Korištenje JavaScript-a kao tehnika vizualizacije podataka

Ungar, Borna

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Humanities and Social Sciences / Sveučilište Josipa Jurja Strossmayera u Osijeku, Filozofski fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:142:172091

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

FILOZOFSKI FAKULTET
SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

Repository / Repozitorij:

FFOS-repository - Repository of the Faculty of Humanities and Social Sciences Osijek

zir.nsk.hr

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

Sveučilište J.J. Strossmayera u Osijeku

Filozofski fakultet

Odsjek za informacijske znanosti

Preddiplomski studij informatologije

Borna Ungar

Korištenje JavaScript-a kao tehnika vizualizacije podataka

Završni rad

Mentor: izv.prof.dr.sc. Boris Badurina
Komentor: dr.sc. Tomislav Jakopec, poslijedoktorand

Osijek, rujan 2018.

Sveučilište J.J. Strossmayera u Osijeku

Filozofski fakultet

Odsjek za informacijske znanosti

Preddiplomski studij informatologije

Borna Ungar

Korištenje JavaScript-a kao tehnika vizualizacije podataka

Završni rad

Društvene znanosti, informacijske i komunikacijske znanosti,
informacijski sustavi i informatologija

Mentor: izv.prof.dr.sc. Boris Badurina
Komentor: dr.sc. Tomislav Jakopec, poslijedoktorand

Osijek, rujan 2018.

Sažetak

Rad se bavi opisom izrade primjera programskih rješenja za vizualizaciju podataka na mrežnim stranicama, usredotočujući se prvenstveno na izradu grafikona kao osnovnim alatom za prikaz i analizu različitih skupova podataka. U početnom dijelu rada objašnjava se koji će alati biti primijenjeni, programski jezici i instalacija, odnosno sve nužno za pravilnu primjenu alata i uspješno prikazivanje sadržaja na mrežnom pregledniku. Daju se upute kako podići vlastiti lokalni server, objašnjavaju se elementi grafikona, nizovi i objekti. Navode se nekoliko poznatijih alata za vizualizaciju podataka i njihove osnovne karakteristike, a u samom radu se koriste dvije JavaScript biblioteke, Highcharts i D3.js (Data-driven documents). Dok je Highcharts fokusiran na grafikone, D3.js ima širi raspon primjene, ali je i teži za svladati. Rad se posebno posvećuje prvim koracima u korištenju JavaScript biblioteka Highcharts i D3 i načinom na koji se mogu povezati s vanjskim izvora koje sadrže podatke od kojih se želi izraditi grafikon. Rad je moguće pratiti i razumijeti bez predznanja o vizualizaciji podataka, ali je poželjno osnovno znanje o primjeni, ulogama i sintaksi HTML, CSS i ponajviše JavaScripta.

Ključne riječi: vizualizacija podataka, JavaScript, Data-driven documents, Highcharts

Sadržaj

1. Uvod 1
2. Instalacija i uvjeti 2
2.1. Elementi grafikona 4
2.2. SVG i DOM 5
2.3. Nizovi, objekti i JSON 6
2.4. Alati za vizualizaciju podataka 8
3. Highcharts 9
4. D3.js 16
5. Zaključak 20
Literatura 21
Prilozi 23

1. Uvod

Kada je riječ o podacima, živimo u vremenu u kojemu je svaki aspekt života okružen podacima. Reći da je stoljeće u kojemu se nalazimo informacijski ispunjen nije dostatno. Štoviše, može se reći da se dogodila informacijska eksplozija i da je postalo teško ovladati svim podacima koje nam pristižu na dnevnoj bazi. Pretrpani kako potrebnim tako i nepotrebnim podacima i informacijama, sada je više nego ikada bitno biti informacijski pismen. Kod velikih količina podataka, na primjer rezultata znanstvenih istraživanja, moguće je da zbog povećeg broja sličnih podataka korisnik izgubi zainteresiranost za tim podacima, izgubi se u njima ili mu promaknu važni detalji. Postoje razni načini kojima se podaci mogu praktičnije predstaviti i pojednostaviti. Jedan od načina na koji se ti podaci mogu pojednostaviti i učiniti pristupačnijim je njihova vizualizacija. Vizualizacija podataka nije ništa novo i revolucionarno, prisutna je u određenim oblicima stotinama, možda i tisućama godina. Ono što je novo je suvremeni opći interes za polje koje je posljednjih dvadesetak godina doživjelo brzi rast, dijelom zbog novih tehnoloških mogućnosti i kulturnog pomaka prema većoj transparentnosti i dostupnosti podataka.[1] Vizualno prikazivanje podataka kao na primjer pravljenje grafikona u kojemu podatke predstavljaju simboli olakšava njihovo organiziranje, proučavanje, uspoređivanje te donošenje zaključaka. Uz to, mapiranje informacija u vizualni format pomaže bržem informiranju mozga i učinkovitije je od čitanja tekstualnih linija informacija. Poznata je izreka: slika govori tisuću riječi.[2] Način na koji gledamo podatke ima veliki utjecajkako ga možemo razumjeti.

Rad se bavi tematikom vizualizacije podataka pomoću skriptnog programskog jezika JavaScript i koristeći JavaScript libraries, tj. različite JavaScript biblioteke, prvenstveno prikazom pravljenja grafikona. Određene vrste grafikona su prikladnije za prikazivanje određenog skupa podataka od drugih vrsta grafikona i rad će pokriti od jednostavnih grafikona kao što su linijski i kružni grafikoni do složenijih s interaktivnošću koje proširuje mogućnosti grafikona i onoga što nudi na mrežnoj stranici. Osim grafikona, vizualizirat će se i drukčiji skupovi podataka, na primjer vizualni, interaktivni prikaz skupa realnih brojeva. Važno je spomenuti koje će se biblioteke

[1] Usp. Kirk, Andy. Data Visualization: A Successful Design Process. URL: https://books.google.hr/books?id=I4qBVLfD3t4C&printsec=frontcover#v=onepage&q&f=false (2018-08-17)
[2] Usp. Finch, L. Jannete, Flenner R. Angela. Using Data Visualization to Examine an Academic Library Collection. // College & Reasearch Libraries 77, 6(2016), str. 765. URL: https://crl.acrl.org/index.php/crl/article/view/16555/18001 (2018-08-17)

koristiti u radu za vizualizaciju podataka – Highcharts i D3. Iako se na obje JavaScript bibliotekegleda kao na alate za vizualizaciju i imaju slična svojstva (koriste JavaScript, vizualizacija podataka, izgrađeno pomoću SVG i CSS), one imaju svoje različitostitako da odabir biblioteke ovisi o potrebama i preferencijama korisnika. Highcharts, kako samo ime spominje, omogućuje implementaciju interaktivnih i dinamičnih grafikona, dok D3 (Data-driven documents) može vizualizirati širi raspon skupova podataka. Ukratko, Highcharts se koristi za brzo i lako pravljenje tradicionalnih grafikona, a D3 za pravljenje kreativnijih, jedinstvenih grafikona te manipulaciju DOM-a (Document Object Model), tj. dodjeljivanje podataka DOM-u, a zatim njegova transformacija. Važno je napomenuti da se Highcharts plaća ako se planira koristiti u komercijalne svrhe što zasigurno utječe na odabir biblioteke, ali njegova popularnost i primjena u najvećim tvrtkama svijeta ukazuje na njegovu kvalitetu.[3]

U svrhe rada, koristit će se mrežni preglednik Google Chrome, ali ne mora nužno biti Chrome jer svi najzastupljeniji preglednici podržavaju JavaScript. Također, koristit će se D3.js i Highcharts.js biblioteke te Atom za uređivanje teksta i koda. Podaci koji će se koristiti za određene grafikone su prezentacijske prirode, a za ostale će se koristiti skupovi podataka preuzeti s Portala otvorenih podataka Republike Hrvatske.[4] Svi korišteni prilozi mogu se pronaći na Githubu.[5]

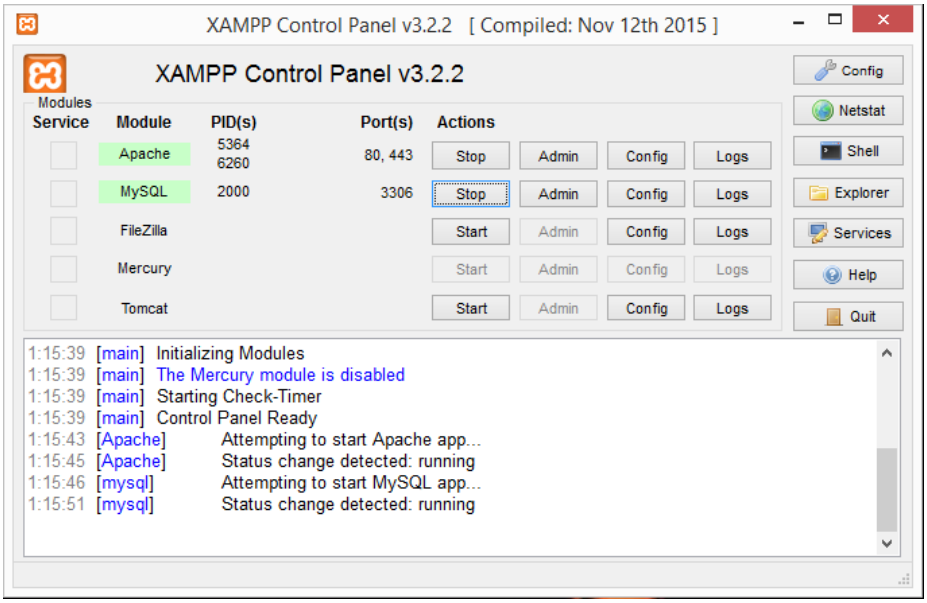
2. Instalacija i uvjeti

Highcharts i D3 se mogu preuzeti s njihovih službenih stranica.[6][7] Kod korištenja skinutih mapa, važno je obratiti pažnju gdje se kreiraju datoteke kako bi se tijekom pisanja programa pravilno pozivale njihove lokacije. U oba slučaja Highchartsa i D3-a instalacija datoteka na računalu nije obvezna. S jednom linijom koda može se povezati na najnovije inačice njihovih biblioteka. Linije se u praksi se stavljaju unutar *<head> </head>* ili *<body> </body>* prije

[3] https://www.highcharts.com/use-cases/ (2018-08-23)
[4] https://data.gov.hr/ (2018-08-20)
[5] https://github.com/Bungar95/datavisualization (2018-8-27)
[6] https://www.highcharts.com/products/highcharts/ (2018-08-20)
[7] https://d3js.org/ (2018-08-20)

JavaScripta.Za Highcharts je *<script src="http://code.highcharts.com/highcharts.js"></script>*, a za D3 *<script src=https://d3js.org/d3.v5.min.js></script>*.

