7.1.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8"/>

    <meta name="description" content="Data Visualisation"/>

    <meta name="keyword" content="HTML, CSS, D3"/>

    <meta name="author" content="Minh Nguyen"/>

    <title>Task 7.1</title>

    <link rel="stylesheet" href="7.1.css">

</head>

<body>

    <h1>Number of Unemployed in Australia</h1>

    <p id="chart"></p>

    <br>

    <bf></bf>

    <footer style="color:grey">COS30045 Data Visualisation<br>Minh Nguyen</footer>

    <script src="https://d3js.org/d3.v7.min.js"></script>

    <script src="7.1.js"></script>

</body>

</html>

7.1.css

.line {

    fill: none;

    stroke: slategray;

    stroke-width: 0.5;

}

.area {

    fill: slategray;

    stroke: slategray;

    stroke-width: 0.5;

}

7.1.js

function init() {

    var w = 600;

    var h = 300;

    var padding = 55;

    var dataset;

    function lineChart(dataset) {

      var svg = d3.select("#chart")

                  .append("svg")

                  .attr("width", w)

                  .attr("height", h);

      var xScale = d3.scaleTime()

                     .domain([

                        d3.min(dataset, function(d) { return d.date; }),

                        d3.max(dataset, function(d) { return d.date; })

                    ])

                    .range([padding, w]);

      var yScale = d3.scaleLinear()

                    .domain([0, d3.max(dataset, function(d) { return d.number; })])

                    .range([h - padding + 10, 0]);

      var xAxis = d3.axisBottom().ticks(10).scale(xScale);

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

      //creating line

      var 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); })

                  //base line for area shape

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

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

      //append line

      svg.append("path")

        .datum(dataset)

        .attr("class", "line")

        .attr("d", line);

      svg.append("path")

        .datum(dataset)

        .attr("class", "area")

        .attr("d", area);

      svg.append("g") //Append the x and y axis to the SVG element as groups

        .attr("transform", "translate(0, "+ (h - padding + 10) +")") //translating them to the appropriate positions

        .call(xAxis);

      svg.append("g")

        .attr("transform", "translate(" + (padding) + ", 0)")

        .call(yAxis);

      svg.append("line")

        .attr("class", "line halfMilMark")

      //start of the line

        .attr("x1", padding)

        .attr("y1", yScale(500000))

      //end of the line

        .attr("x2", w)

        .attr("y2", yScale(500000))

        .style("stroke", "red") // set stroke color to red

        .style("stroke-dasharray", "2"); // set stroke dasharray to create a dotted effect

      svg.append("text")

        .attr("class", "halfMilLabel")

        .attr("x", padding + 10)

        .attr("y", yScale(500000) - 7)

        .text("Half a million unemployed")

        .attr("fill", "red");

    }

    d3.csv("Unemployment\_78-95.csv", function(d) {

      return {

        date: new Date(+d.year, +d.month - 1),

        number: +d.number

      };

    }).then(function(data) {

      dataset = data;

      lineChart(dataset);

      // console.table(dataset, ["date", "number"]); check the DOM for values of CSV

    });

  }

  window.onload = init;

7.2.html donutchart

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8"/>

    <meta name="description" content="Data Visualisation"/>

    <meta name="keyword" content="HTML, CSS, D3"/>

    <meta name="author" content="Minh Nguyen"/>

    <title>Task 7.2</title>

</head>

<body>

    <h1>Donut Chart</h1>

    <p id="chart"></p>

    <br>

    <bf></bf>

    <footer style="color:grey">COS30045 Data Visualisation<br>Minh Nguyen</footer>

    <script src="https://d3js.org/d3.v7.min.js"></script>

    <script src="7.2.1.js"></script>

</body>

</html>

7.2.js donutchart

function init() {

    var w = 300;

    var h = 300;

    var padding = 55;

    var dataset = [5, 10, 15, 20, 25, 30];

    var outerRadius = w/2;

    //Later you can change the value of the inner radius to generate donut charts

    var innerRadius = h/3;

    var arc = d3.arc()

                  .outerRadius(outerRadius)

                  .innerRadius(innerRadius);

    var pie = d3.pie();

    var svg = d3.select("#chart")

                  .append("svg")

                  .attr("width", w)

                  .attr("height", h);

    var arcs = svg.selectAll("g.arc")

                    .data(pie(dataset))

