COS30045: Data Visualisation
Process Book

CO2 EMISSIONS AND RISING SEA-LEVELS

Connor Nee-Salvador
102061410

https://mercury.swin.edu.au/cos30045/s102061410/finalproject/index.html

Table of Contents

Introduction

Background and motivation

Because the main topic for this project is Sustainability: Environment in Crisis, I have decided to base my work towards the current rise in the greenhouse effect and the CO_2 emissions over different countries (preferably the countries that contribute the most CO_2 emissions) that increase the greenhouse effect. Also, I will compare the data to other issues that are caused by the greenhouse effect such as the rise in sea-levels and rise in global temperature as a result. I will be using various sources that have plenty of information on this topic, as well as providing CSV spreadsheets of their data to make it easier for me to select the specific data needed to visualise my topic. The motivation for this topic is simply that while we do need to look out for our own carbon footprint, we also need to acknowledge that maybe it isn't so much up to the common people to fix this issue considering most people's everyday footprint makes up a minimal amount in the overall CO_2 emissions, but rather the large corporations/governments that play the largest of roles in adding CO_2 into the atmosphere.

Project Objectives

The main objective for my visualisation is to inform the audience of the following pieces of information:

- How much **CO_2 emissions** have **increased** over a long period of time between countries.
- How the **current CO_2 emissions** rate compares to the **rising sea-level** data.
- The **correlation** between the 2 data visualisations and confirming the rise in CO_2 emissions is the **causation** for the large rise in sea-levels over time.

If these points become clear to the audience through my visualisations, it will be a good way to showcase how this problem effects the environment around us and will perhaps motivate them into making a change to it. The objectives of this visualisation is also to simply inform the audience that CO_2 emissions are the direct cause for this change in in environment.

Project Schedule

I was able to get started on the project with good time. I had a working prototype of the data visualisations on the mercury server about 3 weeks in advance of the deadline. Having the front-end development of the visualisation completed has made the rest of the designing of the website more efficient as it gave me more time to focus on styling and layouts.

Data

Data Source

I will be using 2 datasets in a .CSV file format. The first data set I will be using comes from the Emissions Database for Global Atmospheric Research (EDGAR) and contains the following sets of data:

Attribute	Description	Data Type
ISO_CODE	This is the unique identifier for every country that consists of 3 letters for each. (i.e. AUS, USA)	Categorical (Text)
ISO_NAME	This is the field that contains for full countries name next to the ISO_CODE.	Categorical (Text)
Sector	This field contains 1 out of 5 sectors from where the data is being gathered from in each country (i.e. Transport, buildings).	Categorical (Text)
substance	This field is supposed to contain what substance is being emitted, for the purpose of this dataset however this field only contains substance type – CO2	Categorical (Text)
Year	There are several fields of this type, from 1970 to 2016. They contain the number value for the amount of emissions from each country.	Ordinal (Number)

The next dataset comes from the EPA's Climate Change Indicators in the United States. The data source comes from both CSIRO (the Commonwealth Scientific and Industrial Research Organisation) and NOAA (National Oceanic and Atmospheric Administration), and contains the following data:

Attribute	**Description**	**Data Type**
Year	This field contains years from 1906 to 2015.	Ordinal (Number)
CSIRO – Adjusted sea level (inches)	This field contains a number value representing how many inches from sea level (0) recorded for each year.	Ordinal (Number)
CSIRO – Lower error bound (inches)	This field contains the same information but accounts for any errors in the lower.	Ordinal (Number)
CSIRO – Upper error bound (inches)	This field contains the same information but accounts for any errors in the upper.	Ordinal (Number)
NOAA – Adjusted sea level (inches)	This field contains a number value representing how many inches from sea level (0) recorded for each year, however this field contains only results from the NOAA.	Ordinal (Number)

Data Processing

For the .csv file I used for the CO2 emissions visualisation, I had to first remove any of the attributes that would not be necessary for what I was creating. essentially all I had to do was copy and paste the data I needed and reformat the structure to fit my purposes.

51	ABW	Aruba	Non-combustion	CO2	0.417657	0.421436	0.425278	0.428903
52	AUS	Australia	Power Industry	CO2	53229	52451.8	57959.4	59157.8
53	AUS	Australia	Transport	CO2	32481.11	33373.3	34341.97	38072.09
54	AUS	Australia	Other industrial combustion	CO2	45932.8	47064.1	45528.8	47818.7
55	AUS	Australia	Non-combustion	CO2	17875.82	17549.5	18362.58	18683.88
56	AUS	Australia	Buildings	CO2	10509.8	10973	11113.1	13216.4

I had to select and copy the relevant data that I would be using. In this case I copied the number data for CO2 emissions each year for the countries I would be visualising (in the above case: Australia).

5											
6	ISO_CODE	ISO_NAME	sector	substance	1970	1971	1972	1973	1974	1975	
7	AFG	Afghanistan	Transport	CO2	503.4713	503.7333	536.897	525.4184	552.5486	666.1144	6
8	AFG	Afghanistan	Other indust	CO2	128.944	128.944	105.463	54.1501	62.1655	55.4417	
9	AFG	Afghanistan	Buildings	CO2	641.589	670.221	518.926	757.527	887.659	590.666	
10	AFG	Afghanistan	Non-combus	CO2	457.8263	455.0624	528.2834	399.1108	562.9699	499.7771	5

Next I copies the years from 1970 up to 2016, so I could create a column attribute for years in the .csv created from this.

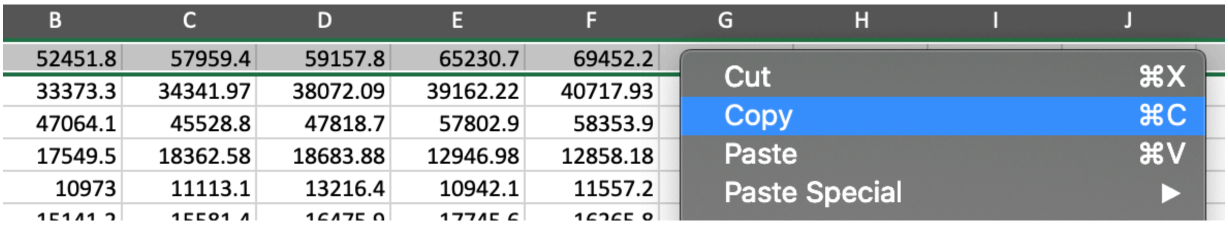
	A	B	C	D	E	F	G	H	I	J	K
1	53229	52451.8	57959.4	59157.8	65230.7	69452.2	71822	80798.9	75538.1	78785.6	89132.
2	32481.11	33373.3	34341.97	38072.09	39162.22	40717.93	42019.44	44744.58	47210.79	48332.47	49573.1
3	45932.8	47064.1	45528.8	47818.7	57802.9	58353.9	57303.7	59170.8	56747.2	58482.6	56990.
4	17875.82	17549.5	18362.58	18683.88	12946.98	12858.18	13178.28	12337	14352.47	13839.31	14701.1
5	10509.8	10973	11113.1	13216.4	10942.1	11557.2	11408.1	11909.7	11587.8	11771.1	11026.
6	14897.7	15141.2	15581.4	16475.9	17745.6	16265.8	16914.1	16460	16888	16625.7	15327.
7	=SUM(A1:A6)										

