

Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek

Vizualizacija podataka
Statistika NK Osijek
Projektni rad

Ivan Ivković

Osijek, 2020.

SADRŽAJ

1. OPIS PROJEKTOG ZADATAKA	1
2. KORIŠTENE TEHNOLOGIJE I OPIS PROGRAMSKOG RJEŠENJA.....	2
2.1. Korištene tehnologije	2
2.2. Programsko rješenje	2
LITERATURA.....	20

1. OPIS PROJEKTNOG ZADATKA

Cilj je projektnog zadatka vizualizirati statističke podatke NK Osijek, koji su trenutačno dostupni samo u tabličnim prikazima. S obzirom na relativno veliku količinu podataka, usporedba istih je jako teška, praktički nemoguća bez većeg zamaranja, a vizualizacija omogućuje lakše donošenje određenih zaključaka. Naime, NK Osijek je od 1992. do 2020. godine, u razdoblju za koje su u ovom radu vizualizirani podatci, prošao put od nekoliko faza, počevši od Domovinskog rata, preko relativno stabilnih godina uz naizmjenične lagane uspone i padove, potom velikog pada, koji je zamalo doveo do prestanka postojanja kluba, pa sve do privatizacije, koja je dovela do velikog uspona. Uzevši u obzir sve te faze i događaje, svakako je zanimljivo usporediti neke podatke među sezonama, ali i prisjetiti se nedavno završene sezone 2019./20., koja je do zadnjeg kola imala potencijal biti najbolja u cijeloj klupskoj povijesti, no na kraju je ipak ostala tek prosječna. Svi podatci korišteni u projektnom zadatku dostupni su na stranici <https://www.hrnogomet.com>, odakle su, zbog nepostojanja gotovih datasetova, ručno unošeni u novostvorene CSV datoteke.

Cjelokupni projekt dostupan je na: <https://github.com/iivkov/VP-projekt>.

2. KORIŠTENE TEHNOLOGIJE I OPIS PROGRAMSKOG RJEŠENJA

2.1. Korištene tehnologije

Za vizualizaciju podataka korišten je JavaScript, točnije njegova biblioteka D3.js, koja se koristi za manipulaciju HTML elementima u svrhu izrade vizualizacija za web [1]. Za izradu web stranice, na kojoj su korištene stvorene vizualizacije, korišteni su HTML za izgradnju same stranice i CSS za stilizaciju. Projekt je razvijen u okruženju Visual Studio Code i testiran pomoću lokalnog web poslužitelja, točnije dodatka Web Server for Chrome.

2.2. Programsko rješenje

Za početak je napravljeno korisničko sučelje na kojem su vidljive vizualizacije podataka. Radi se o web mjestu, kojeg čini nekoliko stranica jednostavne semantičke strukture. Elementom `<header>` opisano je zaglavlje stranice u kojem se nalaze naziv mjesta i logotip. U navigacijskoj traci `<nav>` nalaze se poveznice prema svim stranicama web mjesta, a podnožje stranice predstavljeno je elementom `<footer>`, gdje se nalaze oznaka autorskog prava i godina. Elementom `<main>` predstavljen je glavni dio stranice u kojem se nalazi sadržaj, odnosno vizualizacije podataka. Slikom 2.1. dan je programski kod kojim je realizirana jedna od stranica web mjesta, a slikom 2.2. dobiven izgled iste stranice.

```

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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>NKOStat - Početna</title>
  <link rel="stylesheet" href="styles/style.css">
</head>

<body>

  <header>
    <h1>NKOStat</h1>
  </header>
  <nav>
    <ul>
      <li><a href="index.html">Početna</a></li>
      <li><a href="pages/average_attendance.html">Posjećenost po sezonama</a></li>
      <li><a href="pages/cards.html">Kartoni po sezonama</a></li>
      <li><a href="pages/last_season.html">Sezona 2019./20.</a></li>
    </ul>
  </nav>

  <main>

    <article>
      <h2>Početna</h2>
      <p>Pozdrav!</p>
      <p>Ovo je stranica s vizualiziranom statistikom NK Osijek.</p>
    </article>

  </main>

  <footer>
    <p>&copy; 2020 ivan.ivkovic@etfos.hr</p>
  </footer>
</body>

</html>

