

Vizualizacija podataka

LV2

FORMATIRANJE PODATAKA, JSON, JEDNOSTAVNE VIZUALIZACIJE

Student: Kristijan Koščak

Smjer: DRC, 1.godina

Datum: 02.04.2020.

Zadatak 1:

Budući da DataWrangler više nije aktivan ,a novi komercijalni program mi traži registraciju sa službenim mailom kojemu nemam pristup , podatke sam kopirao ručno i zalijepio u alat za pretvaranje CSV tablice u JSON format.

#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	LEVEL1	LEVEL2	LEVEL3	LEVEL4	TOOLTIP	SOURCE	SOURCE U	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	LEVEL	
2	PRIHODI	F Prihodi	oc Porez	i pri	Porez	i pri	Grad Bjelc	Ministars: http://www	3E+07	3.3E+07	3.5E+07	3.5E+07	3.9E+07	6.4E+07	7E+07	6.8E+07	5.9E+07	5.8E+07	6.3E+07	6.3E+07	6.1E+07	5.5E+07	5.8E+07	5.4E+07	7.1E+07	4
3	PRIHODI	F Prihodi	oc Porez	i pri	Porez	i pri	Grad Bjelc	Ministars: http://www	0	0	0	0	0	0	0	0	0	0	0	0	0	203591	612227	561015	9777	4
4	PRIHODI	F Prihodi	oc Porez	i pri	Dio porez	Grad Bjelc	Ministars: http://www	2662551	4902628	8042120	8904507	9368088	8704904	9420482	9842747	1.1E+07	0	0	0	0	0	0	0	0	0	4
5	PRIHODI	F Prihodi	oc UKUPNO: Porez	i pri	Grad Bjelc	Ministars: http://www	3.3E+07	3.8E+07	4.3E+07	4.4E+07	4.9E+07	7.3E+07	7.9E+07	7.8E+07	7E+07	5.8E+07	6.3E+07	6.3E+07	6.1E+07	5.5E+07	5.8E+07	5.5E+07	7.1E+07	3		
6	PRIHODI	F Prihodi	oc Porez	na c	Porez	na c	Grad Bjelc	Ministars: http://www	2774342	2803976	3652502	3821147	6800066	6276	0	0	0	0	0	0	0	0	0	0	0	4
7	PRIHODI	F Prihodi	oc UKUPNO: Porez	na d	Grad Bjelc	Ministars: http://www	2774342	2803976	3652502	3821147	6800066	6276	0	0	0	0	0	0	0	0	0	0	0	0	0	3
8	PRIHODI	F Prihodi	oc Porez	na	Stalni	pori	Grad Bjelc	Ministars: http://www	403658	435165	609927	441779	930407	989313	1115434	1923138	1132446	842892	1012510	754856	713181	1079055	1074453	923152	830108	4
9	PRIHODI	F Prihodi	oc Porez	na	Povremen	Grad Bjelc	Ministars: http://www	3415890	4233730	2751859	2403455	3707561	3667628	5409419	2951326	3054067	1719985	2271462	2431993	1449530	2206427	4832977	2981640	4802591	4	
10	PRIHODI	F Prihodi	oc UKUPNO: Porez	na i	Grad Bjelc	Ministars: http://www	3819548	4668895	3361786	2845234	4637968	4656941	6524853	4874464	4186513	2562877	3283972	3186849	2162711	3285482	5907430	3904792	5632699	3		
11	PRIHODI	F Prihodi	oc Porez	na	Porez	na	Grad Bjelc	Ministars: http://www	568767	531227	556417	593299	635122	716580	700430	670036	639233	682048	717687	930255	716228	896938	912303	968794	1013301	4

Vrsta prihoda	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
Porez i prirez na dohodak od nesamostalnog rada	30409020	33442187	35279120	34880300	39304535	64079252	70060188	68246588	59021591	58081342	62548883	63166510	61277939	54964305	57505606	54340681	71288848
58081342	62548883	63166510	61277939	54964305	57505606	54340681	71288848										

