadarChart class, which has an array of RadarOutline classes. This array is dynamic. The user can add and remove the outline based on the data they want to see.

I have also added the following:

Popup function – This is a helper function that most other extensions call. It creates a popup next to the mouse pointer showing appropriate data. The data must be an object with the following parameters: label, colour and data. The parameter extraData can also be used if the extension needs to pass more information into the popup.

Loading Screen – I have also created a loading screen that uses each extension's callback function to check if all the data has loaded. This has its own separate class.

Please see app diagram in q2 folder.