

FINAL ASSIGNMENT:

# ENERGY CONSUMPTION

& GDP per capita

# TABLE OF CONTENT



PAGE

3

ABOUT

4

RESEARCH

6

DESIGN

8

LINE GRAPH

9

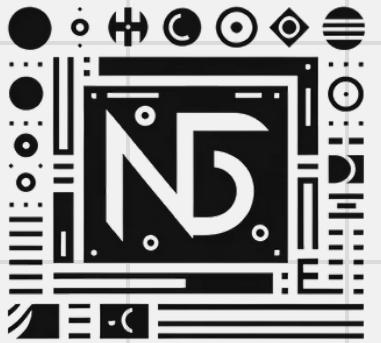
TERNARY GRAPH

10

IDEAS & IMPROVEMENTS

12

REFERENCES



# 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.



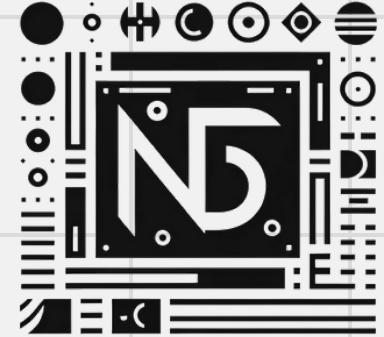
Complementing this, an amount (in tWh) over time graph depicts global energy consumption trends across an amplified variety of energy sources from 1965 to 2022.



## IN CONLUSION...

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

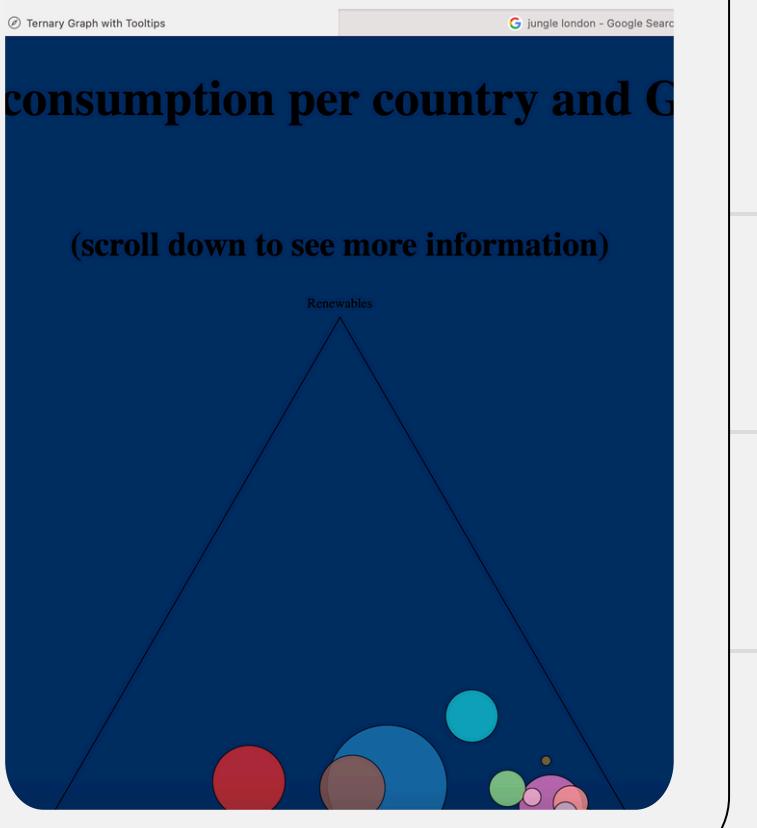
# RESEARCH & DESIGN



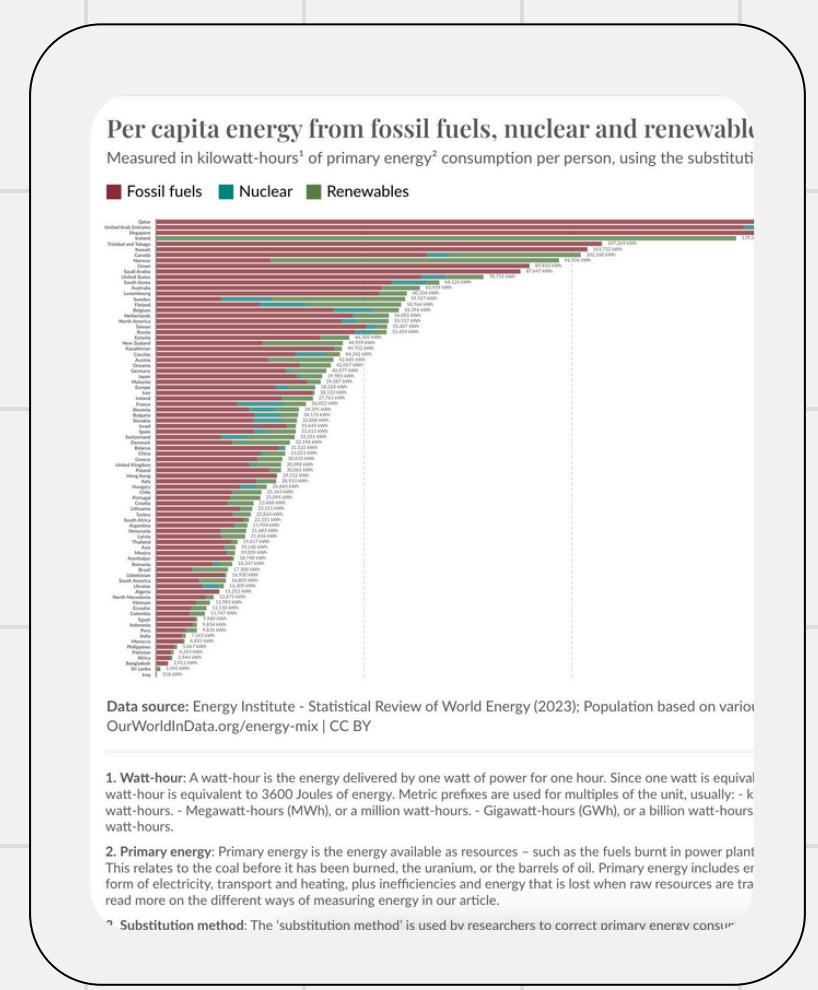
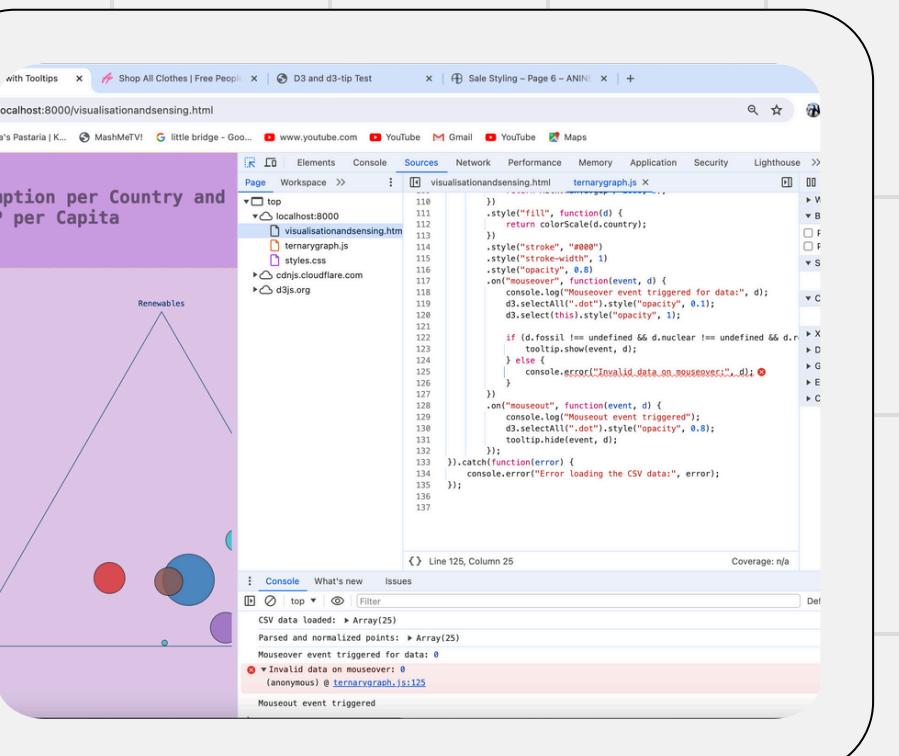
## Research:

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.



```
aph 2 > energyovertime.html ...  
1  <!DOCTYPE html>  
2  <html lang="en">  
3  <head>  
4  |   <meta charset="UTF-8">  
5  |   <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">  
6  |   <title>Global Energy Consumption by Source</title>  
7  |   <link rel="stylesheet" href="styling.css">  
8  </head>  
9  <body>  
10 |   <header>  
11 |   |   <h1>Global Energy Consumption by Source</h1>  
12 |   </header>  
13 |   <div id="chart"></div>  
14 |   <script src="https://d3js.org/d3.v7.min.js"></script>  
15 |   <script src="energycons.js"></script>  
16 |   <footer>&copy; 2024 Nicole's portfolio. All rights reserved.</footer>  
17 </body>  
18 </html>
```



	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	

```

n1> cd data scraping
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'][1]['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)

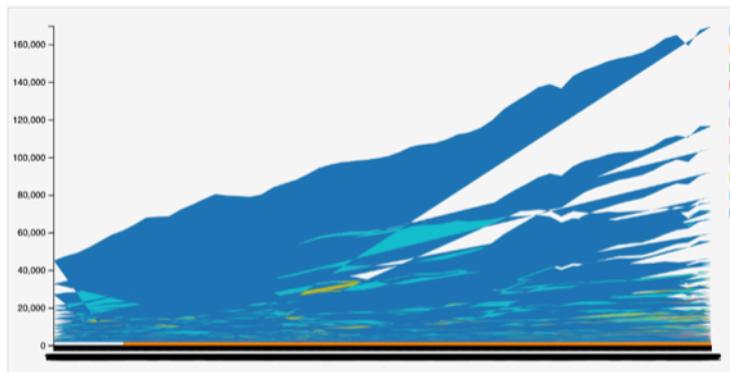
# I used stackoverflow as at the beginning I was only capable of seeing the first and last five countries
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', 1000)
pd.set_option('display.colheader_justify', 'left')

print(gdp_df)

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

```

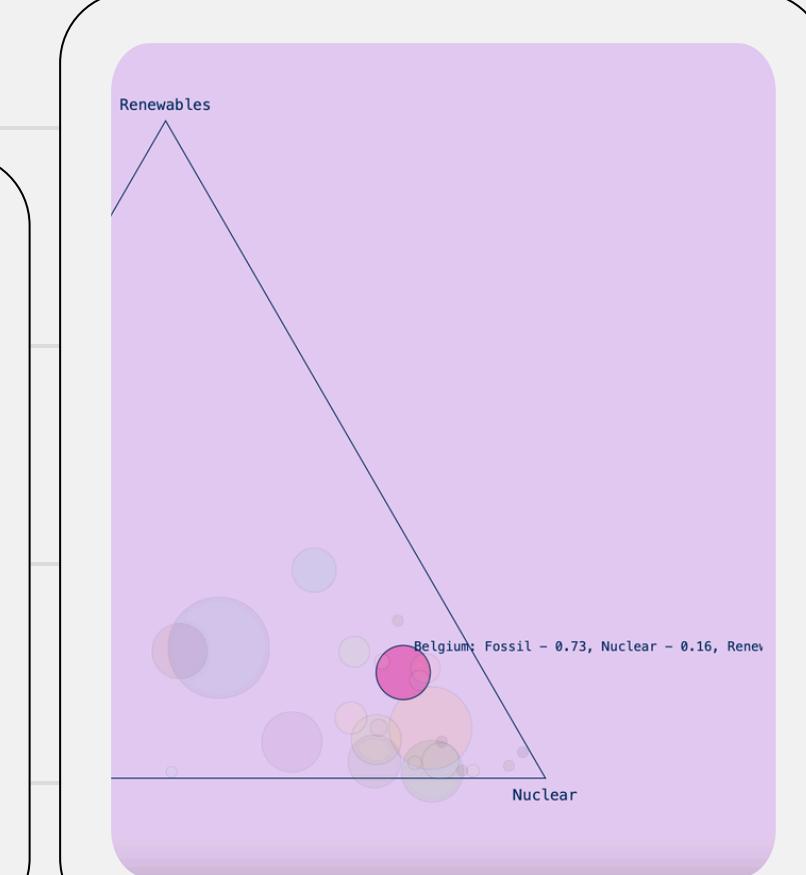
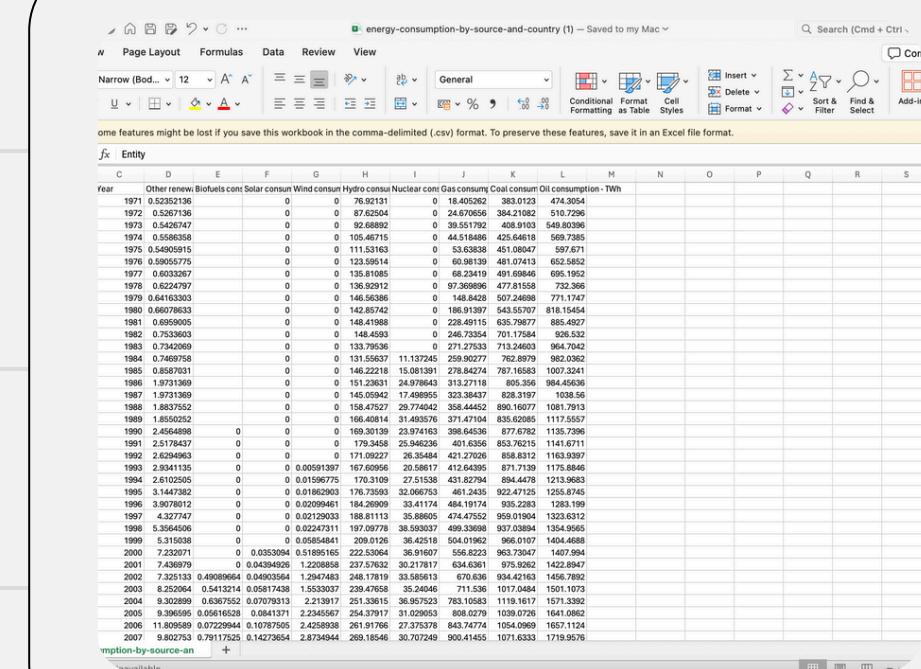
```
1 const margin = { top: 20, right: 150, bottom: 30, left: 50 },
2   width = 1000 - margin.left - margin.right,
3   height = 400 - margin.top - margin.bottom;
4
5 const svg = d3.select("#chart").append("svg")
6   .attr("width", width + margin.left + margin.right)
7   .attr("height", height + margin.top + margin.bottom)
8   .append("g")
9   .attr("transform", `translate(${margin.left},${margin.top})`);
10
11 d3.csv("energy-consumption-by-source-and-country.csv").then(function(data) {
12   console.log("Data loaded:", data);
13
14   const globalData = data.filter(d => d.Entity === "World");
15
16   globalData.forEach(d => {
17     d.Year = +d.Year;
18     d["Other renewables"] = +d["Other renewables (including geothermal and biomass) - TWh"];
19     d.Biofuels = +d["Biofuels consumption - TWh"];
20     d.Solar = +d["Solar consumption - TWh"];
21     d.Wind = +d["Wind consumption - TWh"];
22     d.Hydropower = +d["Hydro consumption - TWh"];
23     d.Nuclear = +d["Nuclear consumption - TWh"];
24     d.Gas = +d["Gas consumption - TWh"];
25     d.Coal = +d["Coal consumption - TWh"];
26     d.Oil = +d["Oil consumption - TWh"];
27   });
28
29   const keys = ["Other renewables", "Biofuels", "Solar", "Wind", "Hydropower", "Nuclear", "Gas"];
30   console.log("Keys:", keys);
31
32   const filteredData = globalData.filter(d => d.Year && !isNaN(d.Year));
33
34   const x = d3.scaleLinear()
35     .domain(d3.extent(filteredData, d => d.Year))
36     .range([0, width]);
37
38   svg.append("g")
39     .attr("transform", `translate(0,${height})`)
40     .call(d3.axisBottom(x).tickFormat(d3.format("d")));
41
42   const y = d3.scaleLinear()
43     .domain([0, d3.max(filteredData, d => Math.max(...keys.map(key => d[key])))])
44     .range([height, 0]);
45
46   svg.append("g")
47     .call(d3.axisLeft(y));
48
49   const color = d3.scaleOrdinal()
50     .domain(keys)
51     .range(d3.schemeCategory10);
52
53   const tooltip = d3.select("body").append("div")
```