Output as **JSON - Properties**

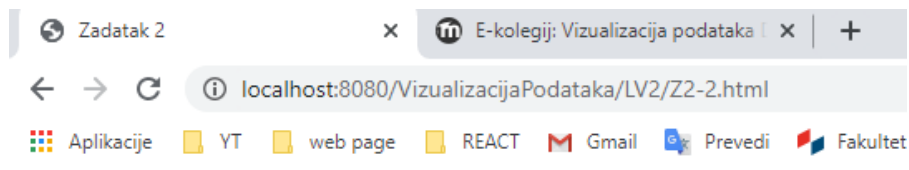
```
[{"Vrsta prihoda": "Porez i prirez na dohodak od nesamostalnog rada", "2002": 30409020, "2003": 33442187, "2004": 35279120, "2005": 34880300, "2006": 39304535, "2007": 64079252, "2008": 70060188, "2009": 68246588, "2010": 59021591, "2011": 58081342, "2012": 62548883, "2013": 63166510, "2014": 61277939, "2015": 54964305, "2016": 57505606, "2017": 54340681, "2018": 71288848}]
```

JSON

```
[{"Vrsta prihoda": "Porez i prirez na dohodak od nesamostalnog rada", "2002": 30409020, "2003": 33442187, "2004": 35279120, "2005": 34880300, "2006": 39304535, "2007": 64079252, "2008": 70060188, "2009": 68246588, "2010": 59021591, "2011": 58081342, "2012": 62548883, "2013": 63166510, "2014": 61277939, "2015": 54964305, "2016": 57505606, "2017": 54340681, "2018": 71288848}]
```

Radi jednostavnosti, izbačeni su nepotrebni stupci. Zadnji stupac predstavlja vrstu prihoda za pojedini redak. Prikazan je prihod od jednog retka odnosno od poreza i prireza na dohodak od nesamostalnog rada za pojedine godine te smo dobili JSON format kao što je navedeno iznad. Također, radi jednostavnosti smanjen je prikaz po broju godina kako bi bilo preglednije.

Zadatak 2:



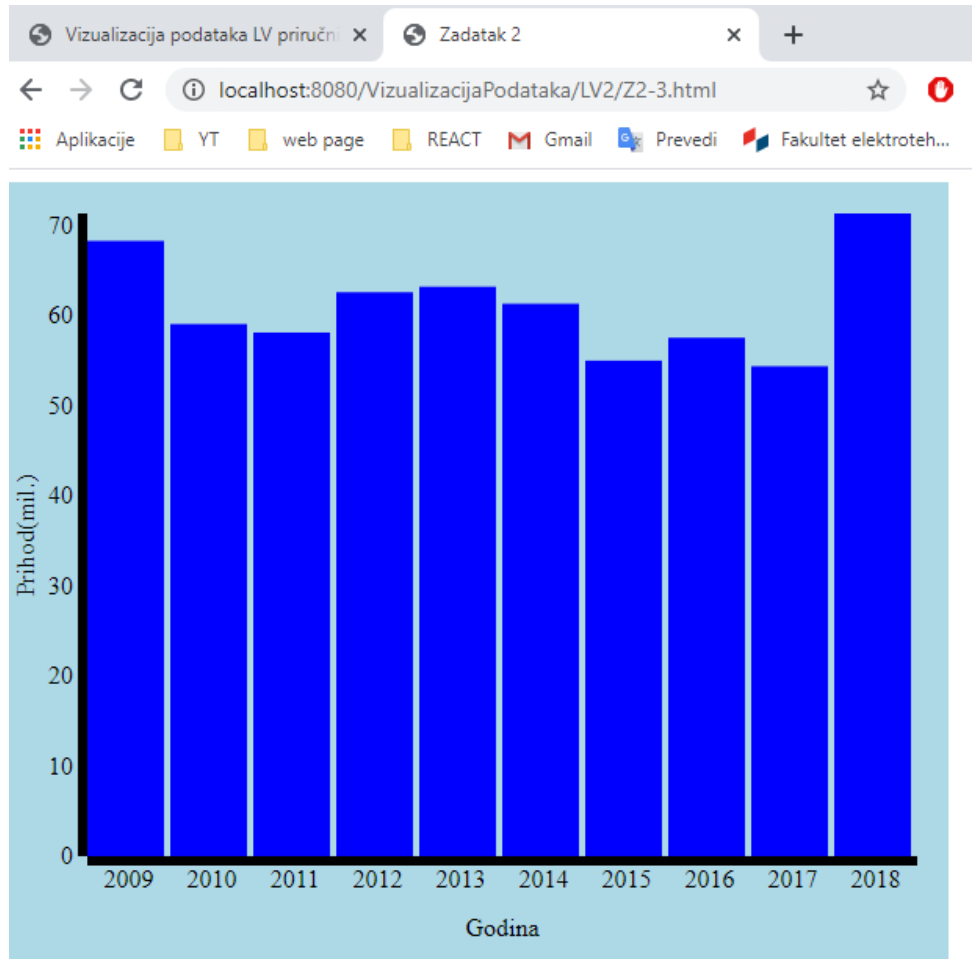
```
var data = []
  var jsonObjArray = [{
    "Vrsta prihoda": "Porez i prirez na dohodak od nesamostalnog rada", "
2009": 68246588, "2010": 59021591, "2011": 58081342,
    "2012": 62548883, "2013": 63166510, "2014": 61277939, "2015": 5496430
5, "2016": 57505606, "2017": 54340681, "2018": 71288848
  }];
  for(var js0 in jsonObjArray){
    for(var item in jsonObjArray[js0])
      if(typeof jsonObjArray[js0][item] == "number")
        data.push(jsonObjArray[js0][item]/1000000);
  }
  console.log(data);

//var data = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

var width = 500;
var height = 100;
var svg = d3.select("body")
  .append("svg")
  .attr("width", width)
  .attr("height", height);
var barchart = svg.selectAll("rect")
  .data(data)
  .enter()
  .append("rect")
  .attr("x", function (d, i) { return 50 * i; })
  .attr("y", function (d) { return height - d ; })
  .attr("width", 40)
  .attr("height", function (d) { return d ; })
  .attr("fill", "blue");
```

Kao što je i traženo u zadatku, napravljen je stupčasti graf s obrađenim vrijednostima u ranijem zadatku. Radi jednostavnosti prikaza, dijelili smo svaku vrijednost s 1 000 000 kako bi lakše prikazali svaki stupac jer su vrijednosti jako velike.

Zadatak 3:



```
var data = [];  
var years = [];  
var jsonObjArray = [{  
    "Vrsta prihoda": "Porez i prirez na dohodak od nesamostalnog rada", "  
2009": 68246588, "2010": 59021591, "2011": 58081342,  
    "2012": 62548883, "2013": 63166510, "2014": 61277939, "2015": 5496430  
5, "2016": 57505606, "2017": 54340681, "2018": 71288848  
}];  
for (var js0 in jsonObjArray) {  
    for (var item in jsonObjArray[js0]) {  
        if (typeof jsonObjArray[js0][item] == "number") {  
            data.push(jsonObjArray[js0][item] / 1000000);  
            years.push(item);  
        }  
    }  
}
```

```

    }
  }
}
console.log(years)
var margin = { top: 20, bottom: 70, left: 50, right: 20 };
var width = 600 - margin.left - margin.right;
var height = 500 - margin.top - margin.bottom;
var barPadding = 4;
var barWidth = width / data.length - barPadding;

var x = d3.scale.ordinal()
    .domain(d3.range(data.length))
    .rangeRoundBands([0, width]);
var y = d3.scale.linear()
    .domain([0, d3.max(data)])
    .range([height, 0]);

var svg = d3.select("body")
    .append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.bottom + margin.top)
    .style("background-color", "lightblue")
    .append("g")
    .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

var xAxis = d3.svg.axis()
    .scale(x)
    .orient("bottom")
    .tickFormat(function(d, i) { return years[i]; });
var yAxis = d3.svg.axis()
    .scale(y)
    .orient("left")
    .ticks(10);
svg.append("g")
    .attr("class", "x axis")
    .attr("transform", "translate(0," + height + ")")
    .call(xAxis)
    .selectAll("text")
    .style("text-anchor", "middle");
svg.append("g")
    .attr("class", "y axis")
    .call(yAxis)
    .append("text")
    .attr("transform", "rotate(-90)")

