BSC CREATIVE COMPUTING
YEAR 2: VISUALISATION AND SENSING

FINAL ASSIGNMENT:

ENERGY CONSUMPTION

& GDP per capita

NICOLE STOTT

TABLE OF CONTENT

ABOUT MY PROJECT

My project dives into the global energy landscape, exploring how different countries use energy sources and how it connects to their economic development.

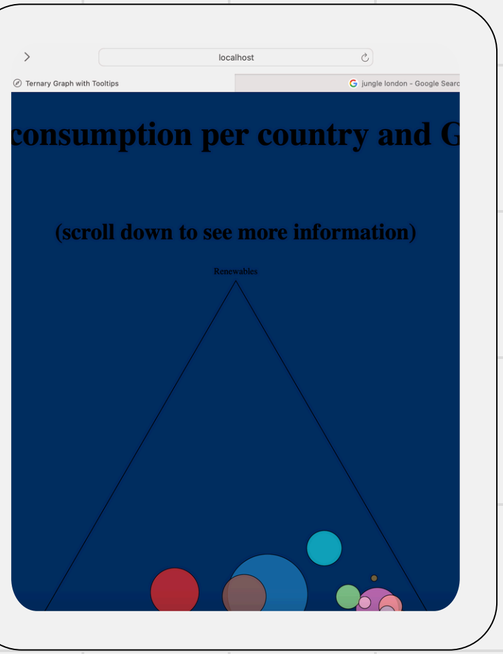
WHAT IS THE PROJECT

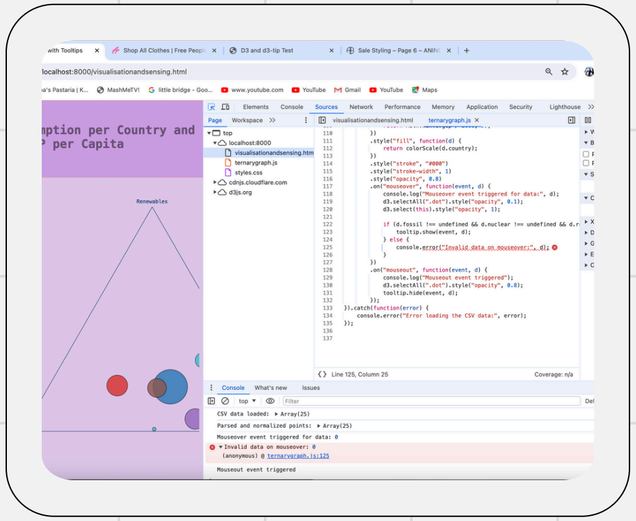
Through a ternary plot, I have displayed data on individual countries, visualising their consumption of fossil fuels, renewable energy, and nuclear energy. The size of each country's data point reflects its GDP per capita.

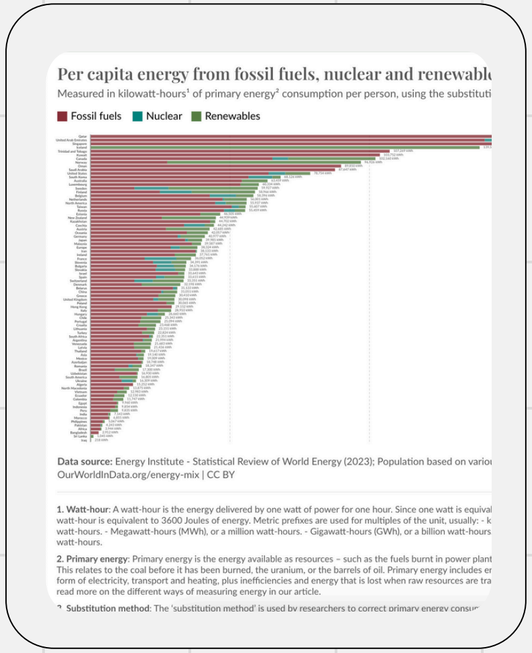
Complementing this, an amount (in tWh) over time graph depicts global energy consumption trends across an amplified variety of energy sources from 1965 to 2022.

IN CONLUSION...

By examining both individual countries and global trends, this project provides a multifaceted view of the world's energy consumption patterns.







RESEARCH & DESIGN

Research:

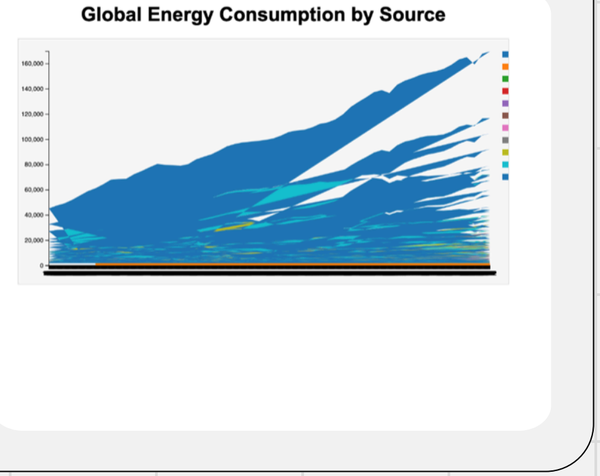
One particular event that started my interest in this field was Spain's decision to shut down its nuclear power plants. Energy use varies greatly across countries, influenced by factors such as resource availability, economic development, and technological advancement. My goal was to try and visualise these diverse data points using statistical methods to get a better understanding of global energy dynamics.