A	B	C	D	E
Country	Code	Fossil Fuels k	Nuclear kWh	Renewables
Larus	BLR	29592.15	1230.5691	309.97775
gium	BEL	42769.266	9386.54	6240.0186
azil	BRA	8699.518	169.1658	8431.066
lgaria	BGR	23789.814	6067.9277	4317.78
nada	CAN	65239.92	5636.829	31283.535
ina	CHN	25344.26	733.05255	4974.1704
echia	CZE	33561.6	7395.684	3284.8643
land	FIN	24868.705	11396.982	22700.47
rance	FRA	19371.66	11409.492	5270.6436
ermany	DEU	31225.268	1041.2916	8710.929
ngary	HUN	20260.748	3968.923	2430.8186
lia	IND	6318.812	81.54943	743.044
an	JPN	33936.066	1044.9875	5003.856
exico	MEX	17093.922	212.88353	1702.3807
etherlands	NLD	47356.742	592.2688	8051.7456
kistan	PAK	3545.5544	236.38889	460.81006
mania	ROU	13614.338	1411.162	3321.3687
ovenia	SVN	22428.342	6615.506	5347.2437
uth Africa	ZAF	21062.77	421.88177	866.2472
ain	ESP	23462.137	3082.4805	7070.7124
eden	SWE	15754.809	12223.56	31948.436
itzerland	CHE	15704.048	6615.6626	11031.041
raine	UKR	11185.414	3911.1982	1212.5131
ited Kingdom	GBR	22509.117	1768.5579	5820.2104
ited States	USA	63835.86	6006.1494	8912.268

```
File | < > v sensing
v sensing } No Selection

1 import pandas as pd
2
3 # Load the datasets
4 energy_file_path = 'energy_consumption_2022_full.csv'
5 gdp_file_path = 'gdp_per_capita_2022.csv'
6
7 energy_data = pd.read_csv(energy_file_path)
8 gdp_data = pd.read_csv(gdp_file_path)
9
10 # Ensure the correct column names
11 energy_data.rename(columns={'Entity': 'Country'}, inplace=True)
12
13 # Merge the datasets on the 'Country' column
14 merged_data = pd.merge(energy_data, gdp_data, on='Country')
15
16 # Filter countries with all energy values greater than 1
17 filtered_data = merged_data[(merged_data['Fossil Fuels'] > 1) &
18                               (merged_data['Nuclear'] > 1) &
19                               (merged_data['Renewables'] > 1)].copy()
20
21 # Save the filtered data to a new CSV file
22 output_path = 'filtered_merged_data_no_percentages.csv'
23 filtered_data.to_csv(output_path, index=False)
24
25 # Print the filtered data to check
26 print(filtered_data.head())
27
```



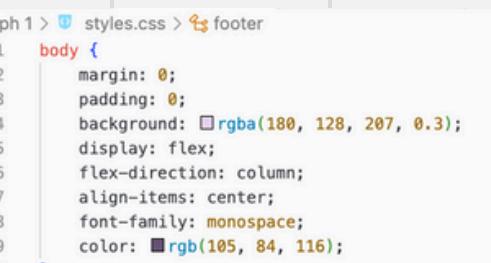
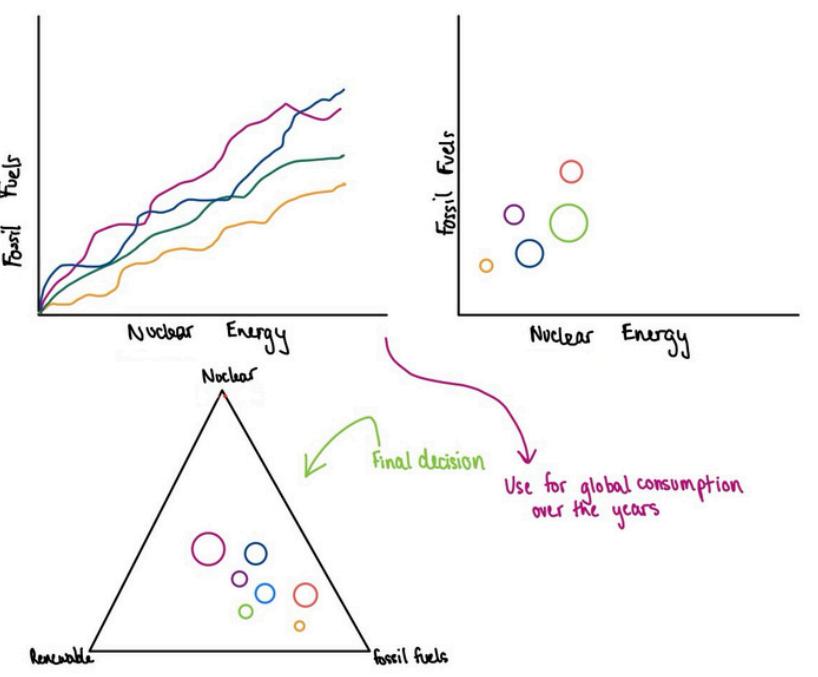
# RESEARCH & DESIGN

## 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 even though we are using other types of fuels which are less harmful, the harmful fuels consumption keeps increasing alarmingly.