```

```

svg.append("text")
  .attr("transform", "rotate(-90)")
  .attr("x", 0 - (height / 2))
  .attr("y", 0 - margin.left)
  .attr("dy", "1em")
  .style("text-anchor", "middle")
  .text("Prihod(mil.)");
svg.append("text")
  .attr("x", (width / 2))
  .attr("y", (height + (margin.bottom / 2)))
  .attr("dy", "1em")
  .style("text-anchor", "middle")
  .text("Godina");
var barchart = svg.selectAll("rect")
  .data(data)
  .enter()
  .append("rect")
  .attr("x", function (d, i) { return x(i); })
  .attr("y", y).attr("height", function (d) { return height - y(d); })
  .attr("width", barWidth)
  .attr("fill", "blue");

```

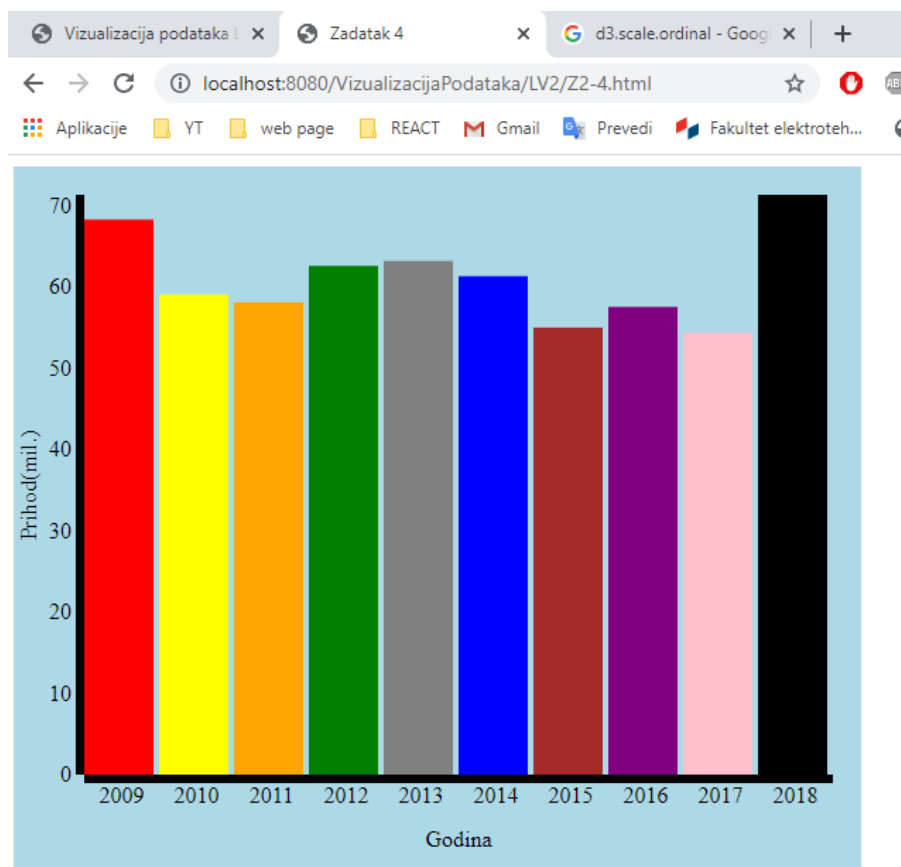
Za razliku od prethodnog zadatka ovdje smo koristili pojedine funkcionalnosti koje nudi D3. Koristili smo linearno skaliranje za y os i ordinalno skaliranje za x os. Dodali smo koordinatne osi kako bi omogućili čitanje vrijednosti koje graf prikazuje. Također, dodali smo i margine za čije vrijednosti se umanjuje veličina grafa te varijable koje predstavljaju širinu stupaca i razmak između njih.

