

Vizualizacija podataka

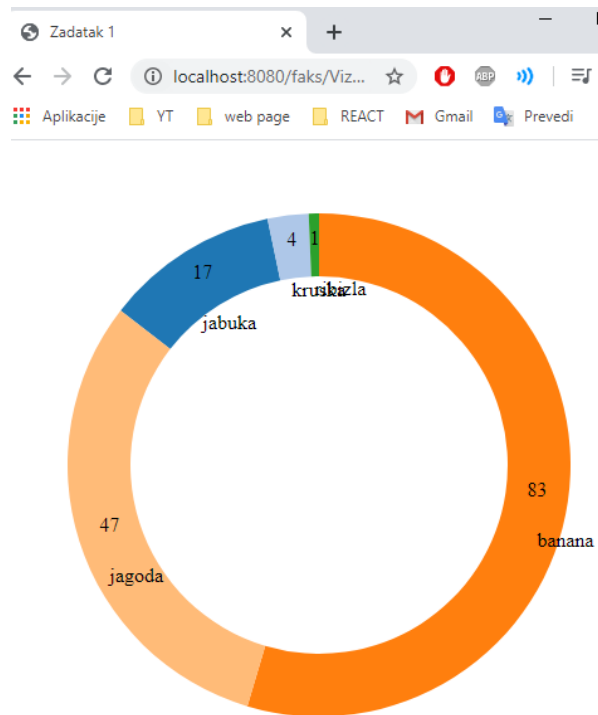
LV4 PRIKAZI

Student: Kristijan Koščak

Smjer: DRC, 1.godina

Datum: 15.04.2020.

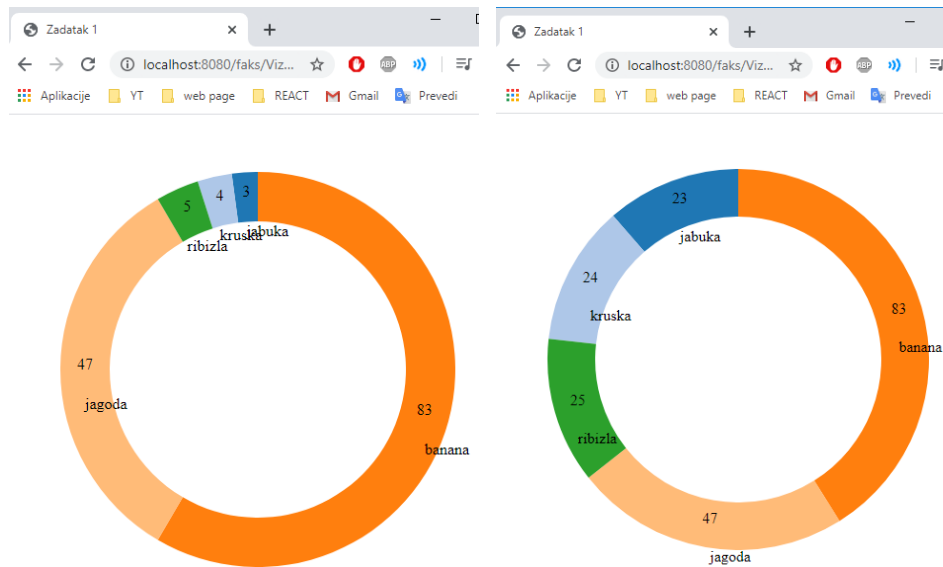
Zadatak 1:



```
pieArcs.append("text")
    .attr("transform", function (d) {
        return "translate(" + arc.centroid(d) + ")";
    })
    .attr("dy", 40)
    .text(function (d,i) { return data[i].name; });
```

Na kod iz predloška dodali smo gore navedeni kod tj. novi tekst koji smo centrirali u središte svakog dijela. Nakon toga smo svakom podesili udaljenost y osi od postavljenog središta i u konačnici ispisali vrstu voća. Možemo uočiti nepravilnosti prilikom iscrtavanja. Navedeno je moguće riješiti rotiranjem teksta kako bi izbjegli preklapanje.

