SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek

Projektni zadatak iz predmeta

VIZUALIZACIJA PODATAKA

Konzumacija alkohola u državama svijeta

Student: Danilo Petrović, Računarstvo - DRC

Mentor: Denis Ivanović

U Osijeku, lipanj 2025.

SADRŽAJ

[**1. KV1 - Definiranje projektnog zadatka 3**](#_4i6s1smptslt)

[1.1. Projektni zadatak 3](#_wgoiizki7jnh)

[1.2. Podatci 3](#_jjnsdui0w5uv)

[1.3. Obrada podataka 3](#_jfu6ybs7kwrw)

[1.4. Relevantne vrste prikaza za korištene podatke 3](#_1g4rlc1z1va6)

[**2. KV2 - Dizajn vizualizacije podataka. 4**](#_smxrkz6d6wy8)

[2.1. Pitanja na koja vizualizacija daje odgovor 4](#_cqmiylhgneug)

[2.2. Skica vizualizacije podataka 4](#_i61spu8qyxmd)

[2.3. Postojeća rješenja i primjeri 4](#_qmiyijy4jsp8)

[2.4. Prilagodba podataka 4](#_cch1vltulvr)

[2.5. Boje i podatci 4](#_w22k3frzbtb2)

[**3. KV3 - Izrada prototipne vizualizacije podataka 5**](#_k5q372z1w040)

[3.1. Osnovne funkcionalnosti i ponašanja 5](#_yaexlmqqdt0u)

[3.2. Napredne funkcionalnosti i ponašanja: 5](#_y300tbmoybm5)

[3.3. Implementacija osnovnih funkcionalnosti 5](#_g3q3h1rhqzyz)

[3.4. Implementacija osnovnog ponašanja 5](#_itz19pnh67g0)

[**4. KV4 - Izrada konačne vizualizacije podataka 6**](#_sj4kshcbtchw)

[4.1. Implementacija osnovnih funkcionalnosti 6](#_dc19rzmsn9k5)

[4.2. Implementacija osnovnog ponašanja 6](#_exvg8zl4eo3w)

[4.3. Implementacija naprednih funkcionalnosti 6](#_ackz82cvngj2)

[4.4. Implementacija naprednog ponašanja 6](#_le6sbg8zsugz)

[**5. KV5 - Dovršetak projektnog zadatka i pisanje dokumentacije 7**](#_woao7m26bf75)

[5.1. Eventualne preinake i dorade rješenja - u dogovoru s nastavnikom 7](#_116xkgvrmrqh)

[5.2. Izrada dokumenta - projektne dokumentacije 7](#_6zplla712qe7)

[**Literatura 8**](#_ahmk8vkny1f9)

[**Prilog I 9**](#_jefyptb5hazf)

# KV1 - Definiranje projektnog zadatka

## Projektni zadatak

Naziv zadatka: Konzumacija alkohola u državama svijeta

Opis problema: Alkohol je jedan od vodećih razloga prerane smrti u svijetu. Razumjeti gdje, koliko i kako ljudi piju preduvjeti su za uspješne preventivne kampanje i ukazivanje na štetnost konzumacije.

Opis zadatka: Zadatak podrazumijeva prikupljanje globalne podatke o potrošnji alkohola, te vizualizaciju pronađenih podataka zbog dolaska do brzog zaključka o tome koje zemlje su najugroženije po pitanju konzumacije alkohola, te preusmjeravanje pažnje na njih kako bi se osigurala potrebna pomoć.

Cilj projekta: prikaz kojim zemljama, tj. njezinim stanovnicima prijeti najviše opasnosti zbog razina alkohola koje unose u prosjeku u toku godine.

Poveznica na git repozitorij projekta: https://github.com/danilo-petrovic/Vizualizacija-projekt.git

## Podatci

* + 1. *Internetska stranica “Our World in Data” je znanstveni portal koji omogućuje pristup skupovima podataka koji omogućuju uvid u podatke vezane uz neki problem, kao što su emisije CO2, siromaštvo, očekivano životno doba i sl. Kao takva, poslužila je za pronalazak pogodnih podatkovnih skupova za implementaciju ovog projektnog zadatka.*
    2. *Na poveznici* [*https://ourworldindata.org/alcohol-consumption*](https://ourworldindata.org/alcohol-consumption) *omogućen je pristup raznim podacima koji omogućuju realizaciju projektnog zadatka, kao što su: konzumacija alkohola po osobi, postotak osoba koje pije alkohol, konzumacija po spolu i sl.*

## Obrada podataka

* + 1. *Korišteni podaci rezultat su istraživanja Svjetske zdravstvene organizacije (engl. WHO), stoga nakon kratkog uvida u podatke moglo se zaključiti da su cjeloviti, konzistentni i kvalitetni za uporabu.*

## Relevantne vrste prikaza za korištene podatke

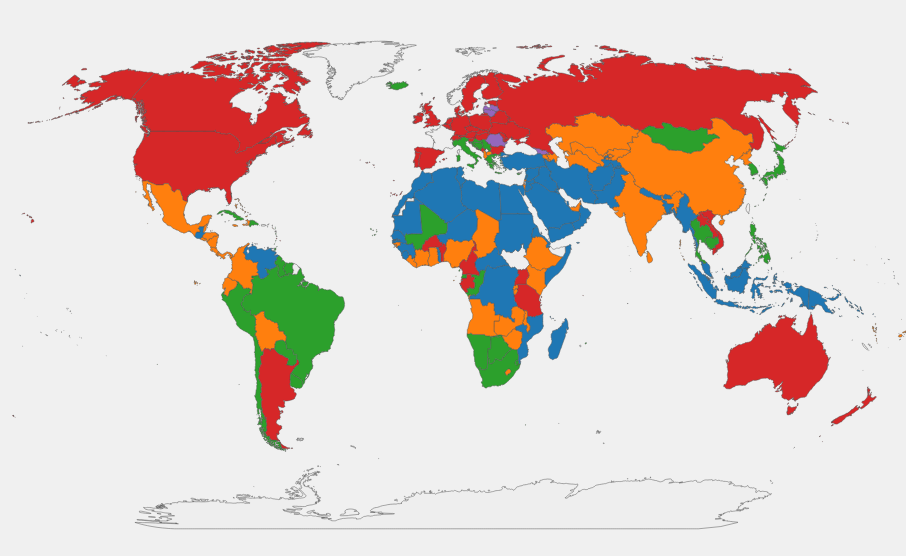
* + 1. *Histogrami, kružni odsječci i sl.*

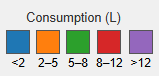
# KV2 - Dizajn vizualizacije podataka.

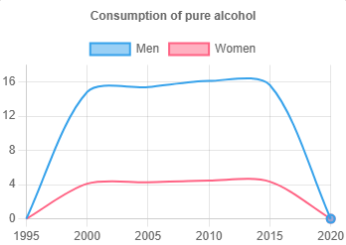
## Pitanja na koja vizualizacija daje odgovor

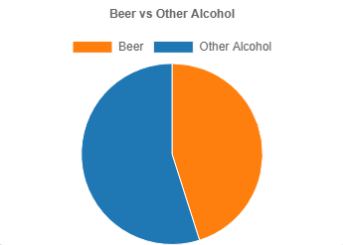
* + 1. Koliko litara alkohola popije prosječan stanovnik neke države. Koliki postotak muškaraca i žena neke države konzumira alkohol. Prikaz konzumacije piva po državama.

## Skica vizualizacije podataka

* + 1. 

 - filter po količini alkohola

 - usporedba konzumacije žena i muškaraca

 - odnos konzumacije piva i ostalih pića

## Postojeća rješenja i primjeri

* + 1. [*https://ourworldindata.org/alcohol-consumption*](https://ourworldindata.org/alcohol-consumption)

<https://global-relocate.com/rankings/alcohol-per-capita-consumption>

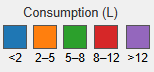
U projektu će se koristiti podaci o konzumaciji alkohola po stanovniku svake države, postotak muškaraca i žena koji konzumiraju alkohol i sl.

## Prilagodba podataka

* + 1. Prilikom istraživanja podatkovnih skupova, pronađeni su oni koji nisu zahtijevali nikakve prilagodbe, osim povezivanja više različitih podatkovnih skupova s ciljem boljeg predstavljanja podataka

## Boje i podatci

* + 1. Korištene boje su plava, narančasta, zelena, crvena i ljubičasta. Svaka od njih predstavlja razinu konzumacije alkohola te su jasno označeni rasponi koje predstavljaju.