Druga bitna stvar koju je potrebno naglasiti prije početka rada s primjerima koji učitavaju podatke s drugih datoteka je da preglednici imaju stroge sigurnosne mjere koje sprječavaju čitanje datoteka iz lokalnog datotečnog sustava. Drugim riječima, da bi se Highcharts ili D3 biblioteka uspješno povezala s nekom datotekom ekstenzije .csv, .json i sl. koje sadrže podatke, potreban je mrežni poslužitelj. XAMPP je instalacijski paket koji sadrži ne samo Apache HTTP Server, najkorišteniji softver za mrežne poslužitelje na svijetu, nego i MySQL te PHP što otvara mogućnosti korištenja Highchartsa ili D3 za vizualizaciju podataka direktno iz baza podataka, tj, php datoteka. Tijekom instalacije treba samo pratiti korake, za korisnike koji se ne razumiju u programe koje XAMPP nudi najbolje ništa ne mijenjati i instalirati kako sam program nudi.[8] Nakon instalacije treba otvoriti XAMPP Control Panel i pokrenuti Apache klikom na Start.

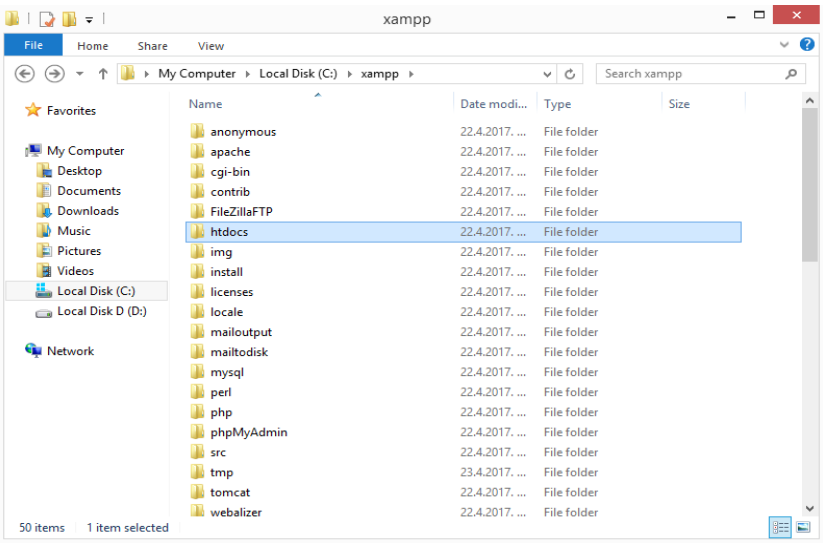


Slika 1. XAMPP Control Panel s upaljenim Apache i MySQL

Za MySQL, driveri su uključeni u instalaciju jer su ugrađeni u sve novije verzije PHP-a i nije ih potrebno tražiti. Ukoliko se koristi neki manje poznati sustav za upravljanje bazama podataka, moguće je da se driveri neće odmah nalaziti unutar PHP instalacije i trebat će ih se samostalno pronaći na internetu, instalirati i omogućiti njihovo korištenje u konfiguracijama, no time se rad

[8]XAMPP Installers and Downloads for Apache Friends. URL: https://www.apachefriends.org/index.html(2018-08-09)

neće baviti. Također, sve datoteke koje će se praviti se moraju nalaziti u specifičnoj mapi XAMPP-a zvanoj htdocs za lakše lociranje i pozivanje tijekom pisanja programa. (vidi Slika 2.)



Slika 2. htdocs mapa u kojoj se spremaju datoteke

2.1. Elementi grafikona

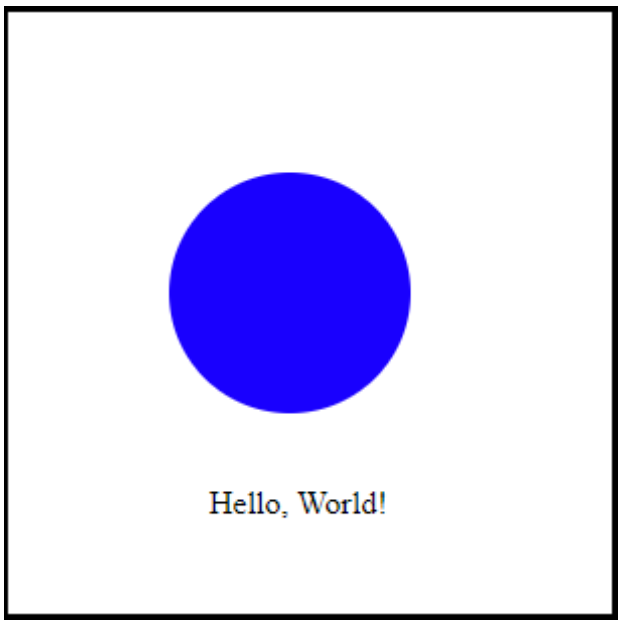
Ukratko pojašnjenje o najbitnijm elementima grafikona s obzirom na to da su ključan dio rada. Grafikoni često imaju dvije okomite linije, osi. Horizontalna linija najčešće predstavlja x os, a vertikalna y os. Svaka os predstavlja skalu koja je numerički ili kategorički podijeljena na segmente koji odgovaraju određenom rasponu vrijednosti. Osi omogućuju korisniku da se referira na vrijednosti koordinata x i y za svaku koordinatnu točku P(x, y), gdje svaka koordinatna točka sadrži podatak. Jednostavno rečeno, svako sjecište osi x i y predstavlja točku koja sadrži podatak. Kako bi se različite serije podataka prikazale u istom grafikonu, kao najčešći pristup koriste se različite boje, a zatim se dodaje tablica koja pokazuje korespondenciju između boja i skupina podataka - legenda.[9] Treba još spomenuti naslov i podnaslov grafikona te tooltip koji opisuje vrijednosti određenog dijela grafikona kada se mišem pređe iznad točke u seriji.

[9] Usp. Nelli, Fabio. Beginning JavaScript Charts : With jqPlot, d3, and Highcharts. CA, USA: Apress, 2013. str. 1-4. URL: https://books.google.hr/books?id=BZ4QAwAAQBAJ&lpg=PP3&dq=highcharts&lr&pg=PP1#v=onepage&q&f=false (2018-8-11)

Prema zadanim postavkama, tooltip prikazuje vrijednosti točke i naziv serije, ali ima i drugih opcija.[10]

2.2. SVG i DOM

SVG, Scalable Vector Graphics, je format zapisa koristeći XML jezik za prikazivanje dvodimenzionalne vektorske grafike. SVG sadržaj može biti statičan, dinamičan, interaktivan ili animiran, što ga čini vrlo fleksibilnim. Elementi SVG-a se mogu oblikovati pomoću CSS-a i dodati im dinamično ponašanje koristeći metode koje pruži SVG-ov DOM. Odabirom takvog formata može se dobiti visoko interaktivna web aplikacija sa skriptiranjem, naprednom animacijom, događajima i što god drugo korisnik poželi.[11] DOM bi se mogao prevesti kao objektni model dokumenta, model pomoću kojeg se definiraju strukture HTML i XML dokumenata. Jednostavan primjer SVG datoteke učitanog u HTML i stiliziran iz CSS datoteke nalazi se u prilogu. (vidi Prilog 1. i Slika 3.)



Slika 3. Krug i tekst napravljeni u SVG datoteci, prikaz iz HTML datoteke

[10] Usp. https://www.highcharts.com/docs/chart-concepts/understanding-highcharts (2018-8-13)
[11] Usp. Nelli, Fabio. Nav. dj., str. 7.

2.3. Nizovi, objekti i JSON

Arrays su nizovi vrijednosti i jedan od najjednostavnijih i najčešćih struktura podataka koji se koriste u JavaScriptu. Odvajaju se zarezom i zatvaraju kvadratnim zagradama [ ]. Može sadržavati bilo kakav tip podatka. U primjeru navedenom dolje, prva vrijednost je integer, druga vrijednost string, kao i treća jer se nalazi u navodnicima stoga predstavlja ne broj tri već tri kao simbol.

```
var array = [ 1, "dva", "3" ];
```

Objekti su složeniji od nizova po tome što se sastoje od svojstava kojima se dodjeljuju vrijednosti. Objekt se zatvara vitičastom zagradom { }, svojstva se definiraju imenom, dvotočkom pa njihovom vrijednošću, a svojstva se odvajaju zarezom.

```
var dan = {
     danas: "nedjelja",
     temperatura: 34,
     padalina: false
};
```

Kombinacijom nizova i objekata dobivaju se kompleksne strukture podataka pomoću kojih se mogu praviti grafikoni.

```
var vikend = [
     {
          dan: "petak",
          temperatura: 28,
          padalina: true
     }, {
          dan: "subota",
          temperatura: 33,
          padalina: false
     }, {
          dan: "nedjelja",
          temperatura: 34,
          padalina: false
     }
];
```

JavaScript Object Notation ili JSON je format datoteke, tj. sintaksa za organizaciju podataka u obliku JavaScript objekata. Koristi ljudski čitljiv tekst i predstavlja alternativu za XML.“ Serijalizacija objekata je proces konvertiranja stanja objekta u niz znakova iz kojega se on kasnije može i vratiti. Dvije funkcije [u JavaScriptu] koje će napraviti serijalizaciju i vraćanje u objekt su JSON.stringify() i JSON.parse(). Serijalizacijom će se objekt prebaciti u format pod nazivom JSON i njegova sintaksa je gotovo identična sintaksi objekata i nizova, ali ipak JSON sintaksa je tek dio JavaScript sintakse tako da to nisu identične stvari. JSON je vrlo bitan u komunikaciji klijenta i poslužitelja, jer se radi o tipičnom formatu za razmjenu podataka. ”[12] Njegova struktura slijedi pravila nizova i objekata definiranih u JavaScriptu te radi dobro s D3 i Highchartsom.[13]

```
var osoba = {
     “ime” : “Borna”,
     “prezime” : “Ungar”,
     “godina” : 23,
     “telBroj”: [
          {
               “tip”: “kucni”,
               “broj”: “031-123-456”
          }, {
               “tip”: “mobilni”,
               “broj”: “091-123-4567”
          }
     ],
     “studentStatus”: true
};
```

[12]Matanović, Antun. Razvoj programskog rješenja za mapiranje shema metapodataka. str. 16. URL: https://repozitorij.ffos.hr/islandora/object/ffos:1594 (2018-08-14)

[13] Usp. Nelli, Fabio. Nav. dj., str. 18.