Zadatak 4:

```

var color = d3.scale.ordinal()
  .domain(data)
  .range(["red", "yellow", "orange", "green", "lightblue", "blue", "brown", "purple", "pink", "black"]);
//console.log(typeof color);
var barchart = svg.selectAll("rect")
  .data(data)
  .enter()
  .append("rect")
  .attr("x", function (d, i) { return x(i); })
  .attr("y", y).attr("height", function (d) { return height - y(d); })
  .attr("width", barWidth)
  .attr("fill", color);

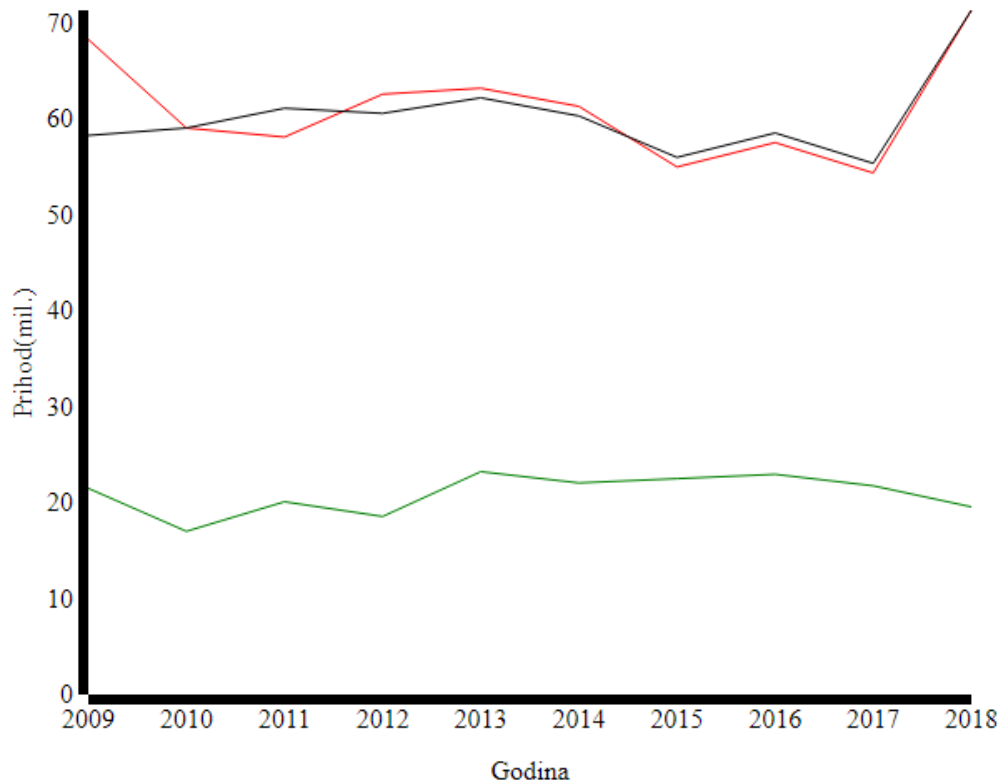
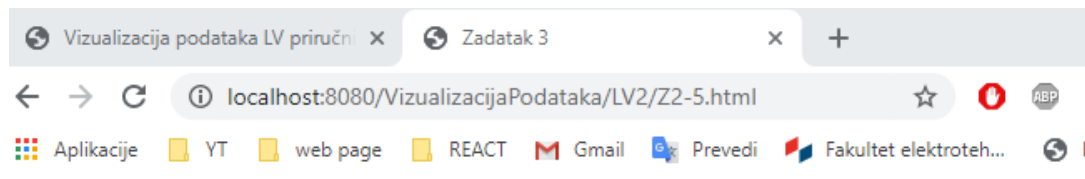
```



Ovaj zadatak je gotovo identičan prethodnom. Razlika je u par linija koda. Napravili smo ordinalnu skalu kao što je traženo te joj postavili domen na skup vrijednosti koje prikazujemo na grafu. Zatim smo raspon postavili na skup imena boja te u konačnici postavili atribut fill na određenu boju jer je color u našem slučaju funkcija, koja svakim pozivom vraća jednu vrijednost iz raspona odnosno ime boje.