For my research I knew ‘Our World In Data.com’ had a wide variety of data for many topics including energy, I spent time looking at the many different tables and articles and finally decided on doing two graphs, one dataset includes data on fossil fuel, renewable energy, and nuclear energy consumption for numerous countries, and the other global energy consumption displaying a variety of fuels over the years, along with historical data spanning from 1965 to 2022. Furthermore, I extracted data from the world bank website and using pandas to extract the GDP per capita of the countries I had energy consumption information for.

	A	B	C
1	Country	GDP	
2	Asia	18532633	
3	Europe	4591100	
4	Asia	4110452	
5	Asia	3937011	
6	Europe	3495261	
7	Europe	3130014	
8	Europe	2328028	
9	Europe	2056844	
10	Oceania	1790348	
11	Asia	1760947	
12	Europe	1647114	
13	Asia	1475690	
14	Europe	1142513	
15	Asia	1113561	
16	Asia	1106015	
17	Europe	938458	
18	Europe	844623	

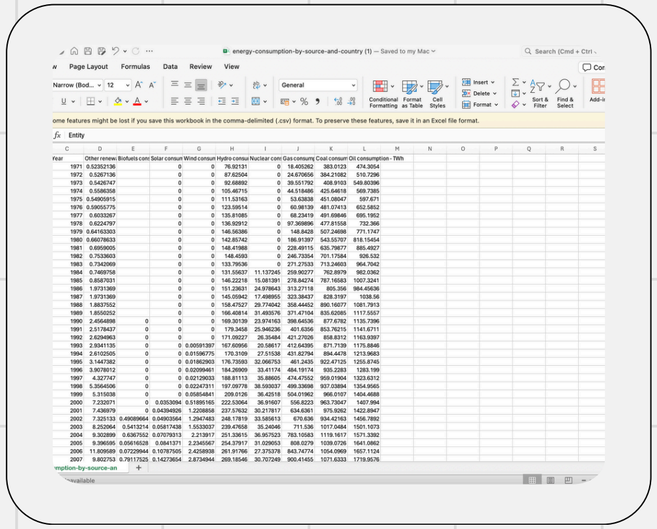
```python
import requests
import pandas as pd

#countries I need so it doesn't print out every gdp
countries = ["Mexico", "Azerbaijan", "Romania", "Brazil", "Uzbekistan", "Ukraine", "Algeria", "North Macedonia",
             "Vietnam", "Ecuador", "Colombia", "Egypt", "Indonesia", "Peru", "India", "Morocco", "Philippines",
             "Pakistan", "Bangladesh", "Sri Lanka", "Iraq", "Qatar", "United Arab Emirates", "Singapore",
             "Iceland", "Trinidad and Tobago", "Kuwait", "Canada", "Norway", "Oman", "Saudi Arabia",
             "United States", "South Korea", "Luxembourg", "Netherlands", "Taiwan", "Russia", "Estonia",
             "New Zealand", "Kazakhstan", "Germany", "Japan", "Malaysia", "Iran", "Ireland", "France",
             "Slovenia", "Bulgaria", "Slovakia", "Israel", "Spain", "Switzerland", "Denmark", "Belarus",
             "China", "Greece", "United Kingdom", "Poland", "Hong Kong", "Italy", "Hungary", "Chile",
             "Portugal", "Croatia", "Lithuania", "Turkey", "South Africa", "Latvia", "Thailand", "Australia",
             "Sweden", "Finland", "Belgium", "Czechia"]

# world bank API for gdp 2022 with help of https://datahelpdesk.worldbank.org/knowledgebase/articles/898599-indicator-api-queries
url = "https://api.worldbank.org/v2/country/all/indicator/NY.GDP.PCAP.CD?date=2022&format=json&per_page=300"

response = requests.get(url)
data = response.json()

# this code is to make sure only the countries I need are printed out
gdp_data = []
for entry in data[1]:
    country_name = entry['country']['value']
    gdp_value = entry['value']
    if country_name in countries:
        gdp_data.append({
            'Country': country_name,
            'GDP per Capita': gdp_value
        })

gdp_df = pd.DataFrame(gdp_data)

# I used stackoverflow as at the beginning I was only capable of seeing the first and last five countries
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', 1000)
pd.set_option('display.colheader_justify', 'left')

print(gdp_df)

gdp_df.to_csv("gdp_per_capita_2022.csv", index=False)
```