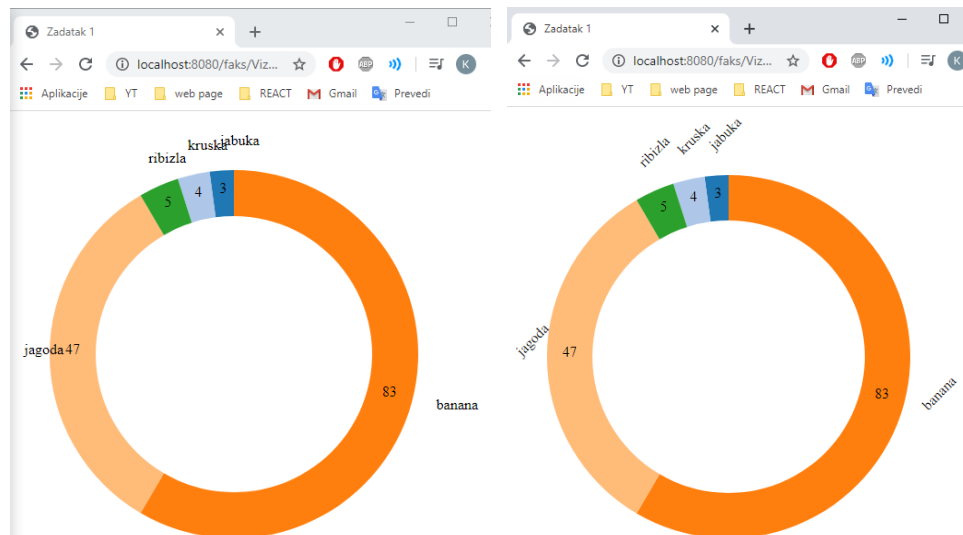
Zadatak 2:



```
pieArcs.append("text")
    .attr("transform", function (d) {
        var center = arc.centroid(d);
        return "translate(" + center[0]*1.3 + "," + center[1]*1.3 + ") rotate(-45)";
    })
    .text(function (d,i) { return data[i].name; });
```

a)

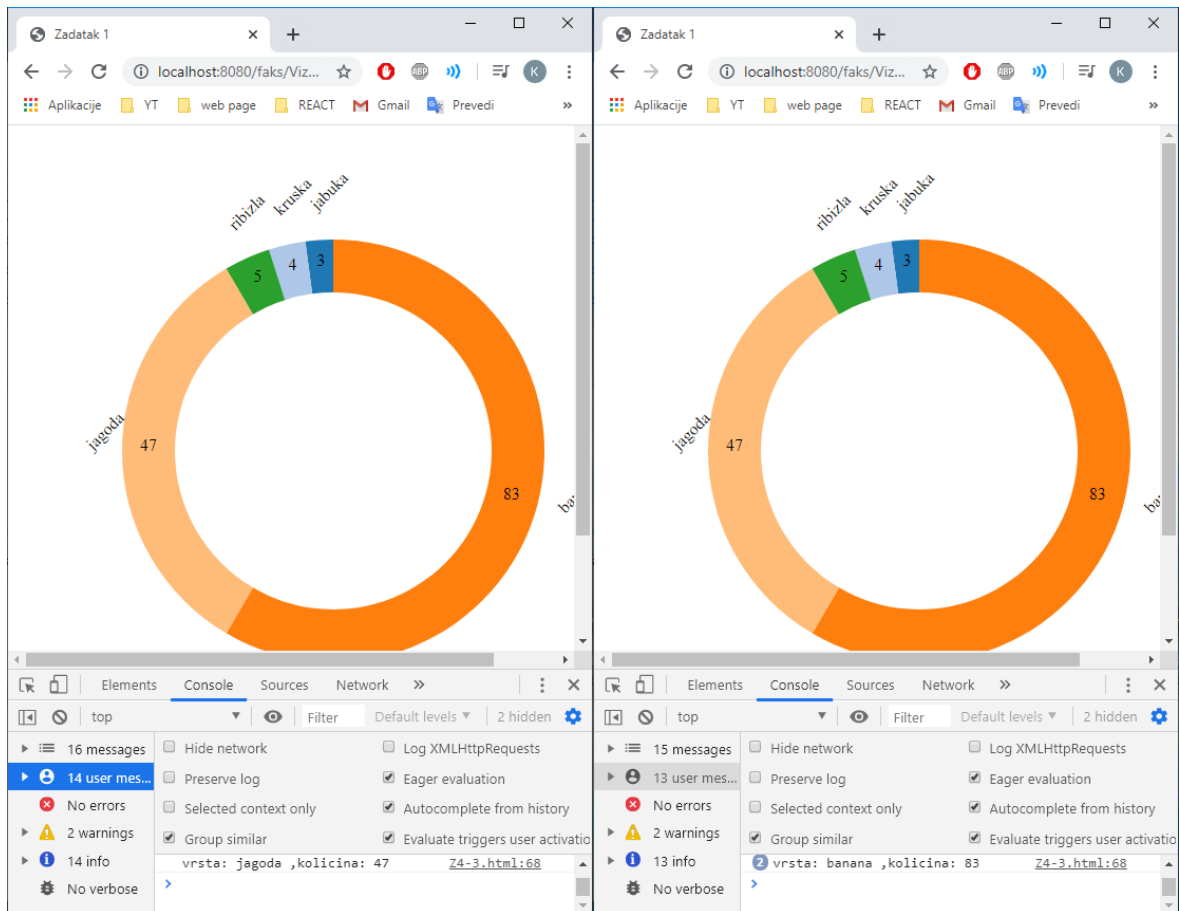
b)