Zadatak 5:

U ovome zadatku napravili smo linijski graf uz male preinake implementacije. Kod stupčastog grafa koristili smo `<rect>` element dok ovdje to nije slučaj već koristimo element `<path>`. Atribut `d` navedenog elementa postavili smo na skup točaka koje smo dobili metodom `svg.line` koja izračunava točke za dvije vrijednosti dok ne dođe do kraja zadanog skupa. Bitno je naglasiti potrebu za postavljanjem fill atributa (path elementa) na `none` jer u suprotnom dobivamo čudan prikaz grafa.



```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Zadatak 5</title>

  <script src="https://d3js.org/d3.v3.min.js"></script>
</head>

<body>
  <script>
    var data1 = [];
    var data2 = [];
    var data3 = [];
    var temp = [];
```



```

var years = [];
var jsonObjArray = [
    {
        "Vrsta prihoda": "Porez i prirez na dohodak od nesamostalnog rada",
        "2009": 68246588, "2010": 59021591, "2011": 58081342,
        "2012": 62548883, "2013": 63166510, "2014": 61277939, "2015": 549
        64305, "2016": 57505606, "2017": 54340681, "2018": 71288848
    },

    {
        "Vrsta prihoda": "Prihodi od poreza", "2009": 21480804, "2010": 1
        7002802, "2011": 20081342,
        "2012": 18548883, "2013": 23215167, "2014": 22046165, "2015": 224
        90152, "2016": 22941095, "2017": 21744225, "2018": 19552467
    },

    {
        "Vrsta prihoda": "Porez i prirez na dohodak", "2009": 58246588, "
        2010": 59021591, "2011": 61081342,
        "2012": 60548883, "2013": 62166510, "2014": 60277939, "2015": 559
        64305, "2016": 58505606, "2017": 55340681, "2018": 71288848
    }
];

for (var item in jsonObjArray[0]) {
    if (typeof jsonObjArray[0][item] == "number") {
        data1.push(jsonObjArray[0][item] / 1000000);
        years.push(item);
    }
}
for (var item in jsonObjArray[1]) {
    if (typeof jsonObjArray[0][item] == "number") {
        data2.push(jsonObjArray[1][item] / 1000000);
    }
}
for (var item in jsonObjArray[2]) {
    if (typeof jsonObjArray[0][item] == "number") {
        data3.push(jsonObjArray[2][item] / 1000000);
    }
}

var margin = { top: 20, bottom: 70, left: 50, right: 20 };
var width = 600 - margin.left - margin.right;
var height = 500 - margin.top - margin.bottom;
var barPadding = 4;
var barWidth = width / data1.length - barPadding;

```

```

var x = d3.scale.ordinal()
    .domain(d3.range(data1.length))
    .rangePoints([0, width]);
var y = d3.scale.linear()
    .domain([0, d3.max(data1)])
    .range([height, 0]);
var xAxis = d3.svg.axis()
    .scale(x)
    .orient("bottom")
    .tickFormat(function (d, i) { return years[i]; });
var yAxis = d3.svg.axis()
    .scale(y)
    .orient("left")
    .ticks(10);
var svg = d3.select("body")
    .append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.bottom + margin.top)
    .append("g")
    .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

svg.append("g")
    .attr("class", "x axis")
    .attr("transform", "translate(0," + height + ")")
    .call(xAxis)
    .selectAll("text")
    .style("text-anchor", "middle");
svg.append("g")
    .attr("class", "y axis")
    .call(yAxis)
    .append("text")
    .attr("transform", "rotate(-90)")

svg.append("text")
    .attr("transform", "rotate(-90)")
    .attr("x", 0 - (height / 2))
    .attr("y", 0 - margin.left)
    .attr("dy", "1em")
    .style("text-anchor", "middle")
    .text("Prihod(mil.)");
svg.append("text")
    .attr("x", (width / 2))
    .attr("y", (height + (margin.bottom / 2)))
    .attr("dy", "1em")