2.4. Alati za vizualizaciju podataka

Rad će pokazati rad dva alata, Highcharts i D3, ali postoji mnogo drugih i vrijedi spomenuti one poznatije.

1. Tableau je alat za vizualizaciju velikih podataka za korporacije. Tableau omogućuje stvaranje grafikona, grafova, karata i mnogih drugih grafika, a podaci se mogu pristupiti od bilo gdje – preglednika, računala, mobitela, ugrađenog u bilo koju aplikaciju i cloud servera.[14]
2. Infogram omogućuje povezivanje njihovih vizualizacija i infografika s podacima u realnom vremenu. Postoje predlošci za brzo pokretanje i mogu se personalizirati s dodatnim vizualizacijama kao što su slike, video zapisi, grafovi i sl.[15]
3. ChartBlocks je jednostavan online alat koji izrađuje grafikone pomoću JavaScript biblioteke D3.js, ali ne zahtijeva od korisnika kodiranje. Također, gradi vizualizacije iz proračunskih tablica, baza podataka i podataka u realnom vremenu. [16] Za programere koji žele sami stvoriti interaktivne vizualizacije i staviti ih na web stranicu, Google Charts je user friendly i na njihovoj stranici ima galerija s raznim vrstama vizualizacije na raspolaganju.[17]
4. Charts.js je jednostavan i open source, a radi na principu <canvas> HTML 5 elementa.[18]
5. Polymaps je besplatna JavaScript biblioteka za izradu dinamičnih i interaktivnih karata koristeći SVG.[19]
6. Spomenimo još jedan, Leaflet, JavaScript biblioteka otvorenog koda za interaktivne karte prilagođene mobilnim uređajima.[20]

[14] https://www.tableau.com/products (2018-08-17)
[15] https://infogram.com/features (2018-08-17)
[16] https://www.chartblocks.com/en/features (2018-08-17)
[17] https://developers.google.com/chart (2018-08-17)
[18] http://www.chartjs.org/docs/latest/ (2018-08-17)
[19] http://polymaps.org/ (2018-08-17)
[20] https://leafletjs.com/ (2018-08-17)

3. Highcharts

Highcharts nudi vrlo privlačan i profesionalan izgled 2D / 3D grafikona na tržištu, što su prepoznale najveće tvrtke na svijetu kao što su Facebook, Twitter, Nokia itd. Kod odabira Highchartsa, uzima se u obzir podrška za izobilje vrsta grafikona, prilagodljivost različitim širinama zaslona, dinamičnost, podrška preglednika, višejezičnost i dr. Podržava dinamično iscrtavanje podataka, za primjer serije podataka se mogu dodavati i brisati i osi se mogu preoblikovati bilo kada nakon izrade grafikona.[21] Sami grafikoni su vizualno ugodni, a stil je jednostavan. Okrugli uglovi savjeta za alat i legende (unutarnji i vanjski ukoliko ih legenda ima) s jednostavnom granicom se ne bore za pozornost gledatelja i lijepo se uklapaju u grafikon. Ukratko, nijedan element u Highchartsu ne natječe se s drugim elementima za pažnju, već jednako dijele teret i rade zajedno kao grafikon.[22] Pokriveno je sve potrebno da bi se napravio grafikon pomoću Highchartsa. Prvi grafikon na kojem će se objasniti gore spomenuti elementi je jednostavni stupčani grafikon s vodoravno orijentiranim pravokutnim stupovima. (vidi Slika 4. i 5.) Za ove prve grafikone neće biti potrebno koristiti lokalnog mrežnog poslužitelja, tj. u ovom slučaju XAMPP.

Slika 4. prikazuje programski kod za izradu stupčastog grafikona. Podcrtano plavom linijom na slici je element `<script>` koji se spominjao u poglavlju instalacije. Preglednik neće prepoznati Highcharts grafikone ukoliko nema te jedne linije stoga je vrlo bitno ne zaboraviti pozvati highcharts.js. Žutom linijom je podcrtan *div* koji pokriva prostor na kojem će se nalaziti grafikon. Crvenom bojom je uokvirena izrada grafikona. Iniciran dodavanjem JavaScript oznake `<script> </script>`, koji može biti bilo gdje na mrežnoj stranici, sadrži poprilično samoobjašnjive elemente. *Series:* element pokriva različite skupove podataka, u ovom slučaju ocjene svakog pojedinca. Element je niz sačinjen od dva objekta, a svaki sadrži ime i *data* koji je također niz od tri vrijednosti, odnosno ocjene. Slika 5. prikazuje izrađen grafikon. Grafikon predstavlja ocjene mature iz tri predmeta za dva studenta. Prozorčić sa slike se pojavljuje prilikom prelaženja miša preko rezultata Studenta2 iz Stranog jezika A. Kod je poprilično

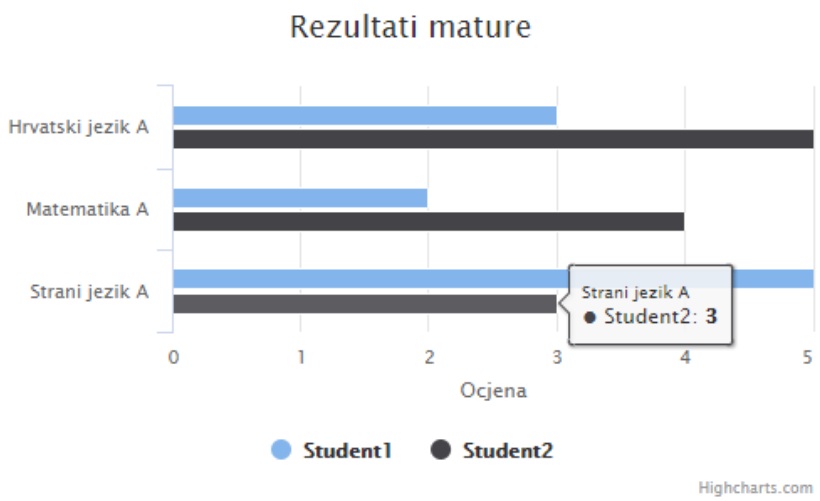
[21] Usp. Shahid, Bilal. Highcharts Essentials. URL: https://books.google.hr/books?id=iNskBQAAQBAJ&printsec=frontcover#v=onepage&q&f=false (2018-08-15)
[22] Usp. Kuan, Joe. Learning Highcharts 4. Birmingham: Packt Publising, 2015. str. 21-22. URL: https://books.google.hr/books?id=M9hpBgAAQBAJ&lpg=PP1&dq=highcharts&lr&pg=PP3#v=onepage&q=highcharts&f=false (2018-08-15)

jednostavan za pratiti. Za složenije grafikone, Highcharts je vrlo *user-friendly* i na svojoj službenoj stranici nudi mnogo primjena raznih grafikona zajedno s njihovim kodom te opširnu dokumentaciju o različitim funkcijama što omogućava pravljenje grafikona bez da se unaprijed mora znati sav kod. Već se gotovi primjeri grafikona s "trial and error" metodom mogu oblikovati u grafikon kakav korisnik ili klijent zahtjeva, ne moraju se više grafikoni praviti od nule.[23]

```
<!DOCTYPE html>

<html lang="en">
<head>
    <meta charset="utf-8" />
    <title>Prvi grafikon</title>
</head>
<body>

  <div id="matura" style="height: 300px; width: 500px"></div>

  <script src="http://code.highcharts.com/highcharts.js"></script>

  <script>
  Highcharts.chart('matura', {
      chart: {
        type: 'bar'
      },
      title: {
        text: 'Rezultati mature'
      },
      xAxis: {
        categories: ['Hrvatski jezik A', 'Matematika A', 'Strani jezik A'],
      },
      yAxis: {
        title: {
          text: 'Ocjena'
        },
        allowDecimals: false,
        tickInterval: 1,
        min: 0,
        max: 5
      },
      series: [{
        name: 'Student1',
        data: [3, 2, 5]
      }, {
        name: 'Student2',
        data: [5, 4, 3]
      }],
    });
  </script>
</body>
</html>
```

Slika 4. Kod – izrada stupčastog grafikona

[23] https://www.highcharts.com/demo (2018-08-16)