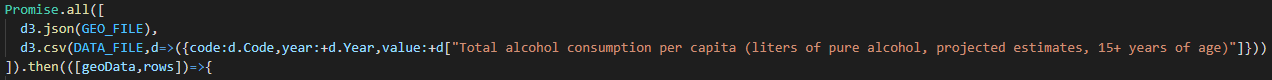
# KV3 - Izrada prototipne vizualizacije podataka

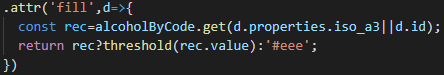
## Osnovne funkcionalnosti i ponašanja

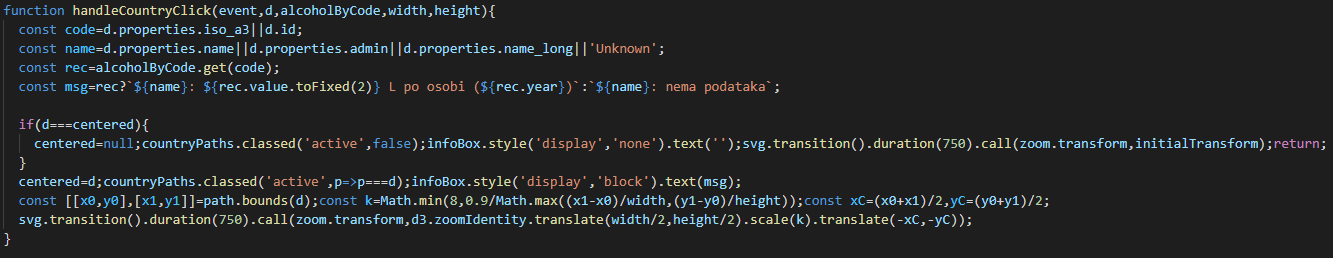
* + 1. Učitavanje geoJSON karte svijeta i CSV podataka o konzumaciji alkohola, prikaz zemalja obojenih prema količini alkohola po osobi, uvećavanje karte i pomicanje po njoj i sl.
    2. Prilikom prijelaza preko države prikazuje se njeno ime. Klikom na neku državu, ona se fokusira i prikazuju se informacije, dok klikom na prazan dio mape vraćamo prikaz na početno stanje.
    3. Korisnik može odabrati zemlju klikom.

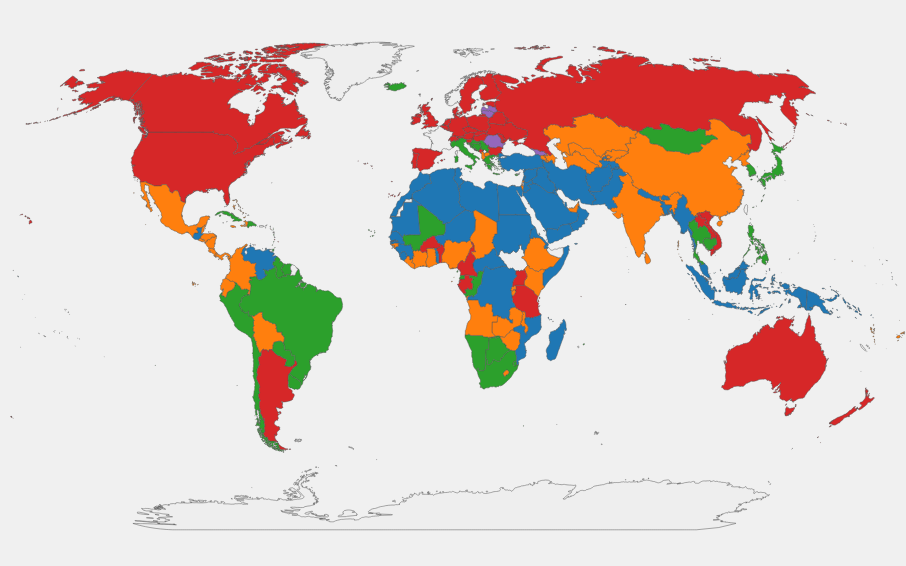
## Napredne funkcionalnosti i ponašanja:

* + 1. GeoJSON i CSV podaci se učitavaju asinkrono pomoću Promise.all funkcije, koja čeka da se sve operacije izvrše prije nastavka izvršavanja koda. Bojanje zemalja se vrši pomoću threshold skale. Funkcija handleCountryClick omogućava uvećanje kliknutih država te prikaz traženih informacija.



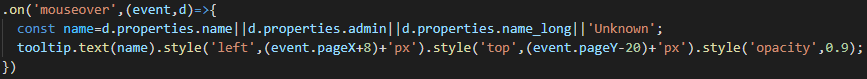




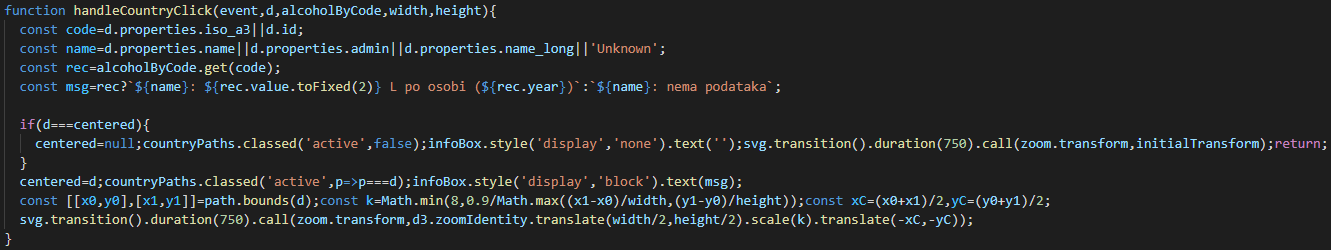


## Implementacija osnovnih funkcionalnosti

* + 1. Kada korisnik prijeđe mišem preko države, pojavljuje se njen naziv.



Kada korisnik klikne na državu, dolazi do ispisivanja podataka o izabranoj državi