I made every line/circle 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.



```
graph 1 > styles.css > footer
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;
}

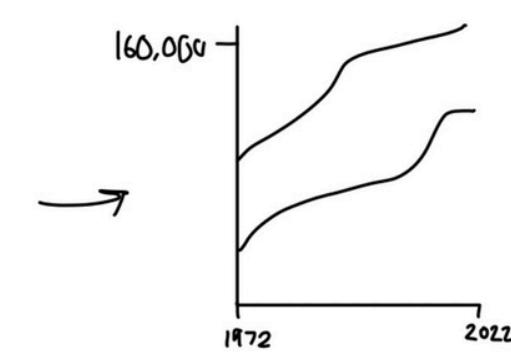
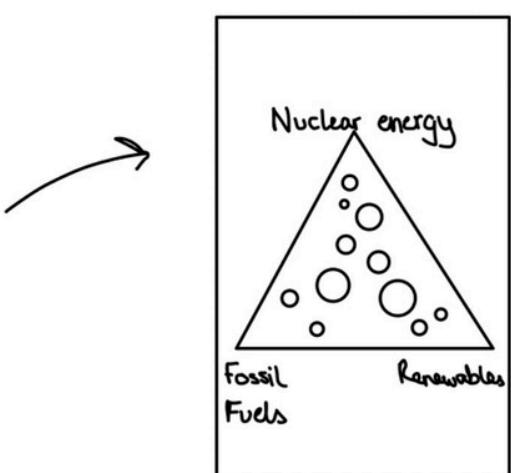
h1 {
  font-family: monospace;
  padding: 10px;
  text-align: center;
  font-size: 32px;
  margin-bottom: 32px;
}

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



CSS file for ternary graph





# DESIGN:

# Interactivity

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 LR
    1 body {
        2     margin: 0;
        3     padding: 30px;
        4     background: #rgba(180, 128, 207, 0.3);
        5     display: flex;
        6     flex-direction: column;
        7     align-items: center;
        8     font-family: monospace;
        9     color: #rgb(105, 84, 116);
    }
    10
    11 header {
        12     width: 100%;
        13     background: #rgb(204, 156, 229);
        14     text-align: center;
        15     padding: 30px 0;
    }
    16
    17 h1 {
        18     font-family: monospace;
        19     color: #rgb(105, 84, 116);
        20     padding: 10px;
        21     text-align: center;
        22     font-size: 32px;
        23     margin-bottom: 32px;
    }
    24
    25 footer {
        26     text-align: center;
        27     width: 100%;
        28     color: #rgb(105, 84, 116);
        29     background: #rgb(204, 156, 229);
        30     padding: 50px;
    }
    31
    32 #chart {
        33     display: flex;
        34     justify-content: center;
        35     margin: 100px;
    }
    36
    37 svg {
        38     background-color: #f9f9f9;
        39     border: 1px solid #000000;
        40     padding: 70px;
    }
    41
    42 .tooltip {
        43     position: absolute;
        44     background: #rgb(204, 156, 229);
        45     color: #rgb(105, 84, 116);
        46     border-radius: 5px;
        47     padding: 10px;
        48     pointer-events: none;
        49     opacity: 0;
    }
```

## CSS file for line graph

```
82   .data(points)
83   .enter().append("circle")
84   .attr("class", "dot")
85   .attr("cx", function(d) {
86     var coords = ternaryToCartesian(d.fossil, d.nuclear, d.renewables);
87     return coords.x;
88   })
89   .attr("cy", function(d) {
90     var coords = ternaryToCartesian(d.fossil, d.nuclear, d.renewables);
91     return coords.y;
92   })
93   .attr("r", function(d) {
94     return Math.max(d.gdp / 2000, 5);
95   })
96   .style("fill", function(d) {
97     return colorScale(d.country);
98   })
99   .style("stroke", "#000")
100  .style("stroke-width", 1)
101  .style("opacity", 0.8)
102  .on("mouseover", function(event, d) {
103    d3.selectAll(".dot").style("opacity", 0.1);
104    d3.select(this).style("opacity", 1);
105
106    var xPosition = parseFloat(d3.select(this).attr("cx")) + 10;
107    var yPosition = parseFloat(d3.select(this).attr("cy")) - 10;
108
109    console.log(`Label Position: (${xPosition}, ${yPosition})`);
110
111    g.append("foreignObject")
112      .attr("id", "hoverValue")
113      .attr("x", xPosition)
114      .attr("y", yPosition)
115      .attr("width", 200)
116      .attr("height", 100)
117      .append("xhtml:div")
118      .attr("class", "tooltip-label")
119      .html(`${d.country}:<br/>Fossil - ${d.fossil.toFixed(2)},<br/>Nuclear - ${d.nuclear.toFixed(2)},<br/>Renewables - ${d.renewables.toFixed(2)},<br/>GDP - ${d.gdp}`);
120  })
121  .on("mouseout", function() {
122    d3.selectAll(".dot").style("opacity", 0.8);
123    g.select("#hoverValue").remove();
124  });
125}).catch(function(error) {
126  console.error("Error loading the CSV data:", error);
127});
```

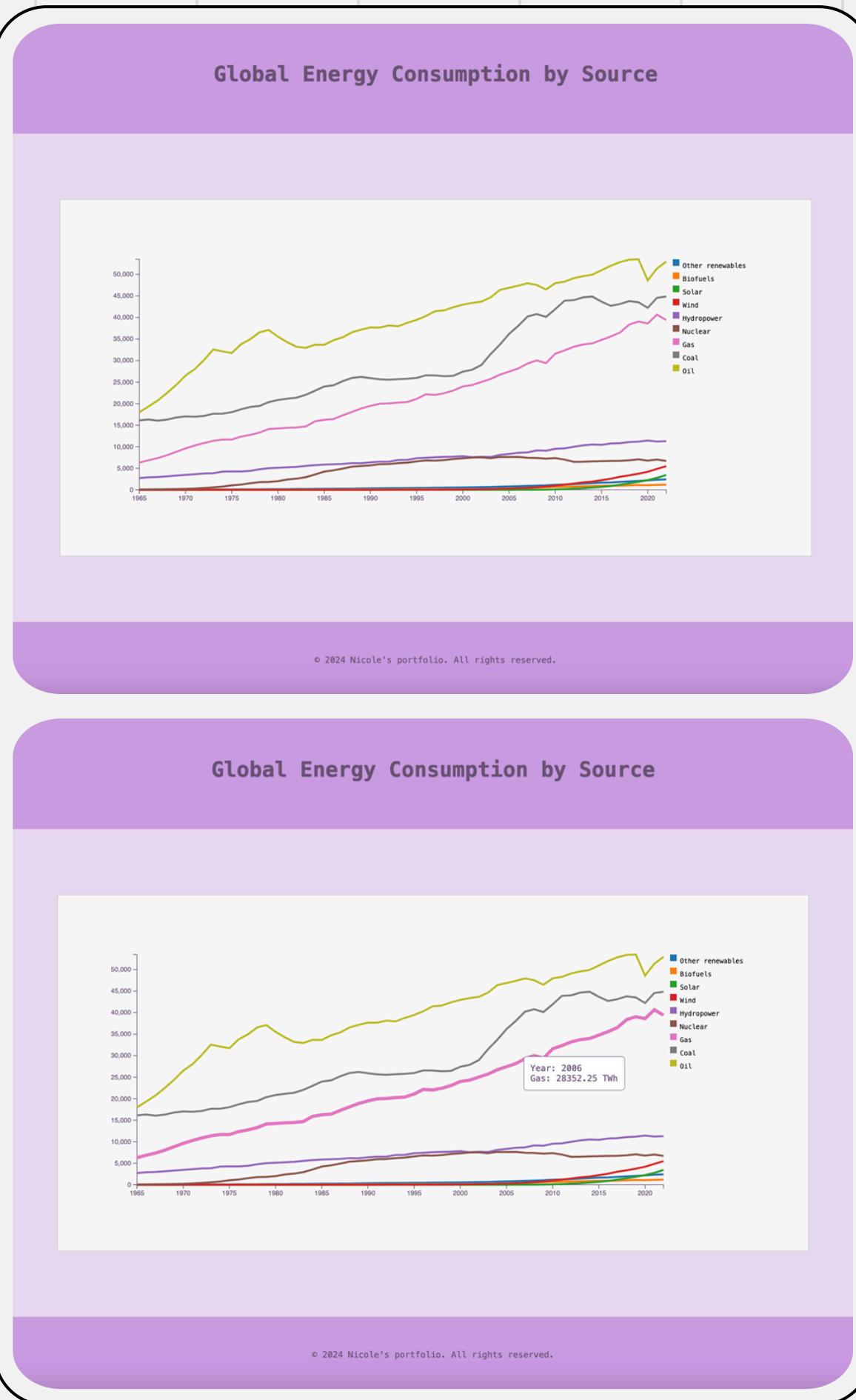
```
15   d3.select(this).attr("stroke-width", 5);
16   tooltip.style("opacity", 1);
17 }
18 .on("mousemove", function(event, d) {
19   const year = x.invert(d3.pointer(event)[0]);
20   const value = y.invert(d3.pointer(event)[1]);
21   tooltip
22     .html(`Year: ${Math.round(year)}  
${key}: ${value.toFixed(2)} TWh`)
23     .style("left", (event.pageX + 10) + "px")
24     .style("top", (event.pageY - 10) + "px");
25 })
26 .on("mouseout", function() {
27   d3.select(this).attr("stroke-width", 3);
28   tooltip.style("opacity", 0);
29 });
30 );
31
32 const legend = svg.append("g")
33   .attr("transform", `translate(${width + 10}, 0)`);
34
35 keys.forEach((key, i) => {
36   const legendRow = legend.append("g")
37     .attr("transform", `translate(0, ${i * 20})`);
38
39   legendRow.append("rect")
40     .attr("width", 10)
41     .attr("height", 10)
42     .attr("fill", color(key));
43
44   legendRow.append("text")
45     .attr("x", 15)
46     .attr("y", 10)
47     .attr("dy", ".35em")
48     .attr("text-anchor", "start")
49     .style("font-size", "10px")
50     .text(key);
51 });
52
53 }).catch(function(error) {
54   console.error("Error loading the CSV data:", error);
55 }
```