```

Sl. 2.1. Programski kod za izradu web stranice



Sl. 2.2. Izgled web stranice

Kao što je već spomenuto, vizualizirani podatci nalaze se unutar elementa `<main>`, i to tako da se unutar njega ugnježđuje element `<div>`, u kojem se izvodi JavaScript kod za vizualizaciju podataka. Svi JavaScript kodovi izdvojeni su u posebne skripte, odvojene od HTML dokumenata, i stoga je u zaglavlju HTML dokumenta potrebno „uključiti“ odgovarajuću skriptu. Prvi grafički prikaz je usporedba prosječnog broja gledatelja na utakmicama u Gradskom vrtu počevši od sezone 1992., pa sve do sezone 2019./20. Za tu je usporedbu korišten stupčasti graf, tzv. *bar chart*, u kojem svaki stupac prikazuje jednu sezonu, točnije prosječan broj gledatelja u sezoni. Također, prelaskom mišem preko određenog stupca pojavljuje se prosječan broj gledatelja u sezoni koju vizualizira promatrani stupac. Na slici 2.3. nalazi se kod HTML dokumenta kojim se, među ostalim, uključuje skripta *average_attendance.js* i pojavljuje element `<div>` u kojem se nalazi graf. Skripta *average_attendance.js* dana je slikom 2.4., a rezultat izvršavanja skripte, odnosno izgled grafa kojim se vizualizira prosječan broj gledatelja po sezonama, prikazan je slikom 2.5.

```

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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>NKOStat - Prosječna posjećenost po sezonama</title>
  <link rel="stylesheet" href="../styles/style.css">
  <script src="http://d3js.org/d3.v4.js" charset="utf-8"></script>
  <script src="../scripts/average_attendance.js"></script>
</head>

<body>

  <header>
    <h1>NKOStat</h1>
  </header>
  <nav>
    <ul>
      <li><a href="../index.html">Početna</a></li>
      <li><a href="average_attendance.html">Posjećenost po sezonama</a></li>
      <li><a href="cards.html">Kartoni po sezonama</a></li>
      <li><a href="last_season.html">Sezona 2019./20.</a></li>
    </ul>
  </nav>

  <main>

    <article>
      <h2>Prosječna posjećenost po sezonama</h2>
      <p>Vizualizacija prosječne posjećenosti Gradskog vrta na utakmicama Prve HNL
od sezone 1992. do sezone 2019./20.</p>
    </article>

    <div id="average_attendance_chart"></div>

  </main>

  <footer>
    <p>&copy; 2020 ivan.ivkovic@etfos.hr</p>
  </footer>
</body>

</html>

```

Sl. 2.3. Stranica *average_attendance.html*

```

document.addEventListener('DOMContentLoaded', function(e) {
  // Dimensions and margins of the graph
  var margin = {
    top: 30,
    right: 30,
    bottom: 70,
    left: 60
  },
  width = 1000 - margin.left - margin.right,
  height = 500 - margin.top - margin.bottom;

  // Appending the svg object to the div called "average_attendance_chart"
  var svg = d3.select("#average_attendance_chart")
    .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 + ")");

  // Fetching and parsing the data
  d3.csv("../data/average_attendance.csv", function(data) {

    // X axis
    var x = d3.scaleBand()
      .range([0, width])
      .domain(data.map(function(d) {
        return d.season;
      }))
      .padding(0.2);
    svg.append("g")
      .attr("transform", "translate(0," + height + ")")
      .call(d3.axisBottom(x))
      .selectAll("text")
      .attr("transform", "translate(-10,0)rotate(-45)")
      .style("text-anchor", "end");

    // Y axis
    var y = d3.scaleLinear()
      .domain([1000, 10000])
      .range([height, 0]);
    svg.append("g")
      .call(d3.axisLeft(y));
  });