I then took the CO2 emissions data for each year for the selected country. However, the data was split into 5 different categories for the different areas of CO2 emissions. I used a simple SUM formula to add the 5 categories together and create a new attribute for my .csv file for the overall emissions for each year.

C	D	E	F	G	H	I	J	K
57959.4	59157.8	65230.7	69452.2	71822	80798.9	75538.1	78785.6	89132.4
34341.97	38072.09	39162.22	40717.93	42019.44	44744.58	47210.79	48332.47	49573.19
45528.8	47818.7	57802.9	58353.9	57303.7	59170.8	56747.2	58482.6	56990.5
18362.58	18683.88	12946.98	12858.18	13178.28	12337	14352.47	13839.31	14701.13
11113.1	13216.4	10942.1	11557.2	11408.1	11909.7	11587.8	11771.1	11026.3
15581.4	16475.9	17745.6	16265.8	16914.1	16460	16888	16625.7	15327.4
182887.25	193424.77	203830.5	209205.21	212645.62	225420.98	222324.36	227836.78	=SUM(K1:K6)

And then simply dragged the formula cell across to calculate overall emissions for all the years.

In order for my JavaScript to read the .csv file, it would have to read in each variable based on columns instead of rows. Luckily, excel has a copy-paste option which allows me to transform a row of data into a column, this method is called: transpose.



	A	B	C	D	E	F	G	H
1	year							
2	53229							
3	52451.8							
4	57959.4							
5	59157.8							
6	65230.7							
7	69452.2							

As for the transformation of the .csv file for the sea-level visualisation, it was already made quite simple since the original .csv file I collected had already been displaying its data in columns instead of rows. I simply only needed to copy paste the data that I required for the visualisation:

5	Year	CSIRO - Adjusted sea level (inches)	CSIRO - Lower error bound (inches)	CSIRO - Upper error bound (inches)	NOAA - Adjuste
6	1906	1.251968503	0.673228346	1.83070866	
7	1907	1.196850392	0.61023622	1.783464565	
8	1908	1.098425196	0.527559055	1.669291337	
9	1909	1.27559055	0.700787401	1.850393699	
10	1910	1.271653542	0.696850393	1.846456691	
11	1911	1.598425195	1.039370078	2.157480313	
12	1912	1.476377951	0.917322834	2.035433069	

However, for the final 2 years recorded in the dataset (2014 and 2015) I had to copy the data from the NOAA column since the CSIRO column did not contain data for those 2 years:

112	2012	9.326771644	8.992125975	9.661417313	8.397377333
113	2013	8.980314951	8.622047235	9.338582668	8.484542336
114	2014				8.59832355
115	2015				8.9433077

Requirements

Must-Have Features

For the final visualisation, I intend for the **features** that it must require be as achievable as possible, whilst the optional features be something that can be added later should I have more time. The must have features for the visualisations are as follows:

- Some form of **interaction** in which the user can control what data is being presented to them.
    - **Buttons**
    - **Scroller**
    - **Zoom in/out**
    - **mouse hover**

  Not all of what is listed will be required however some form of interactivity from that list is a must.
- Graph must contain an **X and Y axis**.
- X and Y axis must be clearly **labelled**.
- **Colour** must be included in the visualisation.
- an **explanation** for what the data is representing (a simple paragraph of text describing the data will suffice).
- The visualisation must have a **meaningful title.**

These are the current goals I have set for the must-have features to my visualisation(s). If they are not met, the visualisation will not be doing its intended purpose and will not fully satisfy the outcome I am hoping for in this project.

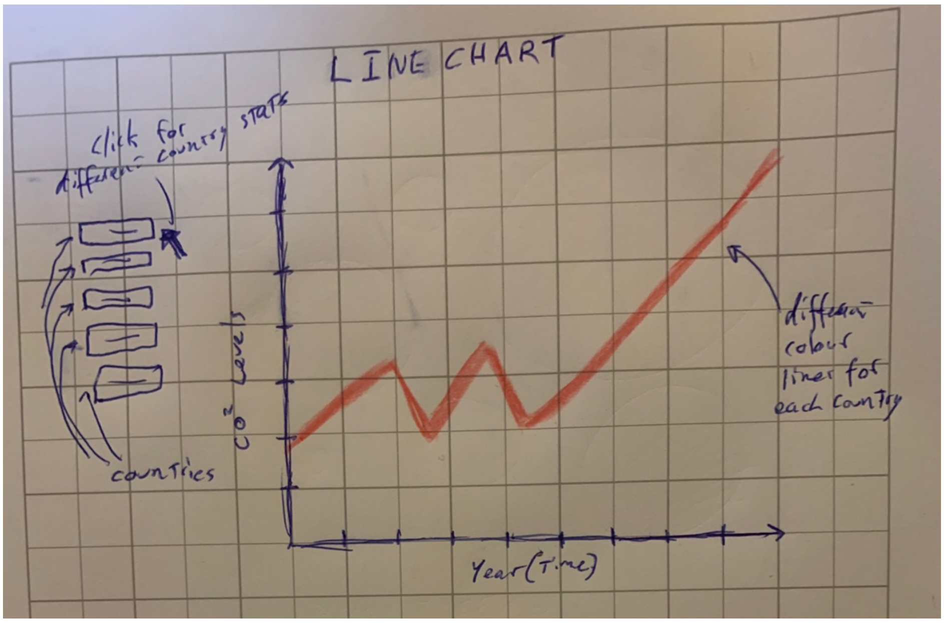
Optional Features

As for the optional features I intend to include in my visualisation(s), the following list describes the most ideal representation on how it should appear:

- **User-interaction** varies between all kinds of different interfaces that should be integrated into the visualisations.
- The **CSS** of the website should be clean, simple, and elegant in a way that compliments how the viewer will see the visualisation(s).
- **Transitions** should be included in the visualisations (Although this is optional as animated transitions between data can sometimes be too complex and confuse the viewer at times).
- **Other webpages** that contain more information on the visualisations (This is optional because I may already be able to contain all the information I need on one page).