Klikanjem van granica, resetira se prikaz

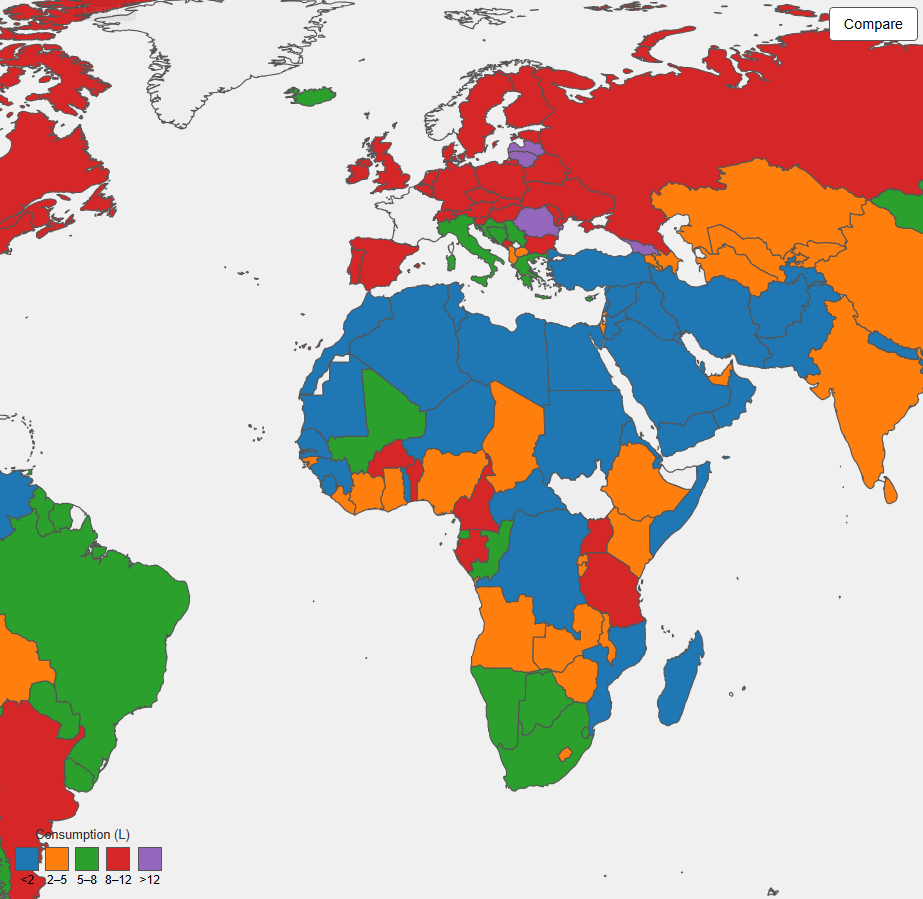


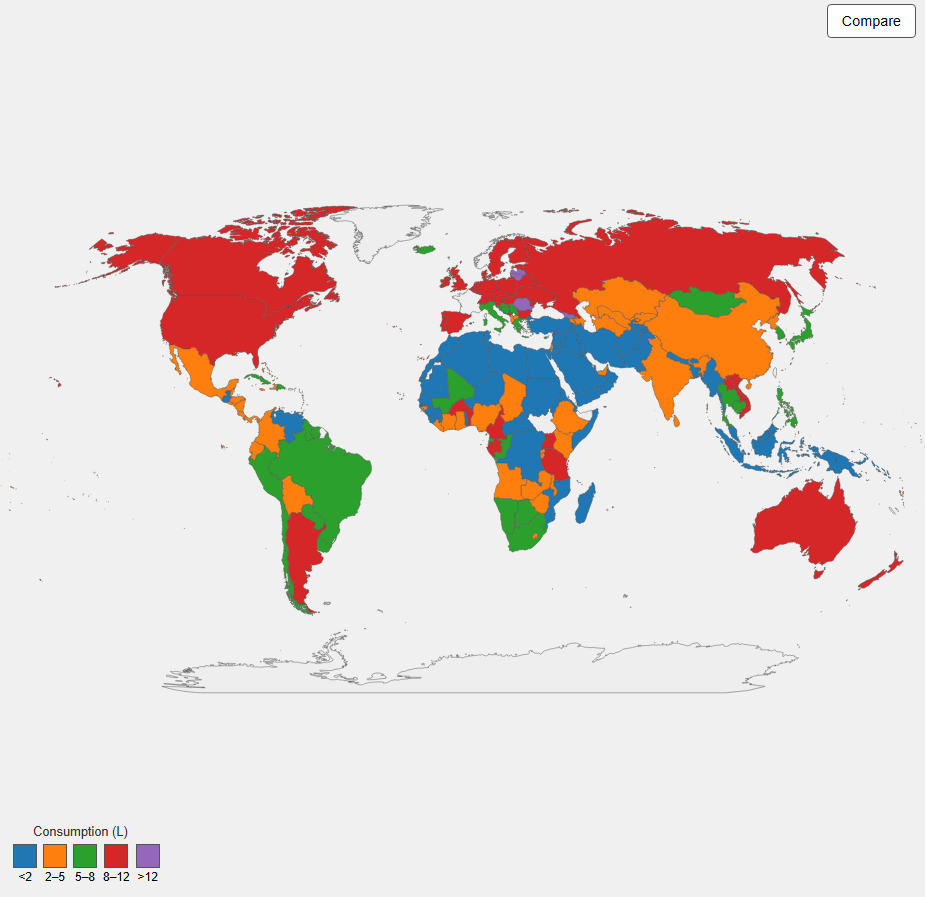
Korisnik može slobodno uvećavati i pomicati kartu koristeći kotač miša i povlačenje



* + 1. Testirati funkcionalnosti i osigurati da su ispravne, tj. da funkcioniraju na očekivani način. Potrebno potvrditi slikom.

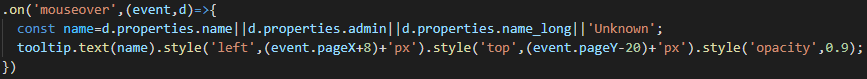




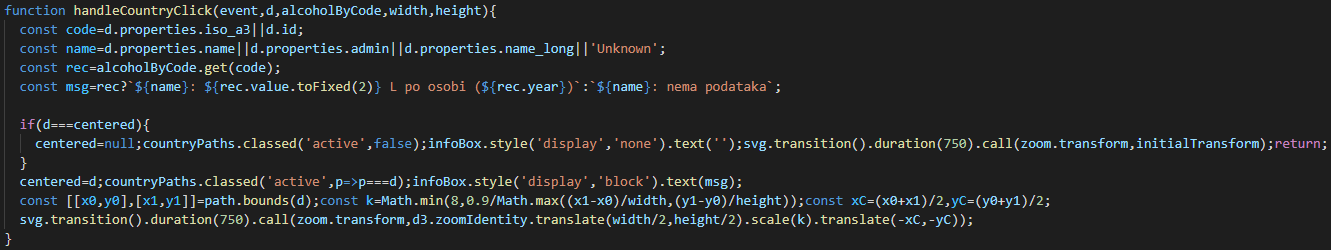


## Implementacija osnovnog ponašanja

* + 1. Kada korisnik prijeđe mišem preko države, pojavljuje se njen naziv.



Kada korisnik klikne na državu, dolazi do ispisivanja podataka o izabranoj državi