```

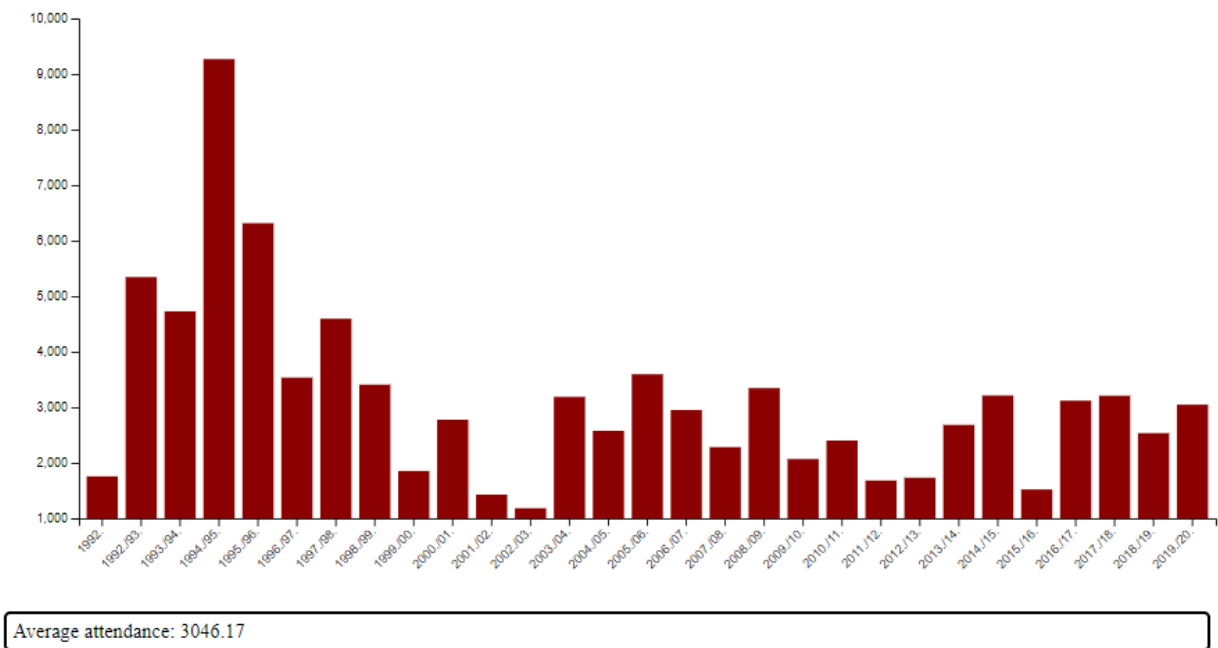


```

var tooltip = d3.select("#average_attendance_chart") // Tooltip
    .append("div")
    .style("opacity", 0)
    .attr("class", "tooltip")
    .style("background-color", "white")
    .style("border", "solid")
    .style("border-width", "2px")
    .style("border-radius", "5px")
    .style("padding", "5px")

// Functions for changing the tooltip when user goes over it
var mouseover = function(d) {
    tooltip
        .style("opacity", 1)
}
var mousemove = function(d) {
    tooltip
        .html("Average attendance: " + d.average_attendance)
        .style("left", (d3.mouse(this)[0] + 70) + "px")
        .style("top", (d3.mouse(this)[1] + "px")
}
var mouseleave = function(d) {
    tooltip
        .style("opacity", 0)
}
// Bars
svg.selectAll()
    .data(data)
    .enter()
    .append("rect")
    .attr("x", function(d) {
        return x(d.season);
    })
    .attr("y", function(d) {
        return y(d.average_attendance);
    })
    .attr("width", x.bandwidth())
    .attr("height", function(d) {
        return height - y(d.average_attendance);
    })
    .attr("fill", "#8B0000")
    .on("mouseover", mouseover)
    .on("mousemove", mousemove)
    .on("mouseleave", mouseleave)
});