Visualisation Design - Proposal

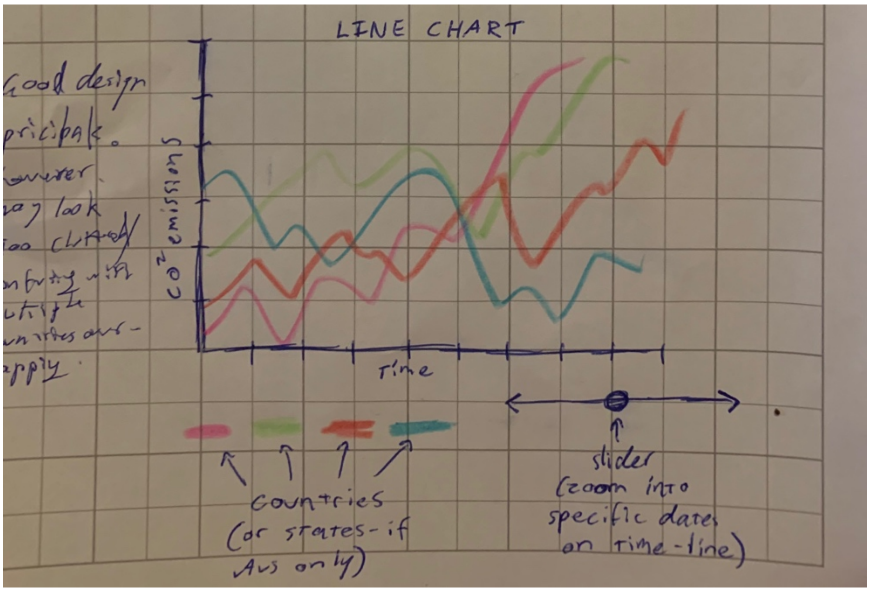
Line Chart



This line chart shows the user in an efficient way through continuous data on how the Co2 emission levels have a rising trend for each year. the interface of buttons switches to different countries so the user can view the continuous data across different countries. The X and Y axis are clearly labelled to show the user what the chart is about, and the buttons will be labelled with the country and possibly colour coordinated with the line graph depending on which country is being viewed.

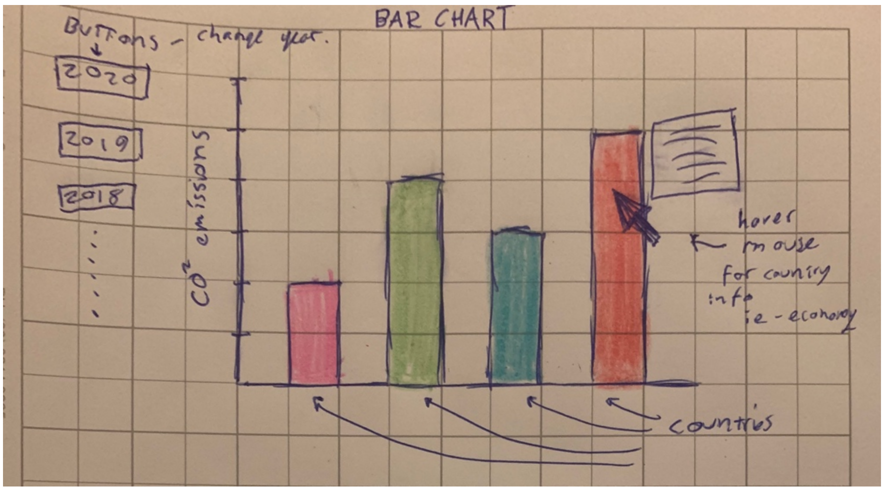
ADVANTAGES	DISADANTAGES
Easy to understand, values are clear and concise	May be too simple with not enough information on one view
Use of colours makes other data stand out / more obvious	Only one line at a time will be displayed
Button layout is user friendly/simple	the Y axis value scales so the user may not know it has changed when pressing button

Multi-Line Chart



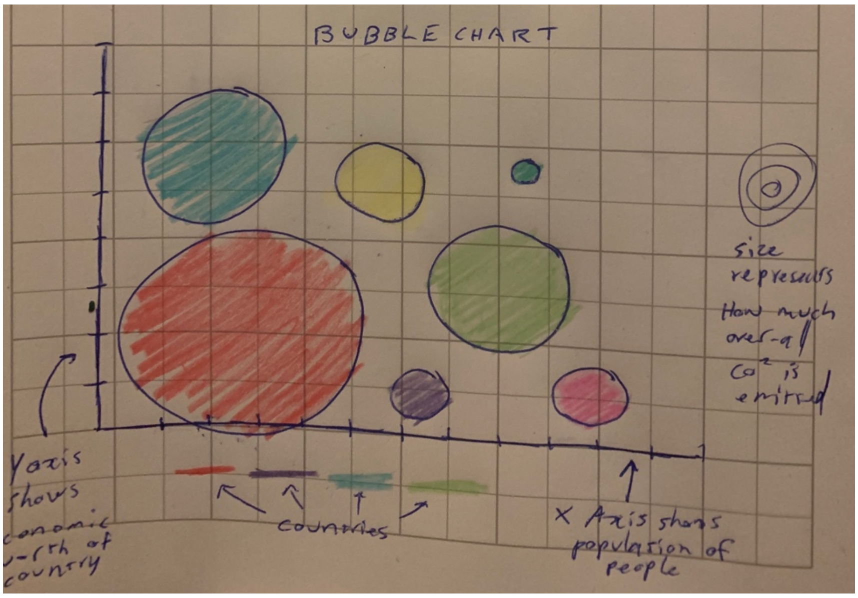
The multi-line chart displays all countries emission data on one graph. However, depending on how many countries are included the data can become cluttered and hard for the viewer to interpret. Each line has a different colour with a key bellow which labels the colours. A slider will also be a good way for the viewer to zoom in through the X axis to display more precise data between specific years on the X axis.