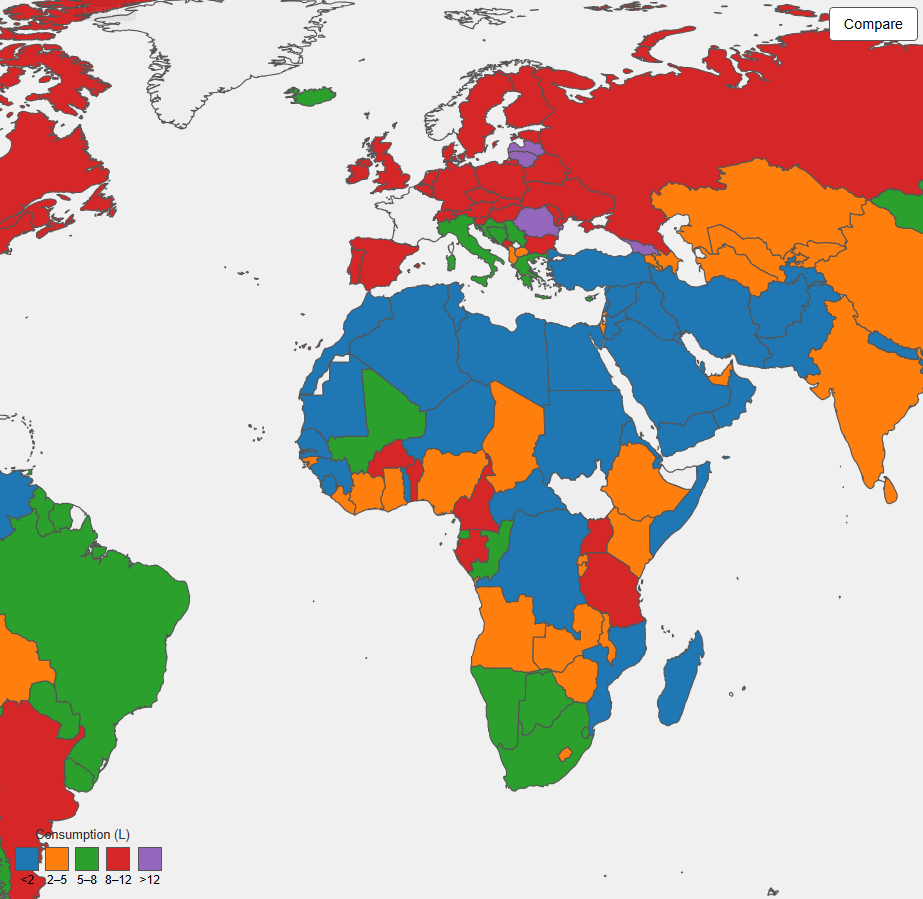


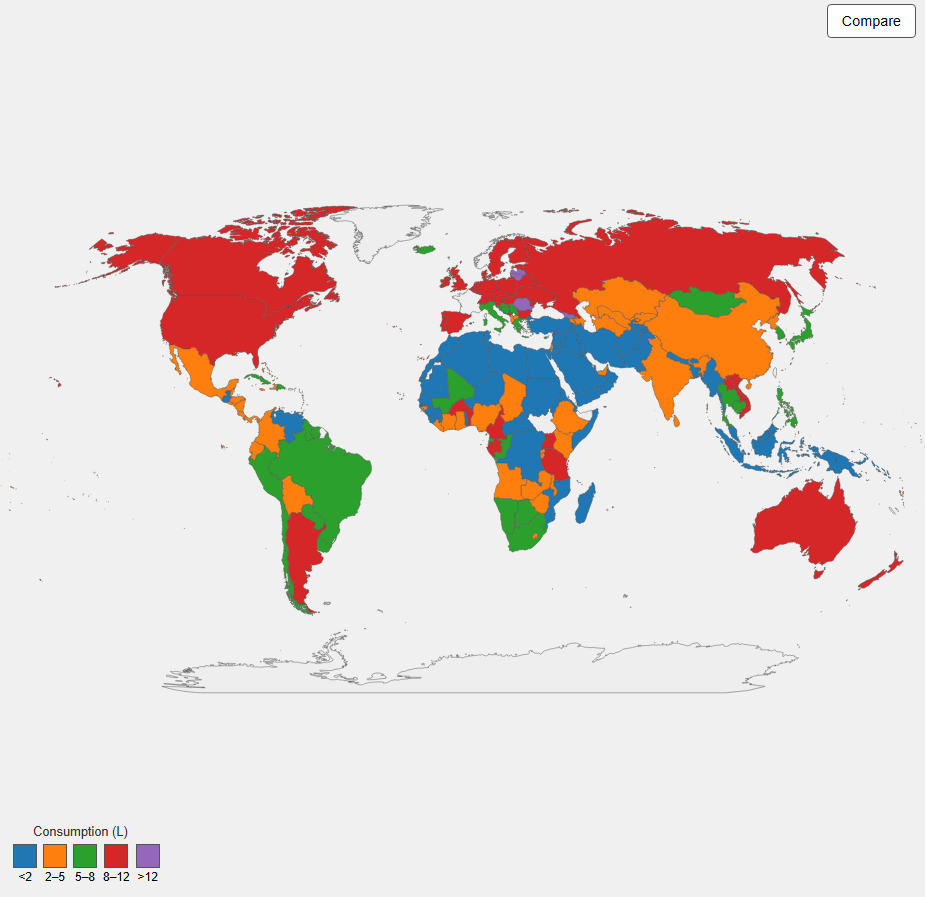
Klikanjem van granica, resetira se prikaz



Korisnik može slobodno uvećavati i pomicati kartu koristeći kotač miša i povlačenje

