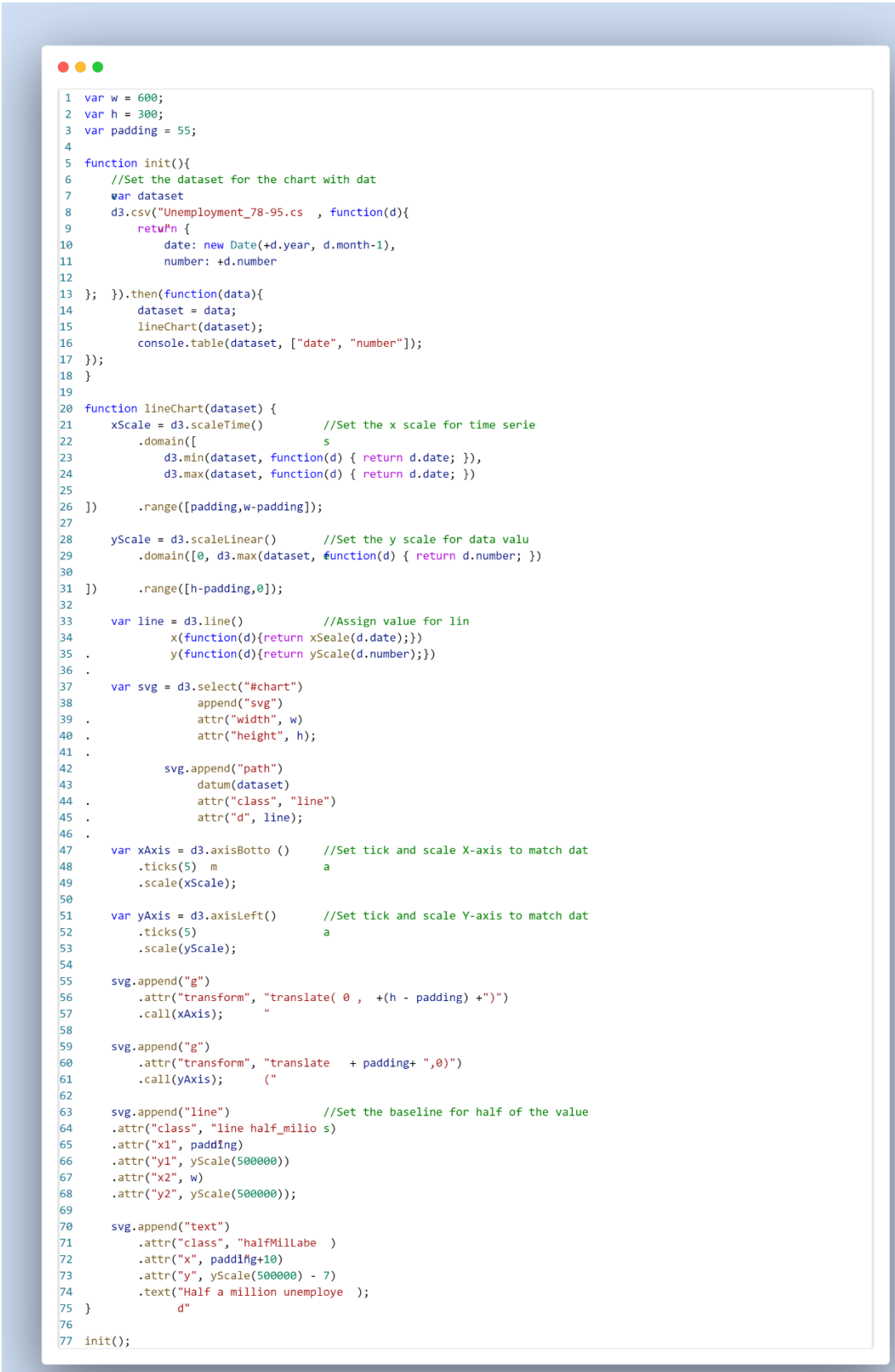
COS30045: Data Visualisation – Lab 7.1 – 8.2