Bar Chart



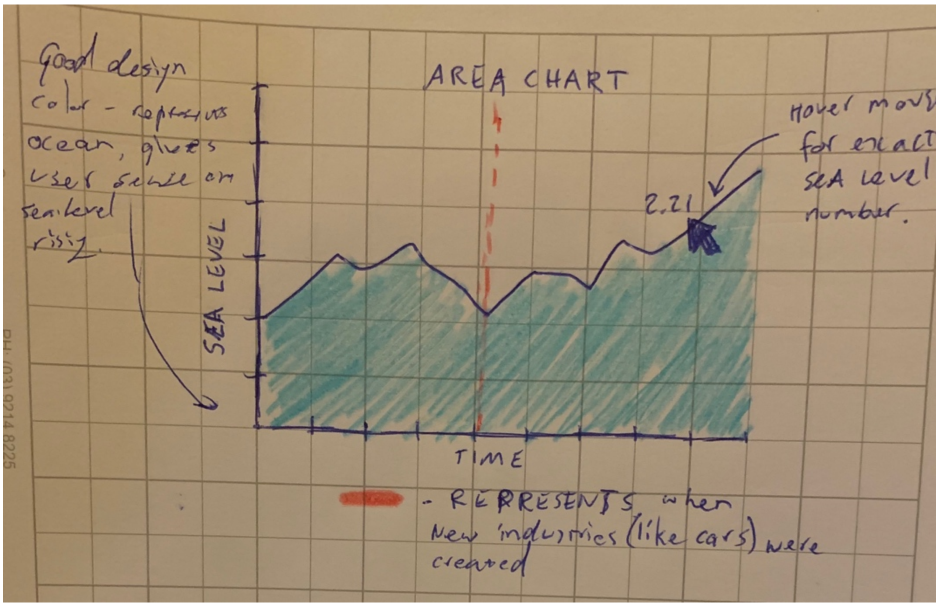
Using a bar chart like this design will compare the Co2 emissions between countries for specific years that can be changed with buttons in the user interface. The bars are colour coded for each country and there is a mouth-hover feature where the user can hover over the bars to display more information on the countries Co2 emissions. the Y axis represents Co2 emissions will the countries are represented on the X axis as coloured bars.

Bubble Chart



The bubble chart is the most complicated visualisation out of the many mock-up ideas I have created. The bubble chart will represent the different countries as coloured bubbles. The size of the bubbles will depend on the amount of Co2 emission that country has (the more emissions the bigger the bubble). The placement of the bubbles on the chart will depend on the X and Y axis. the Y axis represents the country's economic worth and the X axis represents the population density of the country. I had this idea because it will fairly place countries on the graph depending on the X and Y values. If a country has a high emission rate and low population density, it will mean that the country is more pollutive than that of a high population density country. This graph is supposed to accurately place the countries fairly rather than judge them off of CO2 emissions alone. However, the graphs complication is too high, and it will be highly unlikely to be able to create something like this using the d3 library in JavaScript.

Area Chart

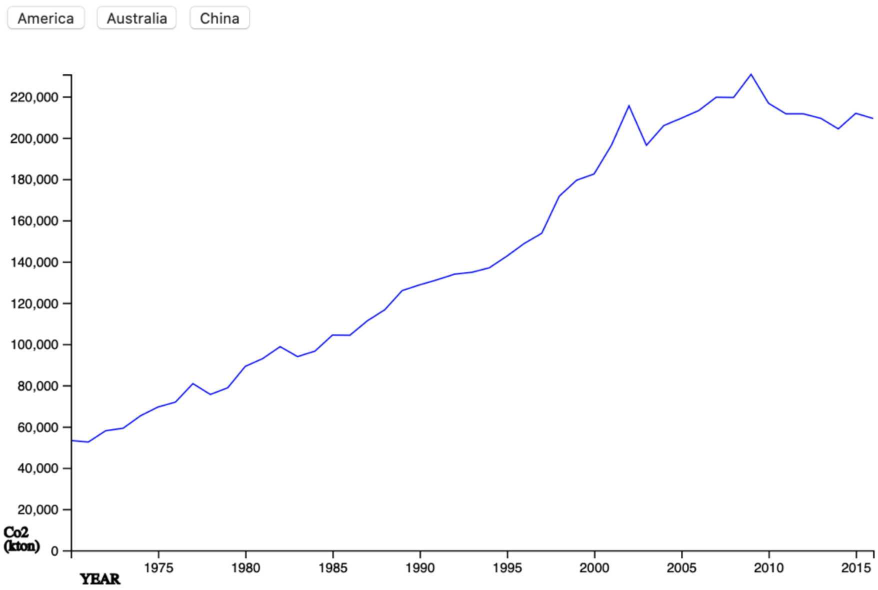


The area chart is the best way for me to convey the rise in sea-level data. The Y axis will represent sea level (in inches) and the X axis represents time. The colour of the area chart will be blue, simply to associate the chart with what it is representing visually (the ocean). because the sea-level data being used starts several years earlier than the CO2 emissions data, there will be a red dotted line placed to show when the CO2 emissions data begins. This will be so the user can see the correlation between the two visualisations.

ADVANTAGES	DISADANTAGES
Use of colour coincides with what the data is representing	Can only display one thing at a time.
The red line will be good for place markers for certain advancements in industrial co2 emissions	Limited user interaction can be implemented
The chart is simple enough to understand with the ability to become more complex	Can only represent 2 different data sets

Visualisation Design – Process

Making the visualisations was somewhat difficult using the d3 libraries, trying to balance the visual appeal of the graphic and the amount of information being displayed. the prototype for the JavaScript code ended up looking something like this:



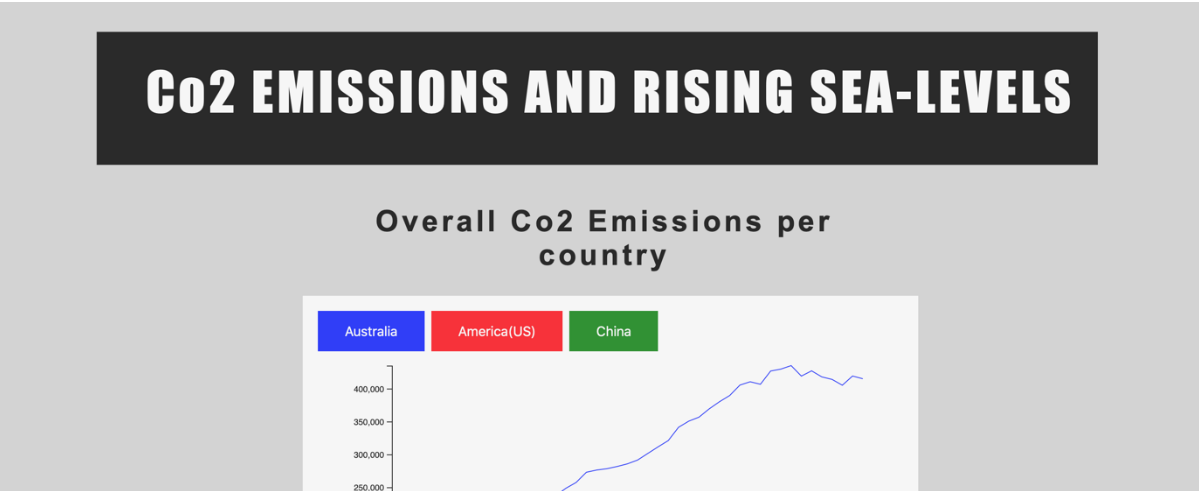
COS30045 Data Visualisation
Connor Nee-Salvador

I essentially worked on a blank canvas without any html and css to go on. once I had a working visualisation, I simply moved the JavaScript over to what was going to be my main webpage to display the visualisations in a more user-friendly way.