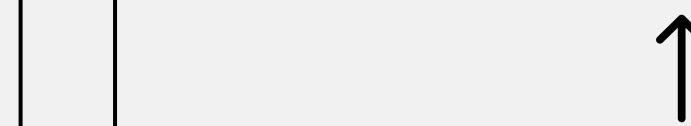
# LINE GRAPH

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.



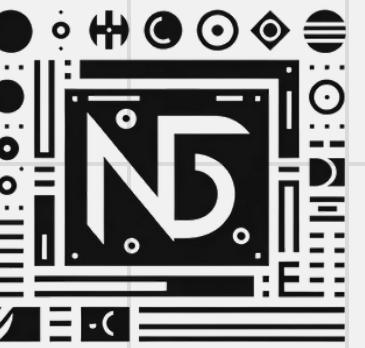
```
1 const margin = { top: 20, right: 150, bottom: 30, left: 50 },
2   width = 1000 - margin.left - margin.right,
3   height = 400 - margin.top - margin.bottom;
4
5 const svg = d3.select("#chart").append("svg")
6   .attr("width", width + margin.left + margin.right)
7   .attr("height", height + margin.top + margin.bottom)
8   .append("g")
9   .attr("transform", `translate(${margin.left},${margin.top})`);
10
11 d3.csv("energy-consumption-by-source-and-country.csv").then(data
12 => {
13   console.log("Data loaded:", data);
14
15   const globalData = data.filter(d => d.Entity === "World");
16
17   const x = d3.scaleLinear()
18     .domain(d3.extent(filteredData, d => d.Year))
19     .range([0, width]);
20
21   svg.append("g")
22     .attr("transform", `translate(0,${height})`)
23     .call(d3.axisBottom(x).tickFormat(d3.format("d")));
24
25   const y = d3.scaleLinear()
26     .domain([0, d3.max(filteredData, d => Math.max(...keys.map
27       (key => d[key])))])
28     .range([height, 0]);
29
30   svg.append("g")
31     .call(d3.axisLeft(y));
32
33 });
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
```



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.

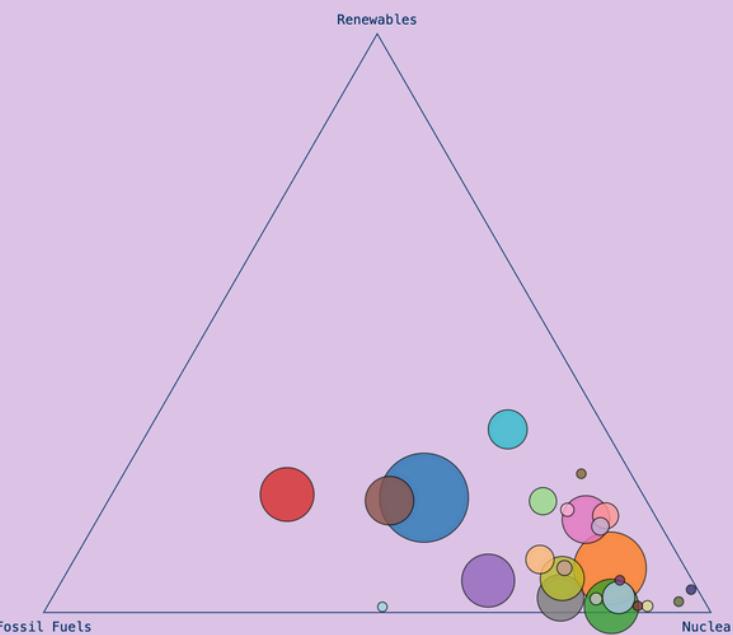


# LINE GRAPH

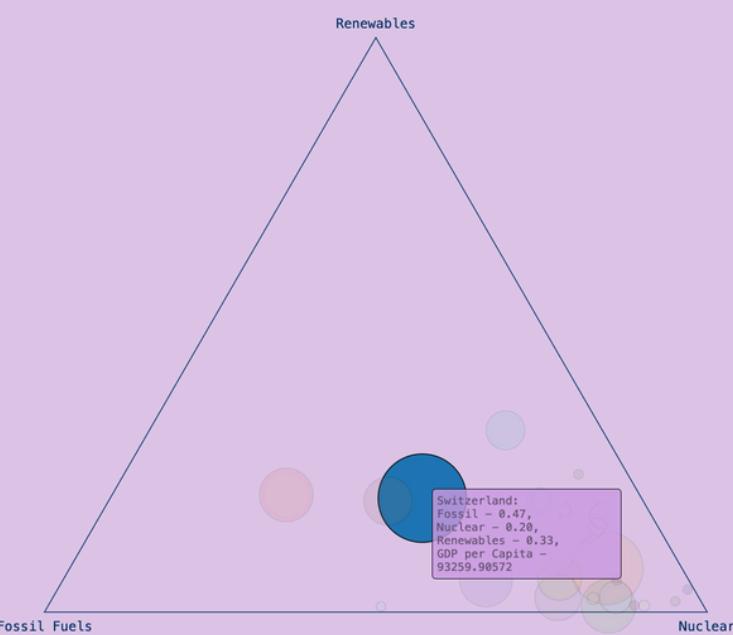
This graph I wanted use to visualise the relation of energy consumption (fossil fuels being the most dangerous and the dirtiest compared to renewable or nuclear) to GDP per capita, showing that even though some countries may be richer and could afford to invest in using more sustainable energy sources,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