```



Sl. 2.5. Grafički prikaz prosječnog broja gledatelja po sezonama

Grafički prikaz dobivenih kartona po sezonama realiziran je na sličan način kao i prethodni, grafički prikaz prosječnog broja gledatelja po sezonama, s tim što se u ovom prikazu, umjesto običnog stupčastog grafa, koristi grupirani stupčasti graf, tzv. *grouped bar chart*, koji omogućuje istovremenu usporedbu broja dobivenih i žutih i crvenih kartona. Skripta *cards.js* dana je slikom 2.6., a rezultat izvršavanja skripte, odnosno izgled grafa kojim se vizualizira broj žutih i crvenih kartona po sezonama, prikazan je slikom 2.7.

```
document.addEventListener('DOMContentLoaded', function(e) {
  // Dimensions and margins of the graph
  var margin = {
    top: 30,
    right: 30,
    bottom: 70,
    left: 60
  },
  width = 1000 - margin.left - margin.right,
  height = 500 - margin.top - margin.bottom;

  // Appending the svg object to the div called "cards_chart"
  var svg = d3.select("#cards_chart")
    .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 + ")");
```

```

// Fetching and parsing the data
d3.csv("../data/cards.csv", function(data) {

    // Lists of subgroups and seasons
    var subgroups = data.columns.slice(1)
    var seasons = d3.map(data, function(d) {
        return (d.season)
    }).keys()

    // X axis
    var x = d3.scaleBand()
        .domain(seasons)
        .range([0, width])
        .padding([0.2])
    svg.append("g")
        .attr("transform", "translate(0," + height + ")")
        .call(d3.axisBottom(x).tickSize(0));

    // Y axis
    var y = d3.scaleLinear()
        .domain([0, 120])
        .range([height, 0]);
    svg.append("g")
        .call(d3.axisLeft(y));

    // Position of subgroup
    var xSubgroup = d3.scaleBand()
        .domain(subgroups)
        .range([0, x.bandwidth()])
        .padding([0.05])

    // Color scale
    var color = d3.scaleOrdinal()
        .domain(subgroups)
        .range(['yellow', 'red'])

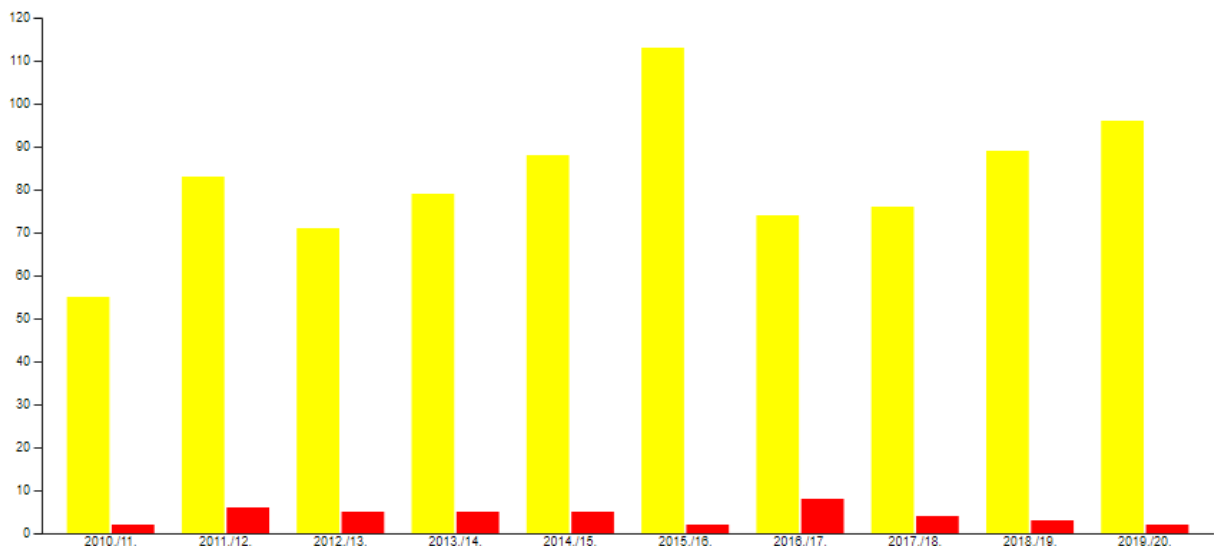
```

```

// Bars
svg.append("g")
  .selectAll("g")
  .data(data)
  .enter()
  .append("g")
  .attr("transform", function(d) {
    return "translate(" + x(d.season) + ",0)";
  })
  .selectAll("rect")
  .data(function(d) {
    return subgroups.map(function(key) {
      return {
        key: key,
        value: d[key]
      };
    });
  })
  .enter().append("rect")
  .attr("x", function(d) {
    return xSubgroup(d.key);
  })
  .attr("y", function(d) {
    return y(d.value);
  })
  .attr("width", xSubgroup.bandwidth())
  .attr("height", function(d) {
    return height - y(d.value);
  })
  .attr("fill", function(d) {
    return color(d.key);
  })
  })
});