The JavaScript file itself is based off of a Doubtfire Task that had already been done in the past. Rewriting the code to fit my own parameters to draw a line chart visualisation based on my new .csv files.

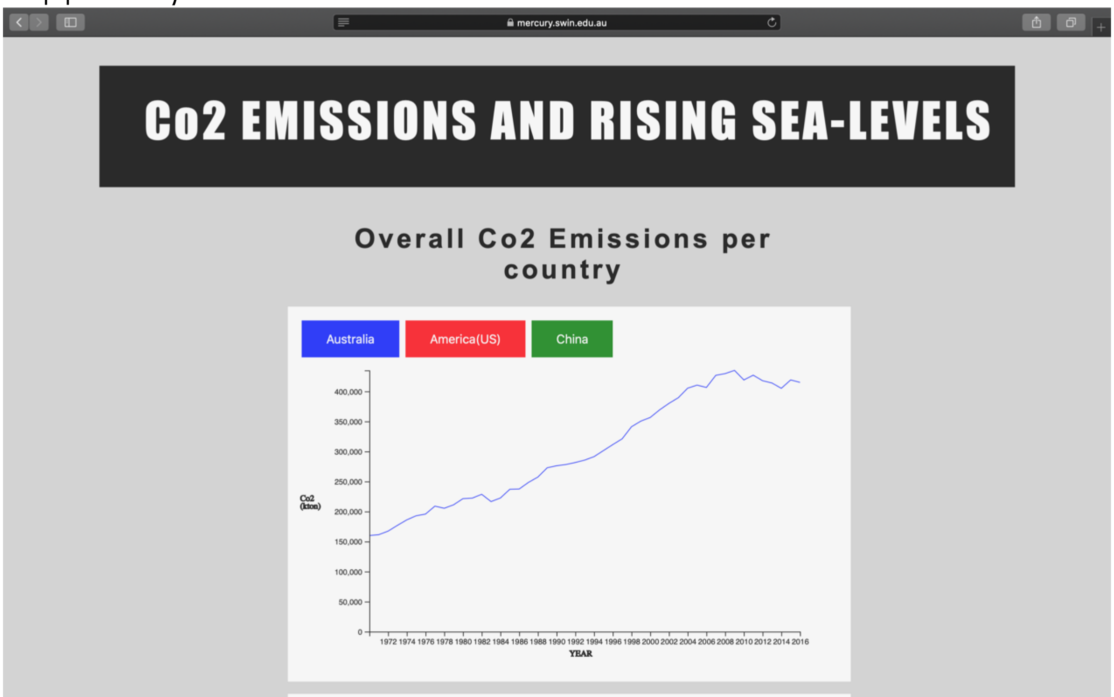
```
function init() {

  var w = 700;
  var h = 400;

  var xPadding = 50;
  var yPadding = 15;

  var dataset;

  var rowConverter = function(d) {
    return {
      date: new Date(d.year),
      number: parseFloat(d.emission)
    };
  }

  d3.csv("australiaCO2Emission.csv", rowConverter).then(function(data) {

      dataset = data;

      lineChart(dataset);
  });

   function lineChart() {



       xScale = d3.scaleTime()
                     .domain ([
                       d3.min(dataset, function(d) { return d.date; }),
                       d3.max(dataset, function(d) { return d.date; }),
                     ])
                     .range([xPadding+40, w - xPadding]);

       yScale = d3.scaleLinear()
                      .domain([0, d3.max(dataset, function(d) { return d.number; })
                       ])
                      .range([h - yPadding - 30, yPadding]);

```

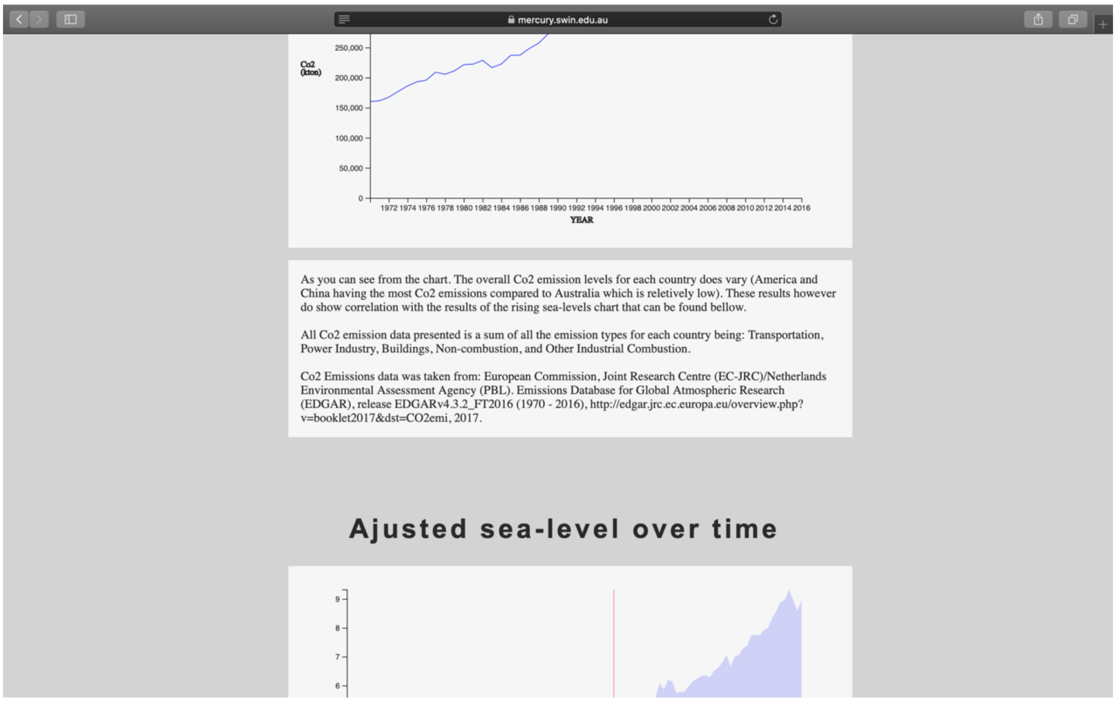
After the JavaScript was used for my main page, I polished up the look with some css and the final product ended up looking something like this:

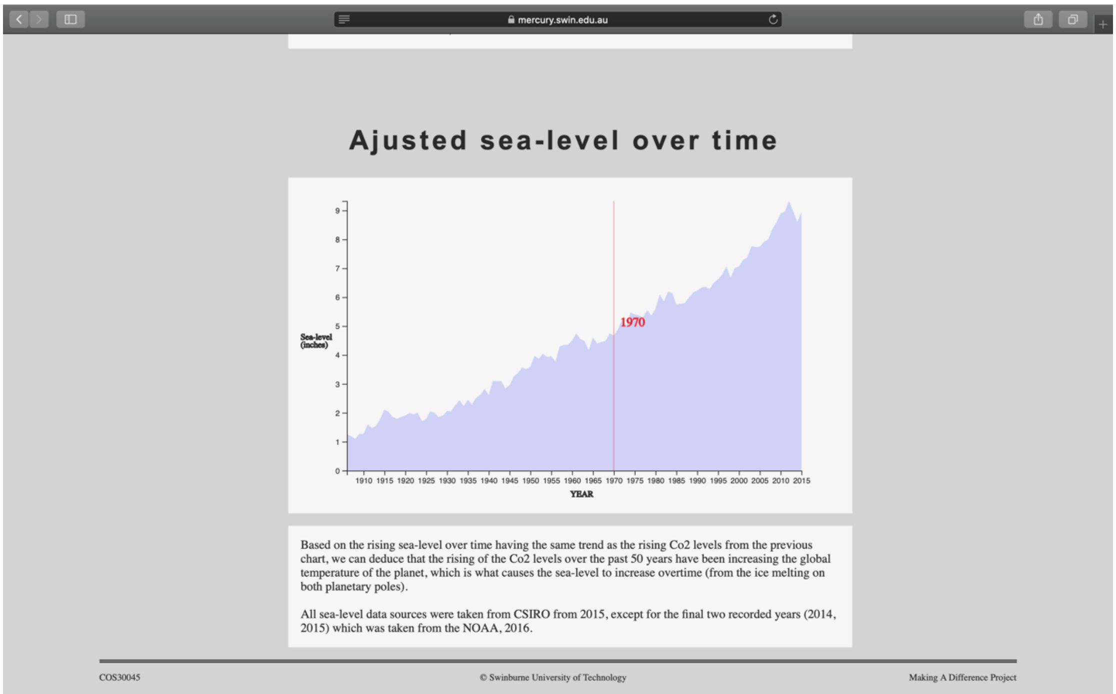


Visualisation Design – Final

The final design for the visualisation project has worked out the way I had expected it to. From the screenshots bellow, you will be able to see what the website is all about and how well it fits all the requirements listed from earlier in the process book (more information bellow the images and can be found in the appendix).







The final design for the Co2 emissions visualisation turned out to be a little simpler than what I had initially intended. However, with the restraints from my knowledge of the d3 library and how time consuming it may have been to work around some issues, I decided to go with my first design to display on a Line Chart the amount of Co2 emissions for each country over every year between 1970 and 2016. the Y axis value is the amount of Co2 emitted (measured in ktons) and the X axis holds the value for the Year those emissions occurred. The line being displayed on the chart is colour coordinated with the

buttons above the chart which is also where the name of the country is displayed. Australia – Blue
America – Red
China – Green

Requirements (Final)

The colour coordination for each country and a labelled X and Y axis were also listed on the Must-Have Features section of the report. And all of the other features listed there have been met as well. while the user interaction is somewhat limited for what it is, it still includes buttons that manipulate the data on screen. Because the countries I chose to display vary in their information greatly, pressing the buttons also causes the y axis to change in values based on how high the Co2 emissions for the countries are. the Line is also being updated by changing values, and colour, depending on the button pressed. The X and Y axis have been clearly labelled. (on both visualisations) as well as the inclusion of colour (the colour combinations also showing the user which country is being shown on screen). There are short descriptions in the paragraphs under the visualisations explaining what the data being represented is and how both visualisations have a relationship and that one correlates with the other in terms of their results. Both visualisations also feature meaningful titles that describe what they are.

As for the optional features that have been included, the css has been added and while it does have a more minimalistic approach, it complements the design of the visualisations and does not impede on any of the information being displayed, the user interaction from the buttons are present, and the second visualisation which is an Area Chart, includes a vertical line located around the middle of the X axis that displays the year when the Co2 emissions data was first being recorded. The area that is coloured in on the chart is also the colour blue, to visually represent the ocean level that the chart is depicting, making it easier for the reader to visually see the amount that the sea-level is rising.

Conclusion

The final outcome for the website turned out a lot better than what I had thought considering the time crunch I was under and how much knowledge I had gained from using the d3 libraries. I also believe my designs could have been greatly improved if I had spent more time on the website through the data visualisations and the way I had presented them using the CSS styling and the html layout of my final website. However, the final website I came up with fits the requirements I had set for myself earlier on, and it presents the required information through the data visualisations to the user from these requirements.