```

```

        .style("text-anchor", "middle")
        .text("Godina");
    var valueline = d3.svg.line()
        .interpolate("linear")
        .x(function (d, i) { return x(i); })
        .y(function (d) { return y(d); });
    var linechart = svg.append("path")
        .attr("class", "line")
        .attr("d", valueline(data1))
        .style("stroke", "red")
        .attr("fill", "none");
    var linechart = svg.append("path")
        .attr("class", "line")
        .attr("d", valueline(data2))
        .style("stroke", "green")
        .attr("fill", "none");
    var linechart = svg.append("path")
        .attr("class", "line")
        .attr("d", valueline(data3))
        .style("stroke", "black")
        .attr("fill", "none");

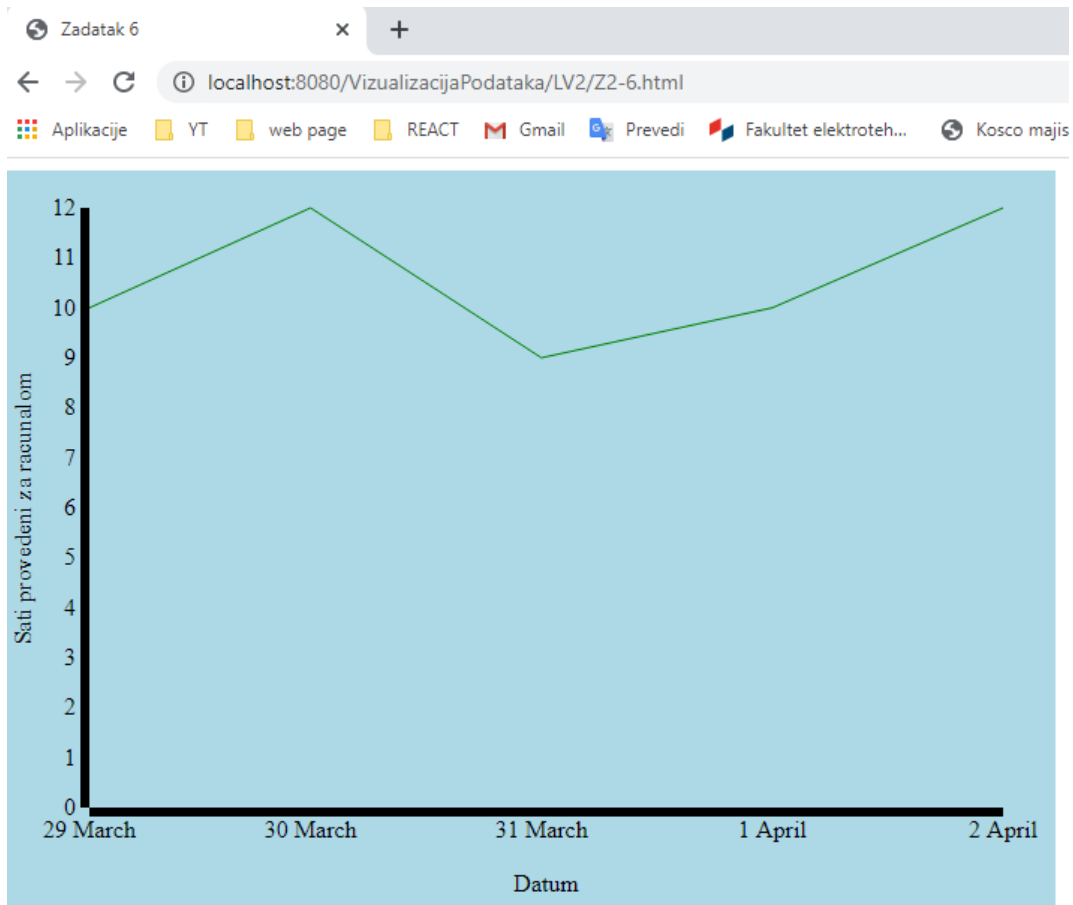
    </script>
</body>

</html>