Energy consumption per country and GDP per capita



Energy consumption per country and GDP per capita



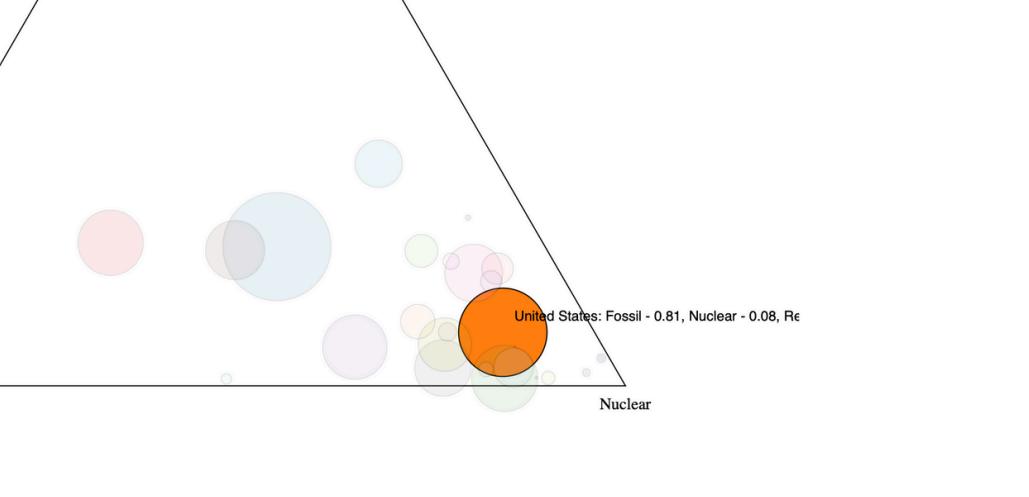
```
1 d3.csv("filtered_merged_data.csv").then(function(data) {  
2   var width = 800,  
3     height = 800,  
4     margin = 50,  
5     additionalMargin = 150;  
6  
7   var side = Math.min(width, height) - margin * 2;  
8   var heightEquilateral = Math.sqrt(3) / 2 * side;  
9  
10  var svg = d3.select("#ternary-plot").append("svg")  
11    .attr("width", width + margin * 2 + additionalMargin)  
12    .attr("height", height + margin * 2)  
13    .style("display", "block")  
14    .style("margin", "0 auto");  
15  
16  g.selectAll(".label")  
17    .data(labels)  
18    .enter().append("text")  
19    .attr("class", "label")  
20    .attr("x", function(d, i) {  
21      return i === 0 ? 0 : (i === 1 ? side : side / 2);  
22    })  
23    .attr("y", function(d, i) {  
24      return i === 0 ? heightEquilateral + 20 : (i === 1 ?  
25        heightEquilateral + 20 : -10);  
26    })  
27    .style("text-anchor", "middle")  
28    .style("font-size", "14px")  
29    .text(function(d) {  
30      return d;  
31    });  
32  
33  g.selectAll("circle")  
34    .data(data)  
35    .enter().append("circle")  
36    .attr("cx", function(d) {  
37      return (d.FossilFuel * side) + margin * 2;  
38    })  
39    .attr("cy", function(d) {  
40      return (d.Nuclear * heightEquilateral) + margin * 2;  
41    })  
42    .attr("r", function(d) {  
43      return d.GDPPerCapita * 10;  
44    })  
45    .style("fill", function(d) {  
46      if (d.FossilFuel > 0.5) {  
47        return "#E69138";  
48      } else if (d.Nuclear > 0.5) {  
49        return "#A52A2A";  
50      } else {  
51        return "#3CB371";  
52      }  
53    })  
54  
55  g.selectAll("text")  
56    .data(data)  
57    .enter().append("text")  
58    .attr("x", function(d) {  
59      return (d.FossilFuel * side) + margin * 2;  
60    })  
61    .attr("y", function(d) {  
62      return (d.Nuclear * heightEquilateral) + margin * 2;  
63    })  
64    .text(function(d) {  
65      return d.Name;  
66    })  
67  
68  g.selectAll("text")  
69    .data(data)  
70    .enter().append("text")  
71    .attr("x", function(d) {  
72      return (d.FossilFuel * side) + margin * 2;  
73    })  
74    .attr("y", function(d) {  
75      return (d.Nuclear * heightEquilateral) + margin * 2;  
76    })  
77    .text(function(d) {  
78      return "Fossil Fuel - " + d.FossilFuel + ",  
79          Nuclear - " + d.Nuclear + ",  
80          Renewables - " + d.Renewables + ",  
81          GDP per Capita - " + d.GDPPerCapita;  
82    })  
83  
84  g.selectAll("text")  
85    .data(data)  
86    .enter().append("text")  
87    .attr("x", function(d) {  
88      return (d.FossilFuel * side) + margin * 2;  
89    })  
90    .attr("y", function(d) {  
91      return (d.Nuclear * heightEquilateral) + margin * 2;  
92    })  
93    .text(function(d) {  
94      return "Renewables  
Fossil Fuels  
Nuclear";  
95    })  
96  
97  g.selectAll("text")  
98    .data(data)  
99    .enter().append("text")  
100   .attr("x", function(d) {  
101     return (d.FossilFuel * side) + margin * 2;  
102   })  
103   .attr("y", function(d) {  
104     return (d.Nuclear * heightEquilateral) + margin * 2;  
105   })  
106   .text(function(d) {  
107     return "GDP per Capita";  
108   })  
109  
110  g.selectAll("text")  
111    .data(data)  
112    .enter().append("text")  
113    .attr("x", function(d) {  
114      return (d.FossilFuel * side) + margin * 2;  
115    })  
116    .attr("y", function(d) {  
117      return (d.Nuclear * heightEquilateral) + margin * 2;  
118    })  
119    .text(function(d) {  
120      return "Fossil Fuel  
Nuclear  
Renewables";  
121    })  
122  
123  g.selectAll("text")  
124    .data(data)  
125    .enter().append("text")  
126    .attr("x", function(d) {  
127      return (d.FossilFuel * side) + margin * 2;  
128    })  
129    .attr("y", function(d) {  
130      return (d.Nuclear * heightEquilateral) + margin * 2;  
131    })  
132    .text(function(d) {  
133      return "Renewables  
Fossil Fuels  
Nuclear";  
134    })  
135  
136  g.selectAll("text")  
137    .data(data)  
138    .enter().append("text")  
139    .attr("x", function(d) {  
140      return (d.FossilFuel * side) + margin * 2;  
141    })  
142    .attr("y", function(d) {  
143      return (d.Nuclear * heightEquilateral) + margin * 2;  
144    })  
145    .text(function(d) {  
146      return "GDP per Capita";  
147    })  
148  
149  g.selectAll("text")  
150    .data(data)  
151    .enter().append("text")  
152    .attr("x", function(d) {  
153      return (d.FossilFuel * side) + margin * 2;  
154    })  
155    .attr("y", function(d) {  
156      return (d.Nuclear * heightEquilateral) + margin * 2;  
157    })  
158    .text(function(d) {  
159      return "Renewables  
Fossil Fuel  
Nuclear";  
160    })  
161  
162  g.selectAll("text")  
163    .data(data)  
164    .enter().append("text")  
165    .attr("x", function(d) {  
166      return (d.FossilFuel * side) + margin * 2;  
167    })  
168    .attr("y", function(d) {  
169      return (d.Nuclear * heightEquilateral) + margin * 2;  
170    })  
171    .text(function(d) {  
172      return "Fossil Fuel  
Nuclear  
Renewables";  
173    })  
174  
175  g.selectAll("text")  
176    .data(data)  
177    .enter().append("text")  
178    .attr("x", function(d) {  
179      return (d.FossilFuel * side) + margin * 2;  
180    })  
181    .attr("y", function(d) {  
182      return (d.Nuclear * heightEquilateral) + margin * 2;  
183    })  
184    .text(function(d) {  
185      return "Renewables  
Fossil Fuels  
Nuclear";  
186    })  
187  
188  g.selectAll("text")  
189    .data(data)  
190    .enter().append("text")  
191    .attr("x", function(d) {  
192      return (d.FossilFuel * side) + margin * 2;  
193    })  
194    .attr("y", function(d) {  
195      return (d.Nuclear * heightEquilateral) + margin * 2;  
196    })  
197    .text(function(d) {  
198      return "GDP per Capita";  
199    })  
200  
201  g.selectAll("text")  
202    .data(data)  
203    .enter().append("text")  
204    .attr("x", function(d) {  
205      return (d.FossilFuel * side) + margin * 2;  
206    })  
207    .attr("y", function(d) {  
208      return (d.Nuclear * heightEquilateral) + margin * 2;  
209    })  
210    .text(function(d) {  
211      return "Renewables  
Fossil Fuel  
Nuclear";  
212    })  
213  
214  g.selectAll("text")  
215    .data(data)  
216    .enter().append("text")  
217    .attr("x", function(d) {  
218      return (d.FossilFuel * side) + margin * 2;  
219    })  
220    .attr("y", function(d) {  
221      return (d.Nuclear * heightEquilateral) + margin * 2;  
222    })  
223    .text(function(d) {  
224      return "Fossil Fuel  
Nuclear  
Renewables";  
225    })  
226  
227  g.selectAll("text")  
228    .data(data)  
229    .enter().append("text")  
230    .attr("x", function(d) {  
231      return (d.FossilFuel * side) + margin * 2;  
232    })  
233    .attr("y", function(d) {  
234      return (d.Nuclear * heightEquilateral) + margin * 2;  
235    })  
236    .text(function(d) {  
237      return "Renewables  
Fossil Fuels  
Nuclear";  
238    })  
239  
240  g.selectAll("text")  
241    .data(data)  
242    .enter().append("text")  
243    .attr("x", function(d) {  
244      return (d.FossilFuel * side) + margin * 2;  
245    })  
246    .attr("y", function(d) {  
247      return (d.Nuclear * heightEquilateral) + margin * 2;  
248    })  
249    .text(function(d) {  
250      return "GDP per Capita";  
251    })  
252  
253  g.selectAll("text")  
254    .data(data)  
255    .enter().append("text")  
256    .attr("x", function(d) {  
257      return (d.FossilFuel * side) + margin * 2;  
258    })  
259    .attr("y", function(d) {  
260      return (d.Nuclear * heightEquilateral) + margin * 2;  
261    })  
262    .text(function(d) {  
263      return "Renewables  
Fossil Fuel  
Nuclear";  
264    })  
265  
266  g.selectAll("text")  
267    .data(data)  