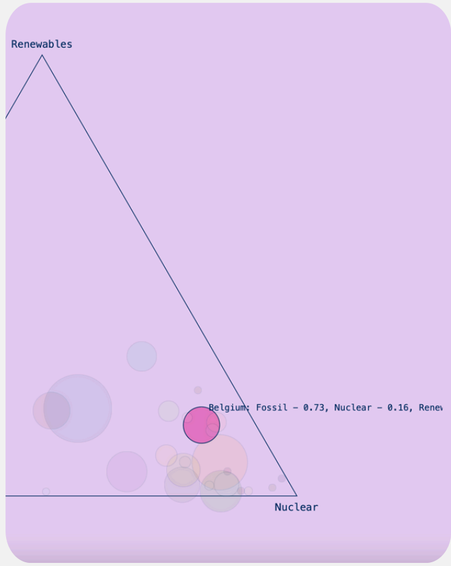
```javascript
const margin = { top: 20, right: 150, bottom: 30, left: 50 },
      width = 1000 - margin.left - margin.right,
      height = 400 - margin.top - margin.bottom;

const svg = d3.select("#chart").append("svg")
  .attr("width", width + margin.left + margin.right)
  .attr("height", height + margin.top + margin.bottom)
  .append("g")
  .attr("transform", `translate(${margin.left},${margin.top})`);

d3.csv("energy-consumption-by-source-and-country.csv").then(function(data) {
    console.log("Data loaded:", data);

    const globalData = data.filter(d => d.Entity === "World");

    globalData.forEach(d => {
        d.Year = +d.Year;
        d["Other renewables"] = +d["Other renewables (including geothermal and biomass) – TWh"];
        d.Biofuels = +d["Biofuels consumption – TWh"];
        d.Solar = +d["Solar consumption – TWh"];
        d.Wind = +d["Wind consumption – TWh"];
        d.Hydropower = +d["Hydro consumption – TWh"];
        d.Nuclear = +d["Nuclear consumption – TWh"];
        d.Gas = +d["Gas consumption – TWh"];
        d.Coal = +d["Coal consumption – TWh"];
        d.Oil = +d["Oil consumption – TWh"];
    });

    const keys = ["Other renewables", "Biofuels", "Solar", "Wind", "Hydropower", "Nuclear", "Gas",
    console.log("Keys:", keys);

    const filteredData = globalData.filter(d => d.Year && !isNaN(d.Year));

    const x = d3.scaleLinear()
        .domain(d3.extent(filteredData, d => d.Year))
        .range([0, width]);

    svg.append("g")
        .attr("transform", `translate(0,${height})`)
        .call(d3.axisBottom(x).tickFormat(d3.format("d")));

    const y = d3.scaleLinear()
        .domain([0, d3.max(filteredData, d => Math.max(...keys.map(key => d[key])))])
        .range([height, 0]);

    svg.append("g")
        .call(d3.axisLeft(y));

    const color = d3.scaleOrdinal()
        .domain(keys)
        .range(d3.schemeCategory10);

    const tooltip = d3.select("body").append("div")
```



	A	B	C	D	E
1	Country	Code	Fossil Fuels k	Nuclear kWh	Renewables (
2	Belarus	BLR	29592.15	1230.5691	309.97775
3	Belgium	BEL	42769.266	9386.54	6240.0186
4	Brazil	BRA	8699.518	169.1658	8431.066
5	Bulgaria	BGR	23789.814	6067.9277	4317.78
6	Canada	CAN	65239.92	5636.829	31283.535
7	China	CHN	25344.26	733.05255	4974.1704
8	Czechia	CZE	33561.6	7395.684	3284.8643
9	Finland	FIN	24868.705	11396.982	22700.47
10	France	FRA	19371.66	11409.492	5270.6436
11	Germany	DEU	31225.268	1041.2916	8710.929
12	Hungary	HUN	20260.748	3968.923	2430.8186
13	India	IND	6318.812	81.54943	743.044
14	Japan	JPN	33936.066	1044.9875	5003.856
15	Mexico	MEX	17093.922	212.88353	1702.3807
16	Netherlands	NLD	47356.742	592.2688	8051.7456
17	Pakistan	PAK	3545.5544	236.38889	460.81006
18	Romania	ROU	13614.338	1411.162	3321.3687
19	Slovenia	SVN	22428.342	6615.506	5347.2437
20	South Africa	ZAF	21062.77	421.88177	866.2472
21	Spain	ESP	23462.137	3082.4805	7070.7124
22	Sweden	SWE	15754.809	12223.56	31948.436
23	Switzerland	CHE	15704.048	6615.6626	11031.041
24	Ukraine	UKR	11185.414	3911.1982	1212.5131
25	United Kingdo	GBR	22509.117	1768.5579	5820.2104
26	United State	USA	63835.86	6006.1494	8912.268

```python
import pandas as pd

# Load the datasets
energy_file_path = 'energy_consumption_2022_full.csv'
gdp_file_path = 'gdp_per_capita_2022.csv'

energy_data = pd.read_csv(energy_file_path)
gdp_data = pd.read_csv(gdp_file_path)

# Ensure the correct column names
energy_data.rename(columns={'Entity': 'Country'}, inplace=True)

# Merge the datasets on the 'Country' column
merged_data = pd.merge(energy_data, gdp_data, on='Country')

# Filter countries with all energy values greater than 1
filtered_data = merged_data[(merged_data['Fossil Fuels'] > 1) &
                            (merged_data['Nuclear'] > 1) &
                            (merged_data['Renewables'] > 1)].copy()

# Save the filtered data to a new CSV file
output_path = 'filtered_merged_data_no_percentages.csv'
filtered_data.to_csv(output_path, index=False)

# Print the filtered data to check
print(filtered_data.head())
```