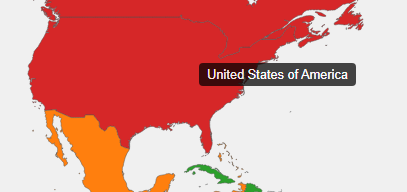


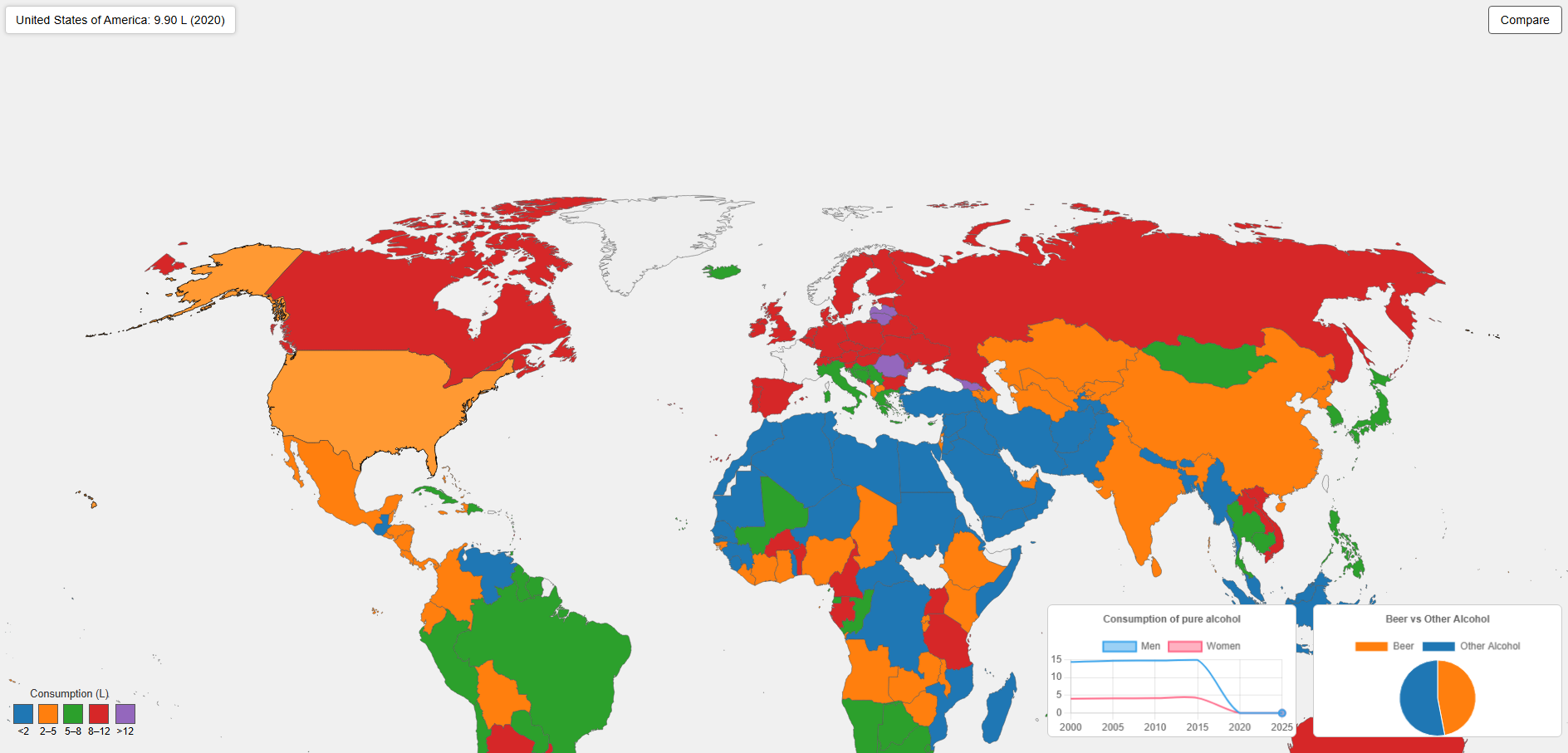


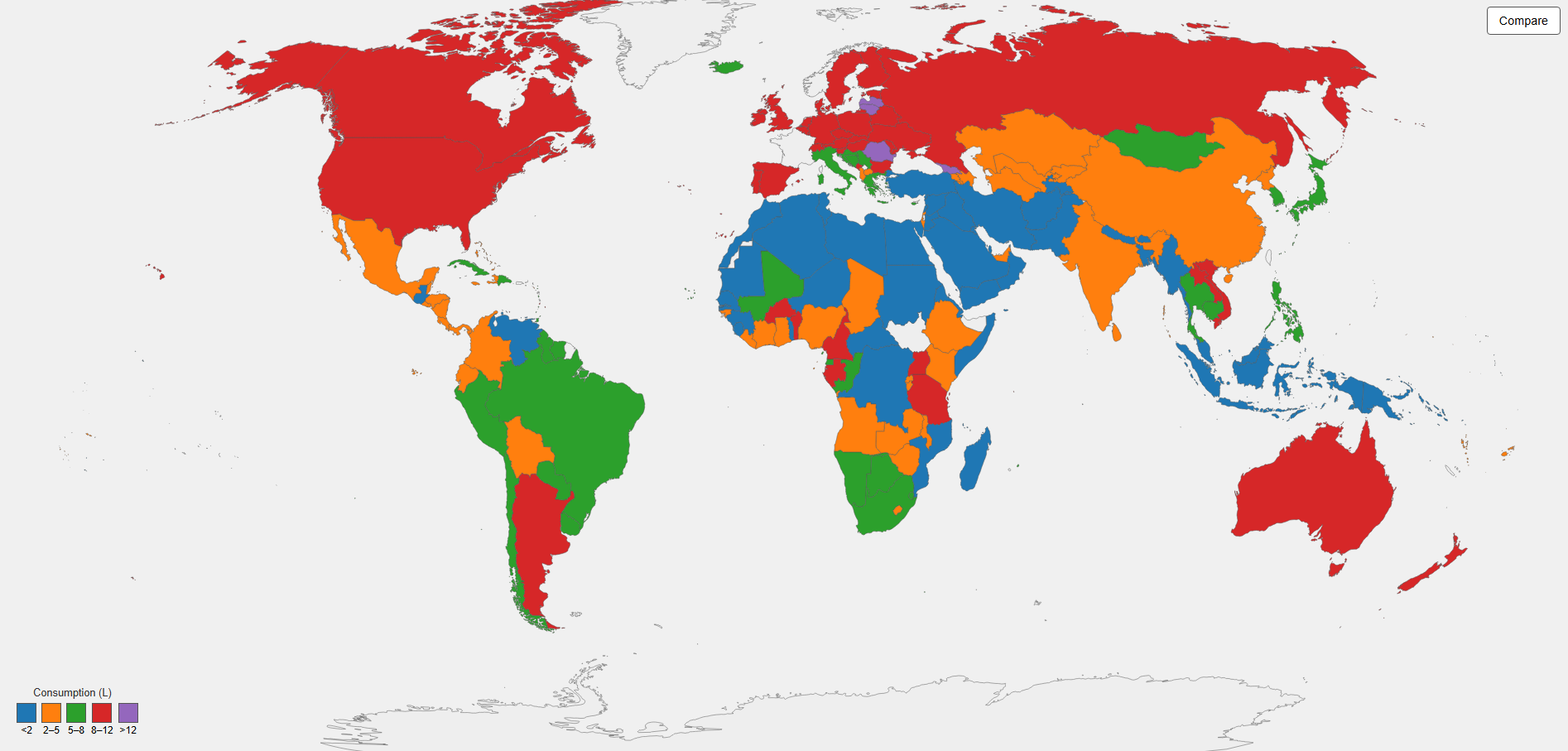


# KV4 - Izrada konačne vizualizacije podataka

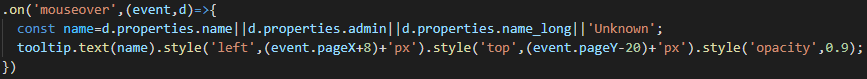
## Implementacija osnovnih funkcionalnosti

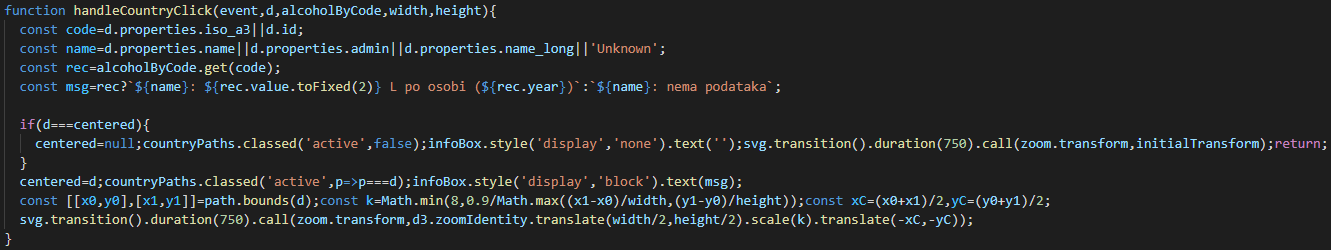
* + 1. 





## Implementacija osnovnog ponašanja

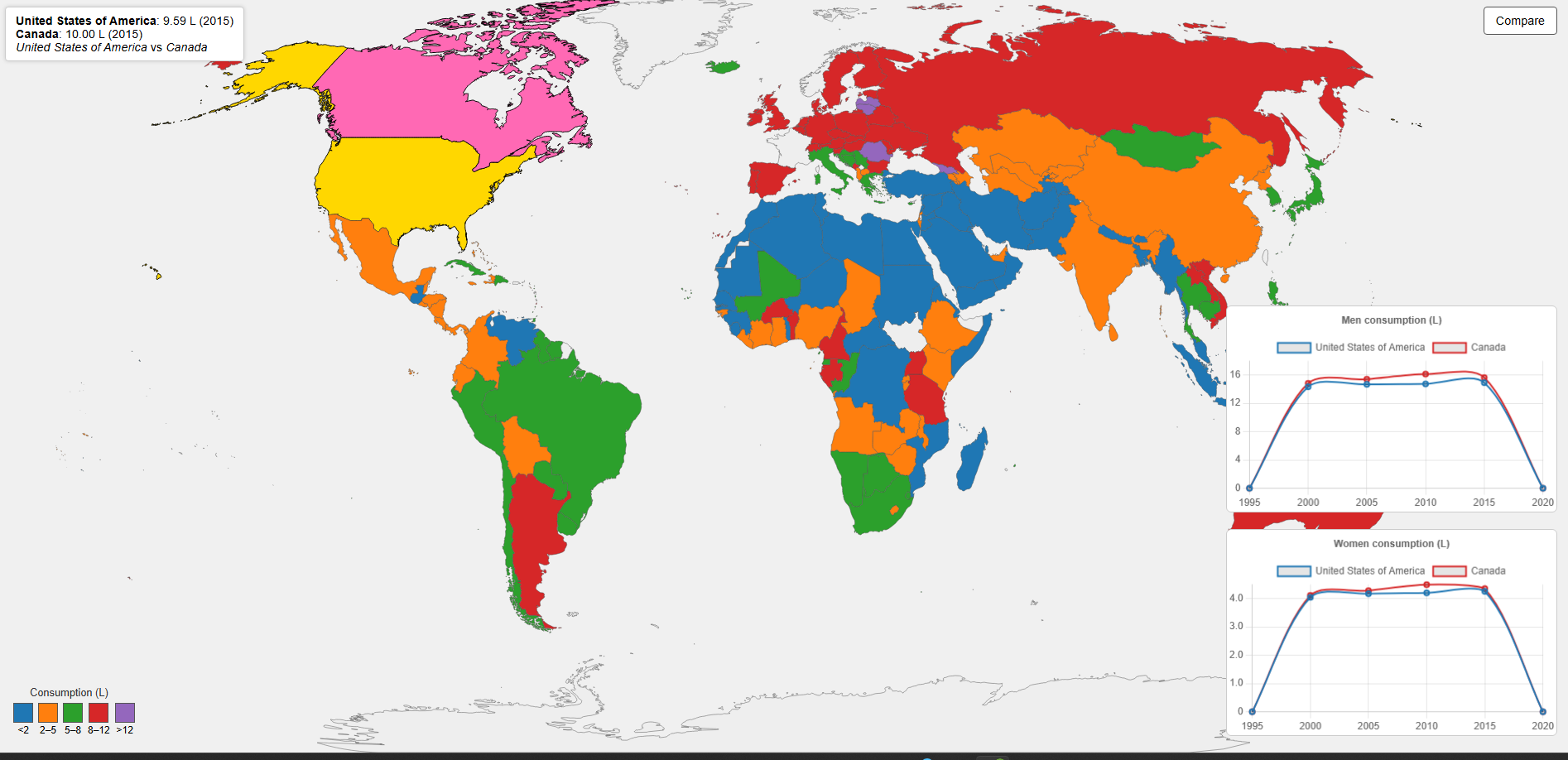
* + 1. 