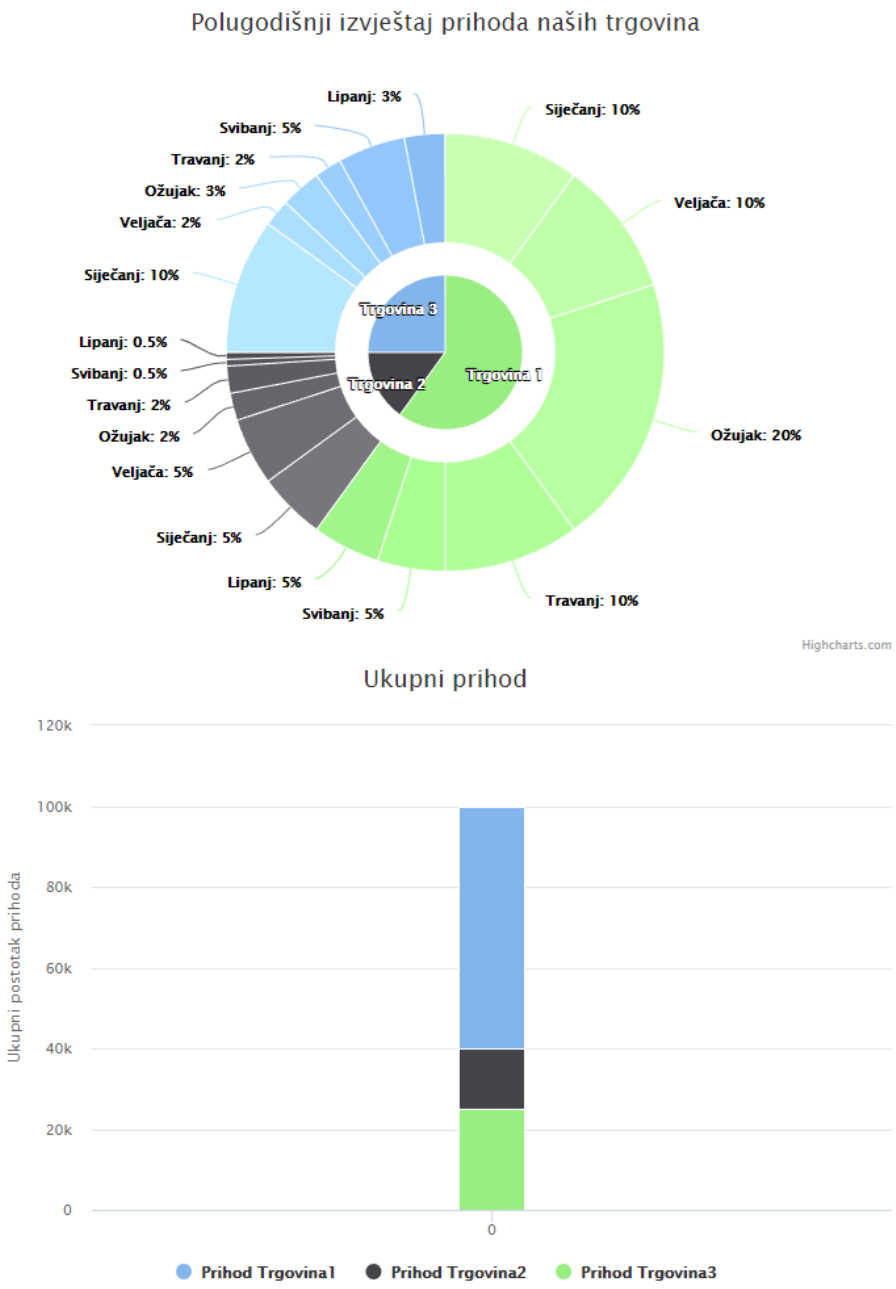
Slika 5. Stupčasti Highcharts grafikon

Sve animacije u Highchartsu fino su podešene: glatke s postupnim usporavanjem. Sljedeći grafikon je također jedan od najčešće korištenih, a to je strukturni krug ili kružni grafikon. Naziva ga se još i tortnim grafikonom. Grafikon je podijeljen na kružne isječke i svaki od njih predstavlja neku djelomičnu vrijednost, a cijeli krug predstavlja zbroj svih tih vrijednosti. Za primjer će se izraditi grafikon za šestomjesečni financijski izvještaj imaginarne tvrtke koja posjeduje tri trgovine i uspoređuje njihov promet te ukupni prihod. (vidi Slika 6.) Zbog veličine koda, cijeli kod se može pronaći na kraju radu među prilozima (vidi Prilog 2.), a ovdje će se izdvojiti dijelovi koda koje vrijedi posebno analizirati.

```
<div id="prihodPoTrgovinama" style="height: 500px; width: 700px;
float: center;"></div>
```

Za popratni, stupčasti grafikon, napravljen je novi div jer bi on u protivnom prebrisao kružni grafikon.

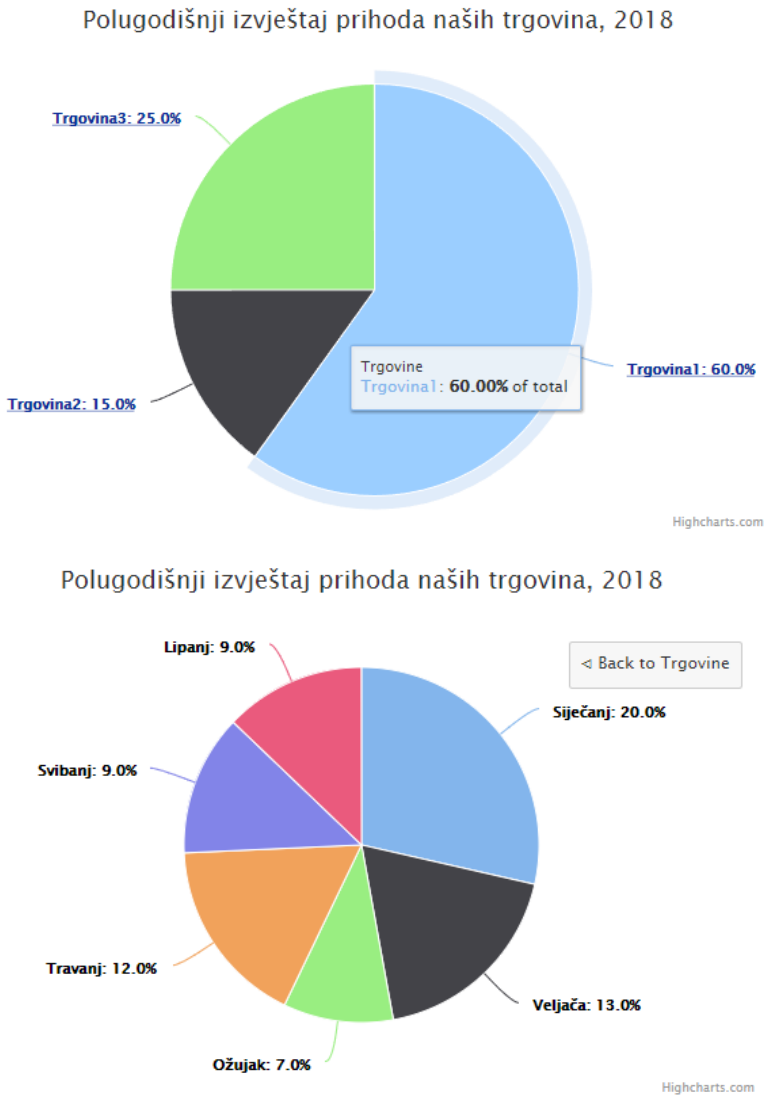
U ovom primjeru se prvo kreiraju varijable *categories* i *data* (nizovi) i daju vrijednosti, s petljama se redom uzima vrijednost iz nizova, stavlja u nove varijable *shopData* i *monthData* i prelazi na sljedeću vrijednost u nizu. Tako šest puta jer je uvjet petlje *i*< *dataLen*, koji je 6, počevši od *i=0*.



Slika 6. Kružni i stupčasti Highcharts grafikon

Unutarnji krug prikazuje udio prihoda po trgovinama, a vanjski krug udio prihoda trgovina po mjesecima. Ipak, pomoću funkcije drilldown, kružni grafikon se može vizualno dotjerati i učiniti ga interaktivnijim. (vidi Prilog 3.) Davanjem dijelu grafikona *drilldown* opciju koji odgovara serijskoj konfiguraciji u drilldown.series nizu, taj dio grafikona se povezuje sa skrivenim

serijama.[24] Laički rečeno, *drilldown* omogućuje dublji pogled u dio grafikona, stvaranjem zasebnog grafikona za taj određeni dio. (vidi Slika 7.)

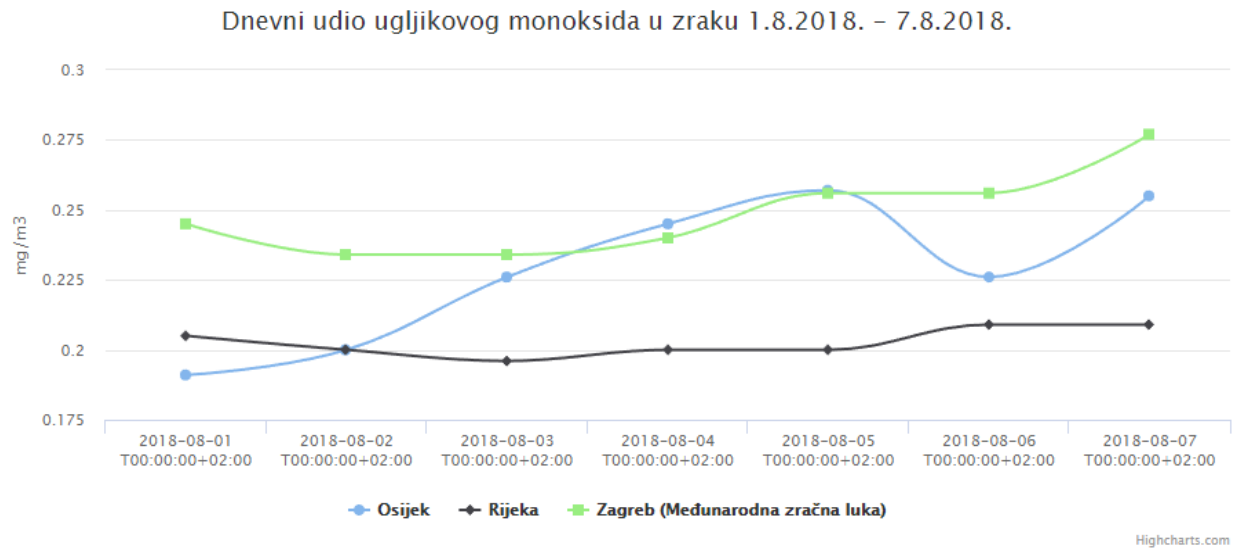


Slika 7. Kružni grafikon s funkcijom drilldown

Područja poput klimatologije i meteorologije zbog volumena, brzine, heterogenosti i složenosti dostupnih podataka su problematična za istraživati i analizirati podatke i nije uvijek moguće analizirati podatke s uobičajenim tehnikama. Sateliti, postaje i senzori svaki dan, svaki sat ili čak konstantno u realnom vremenu skupljaju podatke koji se akumuliraju i rezultat je masovna

[24] https://www.highcharts.com/docs/chart-concepts/drilldown (2018-08-16)

količina podataka koje treba analizirati. Vizualizacija tih podataka može pomoći znanstvenicima da lakše obavljaju svoje zadatke poput korelacijske analize klimatskih faktora i predviđanja događaja.[25] Za sljedeći grafikon koristit će se JSON podaci o kvaliteti zraka u Republici Hrvatskoj objavljene od Hrvatske agencije za okoliš i prirodu, a preuzeti s gov.data.hr.[26] Usporedit će se udio ugljikovog monoksida u Osijeku, Rijeci i Zagrebu[27] na dnevnoj bazi u rasponu od 1.8.2018. do 7.8.2018. (vidi Prilog 4.)



Slika 8. Linijski grafikon napravljen s JSON formatom podataka

```
var co = [{
            "vrijednostOS": "0.191",
        "vrijednostRI": "0.205",
        "vrijednostZG": "0.245",
        "mjernaJedinica":"mg\/m3",
              "Time": "2018-08-01 T00:00:00+02:00"
          },[...]
```

Ovdje se varijabli *co* (kemijska formula za ugljikov monoksid) pridodao niz objekata koji se sastoji od parova svojstvo:vrijednost, ali se u JSON formatu sve smatra tekstom jer se tako podaci izmjenjuju između preglednika i servera.