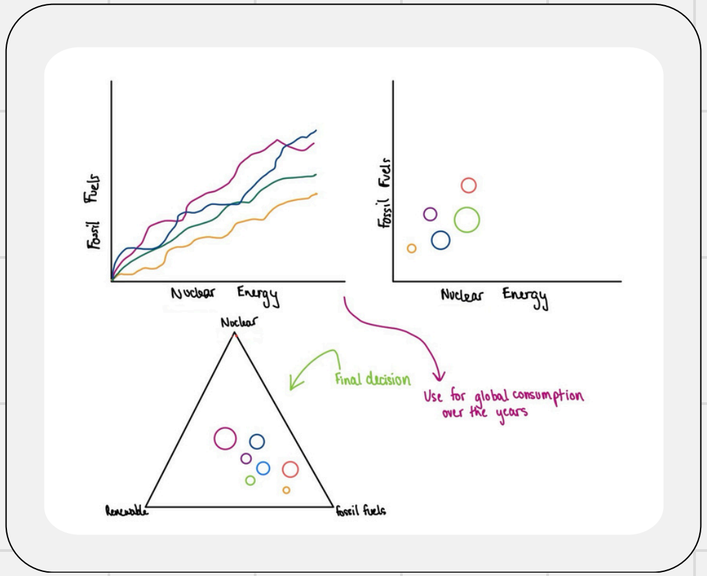


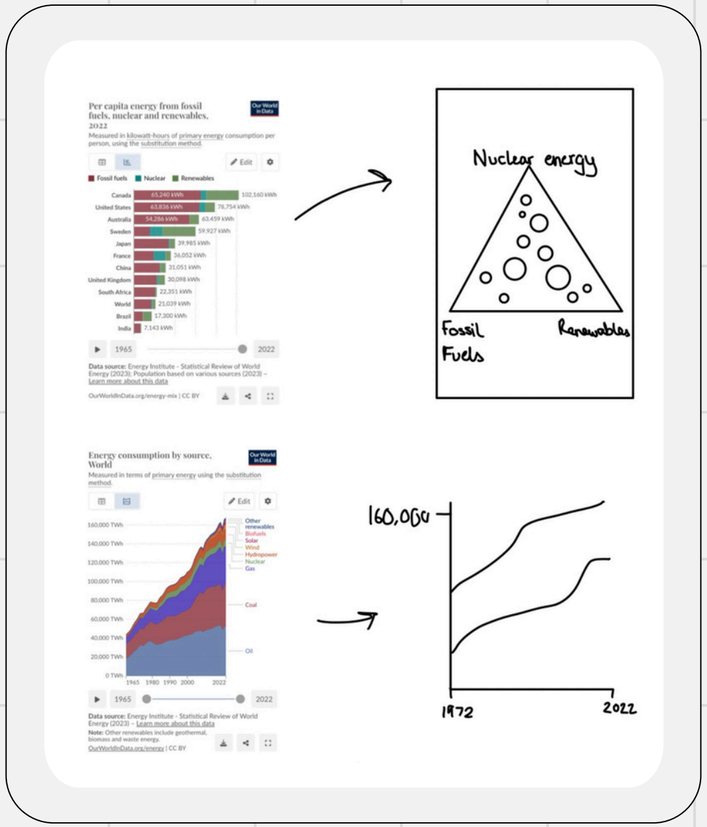


NICOLE STOTT

5/12

on the left is the code for the csv files and on the right are some progress pictures of my graphs





graph 1 > styles.css > footer

```css
body {
    margin: 0;
    padding: 0;
    background: rgba(180, 128, 207, 0.3);
    display: flex;
    flex-direction: column;
    align-items: center;
    font-family: monospace;
    color: rgb(105, 84, 116);
}

header {
    width: 100%;
    background: rgb(204, 156, 229);
    text-align: center;
    padding: 30px 0;
}

main {
    padding: 10px;
    width: 100%;
    max-width: 1200px;
    box-sizing: border-box;
    display: flex;
    flex-direction: column;
    align-items: center;
}

h1 {
    font-family: monospace;
    padding: 10px;
    text-align: center;
    font-size: 32px;
    margin-bottom: 32px;
}

footer {
    text-align: center;
    width: 100%;
    color: rgb(105, 84, 116);
    background: rgb(204, 156, 229);
    padding: 50px;
}

/* Tooltip styles */
.tooltip-label {
    color: rgb(105, 84, 116);
    background: rgb(204, 156, 229, 0.8);
    border: 1px solid #000;
    padding: 5px;
    font-family: monospace;
    font-size: 12px;
    max-width: 200px;
    word-wrap: break-word;
    border-radius: 4px;
}

#ternary-plot {
    display: flex;
    justify-content: center;
    align-items: center;
    width: 100%;
    margin-top: 20px;
}
```

CSS file for ternary graph

RESEARCH & DESIGN

Design :

For my first designs (left above), I used the Ipad to sketch out different options, initially only looking for a way to plot the fuels for each country to compare and contrast the differences, then when I decided I really wanted to include the three energies I was thinking on doing a three-axis graph but I decided that the visualisation would not be clear enough and thats when my research encountered a graph from the economist on modes of transportation which looked really interesting and not only did it allow me to include the three energies but it also let me introduce a new factor which would be represented by the size of the plotted data point.

The second graph I kept similar to the one on ‘Our World In Data.com’ because I wanted a simple graph that really highlighted the use over time, because eventhough we are using other types of fuels which are less harmful, the harmful fuels consumption keeps increasing alarmingly.

I made every line/cirle por the plotted data on the graph a different colour to make sure they were distinctive from each other and all colours contrast well with both white and light purple backgrounds.