                    .enter()

                    .append("g")

                    .attr("class", "arc")

                    .attr("transform", "translate(" + outerRadius + "," + outerRadius + ")");

    var color = d3.scaleOrdinal(d3.schemeCategory10);

    arcs.append("path")

        .attr("fill", function(d, i) {

          return color(i)

        })

        .attr("d", function(d, i){

          return arc(d, i)

        });

    arcs.append("text")

        .text(function(d){

          return d.value

        })

        .attr("transform", function(d){

          return "translate(" + arc.centroid(d) +")";

        })

        .attr("font-size", "14px")

        .attr("font-weight", "bold");

  }

  window.onload = init;

7.2.html piechart

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8"/>

    <meta name="description" content="Data Visualisation"/>

    <meta name="keyword" content="HTML, CSS, D3"/>

    <meta name="author" content="Tyler Dang"/>

    <title>Task 7.2</title>

</head>

<body>

    <h1>Pie Chart</h1>

    <p id="chart"></p>

    <br>

    <bf></bf>

    <footer style="color:grey">COS30045 Data Visualisation<br>Minh Nguyen</footer>

    <script src="https://d3js.org/d3.v7.min.js"></script>

    <script src="7.2.2.js"></script>

</body>

</html>

7.2.js piechart

function init() {

    var w = 300;

    var h = 300;

    var padding = 55;

    var dataset = [5, 10, 15, 20, 25, 30];

    var outerRadius = w/2;

    //Later you can change the value of the inner radius to generate donut charts

    var innerRadius = 0;

    var arc = d3.arc()

                  .outerRadius(outerRadius)

                  .innerRadius(innerRadius);

    var pie = d3.pie();

    var svg = d3.select("#chart")

                  .append("svg")

                  .attr("width", w)

                  .attr("height", h);

    var arcs = svg.selectAll("g.arc")

                    .data(pie(dataset))

                    .enter()

                    .append("g")

                    .attr("class", "arc")

                    .attr("transform", "translate(" + outerRadius + "," + outerRadius + ")");

    var color = d3.scaleOrdinal(d3.schemeCategory10);

    arcs.append("path")

        .attr("fill", function(d, i) {

          return color(i)

        })

        .attr("d", function(d, i){

          return arc(d, i)

        });

    arcs.append("text")

        .text(function(d){

          return d.value

        })

        .attr("transform", function(d){

          return "translate(" + arc.centroid(d) +")";

        })

        .attr("font-size", "14px")

        .attr("font-weight", "bold");

  }

  window.onload = init;

7.3.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8"/>

    <meta name="description" content="Data Visualisation"/>

    <meta name="keyword" content="HTML, CSS, D3"/>

    <meta name="author" content="Minh Nguyen"/>

    <title>Task 7.3</title>

</head>

<body>

    <h1>Stacked Bar Chart</h1>

    <p id="chart"></p>

    <br>

    <bf></bf>

    <script src="https://d3js.org/d3.v6.min.js"></script>

    <script src="7.3.js"></script>

    <footer style="color:grey">COS30045 Data Visualisation<br>Minh Nguyen</footer>

</body>

</html>

7.3.js

function init() {

    var w = 300;

    var h = 300;

    var dataset = [

      {apples: 5, oranges: 10, grapes: 22},

      {apples: 4, oranges: 12, grapes: 28},

      {apples: 2, oranges: 19, grapes: 32},

      {apples: 7, oranges: 23, grapes: 35},

      {apples: 23, oranges: 17, grapes: 43}

    ];

    var stack = d3.stack()

                  .keys(["apples", "oranges", "grapes"]);

    var series = stack(dataset);

    var svg = d3.select("#chart")

                  .append("svg")

                  .attr("width", w)

                  .attr("height", h);

    var color = d3.scaleOrdinal(d3.schemeCategory10);

    var group = svg.selectAll("g")

                    .data(series)

                    .enter()

                    .append("g")

                    .style("fill", function(d, i){

                      return color(i);

                    });

    //Scaleband for xScale when generating stack bar chart

    var xScale = d3.scaleBand()

                    .domain(dataset.map(function(d, i) {

                      return i;

                    }))

                    .range([0, w])

                    .padding(0.1);

    var yScale = d3.scaleLinear()

                    .domain([0, d3.max(dataset, function(d){

                      return d.apples + d.oranges +d.grapes;