Nguyen Khanh Toan – 104180605

Task 7.1

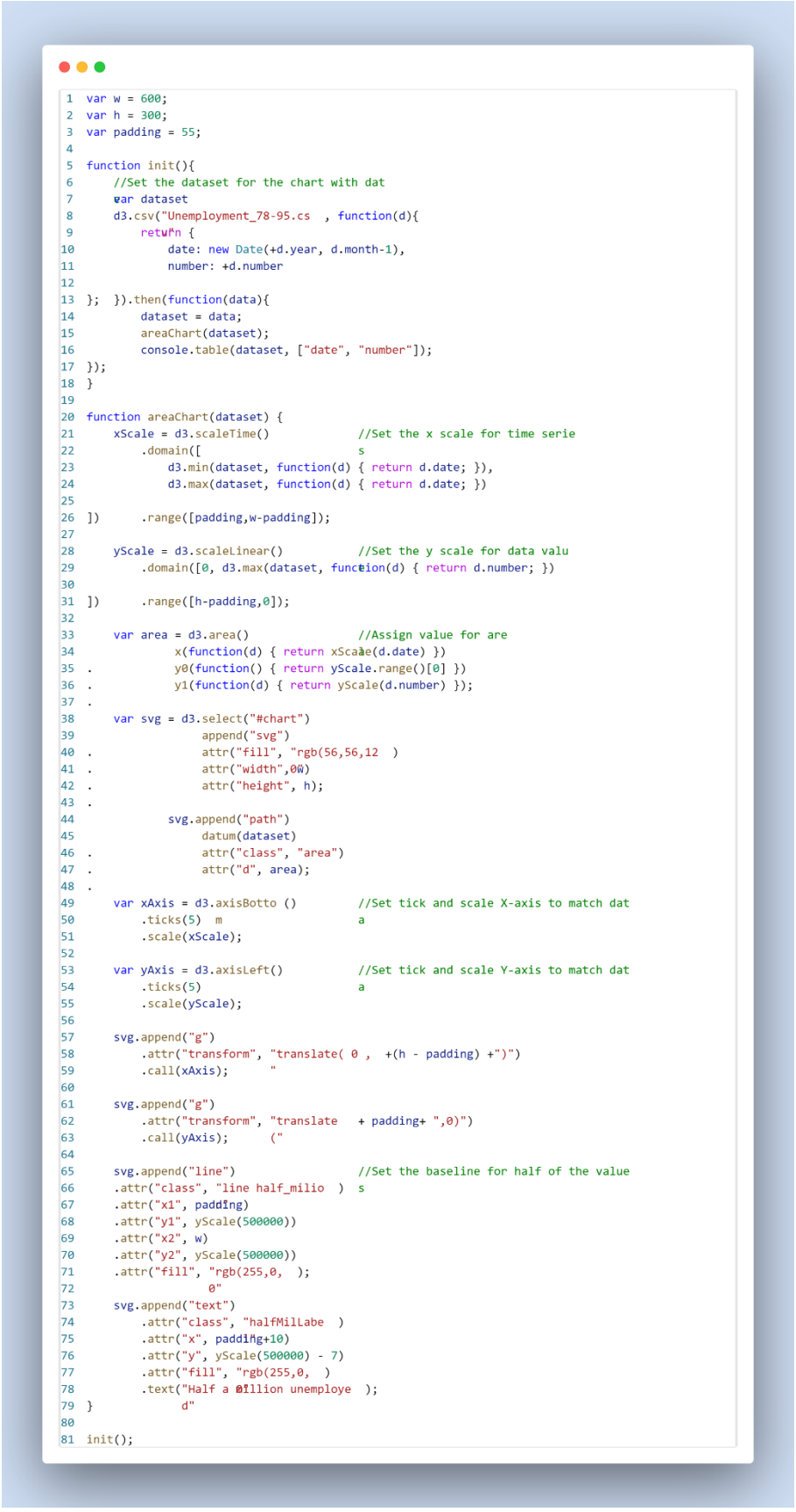
```html
<!DOCTYPE html>
<html lang = "en">
<head>
    <meta charset = "utf-8" />
    <meta name = "description"  content = "Data Visualisation"/>
    <meta name = "keywords"     content = "HTML,CSS,D3" />
    <meta name = "author"       content = "Toan Nguyen" />

    <title> Task 7.1 Drawing with data</title>

    <script src = "https://d3js.org/d3.v7.min.js" ></script>
    <link rel="stylesheet" href="styles.css">
</head>

<body>
    <h1> The D3 Journey Start Here </h1>

    <script src="scripts.js"></script>
    <div id = "chart"></div>

    <br>
    <bf>
    <footer style = "color:grey"> COS30045 Data Visualisation<br>
    Toan Nguyen</footer>
</body>

</html>
```