[25] Bikakis, Nikos. Big Data Visualization Tools*. str. 6-7. URL: https://arxiv.org/pdf/1801.08336.pdf (2018-8-17)
[26] https://data.gov.hr/dataset/kvaliteta-zraka-u-republici-hrvatskoj (2018-08-17)
[27] Zagreb ima višestrukih postaja od kojih su se mogli uzeti podaci, za ovaj primjer se koristi postaja Međunarodne zračne luke Zagreb.

```
var  categories = [],
     points = [],
     points2 = [],
     points3 = [],
     mj = [];

$.each(co, function(i, el) {
     categories.push(el.vrijeme);
     mj.push(el.mjernaJedinica);
     points.push(parseFloat(el.vrijednostOS));
     points2.push(parseFloat(el.vrijednostRI));
     points3.push(parseFloat(el.vrijednostZG));
 });
```

U principu ovo je JQuery iteracija (`$.each`[28]) koja ide kroz niz varijable i redom metodom *push*[29] dodaje nove predmete u prethodno napravljene prazne nizove. Na početku je i=0, uđe u prvi objekt niza varijable, a *el* preuzima vrijednost i kad prođe kroz objekt, odnosno kad prođe kroz sve metode koje su mu zadani, *i*se povećava i tako dok ne prođe kroz cijeli niz. Metoda *parseFloat* će vrijednosti koje su prvotno u JSON-u tekst raščlaniti i vratiti kao realne brojeve.[30] JSON se može nalaziti u zasebnoj .json datoteci umjesto unutar JavaScripta kao u gornjem primjeru. U tom slučaju, prvo treba pokrenuti lokalni server (u ovom slučaju XAMPP-ov Apache). Ajax može čitati sa servera i slati podatke serveru pa će poslužiti za čitanje json.json datoteke koja sadrži iste podatke s istom sintaksom kao prethodna JSON varijabla.[31]

```
var co = $.ajax({
                url: "json.json",
                async: false
            }).responseText;

co = JSON.parse(co);
```

Cijeli kod se nalazi u prilogu (vidi Prilog 5. i 6.). Bitno se podsjetiti da datoteke budu na istoj razini (za XAMPP u htdocs mapi), ili da se pripazi na putanju za *url:*. *responseText* će vratiti

[28] http://api.jquery.com/jquery.each (2018-08-18)
[29] https://www.w3schools.com/jsref/jsref_push.asp (2018-08-18)
[30] https://www.w3schools.com/jsref/jsref_parsefloat.asp (2018-08-18)
[31] http://api.jquery.com/jquery.ajax/ (2018-08-19)

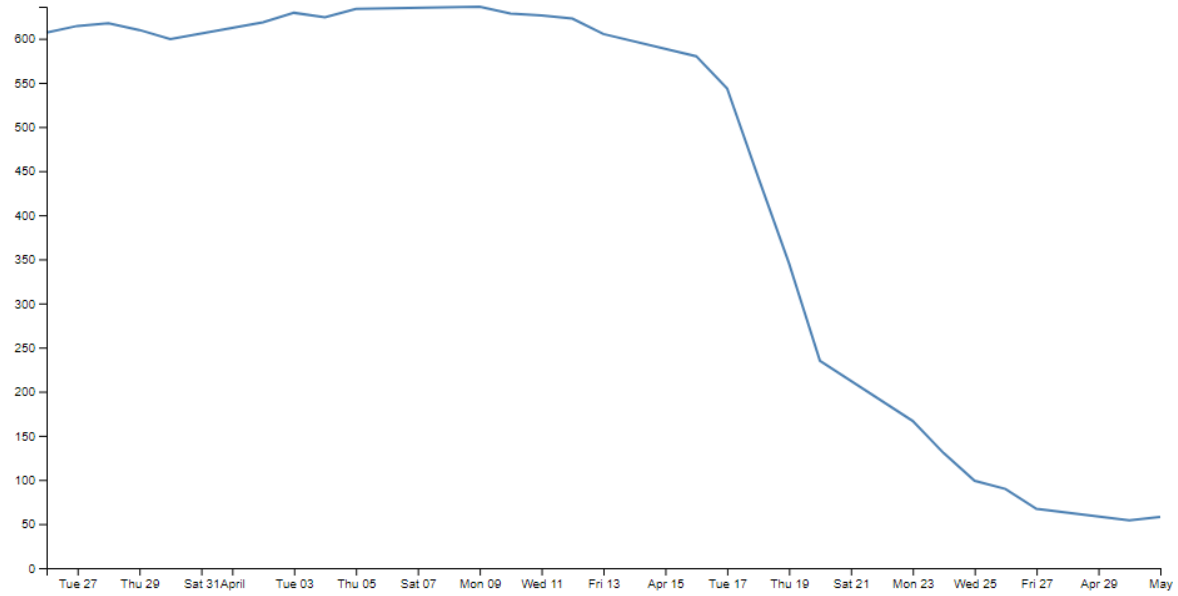
vrijednost kao string, a *JSON.parse(co);* kao JavaScript objekt. Sve ostalo je isto kao u prethodnom primjeru.

4. D3.js

Druga biblioteka koju rad pokriva je D3.js. D3.js pomaže pridružiti podatke na elemente DOM-a, a potom se s HTML, CSS i/ili SVG prikazuju podaci. D3 sadrži različite funkcije koji pomažu u raspoređivanju sadržaja i u oblikovanju podataka tako da se mogu prikazati pomoću popularnih crtačkih metoda. "Ljepota D3-a je da vam omogućuje povezivanje podataka i onoga što se pojavljuje na zaslonu na način koji izravno povezuje to dvoje. Promijenite podatke i promijeni se objekt na zaslonu. D3-ov trik je da vam dopusti postavljanje onoga što se pojavljuje na zaslonu. Krug, linija, točka na karti, grafikon, lopta koja skače, gradijent (i mnogo, mnogo više). Jednom kad se povežu podaci i objekt, mogućnosti su beskrajne".[32]Treba spomenuti kako je krivulja učenja D3-a vrlo strma i lako se izgubiti u kodu. Za razliku od Highchartsa kojim se prvenstveno prave grafikoni, s D3 se može vizualizirati sve što korisnik poželi. Ipak, rad će se držati grafikona i na linijskom grafikonu prikazati osnovne funkcije D3-a. (vidi Slika 9. i Prilog 7. i 8.)[33]

[32]Slobodan prijevod: „The beauty of D3 is that it allows you to associate data and what appears on the screen in a way that directly links the two. Change the data and you change the object on the screen. D3's trick is to let you set what appears on the screen. A circle, a line, a point on a map, a graph, a bouncing ball, a gradient (and way, way more). Once the data and the object are linked the possibilities are endless." // Maclean, Malcolm. D3 Tips and Tricks v4. URL: https://leanpub.com/d3-t-and-t-v4/read (2018-08-20)

[33]Primjer preuzet s: Maclean, Malcolm. D3 Tips and Tricks v4. URL: https://leanpub.com/d3-t-and-t-v4/read#simplegraph (2018-08-20)



Slika 9. Linijski D3 grafikon

U ovom primjeru se učitava .csv datoteka (Comma Seperated Values) stoga je potreban lokalni server. CSS igra veću ulogu nego kod Highchartsa. D3 će napraviti liniju klase *.line*, a onda se s CSS-om ta linija može podebljavati, skraćivati, mijenjati joj boju, spojiti početku i završnu točku i još mnogo toga.

```
var margin = {top: 20, right: 20, bottom: 30, left: 50},
    width = 960 - margin.left - margin.right,
    height = 500 - margin.top - margin.bottom;
```

Prva linija definira margine koje okružuju blok gdje je smješten objekt, a u vitičastoj zagradi je definiralo koliko će određena strana objekta biti udaljena od ruba bloka. Width i Height određuju širinu i visinu bloka.

```
d3.csv("data.csv", function(error, data) { [...] });
```

Ovaj dio koda učitava podatke za grafikon s data.csv datoteke. D3 može zatražiti podatke od običnog teksta, .json, .xml, .html, .csv i .tsv (Tab Separated Values) datoteka.

```
var x = d3.scaleTime().range([0, width]);
var y = d3.scaleLinear().range([height, 0]);
```

```
[…]
     x.domain(d3.extent(data, function(d) { return d.date; }));
     y.domain([0, d3.max(data, function(d) { return d.close;
})]);
```

Prve dvije linije osiguravaju da se podaci koje uzimamo ispravno uklapaju u grafikon. *d3.scaleTime* daje D3-u do znanja da se radi o vrijednostima za datum ili vrijeme..*range* određuje raspon grafikona. Drugi par linija uzima raspon vrijednosti podataka koje imamo i skalira grafikon kako bi ga uklopio u prostor koji je na raspolaganju. Za x os, *function(d) { return d.date; }* sve *date* vrijednosti iz .csv datoteke, *extent* potom traži minimum i maksimum vrijednosti u nizu, a *.domain* vraća te vrijednosti D3 kao raspon za x os. Y os već ima jednu vrijednost, 0 za početnu točku pa se s *max* traži samo najveća vrijednost među *close* vrijednostima u .csv datoteci.

```
var valueline = d3.line()
    .x(function(d) { return x(d.date); })
    .y(function(d) { return y(d.close); });
```

Ovime se ne pravi linija koju vidimo na grafikonu, ona sprema informacije pomoću kojih će se kasnije linijama spojiti koordinatne točke. Samo crtanje obavlja SVG.

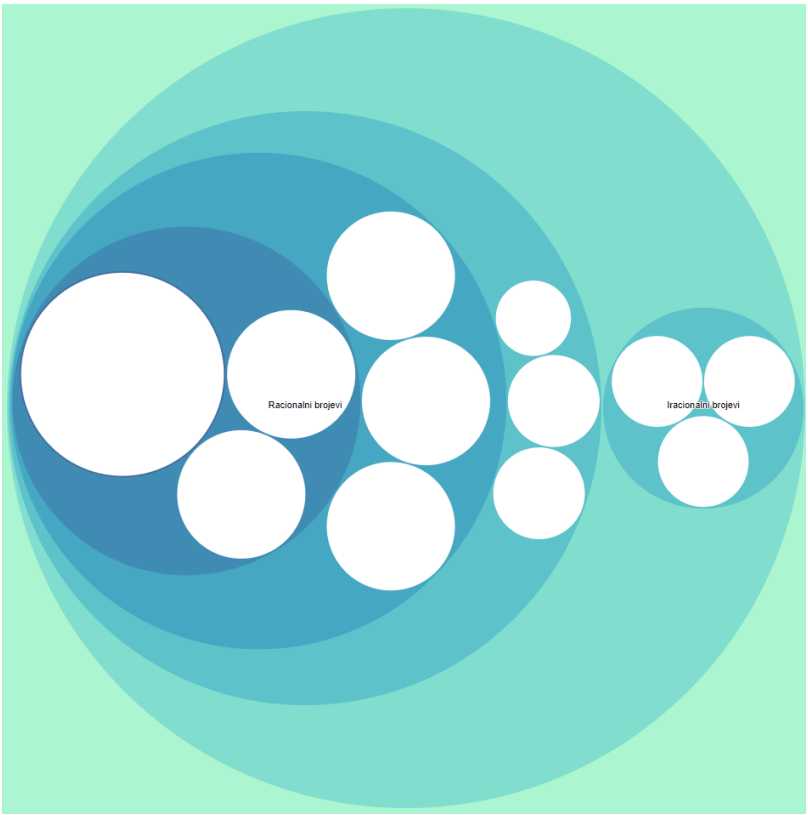
```
svg.append("path")
      .data([data])
      .attr("class", "line")
      .attr("d", valueline);
```

Element *path* predstavlja oblik koji se može manipulirati. *Path* se s *.data([data])* povezuje s podacima koje su se učitali iz datoteke, *line* će primijeniti CSS i *.attr("d", valueline);* će dodati atribut *d path*-u s podacima iz *valueline*. Tako će SVG-ov *path* točno znati kako da nacrta liniju. Na kraju SVG crta x i y os i grafikon je gotov.