```

Sl. 2.6. Skripta cards.js



Sl. 2.7. Grafički prikaz broja žutih i crvenih kartona po sezonama

Za razliku od stranica *average_attendance.html* i *cards.html*, koje su sadržavale po jedan graf, stranica *last_season.html* sadrži tri grafička prikaza, koji vizualiziraju neke podatke iz sezone 2019./20. Prvi u nizu je tortni graf, tzv. *pie chart*, koji vizualizira omjer pobjeda, neriješenih ishoda i poraza u promatranoj sezoni. Pritom su pobjede predstavljene zelenom bojom, neriješeni ishodi žutom bojom, a porazi crvenom bojom. Nad svakim od isječaka grafa nalazi se količina ishoda kojeg predstavlja određeni isječak grafa. Skripta *record.js* dana je slikom 2.8., a rezultat izvršavanja skripte, odnosno izgled grafa kojim se vizualizira omjer pobjeda, neriješenih ishoda i poraza u sezoni 2019./20., prikazan je slikom 2.9.

```
document.addEventListener('DOMContentLoaded', function(e) {
  // Dimensions and margins of the graph
  var width = 500
  height = 500
  margin = 50

  // Radius of the pieplot
  var radius = Math.min(width, height) / 2 - margin

  // Appending the svg object to the div called "record_chart"
  var svg = d3.select("#record_chart")
    .append("svg")
    .attr("width", width)
    .attr("height", height)
    .append("g")
    .attr("transform", "translate(" + width / 2 + "," + height / 2 + ")");
```

```

// Data: number of wins, draws and defeats
var data = {
  "wins": 17,
  "draws": 11,
  "defeats": 8
}

// Color scale
var color = d3.scaleOrdinal()
  .domain(data)
  .range(['#32CD32', '#FFFF00', '#E50000']);

// Position of groups on the pie
var pie = d3.pie()
  .value(function(d) {
    return d.value;
  })
var data_ready = pie(d3.entries(data))

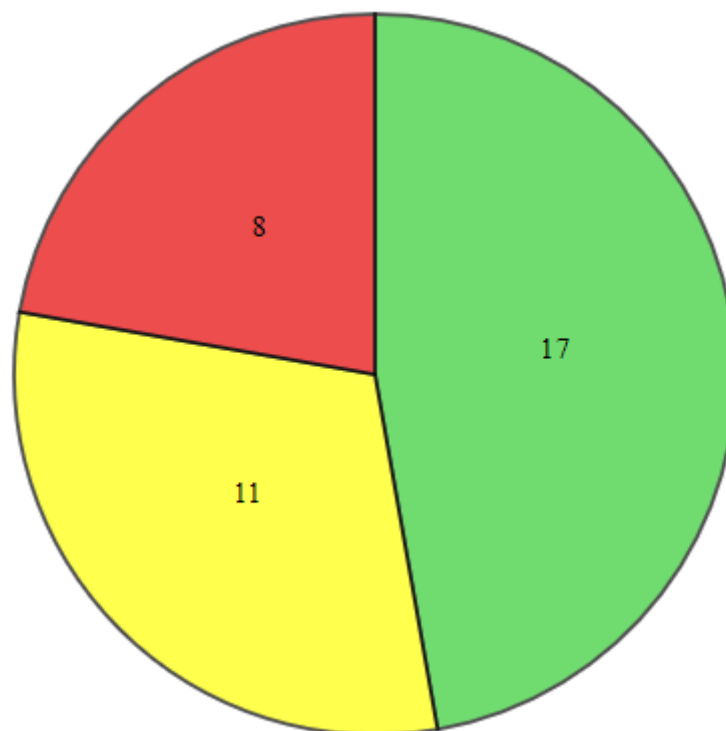
// Building arcs
var arcGenerator = d3.arc()
  .innerRadius(0)
  .outerRadius(radius)

// Building the pie chart
svg
  .selectAll("slices")
  .data(data_ready)
  .enter()
  .append("path")
  .attr("d", arcGenerator)
  .attr("fill", function(d) {
    return color(d.data.key)
  })
  .attr("stroke", "black")
  .style("stroke-width", "2px")
  .style("opacity", 0.7)