Fig 1: HTML code



```
var w = 600;
var h = 300;
var padding = 55;

function init(){
    //Set the dataset for the chart with dat
    var dataset
    d3.csv("Unemployment_78-95.cs  , 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"]);
});
}

function lineChart(dataset) {
    xScale = d3.scaleTime()         //Set the x scale for time serie
        .domain([                   s
            d3.min(dataset, function(d) { return d.date; }),
            d3.max(dataset, function(d) { return d.date; })

])      .range([padding,w-padding]);

    yScale = d3.scaleLinear()       //Set the y scale for data valu
        .domain([0, d3.max(dataset, function(d) { return d.number; })

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

    var line = d3.line()            //Assign value for lin
            x(function(d){return xScale(d.date);})
.           y(function(d){return yScale(d.number);})
.
    var svg = d3.select("#chart")
                append("svg")
.               attr("width", w)
.               attr("height", h);
.
            svg.append("path")
                datum(dataset)
.               attr("class", "line")
.               attr("d", line);
.
    var xAxis = d3.axisBotto ()     //Set tick and scale X-axis to match dat
        .ticks(5)  m                a
        .scale(xScale);

    var yAxis = d3.axisLeft()       //Set tick and scale Y-axis to match dat
        .ticks(5)                   a
        .scale(yScale);

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

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

    svg.append("line")              //Set the baseline for half of the value
    .attr("class", "line half_milio s)
    .attr("x1", padding)
    .attr("y1", yScale(500000))
    .attr("x2", w)
    .attr("y2", yScale(500000));

    svg.append("text")
        .attr("class", "halfMilLabe  )
        .attr("x", padding+10)
        .attr("y", yScale(500000) - 7)
        .text("Half a million unemploye  );
}           d"

init();
```

Fig2: JS code for Line chart



```
var w = 600;
var h = 300;
var padding = 55;

function init(){
    //Set the dataset for the chart with dat
    var dataset
    d3.csv("Unemployment_78-95.cs  , function(d){
        return {
            date: new Date(+d.year, d.month-1),
            number: +d.number

};  }).then(function(data){
        dataset = data;
        areaChart(dataset);
        console.table(dataset, ["date", "number"]);
});
}

function areaChart(dataset) {
    xScale = d3.scaleTime()             //Set the x scale for time serie
        .domain([                       s
            d3.min(dataset, function(d) { return d.date; }),
            d3.max(dataset, function(d) { return d.date; })

])      .range([padding,w-padding]);

    yScale = d3.scaleLinear()           //Set the y scale for data valu
        .domain([0, d3.max(dataset, function(d) { return d.number; })

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

    var area = d3.area()                //Assign value for are
            x(function(d) { return xScale(d.date) })
.           y0(function() { return yScale.range()[0] })
.           y1(function(d) { return yScale(d.number) });
.
    var svg = d3.select("#chart")
                append("svg")
.               attr("fill", "rgb(56,56,12  )
.               attr("width",0)
.               attr("height", h);
.
            svg.append("path")
                datum(dataset)
.               attr("class", "area")
.               attr("d", area);
.
    var xAxis = d3.axisBotto ()         //Set tick and scale X-axis to match dat
        .ticks(5)  m                    a
        .scale(xScale);

    var yAxis = d3.axisLeft()           //Set tick and scale Y-axis to match dat
        .ticks(5)                       a
        .scale(yScale);

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

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

    svg.append("line")                  //Set the baseline for half of the value
    .attr("class", "line half_milio  )  s
    .attr("x1", padding)
    .attr("y1", yScale(500000))
    .attr("x2", w)
    .attr("y2", yScale(500000))
    .attr("fill", "rgb(255,0,  );
                  0"
    svg.append("text")
        .attr("class", "halfMilLabe  )
        .attr("x", padding+10)
        .attr("y", yScale(500000) - 7)
        .attr("fill", "rgb(255,0,  )
        .text("Half a million unemploye  );
}           d"

init();
```