Da se može bilo što napraviti i da nije potrebno sve raditi samostalno, dokazuje galerija s na stotine primjera, sve zahvaljujući aktivnoj D3 zajednici.[34]Upravo jedan od tih primjera[35] je

[34]https://github.com/d3/d3/wiki/Gallery (2018-8-20)

poslužio kao predložak za sljedeći primjer: interaktivni prikaz skupa realnih brojeva. (vidi Slika 10. i 11. te Prilog 9. i 10.) Realni brojevi se sastoje od racionalnih i iracionalnih brojeva, a unutar racionalnih brojeva je podskupina integera, unutar njega podskupina cijelih brojeva, a unutar njega prirodni brojevi. Svaki krug je interaktivan. Klikom na krug racionalnih brojeva započinje animacija kojom će se zoomirati u unutrašnjost kruga koji se sastoji od manjih krugova, jedan veliki koji predstavlja skup integera (koji se također može kliknuti i tako sve do prirodnih brojeva) i tri manja kruga u kojima se nalaze primjeri racionalnih brojeva. Realni brojevi se uče u osnovno i srednjoškolskom obrazovanju te bi ovakva vizualizacija mogla postati pomoćni alat u nastavi.



Slika 10. Vizualni prikaz skupa realnih brojeva

[35] https://bl.ocks.org/mbostock/7607535 (2018-8-20)

5. Zaključak

Vizualizacija podataka nije nikakvo novo otkriće, ali razvojem suvremenih tehnoloških mogućnosti i društvenog pomaka opće svijesti vizualizacija podataka je postala nova norma za kvalitetno prezentiranje i analiziranje skupova podataka. Rad se osvrnuo na korištenje JavaScript biblioteka, Highcharts i D3.js, zajedno s alatima mrežnog okruženja HTML, CSS i naravno JavaScript, da prikaže tek djelić tih tehnoloških mogućnosti. Svrha rada je bila prikazati što su JavaScript biblioteke u mogućnosti izgraditi uz malo korisnikovog znanja o tom skriptnom programskom jeziku. Highcharts nudi širok spektar predložaka grafikona spremni za neposrednuprimjenu, uz bogatu dokumentaciju koja korisnika vodi kroz čitav proces od prvih linija koda do gotovog grafikona. D3 se čini malo kompliciranijim od Highchartsa: krivulja učenja D3-a je strma i potrebno je malo više truda i vremena da se svlada, ali rezultat je vizualiziranje ne samo grafikona, već sve što korisnik zamisli. Uz to, D3 zajednica je vrlo aktivna i spremna za pomoći, a tu je i galerija gotovih primjera. Rad je pokazao početne uvjete u kojima se može početi raditi i pomoći oko prvih koraka u pravljenju vizualnih primjeraka skupova podataka nego objasniti svaku liniju koda i nabrojati sve vrste grafikona i kako ih nacrtati. Također, na kraju rada se ukazalo na mogućnost korištenja D3.js biblioteke, odnosno vizualizacije podataka općenito, u obrazovanju primjerom vizualizacije nastavne jedinice iz matematike, nešto što bi se u budućnosti moglo detaljnije istražiti. Sve u svemu, JavaScript biblioteke pružaju praktičan, jednostavan i moderan način vizualizacije podataka koji je primjenjiv za razne prigode i potrebe korporacija i individualaca.

Literatura

1. Bikakis, Nikos. Big Data Visualization Tools*. URL: https://arxiv.org/pdf/1801.08336.pdf(2018-08-21)

2. Chartblocks. https://www.chartblocks.com/en (2018-08-21)

3. Charts.js. URL: http://www.chartjs.org/docs/latest (2018-08-21)

4. Data-Driven Documents. URL: https://d3js.org(2018-08-21)

5. Finch, L. Jannete, Flenner R. Angela. Using Data Visualization to Examine an Academic Library Collection. // College & Reasearch Libraries 77, 6(2016). URL: https://crl.acrl.org/index.php/crl/article/view/16555/18001(2018-08-21)

6. Github: D3 gallery. URL: https://github.com/d3/d3/wiki/Gallery (2018-08-21)

7. Google charts. URL: https://developers.google.com/chart/ (2018-08-21)

8. Highcharts. URL: https://www.highcharts.com(2018-08-21)

9. Infogram. URL: https://infogram.com/ (2018-08-21)

10. jQuery. URL: https://jquery.com/(2018-08-21)

11. Kirk, Andy. Data Visualization: A Successful Design Process. URL: https://books.google.hr/books?id=I4qBVLfD3t4C&printsec=frontcover#v=onepage&q&f=false(2018-08-21)

12. Kuan, Joe. Learning Highcharts 4. Birmingham: Packt Publising, 2015. URL: https://books.google.hr/books?id=M9hpBgAAQBAJ&lpg=PP1&dq=highcharts&lr&pg=PP3#v=onepage&q=highcharts&f=false(2018-08-21)

13. Leaflet. URL: https://leafletjs.com/ (2018-08-21)

14. Matanović, Antun. Razvoj programskog rješenja za mapiranje shema metapodataka. str. URL: https://repozitorij.ffos.hr/islandora/object/ffos:1594(2018-08-21)

15. Nelli, Fabio. Beginning JavaScript Charts : With jqPlot, d3, and Highcharts. CA, USA: Apress, 2013. URL: https://books.google.hr/books?id=BZ4QAwAAQBAJ&lpg=PP3&dq=highcharts&lr&pg=PP1#v=onepage&q&f=false (2018-08-21)

16. Polymaps. URL: http://polymaps.org (2018-08-21)

17. Portal otvorenih podataka Republike Hrvatske. URL: https://data.gov.hr(2018-08-21)

18. Shahid, Bilal. Highcharts Essentials. URL: https://books.google.hr/books?id=iNskBQAAQBAJ&printsec=frontcover#v=onepage&q&f=false(2018-08-21)

19. Tableau. URL: https://www.tableau.com/ (2018-08-21)

20. W3Schools.com: JavaScript Array Reference. URL: https://www.w3schools.com/jsref/jsref_obj_array.asp (2018-08-21)

21. XAMPP Installers and Downloads for Apache Friends. URL: https://www.apachefriends.org/index.html (2018-08-21)

22. Zoomable Circle Packing. URL: https://bl.ocks.org/mbostock/7607535 (2018-08-21)

Prilozi

```svg
<svg xmlns="http://www.w3.org/2000/svg" width='300px' height='300px'>

  <link xmlns="http://www.w3.org/1999/xhtml" rel="stylesheet" href="svg.css" type="text/css"/>
  <circle id='krug' cx='140' cy='140' r='60'></circle>
  <text id='tekst' x='100' y='250'>Hello, World!</text>

</svg>
```

```html
<!DOCTYPE html>

<html lang="en">
<head>
    <meta charset="utf-8" />
    <title>SVG primjer - Hello World</title>

</head>
<body>
  <object type="image/svg+xml" data="hello.svg"
  width="300" height="300" border="3"></object>

</body>
</html>
```

```css
#krug{
  fill: blue;
}
```

Prilog 1. Kod SVG, HTML i CSS datoteka SVG primjera

```
<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="utf-8" />
    <title>Drugi grafikon</title>
</head>

<body>
  <script src="http://code.highcharts.com/highcharts.js"></script>
  <script src="https://code.highcharts.com/modules/drilldown.js"></script>
  <div id="grafikon" style="height: 500px; width: 700px; float:left;"></div>

  <div id="prihodPoTrgovinama" style="height: 500px;
  width: 700px; float: center;"></div>

<script>

  var colors = Highcharts.getOptions().colors,
    categories = [
        "Trgovina 1",
        "Trgovina 2",
        "Trgovina 3",
    ],
    data = [
        {
            "y": 60,
            "color": colors[2],
            "drilldown": {
                "name": "Trgovina 1",
                "categories": [
                    "Siječanj",
                    "Veljača",
                    "Ožujak",
                    "Travanj",
                    "Svibanj",
                    "Lipanj"
                ],
                "data": [
                    10,
                    10,
                    20,
                    10,
                    5,
                    5
                ]
```

```
            }
        },
        {
            "y": 15,
            "color": colors[1],
            "drilldown": {
                "name": "Trgovina 2",
                "categories": [
                    "Siječanj",
                    "Veljača",
                    "Ožujak",
                    "Travanj",
                    "Svibanj",
                    "Lipanj"
                ],
                "data": [
                    5,
                    5,
                    2,
                    2,
                    0.5,
                    0.5,
                ]
            }
        },
        {
            "y": 25,
            "color": colors[0],
            "drilldown": {
                "name": "Trgovina 3",
                "categories": [
                    "Siječanj",
                    "Veljača",
                    "Ožujak",
                    "Travanj",
                    "Svibanj",
                    "Lipanj"
                ],
                "data": [
                    10,
                    2,
                    3,
                    2,
          5,
          3
                ]
```

```
            }
        }
    ],
    shopData = [],
    monthData = [],
    i,
    j,
    dataLen = data.length,
    drillDataLen,
    brightness;


// Build the data arrays
for (i = 0; i < dataLen; i += 1) {

    // add shop data
    shopData.push({
        name: categories[i],
        y: data[i].y,
        color: data[i].color
    });

    // add month data
    drillDataLen = data[i].drilldown.data.length;
    for (j = 0; j < drillDataLen; j += 1) {
        brightness = 0.2 - (j / drillDataLen) / 5;
        monthData.push({
            name: data[i].drilldown.categories[j],
            y: data[i].drilldown.data[j],
            color: Highcharts.Color(data[i].color).brighten(brightness).get()
        });
    }
}

// Create the chart
Highcharts.chart('grafikon', {
    chart: {
        type: 'pie'
    },
    title: {
        text: 'Polugodišnji izvještaj prihoda naših trgovina'
    },
    yAxis: {
        title: {
            text: 'Ukupni postotak prihoda'
        }
```

```
    },
    plotOptions: {
        pie: {
            shadow: false,
            center: ['50%', '50%']
        }
    },
    tooltip: {
        valueSuffix: '%'
    },
    series: [{
        name: 'Udio trgovine',
        data: shopData,
        size: '25%',
        dataLabels: {
            formatter: function () {
                return this.y > 5 ? this.point.name : null;
            },
            color: '#ffffff',
            distance: -10
        }
    }, {
        name: 'Mjesečni udio',
        data: monthData,
        size: '80%',
        innerSize: '50%',
        dataLabels: {
            formatter: function () {
                // display only if larger or same as 0.5
                return this.y >= 0.5 ? '<b>' + this.point.name + ':</b> ' +
                    this.y + '%' : null;
            }
        }
    }],
    responsive: {
        rules: [{
            condition: {
                maxWidth: 400
            },
            chartOptions: {
                series: [{
                    id: 'versions',
                    dataLabels: {
                        enabled: false
                    }
                }]
```

```
            }
        }]
    }
});

  Highcharts.chart('prihodPoTrgovinama', {
    chart: {
        type: 'column'
    },
  title: {
        text: 'Ukupni prihod'
    },
    yAxis: {
        title: {
            text: 'Ukupni postotak prihoda',
      categories: ['Prihod']
        }
    },
  plotOptions: {
        series: {
            stacking: 'normal',
      pointWidth: 50
        }
    },
  tooltip: {
        valueSuffix: 'kn'
    },
  series: [{
        name: 'Prihod Trgovina1',
        data: [60000],
      }, {
        name: 'Prihod Trgovina2',
        data: [15000]
      }, {
        name: 'Prihod Trgovina3',
        data: [25000]
      }]
  });
  </script>
</body>

</html>
```