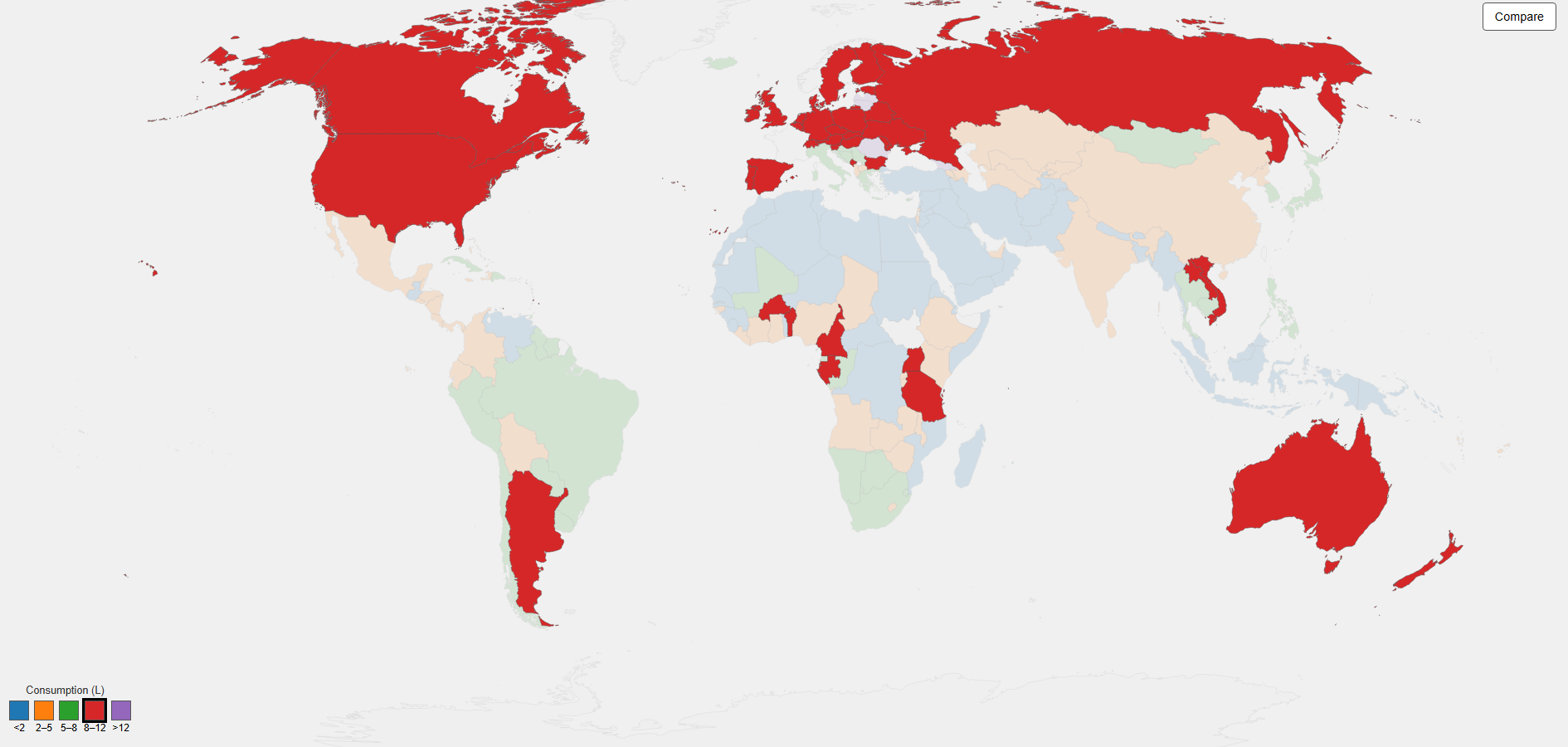




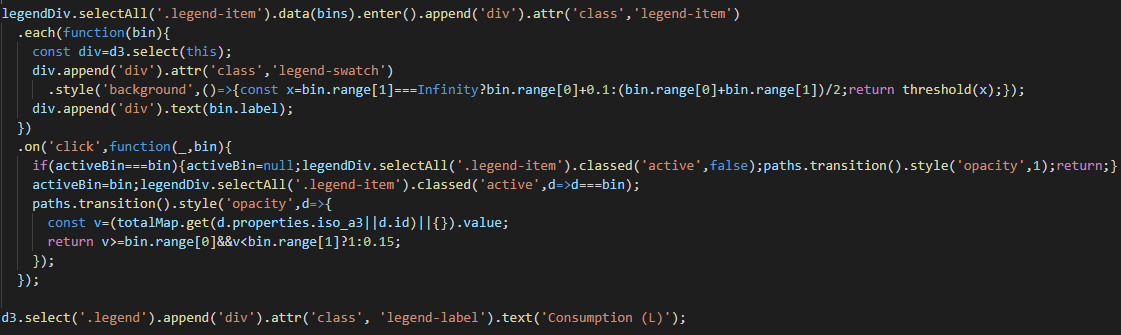
## Implementacija naprednih funkcionalnosti

* + 1. Usporedba dviju odabranih država, filter prema konzumaciji alkohola

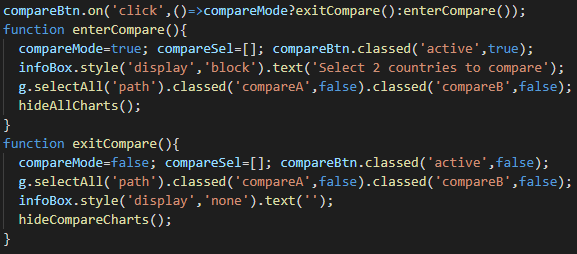


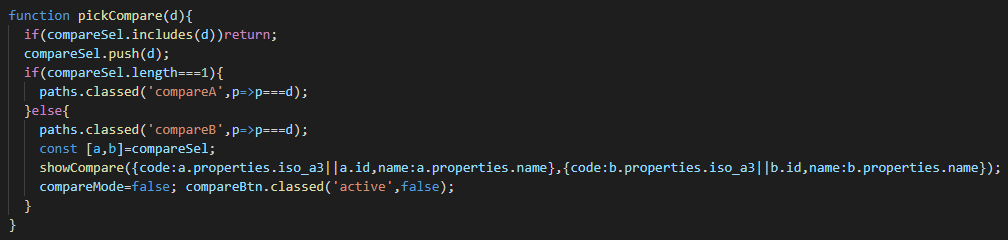


## Implementacija naprednog ponašanja



Ova funkcionalnost omogućava korisniku da klikne na legendu u dnu ekrana kako bi filtrirao zemlje po količini konzumiranog alkohola (izraženo u litrama po osobi). Zemlje koje spadaju u izabrani opseg ostaju jasno vidljive, dok se ostale zasjene. Ponovni klik deaktivira filter.



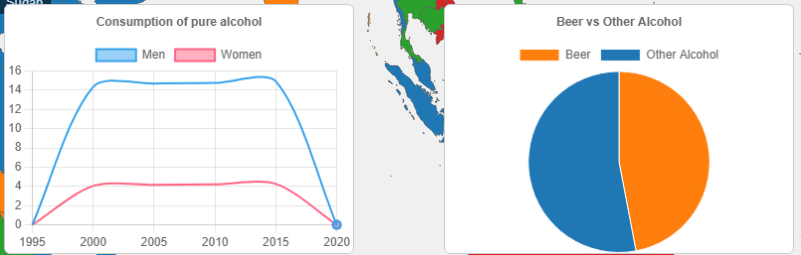


Ova funkcionalnost omogućava korisniku da usporedi konzumaciju alkohola između **dvije zemlje**. Aktivira se klikom na dugme „Compare“. Korisnik zatim klikne na dvije različite zemlje, koje se označe različitim bojama. Nakon drugog izbora, prikazuju se **dva grafa** (za muškarce i žene) koji prikazuju podatke o konzumaciji alkohola od 1995. do 2020. godine.

# KV5 - Dovršetak projektnog zadatka i pisanje dokumentacije

## Eventualne preinake i dorade rješenja - u dogovoru s nastavnikom

Jedina promjena koju je trebalo napraviti je uvećati grafove koji se iscrtavaju prilikom odabira neke države.



## Izrada dokumenta - projektne dokumentacije

Zadatak je bio izraditi interaktivnu web-aplikaciju koja prikazuje prosječnu količinu alkohola koja se konzumira u svim državama svijeta. Pored prikaza prosječne količine za svaku državu, vidljiva je usporedba konzumacije između muškaraca i žena, te odnos konzumacije piva i ostalih alkoholnih pića. Također je dostupan način rada u kojem se mogu usporediti podaci dvije države, te filter koji izdvaja države na osnovu prosječne količine alkohola koja se konzumira.