Fig 3: JS code for Area chart

Task 7.2

```html
<!DOCTYPE html>
<html lang = "en">
<head>
    <meta charset = "utf-8" />
    <meta name = "description"  content = "Data Visualisation"/>
    <meta name = "keywords"     content = "HTML,CSS,D3" />
    <meta name = "author"       content = "Toan Nguyen" />

    <title> Task 7.2 Drawing with data</title>

    <script src = "https://d3js.org/d3.v7.min.js" ></script>
    <link rel="stylesheet" href="styles.css">
</head>

<body>
    <h1> The D3 Journey Start Here </h1>

    <script src="scripts.js"></script>

    <br>
    <bf>
    <footer style = "color:grey"> COS30045 Data Visualisation<br>
    Toan Nguyen</footer>
</body>

</html>
```

Fig 4: HTML code

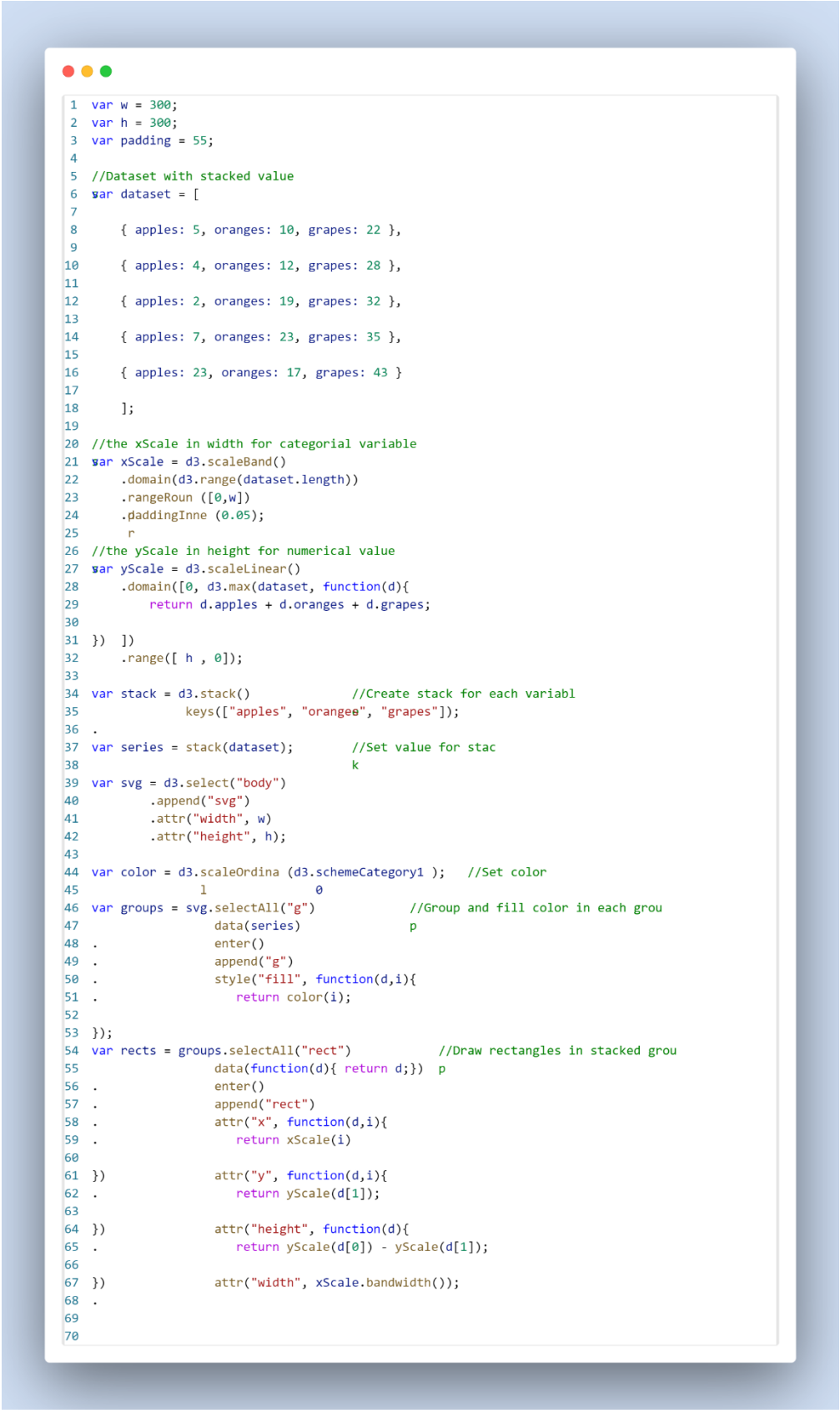
```
var w = 300;
var h = 300;
var padding = 55;

var innerRadius = 0;
var outerRadius = w/2;

var dataset = [5,10,20,45,28,58,42];    //Set the value for pie char
                                        t
var arc = d3.arc()                      //Set the radius for inner and outer circle
             innerRadius(innerRadius)   s
.            outerRadius(outerRadius);
.
var pie = d3.pie();

var svg = d3.select("body")
             append("svg")
.            attr("width", w)
.            attr("height", h);
.
var arcs = svg.selectAll("g.arc")       //Set the curve part for pie char
    .data(pie(dataset))                 t
    .enter()
    .append("g")
    .attr("class", "arc")
    .attr("transform", "translate  + outerRadius+ ", " + outerRadius +")");
                       ("
var color = d3.scaleOrdina (d3.schemeCategory1 );   //Set color for pie char
               l               0                    t
    arcs.append("path")                 //Draw pie and its colo
        .attr("fill", function(d,i){    r
            return color(i);

})      .attr("d", function(d,i){
            return arc(d,i);
        });

    arcs.append("text")                 //Set value in text for each pi
        .text(function(d){              e
            return d.value;

})      .attr("transform",function(d){
            return "translate  + arc.centroid(d) + ")";
        });        ("
```