Prilog 2. Programski kod za kružni i popratni stupčasti grafikon

```
<html>
<head>
<script src="https://code.highcharts.com/highcharts.js"></script>
<script src="https://code.highcharts.com/modules/data.js"></script>
<script src="https://code.highcharts.com/modules/drilldown.js"></script>

<div id="prihodPoTrgovinama" style="min-width: 300px; min-width: 600px;
 height: 400px; margin: 0 auto; float: left"></div>
</head>
<body>
<script language="JavaScript">
// Create the chart
Highcharts.chart('prihodPoTrgovinama', {
    chart: {
        type: 'pie'
    },
    title: {
        text: 'Polugodišnji izvještaj prihoda naših trgovina, 2018'
    },
    plotOptions: {
        series: {
            dataLabels: {
                enabled: true,
                format: '{point.name}: {point.y:.1f}%'
            }
        }
    },
    tooltip: {
        headerFormat: '<span style="font-size:11px">{series.name}</span><br>',
        pointFormat: '<span style="color:{point.color}">{point.name}</span>: <b>{point.y:.2f}%</b> of total<br/>'
    },

    "series": [
        {
            "name": "Trgovine",
            "colorByPoint": true,
            "data": [
                {
                    "name": "Trgovina1",
                    "y": 60,
                    "drilldown": "t1"
                },
                {
                    "name": "Trgovina2",
                    "y": 15,
                    "drilldown": "t2"
                },
                {
                    "name": "Trgovina3",
                    "y": 25,
                    "drilldown": "t3"
                }
            ]
        }
```

```
    ],
    "drilldown": {
        "series": [
            {
                "name": "Trgovina1",
                "id": "t1",
                "data": [
                    [
                        "Siječanj",
                        20
                    ],
                    [
                        "Veljača",
                        13
                    ],
                    [
                        "Ožujak",
                        7
                    ],
                    [
                        "Travanj",
                        12
                    ],
                    [
                        "Svibanj",
                        9
                    ],
                    [
                        "Lipanj",
                        9
                    ]
                ]
            },
            {
                "name": "Trgovina2",
                "id": "t2",
                "data": [
                    [
                        "Siječanj",
                        2
                    ],
                    [
                        "Veljača",
                        4.3
                    ],
                    [
                        "Ožujak",
                        3.8
                    ],
                    [
                        "Travanj",
                        1.4
                    ],
                    [
```

```
                        "Svibanj",
                        1.8
                    ],
                    [
                        "Lipanj",
                        1.7
                    ]
                ]
            },
            {
                "name": "Trgovina3",
                "id": "t3",
                "data": [
                    [
                        "Siječanj",
                        0
                    ],
                    [
                        "Veljača",
                        0
                    ],
                    [
                        "Ožujak",
                        12
                    ],
                    [
                        "Travanj",
                        6
                    ],
                    [
                        "Svibanj",
                        3.5
                    ],
                    [
                        "Lipanj",
                        3.5
                    ]
                ]
            }
        ]
    }
});
</script>
</body>
</html>
```

Prilog 3. Programski kod za kružni grafikon s drilldown funkcijom

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="utf-8" />
    <title>Highcharts data from JSON Response</title>
    <style>
    body{
      margin-top: 30px;
      margin-left:40px;
    }
    </style>
  <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
  <script src="http://code.highcharts.com/highcharts.js"></script>
  <script type="text/javascript">
        $(document).ready(function () {
           var co = [{
                "vrijednostOS": "0.191",
         "vrijednostRI": "0.205",
         "vrijednostZG": "0.245",
         "mjernaJedinica":"mg\/m3",
                "vrijeme": "2018-08-01 T00:00:00+02:00"
           }, {
                "vrijednostOS": "0.2",
         "vrijednostRI": "0.2",
         "vrijednostZG": "0.234",
         "mjernaJedinica":"mg\/m3",
                "vrijeme": "2018-08-02 T00:00:00+02:00"
           }, {
                "vrijednostOS": "0.226",
         "vrijednostRI": "0.196",
         "vrijednostZG": "0.234",
         "mjernaJedinica":"mg\/m3",
                "vrijeme": "2018-08-03 T00:00:00+02:00"
           }, {
                "vrijednostOS": "0.245",
         "vrijednostRI": "0.2",
         "vrijednostZG": "0.24",
         "mjernaJedinica":"mg\/m3",
                "vrijeme": "2018-08-04 T00:00:00+02:00"
           }, {
                "vrijednostOS": "0.257",
         "vrijednostRI": "0.2",
         "vrijednostZG": "0.256",
         "mjernaJedinica":"mg\/m3",
                "vrijeme": "2018-08-05 T00:00:00+02:00"
           }, {
                "vrijednostOS": "0.226",
         "vrijednostRI": "0.209",
         "vrijednostZG": "0.256",
         "mjernaJedinica":"mg\/m3",
                "vrijeme": "2018-08-06 T00:00:00+02:00"
       }, {
                "vrijednostOS": "0.255",
          "vrijednostRI": "0.209",
```

```
            "vrijednostZG": "0.277",
            "mjernaJedinica":"mg\/m3",
                  "vrijeme": "2018-08-07 T00:00:00+02:00"
              }];

              var options = {
                  chart: {
                      renderTo: 'container',
                      type: 'spline'
                  },
             title: {
              text: 'Dnevni udio ugljikovog monoksida u zraku 1.8.2018. - 7.8.2018.'
            },
             tooltip: {
              valueSuffix: {}
            },
                  xAxis: {},
                  series: [{},{},{}],
             yAxis: {
              title: { text: "mg/m3" }
            }
               };

                  var categories = [],
                      points = [];
               points2 = [];
               points3 = [];
               mj = [];

                    $.each(co, function(i, el) {
              categories.push(el.vrijeme);
              mj.push(el.mjernaJedinica);
              points.push(parseFloat(el.vrijednostOS));
              points2.push(parseFloat(el.vrijednostRI));
              points3.push(parseFloat(el.vrijednostZG));
                   });
                  options.xAxis.categories = categories;
                  options.series[0].data = points;
             options.series[0].name = "Osijek";
             options.series[1].data = points2;
             options.series[1].name = "Rijeka";
             options.series[2].data = points3;
             options.series[2].name = "Zagreb (Međunarodna zračna luka)";
             options.tooltip.valueSuffix = mj[0];
                  var chart = new Highcharts.Chart(options);
           });
        </script>
    </head>
    <body>
        <div id="container" style="height: 400px"></div>
    </body>
    </html>
```

Prilog 4. Programski kod linijskog grafikona

```html
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="utf-8" />
    <title>Highcharts data from JSON Response</title>
    <style>
        body{
            margin-top: 30px;
            margin-left:40px;
        }
    </style>
    <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
    <script src="http://code.highcharts.com/highcharts.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {

            //Paziti na dobar link
            var co = $.ajax({
                url: "json.json",
                async: false
            }).responseText;

            //treba dekodirati u obicni array
            co = JSON.parse(co);

            var options = {
                chart: {
                    renderTo: 'container',
                    type: 'spline'
                },
                title: {
                    text: 'Dnevni udio ugljikovog monoksida u zraku 1.8.2018. - 7.8.2018.'
                },
                tooltip: {
                    valueSuffix: {}
                },
                xAxis: {},
                series: [{},{},{}],
                yAxis: {
                    title: { text: "mg/m3" }
                }
            };

            var categories = [],
                points = [];
            points2 = [];
            points3 = [];
            mj = [];

            $.each(co, function(i, el) {
                categories.push(el.vrijeme);
                mj.push(el.mjernaJedinica);
                points.push(parseFloat(el.vrijednostOS));
                points2.push(parseFloat(el.vrijednostRI));
```