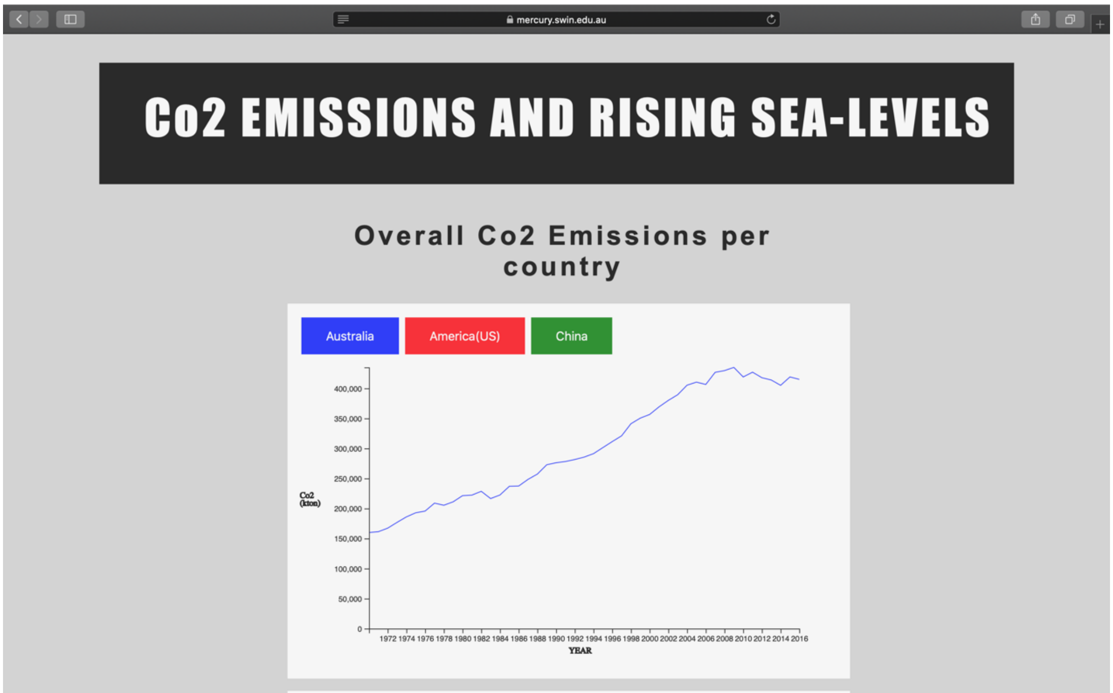
References

- European Commission, Joint Research Centre (EC-JRC)/Netherlands Environmental Assessment Agency (PBL). Emissions Database for Global Atmospheric Research (EDGAR), release EDGARv4.3.2_FT2016 (1970 - 2016), http://edgar.jrc.ec.europa.eu/overview.php?v=booklet2017&dst=CO2emi , 2017

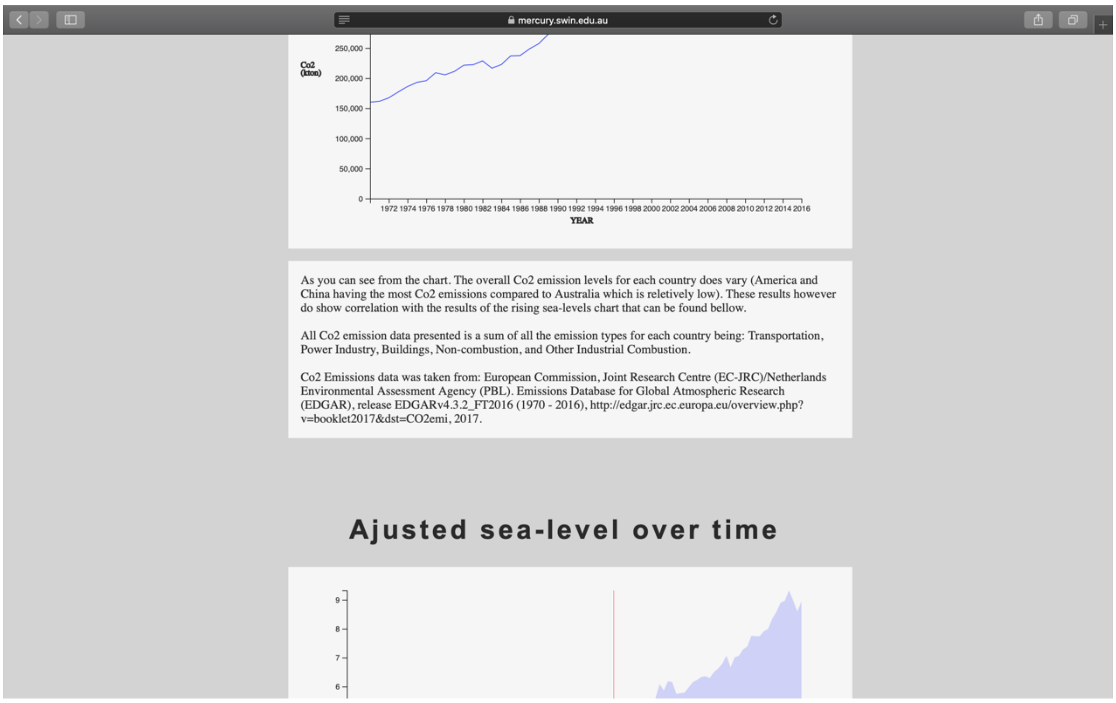
- EPA's Climate Change Indicators in the United States: www.epa.gov/climate-indicators

Appendix

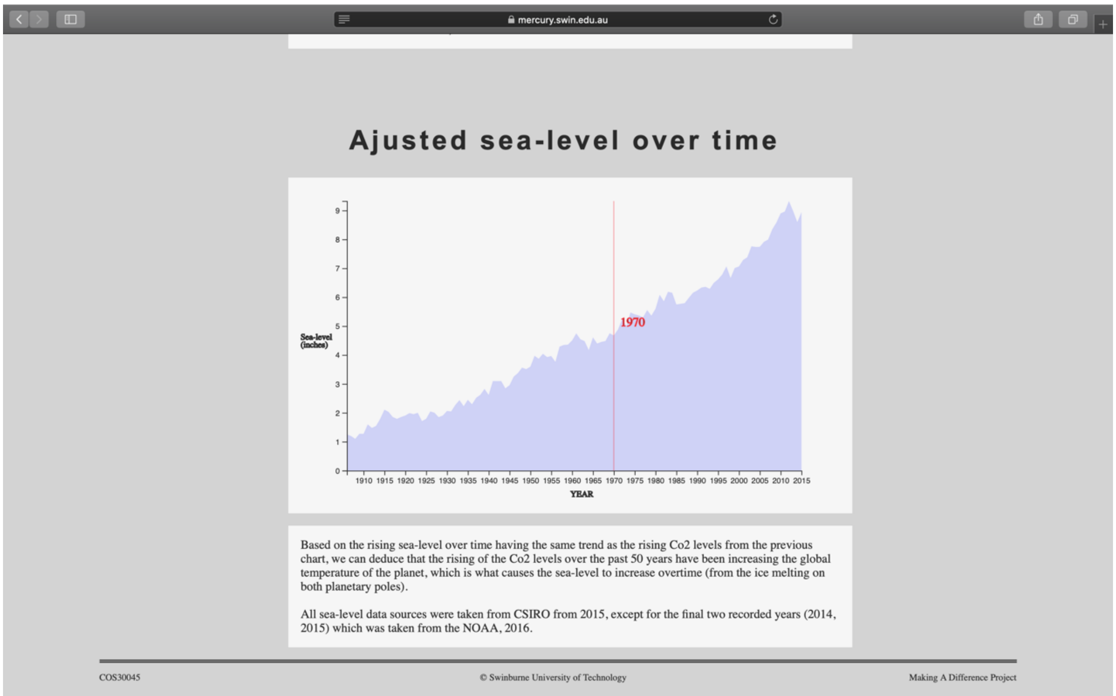
Top part of the page. Title screen, first visualisation (Line Chart) – Overall Co2 Emissions per country.