DESIGN:

Interactivness

I have made both my graphs graph interactive. The top one is the ternary graph which I have made interactive by incorporating hover effects that provide additional details about each data point and makes the rest of the data points fade away. When a user hovers over a dot, it displays a tooltip that displays the information of energy consumption and gdp per capita. In the second graph I have applied the same aesthetic but instead of the other lines fading away, the line that is selected gets thicker, this boosts user engagement and gives a more detailed exploration of the data.

graph 2 > styling.css > ...

```css
body {
    margin: 0;
    padding: 30;
    background: rgba(180, 128, 207, 0.3);
    display: flex;
    flex-direction: column;
    align-items: center;
    font-family: monospace;
    color: rgb(105, 84, 116);
}

header {
    width: 100%;
    background: rgb(204, 156, 229);
    text-align: center;
    padding: 30px 0;
}

h1 {
    font-family: monospace;
    color: rgb(105, 84, 116);
    padding: 10px;
    text-align: center;
    font-size: 32px;
    margin-bottom: 32px;
}

footer {
    text-align: center;
    width: 100%;
    color: rgb(105, 84, 116);
    background: rgb(204, 156, 229);
    padding: 50px;
}

#chart {
    display: flex;
    justify-content: center;
    margin: 100px;
}

svg {
    background-color: #f9f9f9;
    border: 1px solid #000000;
    padding: 70px;
}

.tooltip {
    position: absolute;
    background: rgb(204, 156, 229);
    color:rgb(105, 84, 116);
    border-radius: 5px;
    padding: 10px;
    pointer-events: none;
    opacity: 0;
}
```

CSS file for line graph

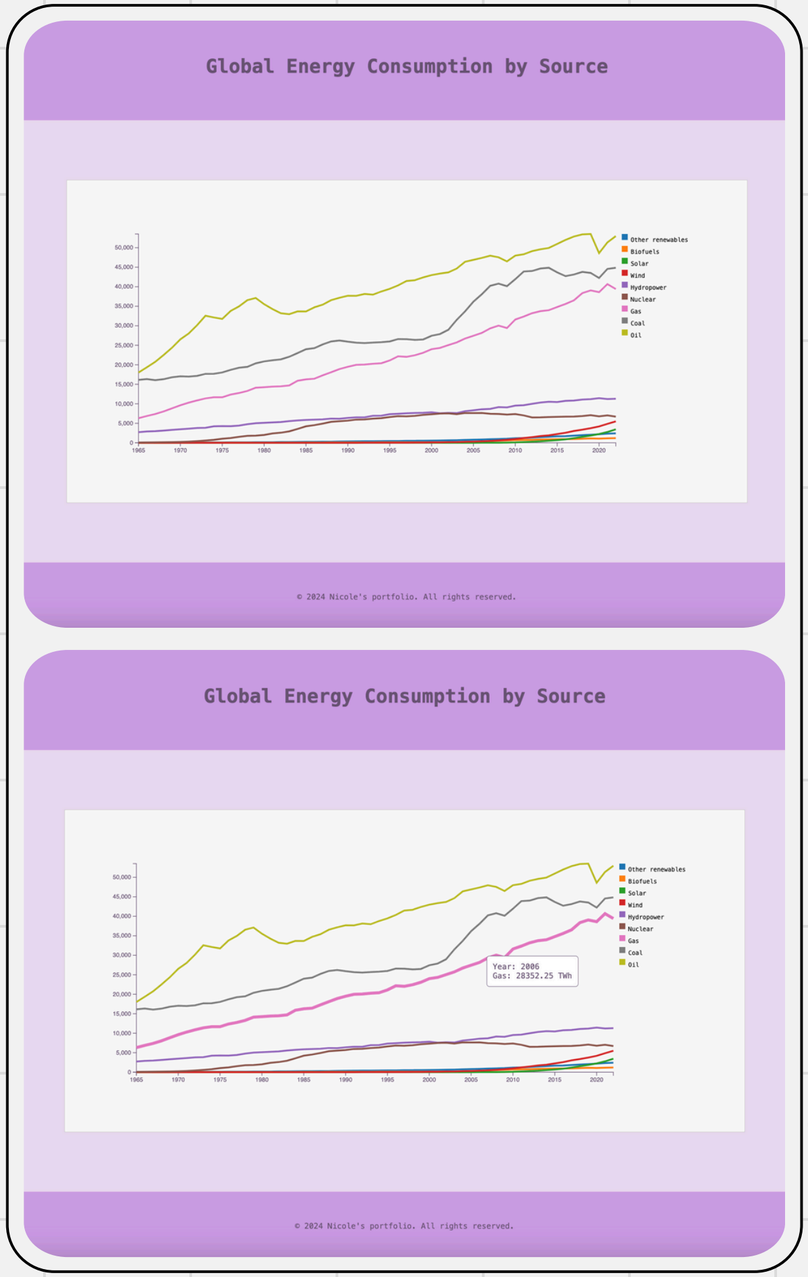
```
        .data(points)
        .enter().append("circle")
        .attr("class", "dot")
        .attr("cx", function(d) {
            var coords = ternaryToCartesian(d.fossil, d.nuclear, d.renewables);
            return coords.x;
        })
        .attr("cy", function(d) {
            var coords = ternaryToCartesian(d.fossil, d.nuclear, d.renewables);
            return coords.y;
        })
        .attr("r", function(d) {
            return Math.max(d.gdp / 2000, 5);
        })
        .style("fill", function(d) {
            return colorScale(d.country);
        })
        .style("stroke", "#000")
        .style("stroke-width", 1)
        .style("opacity", 0.8)
        .on("mouseover", function(event, d) {
            d3.selectAll(".dot").style("opacity", 0.1);
            d3.select(this).style("opacity", 1);

            var xPosition = parseFloat(d3.select(this).attr("cx")) + 10;
            var yPosition = parseFloat(d3.select(this).attr("cy")) - 10;

            console.log(`Label Position: (${xPosition}, ${yPosition})`);

            g.append("foreignObject")
                .attr("id", "hoverValue")
                .attr("x", xPosition)
                .attr("y", yPosition)
                .attr("width", 200)
                .attr("height", 100)
                .append("xhtml:div")
                .attr("class", "tooltip-label")
                .html(`${d.country}:<br/>Fossil - ${d.fossil.toFixed(2)},<br/>Nuclear - ${d.nuclear.toFixed(2)},<br/>Renewables - ${d.renewables.toFixed(2)},<
        })
        .on("mouseout", function() {
            d3.selectAll(".dot").style("opacity", 0.8);
            g.select("#hoverValue").remove();
        });
}).catch(function(error) {
    console.error("Error loading the CSV data:", error);
```

```
                d3.select(this).attr("stroke-width", 5);
                tooltip.style("opacity", 1);
            })
            .on("mousemove", function(event, d) {
                const year = x.invert(d3.pointer(event)[0]);
                const value = y.invert(d3.pointer(event)[1]);
                tooltip
                    .html(`Year: ${Math.round(year)}<br>${key}: ${value.toFixed(2)} TWh`)
                    .style("left", (event.pageX + 10) + "px")
                    .style("top", (event.pageY - 10) + "px");
            })
            .on("mouseout", function() {
                d3.select(this).attr("stroke-width", 3);
                tooltip.style("opacity", 0);
            });
    });

    const legend = svg.append("g")
        .attr("transform", `translate(${width + 10}, 0)`);

    keys.forEach((key, i) => {
        const legendRow = legend.append("g")
            .attr("transform", `translate(0, ${i * 20})`);

        legendRow.append("rect")
            .attr("width", 10)
            .attr("height", 10)
            .attr("fill", color(key));

        legendRow.append("text")
            .attr("x", 15)
            .attr("y", 10)
            .attr("dy", ".35em")
            .attr("text-anchor", "start")
            .style("font-size", "10px")
            .text(key);
    });
}).catch(function(error) {
    console.error("Error loading the CSV data:", error);
```