Već u prvom zadatku uočili smo nepravilnosti, odnosno preklapanje teksta pojedinih segmenata. Segmente smješta od najvećih vrijednosti prema najnižima. Zbog toga na prve dvije slike uočavamo razliku zelenog i plavih segmenata. Kada su vrijednosti male, dolazi do prijelaza između tekstova navedenih segmenata, a kada su velike toga

preklapanja nema. Kao što je i traženo, gornjim kodom smjestili smo tekst iznad svakog segmenta te znatno smanjili preklapanje. Najveći učinak ima rotiranje koje smo primijenili na tekst postavljen iznad segmenata.

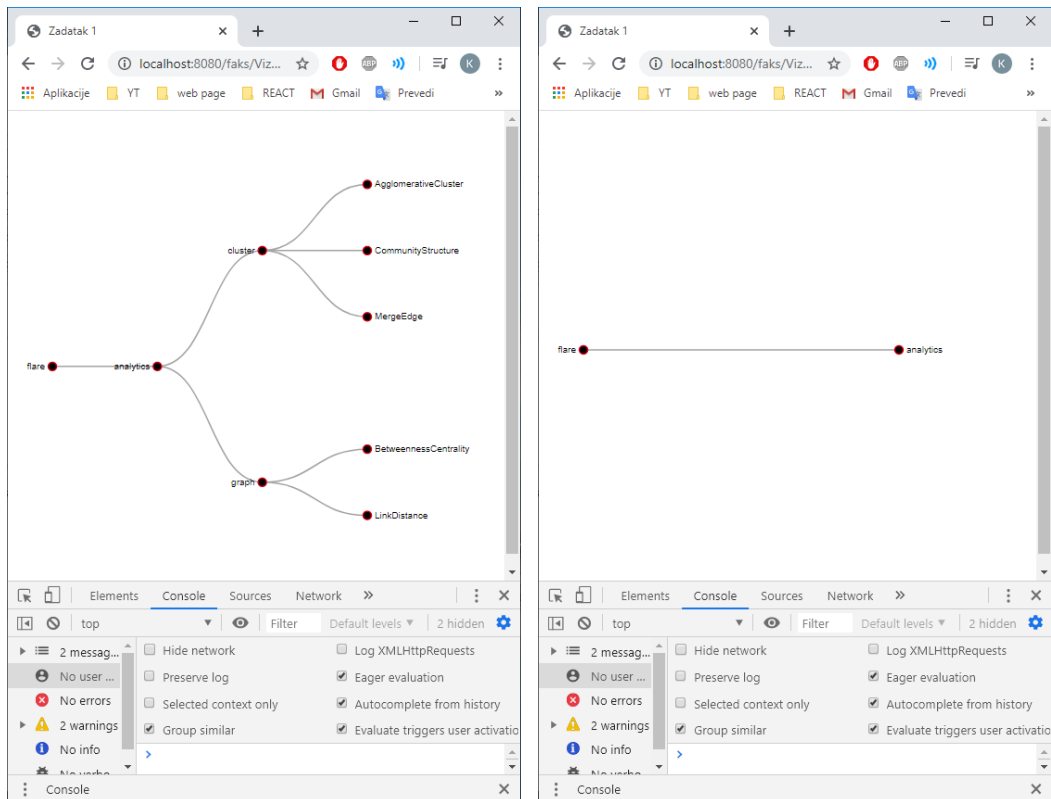
Zadatak 3:



```
svg.selectAll("path").on("mouseover", function (d,i) { console.log("vrsta : "+data[i].name + " ,kolicina: " +d.value)});
```