```
                points3.push(parseFloat(el.vrijednostZG));
            });
            options.xAxis.categories = categories;
            options.series[0].data = points;
            options.series[0].name = "Osijek";
            options.series[1].data = points2;
            options.series[1].name = "Rijeka";
            options.series[2].data = points3;
            options.series[2].name = "Zagreb (Međunarodna zračna luka)";
            options.tooltip.valueSuffix = mj[0];
            var chart = new Highcharts.Chart(options);
        });
    </script>
</head>
<body>
<div id="container" style="height: 400px"></div>
</body>
</html>
```

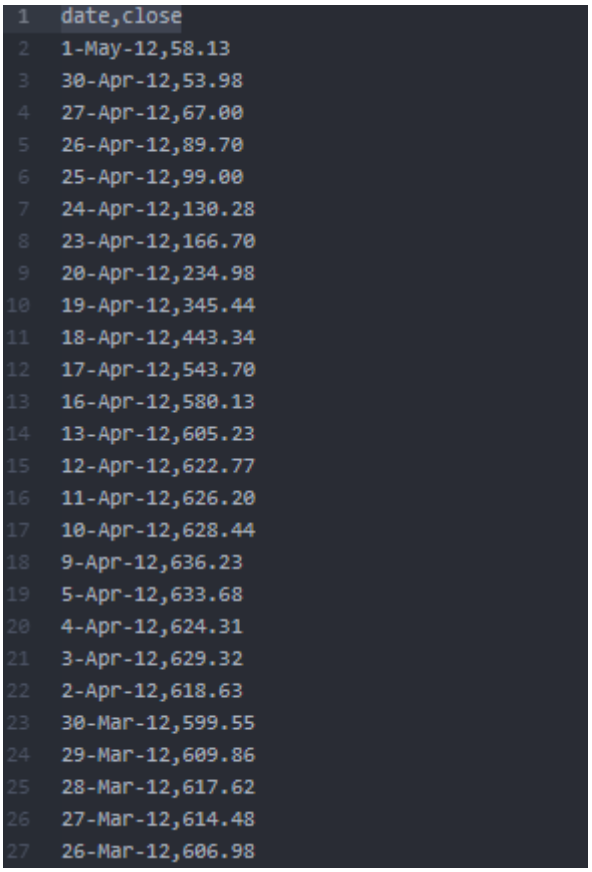
Prilog 5. Programski kod linijskog grafikona s učitavanjem JSON datoteke

```
[{
        "vrijednostOS": "0.191",
        "vrijednostRI": "0.205",
        "vrijednostZG": "0.245",
        "mjernaJedinica":"mg\/m3",
        "vrijeme": "2018-08-01 T00:00:00+02:00"
    }, {
        "vrijednostOS": "0.2",
        "vrijednostRI": "0.2",
        "vrijednostZG": "0.234",
        "mjernaJedinica":"mg\/m3",
        "vrijeme": "2018-08-02 T00:00:00+02:00"
    }, {
        "vrijednostOS": "0.226",
        "vrijednostRI": "0.196",
        "vrijednostZG": "0.234",
        "mjernaJedinica":"mg\/m3",
        "vrijeme": "2018-08-03 T00:00:00+02:00"
    }, {
        "vrijednostOS": "0.245",
        "vrijednostRI": "0.2",
        "vrijednostZG": "0.24",
        "mjernaJedinica":"mg\/m3",
        "vrijeme": "2018-08-04 T00:00:00+02:00"
    }, {
        "vrijednostOS": "0.257",
        "vrijednostRI": "0.2",
        "vrijednostZG": "0.256",
        "mjernaJedinica":"mg\/m3",
        "vrijeme": "2018-08-05 T00:00:00+02:00"
    }, {
        "vrijednostOS": "0.226",
        "vrijednostRI": "0.209",
        "vrijednostZG": "0.256",
        "mjernaJedinica":"mg\/m3",
        "vrijeme": "2018-08-06 T00:00:00+02:00"
    }, {
        "vrijednostOS": "0.255",
        "vrijednostRI": "0.209",
        "vrijednostZG": "0.277",
        "mjernaJedinica":"mg\/m3",
        "vrijeme": "2018-08-07 T00:00:00+02:00"
    }]
```


Prilog 6. Sadržaj JSON datoteke za linijski Highcharts grafikon

```
<!DOCTYPE html>
<meta charset="utf-8">
<style> /* CSS */
.line {
  fill: none;
  stroke: steelblue;
  stroke-width: 2px;
}
</style>
<body>
<!-- Load the d3.js library -->
<script src="https://d3js.org/d3.v4.min.js"></script>
<script>
// set the dimensions and margins of the graph
var margin = {top: 20, right: 20, bottom: 30, left: 50},
    width = 960 - margin.left - margin.right,
    height = 500 - margin.top - margin.bottom;
// parse the date / time
var parseTime = d3.timeParse("%d-%b-%y");
// set the ranges
var x = d3.scaleTime().range([0, width]);
var y = d3.scaleLinear().range([height, 0]);
// define the line
var valueline = d3.line()
    .x(function(d) { return x(d.date); })
    .y(function(d) { return y(d.close); });
// append the svg obgect to the body of the page
// appends a 'group' element to 'svg'
// moves the 'group' element to the top left margin
var svg = d3.select("body").append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
  .append("g")
    .attr("transform",
          "translate(" + margin.left + "," + margin.top + ")");
// Get the data
d3.csv("data.csv", function(error, data) {
  if (error) throw error;
  // format the data
  data.forEach(function(d) {
      d.date = parseTime(d.date);
      d.close = +d.close;
  });
  // Scale the range of the data
  x.domain(d3.extent(data, function(d) { return d.date; }));
  y.domain([0, d3.max(data, function(d) { return d.close; })]);
  // Add the valueline path.
  svg.append("path")
      .data([data])
      .attr("class", "line")
      .attr("d", valueline);
  // Add the X Axis
  svg.append("g")
      .attr("transform", "translate(0," + height + ")")
      .call(d3.axisBottom(x));
  // Add the Y Axis
  svg.append("g")
      .call(d3.axisLeft(y));
});
</script>
</body>
```

Prilog 7. Programski kod linijskog D3 grafikona



```
date,close
1-May-12,58.13
30-Apr-12,53.98
27-Apr-12,67.00
26-Apr-12,89.70
25-Apr-12,99.00
24-Apr-12,130.28
23-Apr-12,166.70
20-Apr-12,234.98
19-Apr-12,345.44
18-Apr-12,443.34
17-Apr-12,543.70
16-Apr-12,580.13
13-Apr-12,605.23
12-Apr-12,622.77
11-Apr-12,626.20
10-Apr-12,628.44
9-Apr-12,636.23
5-Apr-12,633.68
4-Apr-12,624.31
3-Apr-12,629.32
2-Apr-12,618.63
30-Mar-12,599.55
29-Mar-12,609.86
28-Mar-12,617.62
27-Mar-12,614.48
26-Mar-12,606.98
```

Prilog 8. Sadržaj CSV datoteke za linijski D3 grafikon

```
<!DOCTYPE html>
<meta charset="utf-8">
<style>

.node {
  cursor: pointer;
}

.node:hover {
  stroke: #000;
  stroke-width: 1.5px;
}

.node--leaf {
  fill: white;
}

.label {
  font: 11px "Helvetica Neue", Helvetica, Arial, sans-serif;
  text-anchor: middle;
  text-shadow: 0 1px 0 #fff, 1px 0 0 #fff, -1px 0 0 #fff, 0 -1px 0 #fff;
}

.label,
.node--root,
.node--leaf {
  pointer-events: none;
}

</style>
<svg width="960" height="960"></svg>
<script src="https://d3js.org/d3.v4.min.js"></script>
<script>

var svg = d3.select("svg"),
    margin = 20,
    diameter = +svg.attr("width"),
    g = svg.append("g").attr("transform", "translate(" + diameter / 2 + "," + diameter / 2 + ")");

var color = d3.scaleLinear()
    .domain([-1, 5])
    .range(["hsl(152,80%,80%)", "hsl(228,30%,40%)"])
    .interpolate(d3.interpolateHcl);

var pack = d3.pack()
    .size([diameter - margin, diameter - margin])
    .padding(2);

d3.json("mat.json", function(error, root) {
  if (error) throw error;

  root = d3.hierarchy(root)
      .sum(function(d) { return d.size; })
      .sort(function(a, b) { return b.value - a.value; });

  var focus = root,
      nodes = pack(root).descendants(),
      view;

  var circle = g.selectAll("circle")
    .data(nodes)
```

```
    .enter().append("circle")
      .attr("class", function(d) { return d.parent ? d.children ? "node" : "node node--leaf" : "node node--root"; })
      .style("fill", function(d) { return d.children ? color(d.depth) : null; })
      .on("click", function(d) { if (focus !== d) zoom(d), d3.event.stopPropagation(); });

  var text = g.selectAll("text")
    .data(nodes)
    .enter().append("text")
      .attr("class", "label")
      .style("fill-opacity", function(d) { return d.parent === root ? 1 : 0; })
      .style("display", function(d) { return d.parent === root ? "inline" : "none"; })
      .text(function(d) { return d.data.name; });

  var node = g.selectAll("circle,text");

  svg
      .style("background", color(-1))
      .on("click", function() { zoom(root); });

  zoomTo([root.x, root.y, root.r * 2 + margin]);

  function zoom(d) {
    var focus0 = focus; focus = d;

    var transition = d3.transition()
        .duration(d3.event.altKey ? 7500 : 750)
        .tween("zoom", function(d) {
          var i = d3.interpolateZoom(view, [focus.x, focus.y, focus.r * 2 + margin]);
          return function(t) { zoomTo(i(t)); };
        });

    transition.selectAll("text")
      .filter(function(d) { return d.parent === focus || this.style.display === "inline"; })
        .style("fill-opacity", function(d) { return d.parent === focus ? 1 : 0; })
        .on("start", function(d) { if (d.parent === focus) this.style.display = "inline"; })
        .on("end", function(d) { if (d.parent !== focus) this.style.display = "none"; });
  }

  function zoomTo(v) {
    var k = diameter / v[2]; view = v;
    node.attr("transform", function(d) { return "translate(" + (d.x - v[0]) * k + "," + (d.y - v[1]) * k + ")"; });
    circle.attr("r", function(d) { return d.r * k; });
  }
});

</script>
```

Prilog 9. Programski kod za D3 interaktivne krugove

```
{
   "name": "Realni brojevi",
   "children": [
    {
     "name": "Racionalni brojevi",
     "children": [
      {"name": "-3.4", "size": 2047},
      {"name": "2/3", "size": 1375},
      {"name": "Integeri", "size": 10000,
    "children": [
    {"name": "Cijeli brojevi", "size": 10000,
    "children": [
      {"name": "Prirodni brojevi", "size": 10000,
      "children": [
        {"name": "1,2,3,4...n", "size": 10000}
      ]},
      {"name": "-2", "size": 4000},
      {"name": "2", "size": 4000}
      ]},
    {"name": "V49", "size": 4000},
    {"name": "-5", "size": 4000},
    {"name": "0", "size": 4000}
    ]},
      {"name": "9.45454545...", "size": 2042}
     ]
    },
    {"name": "Iracionalni brojevi", "size": 8176,
  "children": [
    {"name": "V2", "size": 2000},
    {"name": "V3", "size": 2000},
    {"name": "PI", "size": 2000}
  ]}
   ]
 }
```

Prilog 10. Sadržaj JSON datoteke za interaktivne krugove