```

Zadatak 6:

Ovaj zadatak smo riješili koristeći se internetom i danim uputama u predlošku. Problem je predstavljao datum jer je vraćao jedan mjesec više od traženog. Razlog je što u JS-u, Date mjesec gleda kao vrijednosti od 0 do 11, a ne od 1 do 12. Stoga, svaki mjesec smo umanjili za jedan kako bi dobili željeni. Ovaj zadatak nije u potpunosti različit od prethodnog, novitet je `time.scale` koji smo koristili za skaliranje, čiju domenu smo postavili na raspon od najmanjeg datuma do najvećeg u danom skupu datuma. Također, format kojim određujemo prikaz datuma je novitet s kojim se nismo susretali i njega koristimo na x osi. Većina ostatka je ista kao iz prethodnog zadatka koja se koristi za prikaz linijskog grafa.



```
var data = [  
  { timeSpentAtComp: new Date(2020, 2, 29), value: 10 },  
  { timeSpentAtComp: new Date(2020, 2, 30), value: 12 },  
  { timeSpentAtComp: new Date(2020, 2, 31), value: 9 },  
  { timeSpentAtComp: new Date(2020, 3, 1), value: 10 },  
  { timeSpentAtComp: new Date(2020, 3, 2), value: 12 }  
];  
  
var width = 600;  
var height = 400;  
  
var margin = { top: 20, bottom: 70, left: 50, right: 30 };  
var width = 700 - margin.left - margin.right;  
var height = 500 - margin.top - margin.bottom;  
var barPadding = 5;  
  
var svg = d3.select("body")  
  .append("svg")
```

```

        .attr("width", width + margin.left + margin.right)
        .attr("height", height + margin.bottom + margin.top)
        .style("background-color", "lightblue")
        .append("g")
        .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

var minDate = d3.min(data, function (d) { return d.timeSpentAtComp; });
var maxDate = d3.max(data, function (d) { return d.timeSpentAtComp; });

var xScale = d3.time.scale()
    .domain([minDate, maxDate])
    .range([barPadding, width - barPadding]);

var yScale = d3.scale.linear()
    .domain([0, d3.max(data, function (d) { return d.value; })])
    .range([height - barPadding, barPadding]);

var format = d3.time.format("%e %B");

var xAxis = d3.svg.axis()
    .scale(xScale)
    .orient("bottom")
    .tickFormat(format)
    .ticks(3);

svg.append("g")
    .attr("class", "axis x-axis")
    .attr("transform", "translate(0," + (height - barPadding) + ")")
    .call(xAxis);

var yAxis = d3.svg.axis()
    .scale(yScale)
    .orient("left")
    .tickFormat(function (d) { return d; })
    .ticks(10);

svg.append("g")
    .attr("class", "axis y-axis")
    .attr("transform", "translate(" + barPadding + ",0)")
    .call(yAxis);

var valueline = d3.svg.line()
    .x(function (d) {
        return xScale(d.timeSpentAtComp);
    });

```

```
    })
    .y(function (d) {
        return yScale(d.value);
    });

var linechart = svg.append("path")
    .attr("class", "line")
    .attr("d", valueline(data))
    .style("stroke", "green")
    .attr("fill", "none");

svg.append("text")
    .attr("transform", "rotate(-90)")
    .attr("x", 0 - (height / 2))
    .attr("y", 0 - margin.left)
    .attr("dy", "1em")
    .style("text-anchor", "middle")
    .text("Sati provedeni za racunalom");
svg.append("text")
    .attr("x", (width / 2))
    .attr("y", (height + (margin.bottom / 2)))
    .attr("dy", "1em")
    .style("text-anchor", "middle")
    .text("Datum");
```