Short description of first visualisation and explanation of data being displayed.



Second visualisation (Area Chart) – Adjusted sea-level over time, And the short description bellow explaining the visualisation and its data.



(You can also find the original .csv files, my .csv files with the transformed data, and any other resources in the mercury folder within the website link)

JavaScript – emission.js

```
function init() {

  var w = 700;
  var h = 400;

  var xPadding = 50;
  var yPadding = 15;

  var dataset;

  var rowConverter = function(d) {
    return {
      date: new Date(d.year),
      number: parseFloat(d.emission)
    };
  }

  d3.csv("australiaCO2Emission.csv", rowConverter).then(function(data) {

      dataset = data;

      lineChart(dataset);
  });

   function lineChart() {

       xScale = d3.scaleTime()
                  .domain ([
                    d3.min(dataset, function(d) { return d.date; }),
                    d3.max(dataset, function(d) { return d.date; }),
                  ])
                  .range([xPadding+40, w - xPadding]);

       yScale = d3.scaleLinear()
                   .domain([0, d3.max(dataset, function(d) { return d.number; })
                    ])
                   .range([h - yPadding - 30, yPadding]);


        line = d3.line()
                .x(function(d) { return xScale(d.date); })
                .y(function(d) { return yScale(d.number); });
```

```
area = d3.area()
        .x(function(d) { return xScale(d.date); })

        .y0(function() { return yScale.range()[0]; })

        .y1 (function(d) { return yScale(d.number); });


var svg = d3.select("#co2chart")
            .append("svg")
            .attr("id", "mysvg")
            .attr("width", w)
            .attr("height", h);

        svg.append("path")
            .datum(dataset)
            .attr("class", "line")
            .attr("d", line)
            .attr("stroke", "blue");


var xAxis = d3.axisBottom()
  .ticks(15)
  .scale(xScale);

svg.append("g")
  .attr("transform", "translate(0, "+ (h - yPadding -30) +")")
  .call(xAxis);

var yAxis = d3.axisLeft()
      .ticks(10)
      .scale(yScale);

svg.append("g")
  .attr("transform", "translate("+ (xPadding + 40) + ",0)")
  .call(yAxis);


svg.append("text")
    .attr("class", "halfMilLabel")
    .attr("x", xPadding + 300)
    .attr("y", yScale(0) + 32)
    .text("YEAR");
```

```
svg.append("text")
    .attr("class", "halfMilLabel")
    .attr("x", xPadding - 50)
    .attr("y", yScale(0) - 170)
    .text("Co2");

 svg.append("text")
    .attr("class", "halfMilLabel")
    .attr("x", xPadding - 50)
    .attr("y", yScale(0) - 160)
    .text("(kton)");


 d3.select("#buttonusa")
  .on("click", function() {

    d3.csv("americaCO2Emission.csv", rowConverter).then(function(data) {

      dataset = data;

      d3.select("#mysvg").remove();

      lineChart(dataset);

      d3.select(".line").attr("stroke", "red");
    });

  });


  d3.select("#buttonaus")
  .on("click", function() {

    d3.csv("australiaCO2Emission.csv", rowConverter).then(function(data) {

      dataset = data;

      d3.select("#mysvg").remove();

      lineChart(dataset);
    });

  });
```

```
      d3.select("#buttonchina")
      .on("click", function() {

        d3.csv("chinaCO2Emission.csv", rowConverter).then(function(data) {

          dataset = data;

          d3.select("#mysvg").remove();

          lineChart(dataset);

          d3.select(".line").attr("stroke", "green");
        });

      });

}


var datasetsea;

d3.csv("seaLevels.csv", rowConverter).then(function(data) {

    datasetsea = data;

    sealineChart(datasetsea);
});


 function sealineChart() {

    xScale = d3.scaleTime()
                .domain ([
                  d3.min(datasetsea, function(d) { return d.date; }),
                  d3.max(datasetsea, function(d) { return d.date; }),
                ])
                .range([xPadding+10, w - xPadding]);

    yScale = d3.scaleLinear()
                 .domain([0, d3.max(datasetsea, function(d) { return d.number; })
                  ])
                 .range([h - yPadding - 20, yPadding]);
```

```
line = d3.line()
           .x(function(d) { return xScale(d.date); })
           .y(function(d) { return yScale(d.number); });

area = d3.area()
       .x(function(d) { return xScale(d.date); })

       .y0(function() { return yScale.range()[0]; })

       .y1 (function(d) { return yScale(d.number); });


var svg = d3.select("#seachart")
          .append("svg")
          .attr("id", "mysvg")
          .attr("width", w)
          .attr("height", h);

      svg.append("path")
          .datum(datasetsea)
          .attr("class", "area")
          .attr("d", area)
          .attr("fill", "blue");


var xAxis = d3.axisBottom()
  .ticks(20)
  .scale(xScale);

svg.append("g")
  .attr("transform", "translate(0, "+ (h - yPadding -20) +")")
  .call(xAxis);

var yAxis = d3.axisLeft()
      .ticks(10)
      .scale(yScale);

svg.append("g")
  .attr("transform", "translate("+ (xPadding + 10) + "-0.1)")
  .call(yAxis);


svg.append("text")
    .attr("class", "halfMilLabel")
    .attr("x", xPadding + 300)
    .attr("y", yScale(0) + 34)
    .text("YEAR");
```

```
      svg.append("text")
         .attr("class", "halfMilLabel")
         .attr("x", xPadding - 50)
         .attr("y", yScale(0) - 170)
         .text("Sea-level");

      svg.append("text")
         .attr("class", "halfMilLabel")
         .attr("x", xPadding - 50)
         .attr("y", yScale(0) - 160)
         .text("(inches)");


      svg.append("line")
         .attr("class", "linehalfMilMark")

         .attr("x1", xScale(7))
         .attr("y1", yPadding)

         .attr("x2", xScale(7))
         .attr("y2", h - 35);

      svg.append("text")

         .attr("x", xPadding + 365)
         .attr("class", "mylabel")
         .attr("y", yScale(5))
         .text("1970")
         .attr("stroke", "red");


  }

}

window.onload = init;
```