Na slikama je vidljiv ispis vrste i količine pojedinog segmenta. Navedeno je ostvareno gornjim kodom. Dohvatili smo sve segmente te smo pratili položaj miša. Ukoliko se on nalazi iznad nekog segmenta, ispisujemo njegovu vrstu i količinu.

Zadatak 4:



```
var d = {
  "name": "flare",
  "children": [
    {
      "name": "analytics",
      "children": [
        {
          "name": "cluster",
          "children": [
            { "name": "AgglomerativeCluster", "size": 3938 },
            { "name": "CommunityStructure", "size": 3812 },
            { "name": "MergeEdge", "size": 743 }
          ]
        },
        {
          "name": "graph",
          "children": [
            { "name": "BetweennessCentrality", "size": 3534 },
            { "name": "LinkDistance", "size": 5731 }
          ]
        }
      ]
    }
  ]
}
```

```

    ]
  }
]
};

update(d);

function update(data) {
  var width = 500;
  var height = 500;

  var cluster = d3.layout.cluster()
    .size([height, width - 160]);

  var diagonal = d3.svg.diagonal()
    .projection(function (d) { return [d.y, d.x]; });

  var svg = d3.select("body").append("svg")
    .attr("width", width)
    .attr("height", height)
    .append("g")
    .attr("transform", "translate(40,0)");

  var nodes = cluster.nodes(data),
      links = cluster.links(nodes);

  var link = svg.selectAll(".link")
    .data(links)
    .enter().append("path")
    .attr("class", "link")
    .attr("d", diagonal);

  var node = svg.selectAll(".node")
    .data(nodes)
    .enter().append("g")
    .attr("class", "node")
    .attr("transform", function (d) {
      return "translate(" + d.y + "," + d.x + ")";
    })
    .append("circle").attr("r", 4.5);

  node.append("text")
    .attr("dx", function (d) { return d.children ? -8 : 8; })
    .attr("dy", 3)
    .style("text-anchor", function (d) {

```

```

        return d.children ? "end" : "start";
    })
    .text(function (d) { return d.name; });

node.on("click", function (d) {
    if (d.children !== undefined) {
        d.children = null;
        d3.select("svg").remove();
        update(data);
    }
});
}

```

Koristeći primjer stabla iz predloška laboratorijskih vježbi riješili smo zadani zadatak. Prelaskom pokazivača miša preko određenog čvora sakrivaju se sva djeca tog čvora ukoliko ih ima. Navedeno je ostvareno danim kodom. Kod iz predloška smo stavili u funkciju update koja kao parametar prima podatke koje prikazuje. Prelaskom preko određenog čvora brišu mu se djeca te se poziva ta ista funkcija s novo uređenim podacima.