Fig 5: JS code

Task 7.3

```html
<!DOCTYPE html>
<html lang = "en">
<head>
    <meta charset = "utf-8" />
    <meta name = "description"  content = "Data Visualisation"/>
    <meta name = "keywords"     content = "HTML,CSS,D3" />
    <meta name = "author"       content = "Toan Nguyen" />

    <title> Task 7.3 Drawing with data</title>

    <script src = "https://d3js.org/d3.v7.min.js" ></script>
    <link rel="stylesheet" href="styles.css">
</head>

<body>
    <h1> The D3 Journey Start Here </h1>

    <script src="scripts.js"></script>

    <br>
    <bf>
    <footer style = "color:grey"> COS30045 Data Visualisation<br>
    Toan Nguyen</footer>
</body>

</html>
```

Fig 6: HTML code



```
var w = 300;
var h = 300;
var padding = 55;

//Dataset with stacked value
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 }

    ];

//the xScale in width for categorial variable
var xScale = d3.scaleBand()
    .domain(d3.range(dataset.length))
    .rangeRoun ([0,w])
    .paddingInne (0.05);
     r
//the yScale in height for numerical value
var yScale = d3.scaleLinear()
    .domain([0, d3.max(dataset, function(d){
        return d.apples + d.oranges + d.grapes;

})  ])
    .range([ h , 0]);

var stack = d3.stack()              //Create stack for each variabl
             keys(["apples", "orangee", "grapes"]);
.
var series = stack(dataset);        //Set value for stac
                                    k
var svg = d3.select("body")
        .append("svg")
        .attr("width", w)
        .attr("height", h);

var color = d3.scaleOrdina (d3.schemeCategory1 );   //Set color
               l              0
var groups = svg.selectAll("g")             //Group and fill color in each grou
                data(series)                p
.               enter()
.               append("g")
.               style("fill", function(d,i){
.                  return color(i);

});
var rects = groups.selectAll("rect")            //Draw rectangles in stacked grou
                data(function(d){ return d;})  p
.               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());
.