                    })])

                    .range([h, 0]);

    var rects = group.selectAll("rect")

                      .data(function(d){ return d;})

                      .enter()

                      .append("rect")

                      .attr("x", function(d, i) {

                        return xScale(i);

                      })

                      .attr("y", function(d, i){

                        return yScale(d[1]);

                      })

                      .attr("height", function(d) {

                        return yScale(d[0]) - yScale(d[1]);

                      })

                      .attr("width", xScale.bandwidth());

  }

  window.onload = init;

8.1.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8"/>

    <meta name="description" content="Data Visualisation"/>

    <meta name="keyword" content="HTML, CSS, D3"/>

    <meta name="author" content="Minh Nguyen"/>

    <title>Task 8.1</title>

</head>

<body>

    <h1>Victorian LGA Map</h1>

    <p id="chart"></p>

    <br>

    <bf></bf>

    <footer style="color:grey">COS30045 Data Visualisation<br>Minh Nguyen</footer>

    <script src="https://d3js.org/d3.v6.min.js"></script>

    <script src="8.1.js"></script>

</body>

</html>

8.1.js

function init() {

    var w = 500;

    var h = 300;

    var projection = d3.geoMercator()

                      .center([145, -36.5])

                      .translate([w/2, h/2])

                      .scale(2450);

    var path = d3.geoPath().projection(projection)

    var svg = d3.select("#chart")

                  .append("svg")

                  .attr("width", w)

                  .attr("height", h)

                  .attr("fill", "grey");

    d3.json("ABS\_LGA\_2011.json").then(function(json) {

      svg.selectAll("path")

      .data(json.features)

      .enter()

      .append("path")

      .attr("d", path);

    });

}

  window.onload = init;

8.2.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8"/>

    <meta name="description" content="Data Visualisation"/>

    <meta name="keyword" content="HTML, CSS, D3"/>

    <meta name="author" content="Minh Nguyen"/>

    <title>Task 8.2</title>

</head>

<body>

    <h1>Victorian Number Unemployed by LGA</h1>

    <p id="chart"></p>

    <br>

    <bf></bf>

    <footer style="color:grey">COS30045 Data Visualisation<br>Minh Nguyen</footer>

    <script src="https://d3js.org/d3.v6.min.js"></script>

    <script src="8.2.js"></script>

</body>

</html>

8.2.js

function init() {

    var w = 500;

    var h = 300;

    var projection = d3.geoMercator()

        .center([145, -36.5])

        .translate([w / 2, h / 2])

        .scale(2450);

    var color = d3.scaleQuantize().range(['#f2f0f8', '#ccc9e4', '#9f99cc', '#7869b6', "#cccccc", '#5b1f95' ]);

    var path = d3.geoPath().projection(projection);

    var svg = d3.select("#chart")

        .append("svg")

        .attr("width", w)

        .attr("height", h)

    d3.csv("VIC\_LGA\_unemployment.csv", function (d) {

        return {

            LGA: d.LGA,

            unemployed: +d.unemployed

        };

    }).then(function(data) {

        d3.json("LGA\_VIC.json").then(function(json) {

            // merge the data from VIC\_LGA\_unemployment.csv and LGA\_VIC.json

            // loop through once for each LGA value

            for (var i = 0; i < data.length; i++) {

                //grab LGA name

                var dataState = data[i].LGA;

                // grab data LGA, and convert from string to float

                var dataValue = parseFloat(data[i].unemployed);

                // find the corresponding LGA inside the LGA\_VIC.json

                for(var j = 0; j < json.features.length; j++){

                  var jsonState = json.features[j].properties.LGA\_name;

                  if(dataState == jsonState){

                      // copy the LGA data value into JSON

                      json.features[j].properties.unemployed = dataValue;

                      break; // stop looking through the JSON

                  }

                }

            }

            // set the domain of the color scale based on the unemployment data

            color.domain([d3.min(json.features, function(d) { return d.properties.unemployed; }), d3.max(json.features, function(d) { return d.properties.unemployed; })]);

            // create the path elements and set the fill color based on the unemployment data

            svg.selectAll("path")

              .data(json.features)

              .enter()

              .append("path")

              .attr("fill", "#cccccc") //color range didn't work, so use this in the svg to fix it

              .style("fill", function(d) { return color(d.properties.unemployed); })

              .attr("d", path);