LINE GRAPH

I wanted to use this graph to remind people that even though we are making many efforts to use more sustainable fuels the increase of these are still very alarming and people are also focusing on short-term not long-term solutions. For example, for oil there is a 5.4% increase in usage between 2010 and 2022 whereas for nuclear there is a 9.1% decrease when there is even an article displaying the information proving nuclear is more than 600 times safer and more than 100 times cleaner.

I made the decision to keep the background of this graph white because I wanted to highlight even the most minimal changes in the energy consumption.

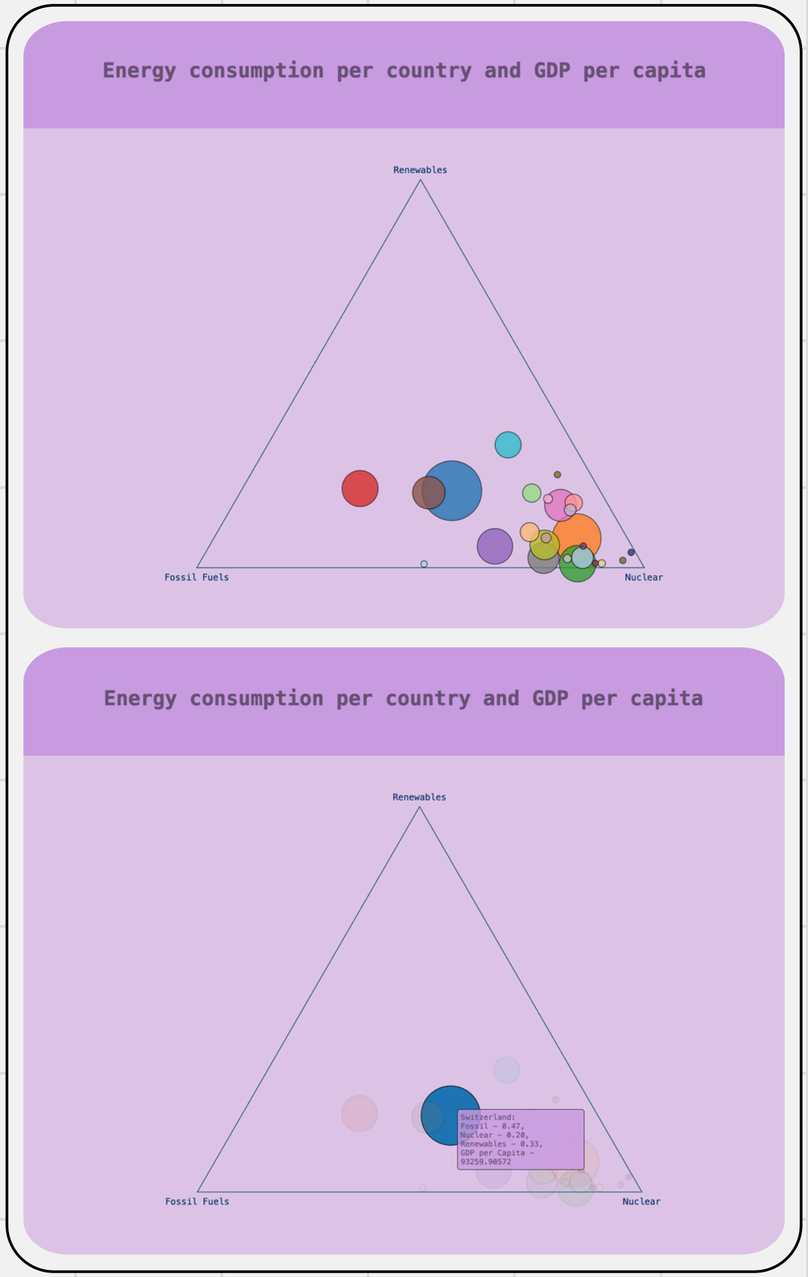



```
const margin = { top: 20, right: 150, bottom: 30, left: 50 },
      width = 1000 - margin.left - margin.right,
      height = 400 - margin.top - margin.bottom;

const svg = d3.select("#chart").append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
    .append("g")
    .attr("transform", `translate(${margin.left},${margin.top})`);

d3.csv("energy-consumption-by-source-and-country.csv").then(data
=> {
    console.log("Data loaded:", data);

    const globalData = data.filter(d => d.Entity === "World");
```

```
    const x = d3.scaleLinear()
        .domain(d3.extent(filteredData, d => d.Year))
        .range([0, width]);

    svg.append("g")
        .attr("transform", `translate(0,${height})`)
        .call(d3.axisBottom(x).tickFormat(d3.format("d")));

    const y = d3.scaleLinear()
        .domain([0, d3.max(filteredData, d => Math.max(...keys.map
        (key => d[key])))])
        .range([height, 0]);

    svg.append("g")
        .call(d3.axisLeft(y));
```