**Hijerarhija projekta.**

Projekt se sastoji od sljedećih datoteka:

* alcohol-consumption-per-capita-men-women.csv
* beer-as-share-alcohol-consumption.csv
* custom.geo.json
* project.html
* style.css
* total-alcohol-consumption-per-capita-litres-of-pure-alcohol.csv

**Popis korištenih tehnologija, bez opisa.**

* HTML5
* CSS
* JavaScript
* D3.js

**Upute za postavljanje.**

* Preuzeti sve datoteke koje su navedene ranije
* Pokrenuti projekt (project.html datoteku) pomoću ekstenzije Live Server u VS Code

**Upute za korištenje.**

* Nakon učitavanja stranice, odaberite državu po želji. Pojavit će se svi dostupni podaci o prosječnoj konzumaciji alkohola u državi, te 2 grafa koji prikazuju odnos konzumacije između žena i muškaraca i odnos konzumacije piva u odnosu na druga alkoholna pića.
* U donjem lijevom kutu nalazi se filter i prilikom odabira neke boje, prikazuju se države koje spadaju u određeni raspon konzumacije (npr. zelena boja označava države koje u prosjeku konzumiraju između 5-8 L alkohola). Ponovnim klikom na boju, filter se gasi.
* U gornjem desnom kutu nalazi se „Compare“ gumb koji omogućuje usporedbu 2 različite države.
* Klikom miša na prazni dio mape, prikaz se vraća u početno stanje, bez obzira na trenutni način rada i poziciju na mapi.

# Literatura

1. https://d3js.org/
2. https://worldpopulationreview.com/country-rankings/alcohol-consumption-by-country
3. https://ourworldindata.org/alcohol-consumption
4. https://developer.mozilla.org/en-US/

# Prilog I

Poveznica na git repozitorij projekta: https://github.com/danilo-petrovic/Vizualizacija-projekt.git

Programski kod

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>World Alcohol Consumption Map</title>

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

</head>

<body>

<div id="wrapper">

  <svg class="map"></svg>

  <div class="info-box"></div>

  <div class="legend"></div>

  <button id="compare-btn">Compare</button>

  <div class="tooltip"></div>

  <canvas id="single-chart" width="350" height="250" class="chart"></canvas>

  <canvas id="pie-chart" width="350" height="250" class="chart"></canvas>

  <canvas id="male-chart" width="400" height="250" class="chart"></canvas>

  <canvas id="female-chart" width="400" height="250" class="chart"></canvas>

  <div id="year-control">

    <input type="range" id="year-slider" step="5" min="1995" max="2020" value="2020">

    <span id="year-label"></span>

    <div id="year-values"></div>

  </div>

</div>

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

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script>

(function(){

const GEO\_FILE='custom.geo.json';

const TOTAL\_FILE='total-alcohol-consumption-per-capita-litres-of-pure-alcohol.csv';

const GENDER\_FILE='alcohol-consumption-per-capita-men-women.csv';

const BEER\_FILE='beer-as-share-alcohol-consumption.csv';

const MIN\_YEAR=1995, MAX\_YEAR=2020;

const colors=['#1f77b4','#ff7f0e','#2ca02c','#d62728','#9467bd'];

const threshold=d3.scaleThreshold().domain([2,5,8,12]).range(colors);

const bins=[

  {range:[0,2],label:'<2'},

  {range:[2,5],label:'2–5'},

  {range:[5,8],label:'5–8'},

  {range:[8,12],label:'8–12'},

  {range:[12,Infinity],label:'>12'}

];

const keep5y = rows => rows.filter(r => r.Year >= MIN\_YEAR && r.Year <= MAX\_YEAR && (r.Year - MIN\_YEAR) % 5 === 0);

const svg=d3.select('svg.map'), g=svg.append('g');

const projection=d3.geoNaturalEarth1(), path=d3.geoPath(projection);

const tooltip=d3.select('.tooltip'), infoBox=d3.select('.info-box'), legendDiv=d3.select('.legend'), compareBtn=d3.select('#compare-btn');

const singleC=document.getElementById('single-chart'), pieC=document.getElementById('pie-chart'), maleC=document.getElementById('male-chart'), femC=document.getElementById('female-chart');

const yearCtl=document.getElementById('year-control'), slider=document.getElementById('year-slider'), yearLbl=document.getElementById('year-label'), yearVals=document.getElementById('year-values');

let initialTransform=d3.zoomIdentity, centered=null, activeBin=null;

let compareMode=false, compareSel=[];

let singleChart=null, pieChart=null, maleChart=null, femaleChart=null, genderMap=null, beerMap=null;

const zoom=d3.zoom().scaleExtent([1,8]).on('zoom',e=>g.attr('transform',e.transform));

svg.call(zoom);

compareBtn.on('click',()=>compareMode?exitCompare():enterCompare());

function enterCompare(){

  compareMode=true; compareSel=[]; compareBtn.classed('active',true);

  infoBox.style('display','block').text('Select 2 countries to compare');

  g.selectAll('path').classed('compareA',false).classed('compareB',false);

  hideAllCharts();

}

function exitCompare(){

  compareMode=false; compareSel=[]; compareBtn.classed('active',false);

  g.selectAll('path').classed('compareA',false).classed('compareB',false);

  infoBox.style('display','none').text('');

  hideCompareCharts();

}

function showSingle(code){

  let rows=genderMap.get(code)||[];

  rows=keep5y(rows.map(r=>({...r,Year:+r.Year}))).sort((a,b)=>a.Year-b.Year);

  const beerData=beerMap.get(code)||[];

  if(!rows.length && !beerData.length){hideSingle();return;}

  const mKey=Object.keys(rows[0] || {}).find(k=>k.includes('Male'));

  const fKey=Object.keys(rows[0] || {}).find(k=>k.includes('Female'));

  const years=rows.map(r=>r.Year);

  const males=rows.map(r=>+r[mKey]);

  const fems=rows.map(r=>+r[fKey]);

  singleChart&&singleChart.destroy();

  if(rows.length){

    singleChart=new Chart(singleC.getContext('2d'),{

      type:'line',

      data:{labels:years,datasets:[

        {label:'Men',data:males,borderWidth:2,fill:false,tension:0.3,pointRadius:Array(years.length).fill(0)},

        {label:'Women',data:fems,borderWidth:2,fill:false,tension:0.3,pointRadius:Array(years.length).fill(0)}

      ]},

      options:{responsive:false,plugins:{legend:{position:'top'},title:{display:true,text:'Consumption of pure alcohol'}},scales:{y:{beginAtZero:true}}}

    });

    singleC.style.display='block';

  }else{

    singleC.style.display='none';

  }

  pieChart&&pieChart.destroy();

  if(beerData.length){

    const beerValue=+beerData[0]["Alcohol, consumption of pure alcohol by type of beverage (%) - Beverage types: Beer"];

    if(!isNaN(beerValue)){

      pieChart=new Chart(pieC.getContext('2d'),{

        type:'pie',

        data:{

          labels:['Beer','Other Alcohol'],

          datasets:[{

            data:[beerValue,100-beerValue],

            backgroundColor:['#ff7f0e','#1f77b4'],

            borderWidth:1

          }]

        },

        options:{

          responsive:false,

          plugins:{

            legend:{position:'top'},

            title:{display:true,text:'Beer vs Other Alcohol'}

          }

        }

      });

      pieC.style.display='block';

    }else{

      pieC.style.display='none';

    }

  }else{

    pieC.style.display='none';

  }

  if(rows.length){

    if(!yearCtl.parentNode) infoBox.node().appendChild(yearCtl);

    slider.min=years[0]; slider.max=years.at(-1); slider.value=slider.max; yearCtl.style.display='block';

    slider.oninput=()=>updateYear(+slider.value); updateYear(+slider.value);

  }else{

    yearCtl.style.display='none';

  }

  function updateYear(val){

    yearLbl.textContent=val;

    const i=years.indexOf(val);

    if(i>-1){

      yearVals.textContent=`M: ${males[i].toFixed(2)} L | Ž: ${fems[i].toFixed(2)} L`;

      singleChart.data.datasets.forEach(ds=>ds.pointRadius=ds.data.map((\_,j)=>j===i?4:0));

    }else{

      yearVals.textContent='no data';

      singleChart.data.datasets.forEach(ds=>ds.pointRadius=Array(ds.data.length).fill(0));

    }

    singleChart&&singleChart.update('none');

  }

}

function hideSingle(){singleC.style.display='none'; pieC.style.display='none'; yearCtl.style.display='none';}

function hideCompareCharts(){maleC.style.display='none'; femC.style.display='none';}

function hideAllCharts(){hideSingle(); hideCompareCharts();}

Promise.all([

  d3.json(GEO\_FILE),

  d3.csv(TOTAL\_FILE,d=>({code:d.Code,year:+d.Year,value:+d["Total alcohol consumption per capita (liters of pure alcohol, projected estimates, 15+ years of age)"]})),

  d3.csv(GENDER\_FILE),

  d3.csv(BEER\_FILE)