            // load VIC\_city.csv data and create heatmap

            d3.csv("VIC\_city.csv", function (d) {

                return {

                    place: d.place, //for string

                    lat: +d.lat, //for int

                    lon: +d.lon

                };

            }).then(function(data) {

              svg.selectAll("circle")

              .data(data)

              .enter()

              .append("circle")

              .attr("cx", function(d){

                return projection([d.lon, d.lat])[0]; //take the first set in the csv

              })

              .attr("cy", function(d){

                return projection([d.lon, d.lat])[1]; ////take the second set in the csv

              })

              .attr("r", 2)

              .style("fill", "red");

              //text for victorian town and city

              svg.selectAll("text")

              .data(data)

              .enter()

              .append("text")

              .attr("x", function(d){

                  return projection([d.lon, d.lat])[0];

              })

              .attr("y", function(d){

                  return projection([d.lon, d.lat])[1];

              })

              .style("fill", "black")

              //iterate over each text svg

              .text(function(d){

                return d.place;

              });

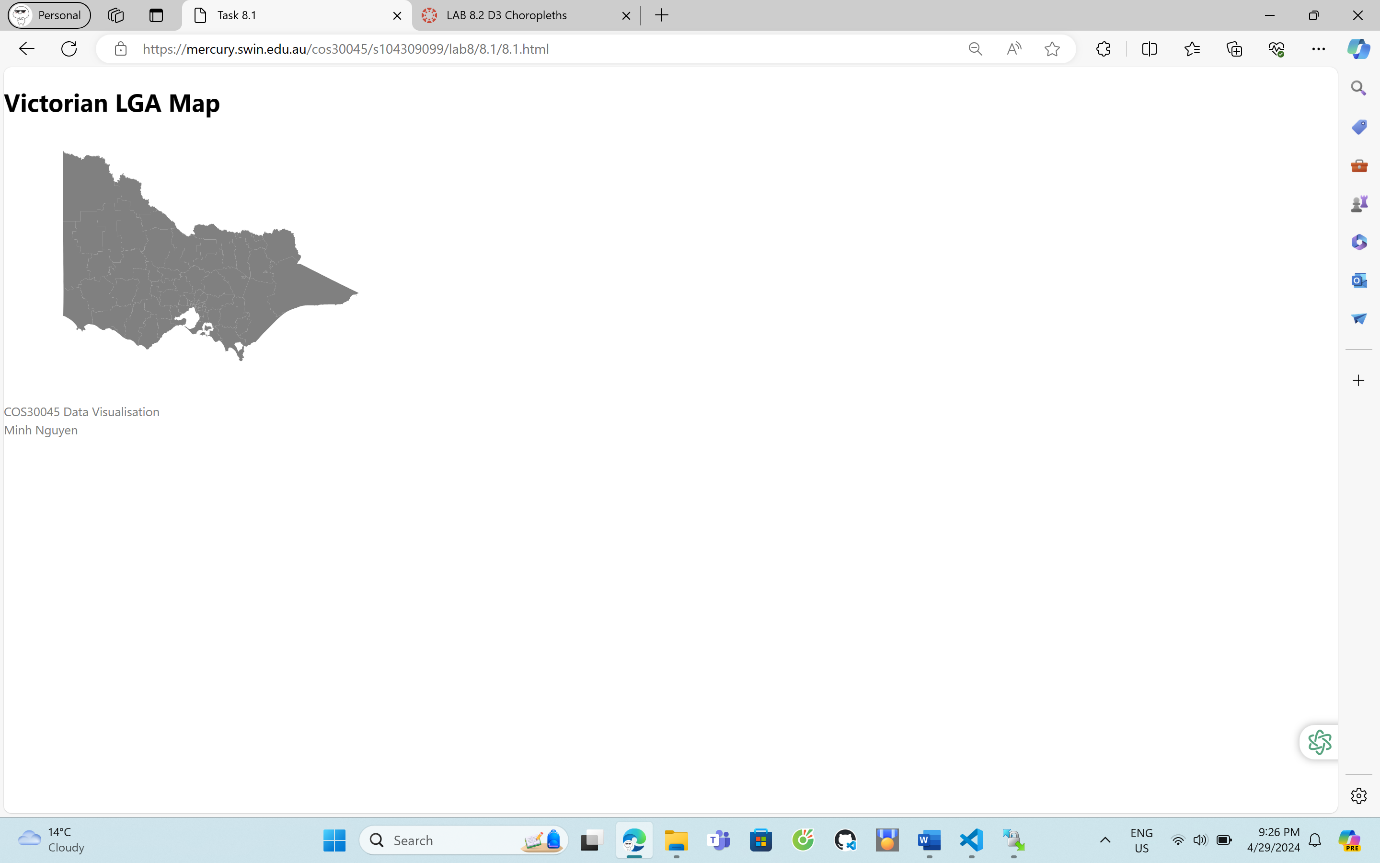
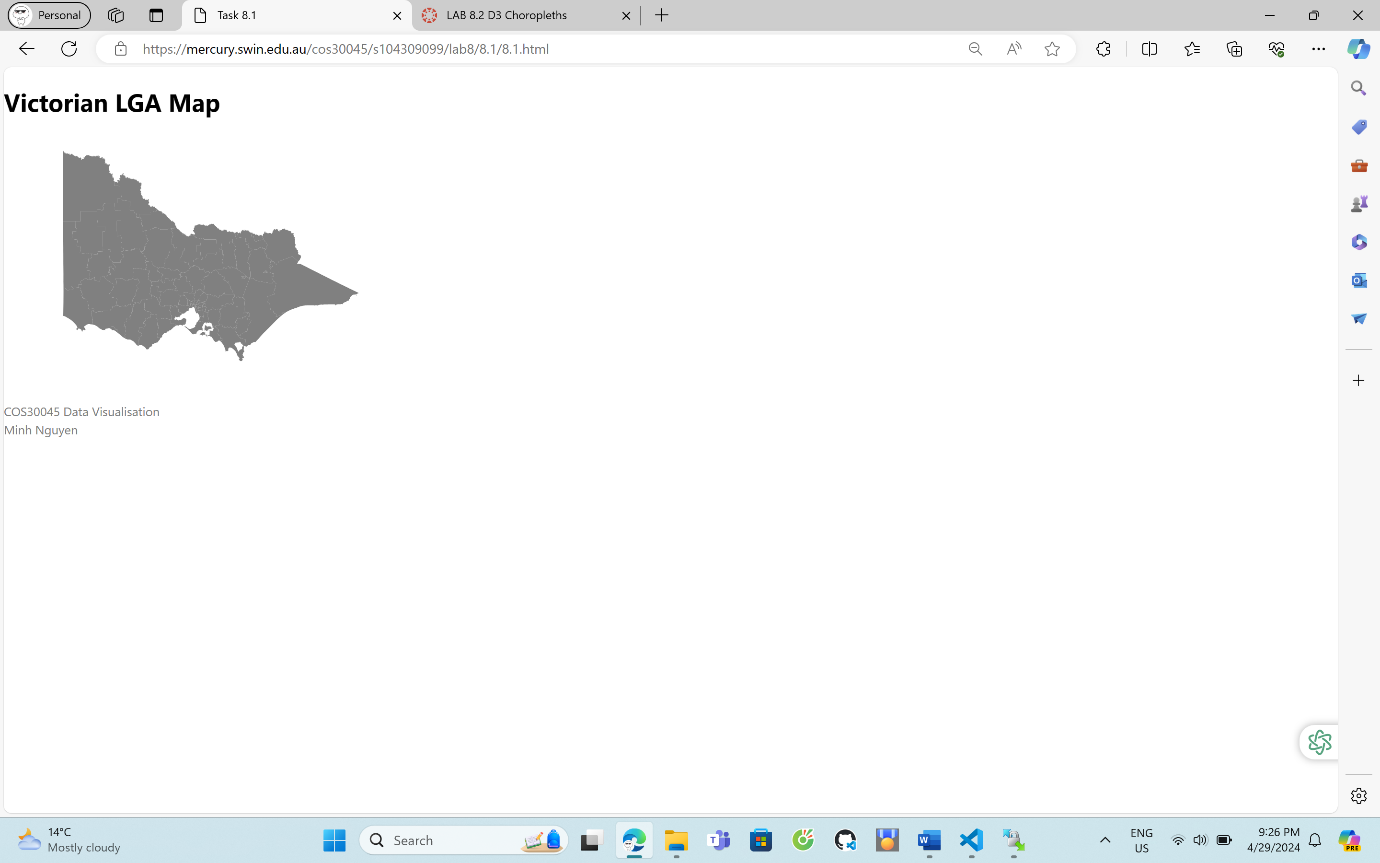
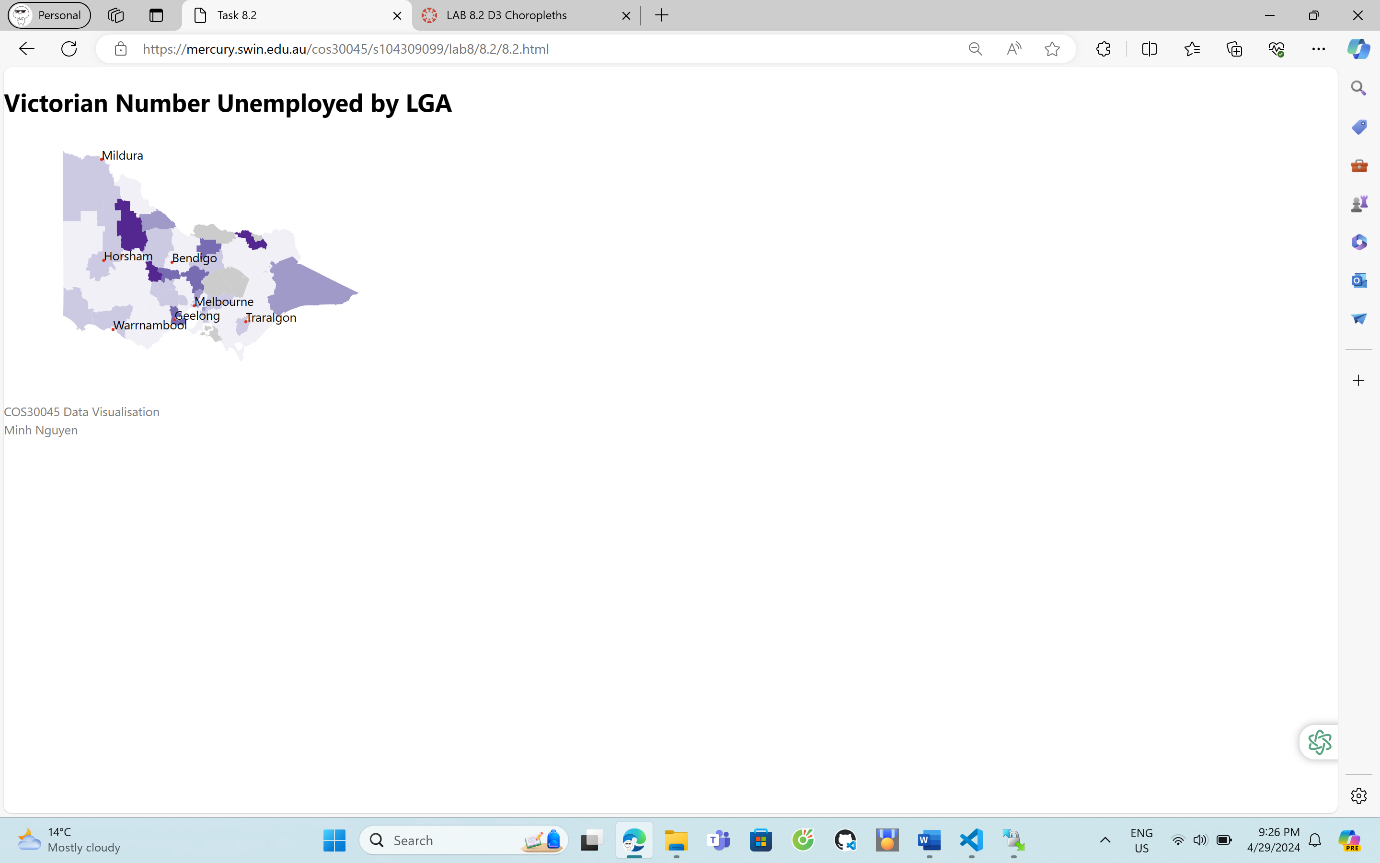
            });

        });

    });

  }

  window.onload = init;

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự độngẢnh có chứa văn bản, phần mềm, Biểu tượng máy tính, Hệ điều hành

Mô tả được tạo tự độngẢnh có chứa ảnh chụp màn hình, văn bản, phần mềm, Hệ điều hành

Mô tả được tạo tự động