268    .enter().append("text")  
269    .attr("x", function(d) {  
270      return (d.FossilFuel * side) + margin * 2;  
271    })  
272    .attr("y", function(d) {  
273      return (d.Nuclear * heightEquilateral) + margin * 2;  
274    })  
275    .text(function(d) {  
276      return "Fossil Fuel  
Nuclear  
Renewables";  
277    })  
278  
279  g.selectAll("text")  
280    .data(data)  
281    .enter().append("text")  
282    .attr("x", function(d) {  
283      return (d.FossilFuel * side) + margin * 2;  
284    })  
285    .attr("y", function(d) {  
286      return (d.Nuclear * heightEquilateral) + margin * 2;  
287    })  
288    .text(function(d) {  
289      return "Renewables  
Fossil Fuels  
Nuclear";  
290    })  
291  
292  g.selectAll("text")  
293    .data(data)  
294    .enter().append("text")  
295    .attr("x", function(d) {  
296      return (d.FossilFuel * side) + margin * 2;  
297    })  
298    .attr("y", function(d) {  
299      return (d.Nuclear * heightEquilateral) + margin * 2;  
300    })  
301    .text(function(d) {  
302      return "GDP per Capita";  
303    })  
304  
305  g.selectAll("text")  
306    .data(data)  
307    .enter().append("text")  
308    .attr("x", function(d) {  
309      return (d.FossilFuel * side) + margin * 2;  
310    })  
311    .attr("y", function(d) {  
312      return (d.Nuclear * heightEquilateral) + margin * 2;  
313    })  
314    .text(function(d) {  
315      return "Renewables  
Fossil Fuel  
Nuclear";  
316    })  
317  
318  g.selectAll("text")  
319    .data(data)  
320    .enter().append("text")  
321    .attr("x", function(d) {  
322      return (d.FossilFuel * side) + margin * 2;  
323    })  
324    .attr("y", function(d) {  
325      return (d.Nuclear * heightEquilateral) + margin * 2;  
326    })  
327    .text(function(d) {  
328      return "Fossil Fuel  
Nuclear  
Renewables";  
329    })  
330  
331  g.selectAll("text")  
332    .data(data)  
333    .enter().append("text")  
334    .attr("x", function(d) {  
335      return (d.FossilFuel * side) + margin * 2;  
336    })  
337    .attr("y", function(d) {  
338      return (d.Nuclear * heightEquilateral) + margin * 2;  
339    })  
340    .text(function(d) {  
341      return "Renewables  
Fossil Fuels  
Nuclear";  
342    })  
343  
344  g.selectAll("text")  
345    .data(data)  
346    .enter().append("text")  
347    .attr("x", function(d) {  
348      return (d.FossilFuel * side) + margin * 2;  
349    })  
350    .attr("y", function(d) {  
351      return (d.Nuclear * heightEquilateral) + margin * 2;  
352    })  
353    .text(function(d) {  
354      return "GDP per Capita";  
355    })  
356  
357  g.selectAll("text")  
358    .data(data)  
359    .enter().append("text")  
360    .attr("x", function(d) {  
361      return (d.FossilFuel * side) + margin * 2;  
362    })  
363    .attr("y", function(d) {  
364      return (d.Nuclear * heightEquilateral) + margin * 2;  
365    })  
366    .text(function(d) {  
367      return "Renewables  
Fossil Fuel  
Nuclear";  
368    })  
369  
370  g.selectAll("text")  
371    .data(data)  
372    .enter().append("text")  
373    .attr("x", function(d) {  
374      return (d.FossilFuel * side) + margin * 2;  
375    })  
376    .attr("y", function(d) {  
377      return (d.Nuclear * heightEquilateral) + margin * 2;  
378    })  
379    .text(function(d) {  
380      return "Fossil Fuel  
Nuclear  
Renewables";  
381    })  
382  
383  g.selectAll("text")  
384    .data(data)  
385    .enter().append("text")  
386    .attr("x", function(d) {  
387      return (d.FossilFuel * side) + margin * 2;  
388    })  
389    .attr("y", function(d) {  
390      return (d.Nuclear * heightEquilateral) + margin * 2;  
391    })  
392    .text(function(d) {  
393      return "Renewables  
Fossil Fuels  
Nuclear";  
394    })  
395  
396  g.selectAll("text")  
397    .data(data)  
398    .enter().append("text")  
399    .attr("x", function(d) {  
400      return (d.FossilFuel * side) + margin * 2;  
401    })  
402    .attr("y", function(d) {  
403      return (d.Nuclear * heightEquilateral) + margin * 2;  
404    })  
405    .text(function(d) {  
406      return "GDP per Capita";  
407    })  
408  
409  g.selectAll("text")  
410    .data(data)  
411    .enter().append("text")  
412    .attr("x", function(d) {  
413      return (d.FossilFuel * side) + margin * 2;  
414    })  
415    .attr("y", function(d) {  
416      return (d.Nuclear * heightEquilateral) + margin * 2;  
417    })  
418    .text(function(d) {  
419      return "Renewables  
Fossil Fuel  
Nuclear";  
420    })  
421  
422  g.selectAll("text")  
423    .data(data)  
424    .enter().append("text")  
425    .attr("x", function(d) {  
426      return (d.FossilFuel * side) + margin * 2;  
427    })  
428    .attr("y", function(d) {  
429      return (d.Nuclear * heightEquilateral) + margin * 2;  
430    })  
431    .text(function(d) {  
432      return "Fossil Fuel  
Nuclear  
Renewables";  
433    })  
434  
435  g.selectAll("text")  
436    .data(data)  
437    .enter().append("text")  
438    .attr("x", function(d) {  
439      return (d.FossilFuel * side) + margin * 2;  
440    })  
441    .attr("y", function(d) {  
442      return (d.Nuclear * heightEquilateral) + margin * 2;  
443    })  
444    .text(function(d) {  
445      return "Renewables  
Fossil Fuels  
Nuclear";  
446    })  
447  
448  g.selectAll("text")  
449    .data(data)  
450    .enter().append("text")  
451    .attr("x", function(d) {  
452      return (d.FossilFuel * side) + margin * 2;  
453    })  
454    .attr("y", function(d) {  
455      return (d.Nuclear * heightEquilateral) + margin * 2;  
456    })  
457    .text(function(d) {  
458      return "GDP per Capita";  
459    })  
460  
461  g.selectAll("text")  
462    .data(data)  
463    .enter().append("text")  
464    .attr("x", function(d) {  
465      return (d.FossilFuel * side) + margin * 2;  
466    })  
467    .attr("y", function(d) {  
468      return (d.Nuclear * heightEquilateral) + margin * 2;  
469    })  
470    .text(function(d) {  
471      return "Renewables  
Fossil Fuel  
Nuclear";  
472    })  
473  
474  g.selectAll("text")  
475    .data(data)  
476    .enter().append("text")  
477    .attr("x", function(d) {  
478      return (d.FossilFuel * side) + margin * 2;  
479    })  
480    .attr("y", function(d) {  
481      return (d.Nuclear * heightEquilateral) + margin * 2;  
482    })  
483    .text(function(d) {  
484      return "Fossil Fuel  
Nuclear  
Renewables";  
485    })  
486  
487  g.selectAll("text")  
488    .data(data)  
489    .enter().append("text")  
490    .attr("x", function(d) {  
491      return (d.FossilFuel * side) + margin * 2;  
492    })  
493    .attr("y", function(d) {  
494      return (d.Nuclear * heightEquilateral) + margin * 2;  
495    })  
496    .text(function(d) {  
497      return "Renewables  
Fossil Fuels  
Nuclear";  
498    })  
499  
500  g.selectAll("text")  
501    .data(data)  
502    .enter().append("text")  
503    .attr("x", function(d) {  
504      return (d.FossilFuel * side) + margin * 2;  
505    })  
506    .attr("y", function(d) {  
507      return (d.Nuclear * heightEquilateral) + margin * 2;  
508    })  
509    .text(function(d) {  
510      return "GDP per Capita";  
511    })  
512  
513  g.selectAll("text")  
514    .data(data)  
515    .enter().append("text")  
516    .attr("x", function(d) {  
517      return (d.FossilFuel * side) + margin * 2;  
518    })  
519    .attr("y", function(d) {  
520      return (d.Nuclear * heightEquilateral) + margin * 2;  
521    })  
522    .text(function(d) {  
523      return "Renewables  
Fossil Fuel  
Nuclear";  
524    })  
525  
526  g.selectAll("text")  
527    .data(data)  
528    .enter().append("text")  
529    .attr("x", function(d) {  
530      return (d.FossilFuel * side) + margin * 2;  
531    })  
532    .attr("y", function(d) {  
533      return (d.Nuclear * heightEquilateral) + margin * 2;  
534    })  
535    .text(function(d) {  
536      return "Fossil Fuel  
Nuclear  
Renewables";  
537    })  
538  
539  g.selectAll("text")<
```