]).then(([geo,total,gRows,beerRows])=>{

  const totalMap=new Map();

  total.forEach(r=>{

    const c=totalMap.get(r.code);

    if(!c||r.year>c.year)totalMap.set(r.code,{value:r.value,year:r.year});

  });

  genderMap=d3.group(gRows,d=>d.Code);

  beerMap=d3.group(beerRows,d=>d.Code);

  projection.fitSize([svg.node().clientWidth,svg.node().clientHeight],geo);

  const paths=g.selectAll('path').data(geo.features).enter().append('path')

    .attr('d',path)

    .attr('fill',d=>{

      const rec=totalMap.get(d.properties.iso\_a3||d.id);

      return rec?threshold(rec.value):'#eee';

    })

    .on('mouseover',(e,d)=>tooltip.text(d.properties.name).style('left',(e.pageX+8)+'px').style('top',(e.pageY-20)+'px').style('opacity',0.9))

    .on('mousemove',e=>tooltip.style('left',(e.pageX+8)+'px').style('top',(e.pageY-20)+'px'))

    .on('mouseout',()=>tooltip.style('opacity',0))

    .on('click',(e,d)=>{e.stopPropagation();compareMode?pickCompare(d):focusCountry(d);});

  legendDiv.selectAll('.legend-item').data(bins).enter().append('div').attr('class','legend-item')

    .each(function(bin){

      const div=d3.select(this);

      div.append('div').attr('class','legend-swatch')

        .style('background',()=>{const x=bin.range[1]===Infinity?bin.range[0]+0.1:(bin.range[0]+bin.range[1])/2;return threshold(x);});

      div.append('div').text(bin.label);

    })

    .on('click',function(\_,bin){

      if(activeBin===bin){activeBin=null;legendDiv.selectAll('.legend-item').classed('active',false);paths.transition().style('opacity',1);return;}

      activeBin=bin;legendDiv.selectAll('.legend-item').classed('active',d=>d===bin);

      paths.transition().style('opacity',d=>{

        const v=(totalMap.get(d.properties.iso\_a3||d.id)||{}).value;

        return v>=bin.range[0]&&v<bin.range[1]?1:0.15;

      });

    });

  d3.select('.legend').append('div').attr('class', 'legend-label').text('Consumption (L)');

  initialTransform=d3.zoomTransform(svg.node());

  svg.on('click',()=>{

    hideAllCharts();

    infoBox.style('display','none').text('');

    paths.classed('active',false).classed('compareA',false).classed('compareB',false);

    centered=null;

    if(compareMode) exitCompare();

    svg.transition().duration(750).call(zoom.transform,initialTransform);

  });

  function focusCountry(d){

    const code=d.properties.iso\_a3||d.id;

    const rec=totalMap.get(code);

    infoBox.style('display','block').text(rec?`${d.properties.name}: ${rec.value.toFixed(2)} L (${rec.year})`:`${d.properties.name}: no data`);

    centered=d; paths.classed('active',p=>p===d);

    const [[x0,y0],[x1,y1]]=path.bounds(d), {clientWidth:W,clientHeight:H}=svg.node();

    const k=Math.min(8,0.9/Math.max((x1-x0)/W,(y1-y0)/H));

    svg.transition().duration(750)

      .call(zoom.transform,d3.zoomIdentity.translate(W/2,H/2).scale(k).translate(-(x0+x1)/2,-(y0+y1)/2));

    showSingle(code);

  }

  function pickCompare(d){

    if(compareSel.includes(d))return;

    compareSel.push(d);

    if(compareSel.length===1){

      paths.classed('compareA',p=>p===d);

    }else{

      paths.classed('compareB',p=>p===d);

      const [a,b]=compareSel;

      showCompare({code:a.properties.iso\_a3||a.id,name:a.properties.name},{code:b.properties.iso\_a3||b.id,name:b.properties.name});

      compareMode=false; compareBtn.classed('active',false);

    }

  }

  function showCompare(a, b) {

    const rowsA = keep5y(genderMap.get(a.code) || []).map(r => ({ ...r, Year: +r.Year }));

    const rowsB = keep5y(genderMap.get(b.code) || []).map(r => ({ ...r, Year: +r.Year }));

    const mKeyA = Object.keys(rowsA[0] || {}).find(k => k.includes('Male'));

    const fKeyA = Object.keys(rowsA[0] || {}).find(k => k.includes('Female'));

    const mKeyB = Object.keys(rowsB[0] || {}).find(k => k.includes('Male'));

    const fKeyB = Object.keys(rowsB[0] || {}).find(k => k.includes('Female'));

    function getValidAvgForYear(rows, mKey, fKey, year) {

      const preferred = rows.find(r => r.Year === year);

      if (preferred) {

        const m = parseFloat(preferred[mKey]);

        const f = parseFloat(preferred[fKey]);

        if (!isNaN(m) && !isNaN(f)) return { value: ((m + f) / 2).toFixed(2), year };

      }

      const fallback = [...rows].reverse().find(r => r.Year < year && !isNaN(parseFloat(r[mKey])) && !isNaN(parseFloat(r[fKey])));

      if (fallback) {

        const m = parseFloat(fallback[mKey]);

        const f = parseFloat(fallback[fKey]);

        return { value: ((m + f) / 2).toFixed(2), year: fallback.Year };

      }

      return null;

    }

    const aData = getValidAvgForYear(rowsA, mKeyA, fKeyA, 2020);

    const bData = getValidAvgForYear(rowsB, mKeyB, fKeyB, 2020);

    const aDisplay = aData ? `${aData.value} L (${aData.year})` : 'no data';

    const bDisplay = bData ? `${bData.value} L (${bData.year})` : 'no data';

    infoBox.html(`

      <strong>${a.name}</strong>: ${aDisplay}<br>

      <strong>${b.name}</strong>: ${bDisplay}<br>

      <em>${a.name}</em> vs <em>${b.name}</em>

    `).style('display', 'block');

    const yearsSet = new Set([...rowsA.map(r => r.Year), ...rowsB.map(r => r.Year)]);

    const years = Array.from(yearsSet).filter(year => (year - MIN\_YEAR) % 5 === 0).sort((a, b) => a - b);

    if (!years.length) {

      hideCompareCharts();

      return;

    }

    const aM = years.map(y => {

      const row = rowsA.find(r => r.Year === y);

      return row ? +row[mKeyA] : null;

    });

    const bM = years.map(y => {

      const row = rowsB.find(r => r.Year === y);

      return row ? +row[mKeyB] : null;

    });

    const aF = years.map(y => {

      const row = rowsA.find(r => r.Year === y);

      return row ? +row[fKeyA] : null;

    });

    const bF = years.map(y => {

      const row = rowsB.find(r => r.Year === y);

      return row ? +row[fKeyB] : null;

    });

    maleChart && maleChart.destroy();

    maleChart = new Chart(maleC.getContext('2d'), {

      type: 'line',

      data: {

        labels: years,

        datasets: [

          { label: `${a.name}`, data: aM, borderColor: '#1f77b4', borderWidth: 2, fill: false, tension: .3 },

          { label: `${b.name}`, data: bM, borderColor: '#d62728', borderWidth: 2, fill: false, tension: .3 }

        ]

      },

      options: {

        responsive: false,

        plugins: {

          legend: { position: 'top' },

          title: { display: true, text: 'Men consumption (L)' }

        },

        scales: { y: { beginAtZero: true } }

      }

    });

    femaleChart && femaleChart.destroy();

    femaleChart = new Chart(femC.getContext('2d'), {

      type: 'line',

      data: {

        labels: years,

        datasets: [

          { label: `${a.name}`, data: aF, borderColor: '#1f77b4', borderWidth: 2, fill: false, tension: .3 },

          { label: `${b.name}`, data: bF, borderColor: '#d62728', borderWidth: 2, fill: false, tension: .3 }

        ]

      },

      options: {

        responsive: false,

        plugins: {

          legend: { position: 'top' },

          title: { display: true, text: 'Women consumption (L)' }

        },

        scales: { y: { beginAtZero: true } }

      }

    });

    maleC.style.display = 'block';

    femC.style.display = 'block';

  }

}

).catch(err=>alert('Data load error: '+err));

})();

</script>

</body>

</html>