The top image is defining margins so all the axis and labels of the colour chart, below the axis are defined and translated if not in the correct position.

The image on the left top, is the graph normally and the left bottom is how it looks when you hover the mouse over one of the lines and the information of that data point is displayed.




```d3
d3.csv("filtered_merged_data.csv").then(function(data) {
    var width = 800,
        height = 800,
        margin = 50,
        additionalMargin = 150;

    var side = Math.min(width, height) - margin * 2;
    var heightEquilateral = Math.sqrt(3) / 2 * side;

    var svg = d3.select("#ternary-plot").append("svg")
        .attr("width", width + margin * 2 + additionalMargin)
        .attr("height", height + margin * 2)
        .style("display", "block")
        .style("margin", "0 auto");
```

```d3
    g.selectAll(".label")
        .data(labels)
        .enter().append("text")
        .attr("class", "label")
        .attr("x", function(d, i) {
            return i === 0 ? 0 : (i === 1 ? side : side / 2);
        })
        .attr("y", function(d, i) {
            return i === 0 ? heightEquilateral + 20 : (i === 1 ?
            heightEquilateral + 20 : -10);
        })
        .style("text-anchor", "middle")
        .style("font-size", "14px")
        .text(function(d) {
            return d;
        });
```

The top image is defining margins so all the axis and information labels are visible, below, the axis are defined and I styled the font size and placement ther because I was struggling in the CSS file.

The image on the left top, is the graph normally and the left bottom is how it looks when you hover the mouse over one of the circles and the information of that data point is displayed.

LINE GRAPH

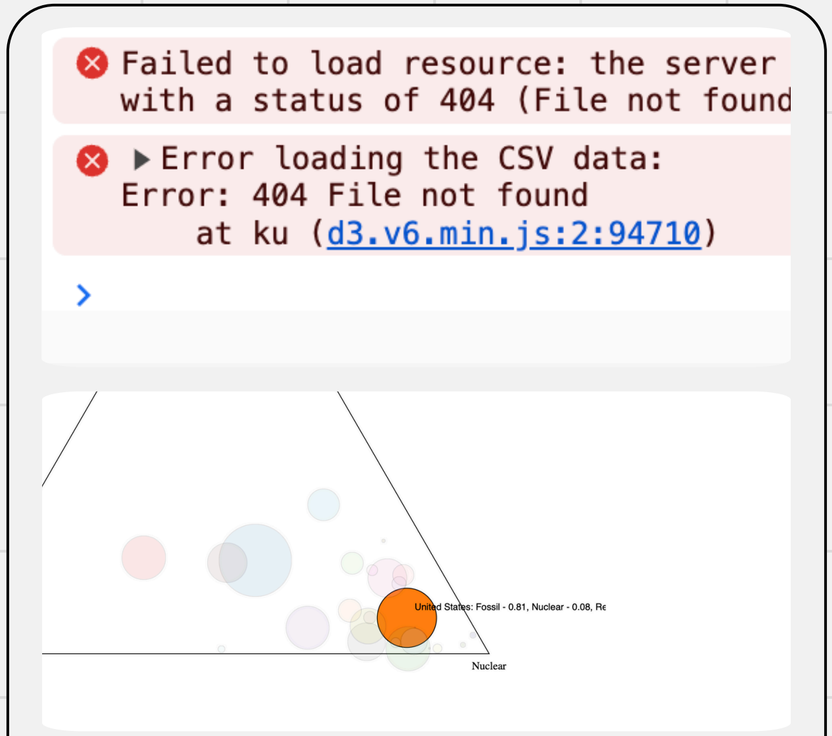
This graph I wanted use to visualise the relation of energy consumption (fossil fuels being the most dangerous and the dirtiest compared to renewable or nuclear) to GDP per capita, showing that even though some countries may be richer and could afford to invest in using more sustainable energy sources, such as the United States (Orange).