```

```
// Annotation with usage of the centroid method to get the best coordinates
svg
  .selectAll("slices")
  .data(data_ready)
  .enter()
  .append("text")
  .text(function(d) {
    return d.data.value
  })
  .attr("transform", function(d) {
    return "translate(" + arcGenerator.centroid(d) + ")";
  })
  .style("text-anchor", "middle")
  .style("font-size", 17)
});
```

Sl. 2.8. *Skripta record.js*



Sl. 2.9. *Grafički prikaz omjera pobjeda, neriješenih ishoda i poraza u sezoni 2019./20.*

Podatci o strijelcima u sezoni 2019./20. vizualizirani su uporabom stupčastog grafa, pri čemu svaki stupac predstavlja jednog igrača koji je u HNL-u postigao barem jedan pogodak. Prelaskom mišem preko stupca određenog strijelca pojavljuje se točan broj golova koje je taj strijelac postignuo. Skripta *goalscorers.js* dana je slikom 2.10., a rezultat izvršavanja skripte, odnosno izgled grafa kojim se vizualizira broj postignutih golova po strijelcima u sezoni 2019./20., prikazan je slikom 2.11.

```
document.addEventListener('DOMContentLoaded', function(e) {
  // Dimensions and margins of the graph
  var margin = {
    top: 30,
    right: 30,
    bottom: 70,
    left: 60
  },
  width = 1000 - margin.left - margin.right,
  height = 500 - margin.top - margin.bottom;

  // Appending the svg object to the div called "goalscorers_chart"
  var svg = d3.select("#goalscorers_chart")
    .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 + ")");

  // Fetching and parsing the data
  d3.csv("../data/goalscorers.csv", function(data) {

    // X axis
    var x = d3.scaleBand()
      .range([0, width])
      .domain(data.map(function(d) {
        return d.player;
      }))
      .padding(0.2);
    svg.append("g")
      .attr("transform", "translate(0," + height + ")")
      .call(d3.axisBottom(x))
      .selectAll("text")
      .attr("transform", "translate(-10,0)rotate(-45)")
      .style("text-anchor", "end");
```