```

Fig 7: JS code

Task 8.1

```html
<!DOCTYPE html>
<html lang = "en">
<head>
    <meta charset = "utf-8" />
    <meta name = "description"  content = "Data Visualisation"/>
    <meta name = "keywords"     content = "HTML,CSS,D3" />
    <meta name = "author"       content = "Toan Nguyen" />

    <title> Task 8.1 Drawing with data</title>

    <script src = "https://d3js.org/d3.v7.min.js" ></script>
    <link rel="stylesheet" href="styles.css">
</head>

<body>
    <h1> The D3 Journey Start Here </h1>

    <script src="scripts.js"></script>

    <br>
    <bf>
    <footer style = "color:grey"> COS30045 Data Visualisation<br>
    Toan Nguyen</footer>
</body>

</html>
```

Fig 8: HTML code

```
var w = 600;
var h = 300;
var padding = 55;

//Specifying the projection
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("body")
             .append("svg")
             .attr("width", w)
             .attr("height", h)
             .attr("fill", "grey");

//Read and Set the GeoJSON file to draw path
d3.json("LGA_VIC.json").then(function(json){

    svg.selectAll("path")
        .data(json.features)
        .enter()
        .append("path")
        .attr("d", path);
});


```

Fig 9: JS code

Task 8.2

```html
<!DOCTYPE html>
<html lang = "en">
<head>
    <meta charset = "utf-8" />
    <meta name = "description"  content = "Data Visualisation"/>
    <meta name = "keywords"     content = "HTML,CSS,D3" />
    <meta name = "author"       content = "Toan Nguyen" />

    <title> Task 8.2 Drawing with data</title>

    <script src = "https://d3js.org/d3.v7.min.js" ></script>
    <link rel="stylesheet" href="styles.css">
</head>

<body>
    <h1> The D3 Journey Start Here </h1>

    <script src="scripts.js"></script>

    <br>
    <bf>
    <footer style = "color:grey"> COS30045 Data Visualisation<br>
    Toan Nguyen</footer>
</body>

</html>
```

Fig 10: HTML code

```
var w = 600;
var h = 300;
var padding = 55;

//Set a range color for each patia
var color = d3.scaleQuantiz ()
             range(['#ffffcc','#c2e699','#78c679','#31a354','#006837'])
.
//Specifying the projection and Set the Geometry of the Ma
var projectio  = d3.geoMercator()
    n               center([145,-36.5])
.                   translate([w/2,h/2])
.                   scale(2450);
.

var path = d3.geoPath()
             projectio (projectio );
.            n          n
var svg = d3.select("body")
             append("svg")
.            attr("width", w)
.            attr("height", h)
.            attr("fill", "grey");
.
//Read the data form CSV fil
var data = d3.csv("VIC_LGA_unemployment.cs  ).then(function(data) {
    color.domain([v"
        d3.min(data, function(d){   return d.unemploye ; }),
        d3.max(data, function(d){   return d.unemploye ; })
    ])                                        d
    return data;
});

//Read and Set the GeoJSON file to draw pat
json = d3.json("LGA_VIC.jso  );
Promise.all([data, json]).then(function(values) {
    var data = values[0];
    var json = values[1];

    for (var i = 0; i <data.length; i++) {
        var dataState = data[i];
        for (var j = 0; j < json.features.length; j++) {

            var jsonState = json.features[j];
            if (jsonState.propertie .LGA_name === dataState.LGA) {
                jsonState.propertie .value = +dataState.unemploye
                break;    s                              d

}       }
    }
    //Draw the geometry and set its color properties coresponding to the dat
    avg.selectAll("path")
        .data(json.features)
        .enter()
        .append("path")
        .attr("d", path)
        .style("fill", function(d){
            var value = d.propertie .value;
            console.log("COLOR "+value);
            if (value) {
                return color(value);
            } else {
                return "#364";

}       });
});
```

Fig 11: JS code