In the information label I have displayed the energy consumptions as percentages (of the number one) to make it simpler to understand the consumption of each country as an individual because the visual graphs is what should be used to compare the differences between countries.

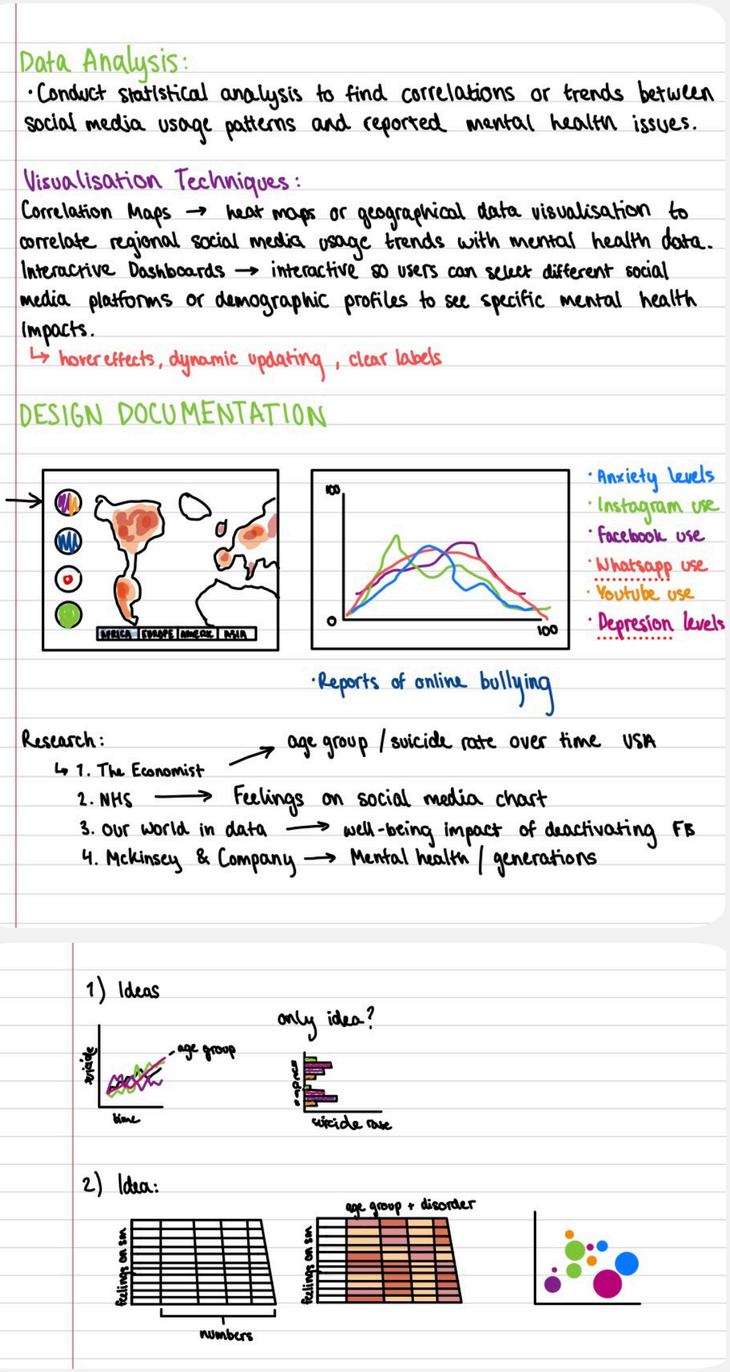
IDEAS & IMPROVEMENTS

My initial idea was to create a variety of graphs that would showcase social media's influence on mental health. I started my research only to encounter a wide range of artciles which had different conclusions on how social media had improved mental health by creating awareness and making mental health services more accessible, and other stating that it had negatively impacted mental health as cyber-bullying had increased or by showing unrealistic body types. I discarded this Idea as there was no precise conclusion and my research also started to show me how many aspects actually affect our mental health which would mean the data isnt strongly correlated.

NICOLE STOTT



The line graph was fairly straightforward to make, despite some bumps, but I eally struggled when adding the labels to the ternary graph. I spent quite a lot of time and managed to solve the issue by adding larger margins to the graph so labels weren't cut off and adding word-wrap to the css styling of the d3.tooltip. Something I would've liked to do but didn't realise until later would be creating the graphs in colours that are easier to read for the visually impaired, or making the two purples more contrasting.



JUNE 2024

THANK YOU

1

Pfenninger, S., & Kirstead, J. (2015, May 15). *Renewables, nuclear, or fossil fuels? scenarios for Great Britain's power system considering costs, emissions and Energy Security*. science direct.

2

Holtz, Y. (n.d.). *Building tooltips with d3.js*. The D3 Graph Gallery – Simple charts made with d3.js.

3

Ritchie, H., & Roser, M. (2024, March 20). *What are the safest and cleanest sources of energy?*. Our World in Data.

4

Ritchie, H., Rosado, P., & Roser, M. (2024, March 25). *Energy mix*. Our World in Data.

REFERENCES

AI tool used: OpenAI ChatGPT 3.5
Purpose: creating a varied colour palette
Section: const customColors = [
 "#1f77b4", ...];

AI tool used: OpenAI ChatGPT 3.5
Purpose: struggling to plot coordinates on ternary graph
Section: var y = heightEquilateral - (heightEquilateral * nuclear);
var x = (fossil + 0.5 * nuclear) * side;