  // If you look closely, you can also see that the values associated with these properties are all strings. This is probably not what you want in the case of numbers. When loading CSVs and other flat files, you have to do the type conversion.

  // We will see more of this in other tasks, but a simple way to do this is to use the + operator (unary plus). forEach can be used to iterate over the data array.

   // with json no need to put + for numeric , no need conversion

   const width = window.innerWidth \* 0.8,

   height = window.innerHeight \* 0.8,

   margin = { top: 20, bottom: 50, left: 60, right: 40 };

       let svg;

    // inisiate the object state

           let state = {

               datageojson: null,

               datahappy: null,

              // dataheat: null,

               hover: { // hover is an object and it has two properties latitude and

                 latitude1: null,

                 longitude1: null,

               },

             };

           //read in topojson

           /\*d3.queue()

           .defer(d3.json,"usState.json")

           .await(ready)\*/

           Promise.all([

               d3.json("world.topojson"),d3.csv("happiness.csv"),  // pull a new file in promises

           ]).then( ([datageojson,datahappy])=>{ // it returns an object

            var countries = topojson.feature(datageojson,datageojson.objects.countries).features;

            console.log(countries)

               state.datageojson = countries;

               state.datahappy= datahappy;

            //  state.dataheat= dataheat; //instead of data.object.countries.features

               console.log(" data state: ", state.datageojson);

               console.log("data happy", state.datahappy);

               // 52 states it will return an array features :52 lines

             // console.log("data heat", state.dataheat);

               init();

           });

           function init() {

   var zoom = d3.zoom();

               svg=d3.select("#map")

               .append("svg")

               .attr("height",height  )

               .attr("width",width)

               .call(d3.zoom().on('zoom',()=>{

                console.log('zoom');

                svg.attr("transform", d3.event.transform)

               }))

                .append("g")

   const projection = d3.geoMercator();

   var path=d3.geoPath()

   .projection(projection)

   /\*state.datageojson.foreach( d=>{

     Object.assign

   });

   \*/

          //create the element

           svg.selectAll(".state")

              .data(state.datageojson)

              .enter().append("path")

              .attr("class","state")

             .attr("d",path)

           .append('title')

          // .text('hello');

          .text(d=> d.properties.name );

          svg.selectAll(".state\_label") // we hace the starting condition enter (0)

          .data(state.datahappy)

          .enter().append("text")

          .attr("class","state\_label")

          .transition().duration(1000) // and we have what we want to do transiition to in the othe r exemple we tart with h 0 and we did transition to the good height

          .text(function(d){

              return d.Ladder\_score

          })

          .attr("dx",5)

          .attr("dy",6)

           }