# 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 articles 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 isn't strongly correlated.



- ✖ Failed to load resource: the server with a status of 404 (File not found)
- ✖ ► Error loading the CSV data:  
Error: 404 File not found  
at ku ([d3.v6.min.js:2:94710](#))



The line graph was fairly straightforward to make, despite some bumps, but I really 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.

## Data Analysis:

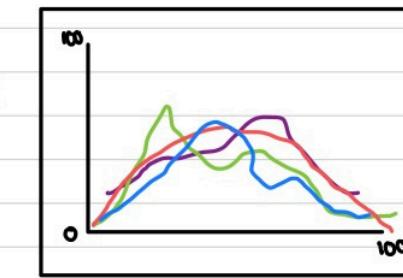
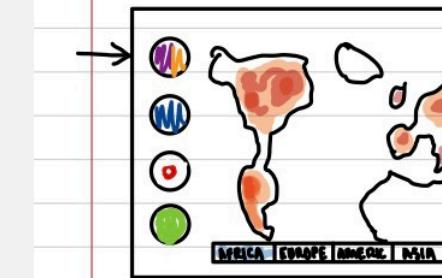
Conduct statistical analysis to find correlations or trends between social media usage patterns and reported mental health issues.

## Visualisation Techniques:

Correlation Maps → heat maps or geographical data visualisation to correlate regional social media usage trends with mental health data.  
Interactive Dashboards → interactive so users can select different social media platforms or demographic profiles to see specific mental health impacts.

↳ hover effects, dynamic updating, clear labels

## DESIGN DOCUMENTATION



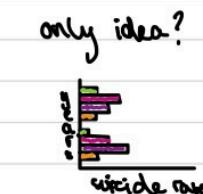
- Anxiety levels
- Instagram use
- Facebook use
- WhatsApp use
- YouTube use
- Depression levels

↳ Reports of online bullying

## Research:

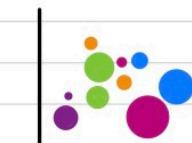
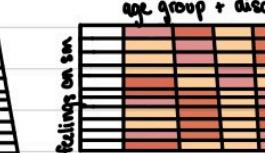
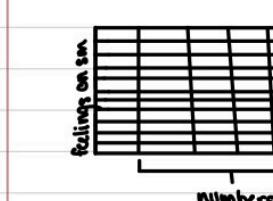
- ↳ 1. The Economist → age group / suicide rate over time USA
- ↳ 2. NHS → Feelings on social media chart
- ↳ 3. Our World in Data → well-being impact of deactivating FB
- ↳ 4. McKinsey & Company → Mental health / generations

## 1) Ideas



only idea?

## 2) Idea:

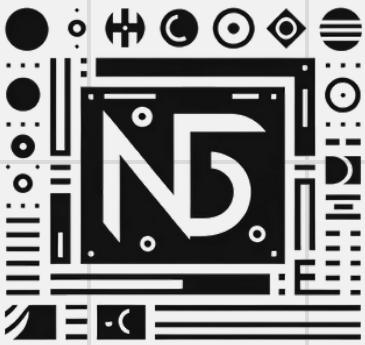


JUNE 2024



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;