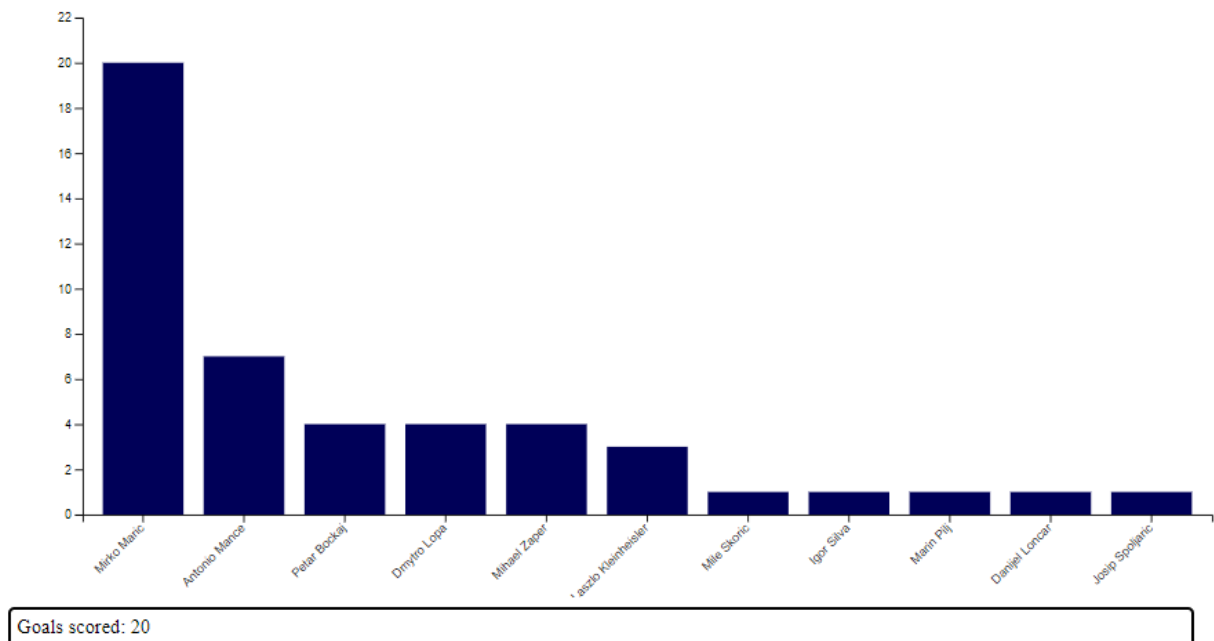
```

var y = d3.scaleLinear() // Y axis
    .domain([0, 22])
    .range([height, 0]);
svg.append("g")
    .call(d3.axisLeft(y));

var tooltip = d3.select("#goalscorers_chart") // Tooltip
    .append("div")
    .style("opacity", 0)
    .attr("class", "tooltip")
    .style("background-color", "white")
    .style("border", "solid")
    .style("border-width", "2px")
    .style("border-radius", "5px")
    .style("padding", "5px")

// Functions for changing the tooltip when user goes over it
var mouseover = function(d) {
    tooltip
        .style("opacity", 1)
}
var mousemove = function(d) {
    tooltip
        .html("Goals scored: " + d.goals)
        .style("left", (d3.mouse(this)[0] + 70) + "px")
        .style("top", (d3.mouse(this)[1] + "px")
}
var mouseleave = function(d) {
    tooltip
        .style("opacity", 0)
}
// Bars
svg.selectAll("bar")
    .data(data)
    .enter()
    .append("rect")
    .attr("x", function(d) { return x(d.player) })
    .attr("y", function(d) { return y(d.goals); })
    .attr("width", x.bandwidth())
    .attr("height", function(d) { return height - y(d.goals); })
    .attr("fill", "#000058")
    .on("mouseover", mouseover)
    .on("mousemove", mousemove)
    .on("mouseleave", mouseleave)
})
});

```



Sl. 2.11. Grafički prikaz postignutih golova po strijelcima u sezoni 2019./20.

Treća u nizu vizualizacija za sezonu 2019./20. grafički je prikaz igrača po broju nastupa u promatranoj sezoni. Igrač koji nastupa na utakmici istu može odigrati ili kao dio početne postave ili kao izmjena, a zbroj ta dva elementa čini ukupan broj nastupa. Za potrebu vizualizacije ove činjenice koristi se složeni stupčasti graf, tzv. *stacked bar chart*, kod kojeg je svaki igrač predstavljen jednim stupcem sastavljenim od dva dijela, pri čemu tamniji dio označava nastupe od početka utakmice, a svjetliji dio nastupe ulaskom u igru nakon izmjene. Skripta *appearances.js* dana je slikom 2.12., a rezultat izvršavanja skripte, odnosno izgled grafa kojim se vizualizira broj nastupa po igračima u sezoni 2019./20., prikazan je slikom 2.13.

```
document.addEventListener('DOMContentLoaded', function(e) {
  // Dimensions and margins of the graph
  var margin = {
    top: 30,
    right: 30,
    bottom: 70,
    left: 60
  },
  width = 1000 - margin.left - margin.right,
  height = 500 - margin.top - margin.bottom;
```

```

// Appending the svg object to the div called "appearances_chart"
var svg = d3.select("#appearances_chart")
    .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 + ")");

// Fetching and parsing the data
d3.csv("../data/appearances.csv", function(data) {

    // Lists of subgroups and players
    var subgroups = data.columns.slice(1)
    var players = d3.map(data, function(d) {
        return (d.player)
    }).keys()

    // X axis
    var x = d3.scaleBand()
        .domain(players)
        .range([0, width])
        .padding([0.2])
    svg.append("g")
        .attr("transform", "translate(0," + height + ")")
        .call(d3.axisBottom(x).tickSizeOuter(0))
        .selectAll("text")
        .attr("transform", "translate(-10,0)rotate(-45)")
        .style("text-anchor", "end");

    // Y axis
    var y = d3.scaleLinear()
        .domain([0, 40])
        .range([height, 0]);
    svg.append("g")
        .call(d3.axisLeft(y));

    // Color scale
    var color = d3.scaleOrdinal()
        .domain(subgroups)
        .range(['#8B0000', '#FF3333'])

    // Stacking the data per subgroups
    var stackedData = d3.stack()
        .keys(subgroups)
        (data)

```

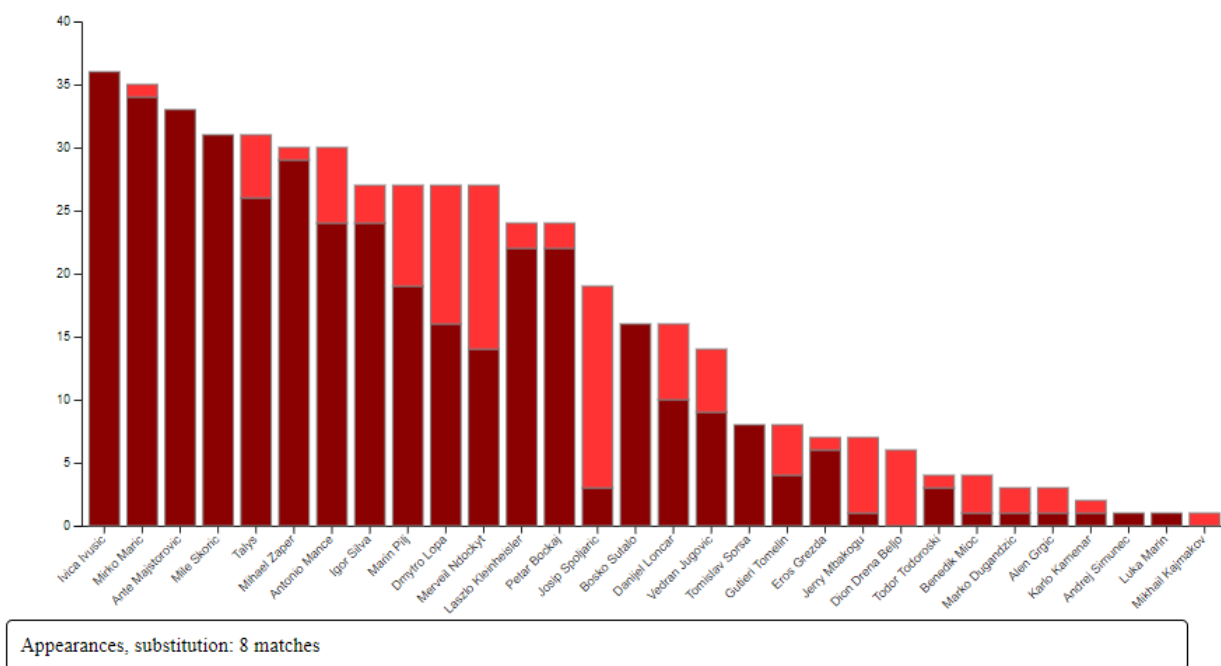
```

var tooltip = d3.select("#appearances_chart") // Tooltip
    .append("div")
    .style("opacity", 0)
    .attr("class", "tooltip")
    .style("background-color", "white")
    .style("border", "solid")
    .style("border-width", "1px")
    .style("border-radius", "5px")
    .style("padding", "10px")

// Functions for changing the tooltip when user goes over it
var mouseover = function(d) {
    var subgroupName = d3.select(this.parentNode).datum().key;
    var subgroupValue = d.data[subgroupName];
    tooltip
        .html("Appearances, " + subgroupName + ": " + subgroupValue + " matches")
        .style("opacity", 1)
}
var mousemove = function(d) {
    tooltip
        .style("left", (d3.mouse(this)[0] + 90) + "px")
        .style("top", (d3.mouse(this)[1] + "px")
}
var mouseleave = function(d) {
    tooltip
        .style("opacity", 0)
}
// Bars
svg.append("g")
    .selectAll("g")
    .data(stackedData)
    .enter().append("g")
    .attr("fill", function(d) { return color(d.key); })
    .selectAll("rect")
    .data(function(d) { return d; })
    .enter().append("rect")
    .attr("x", function(d) { return x(d.data.player); })
    .attr("y", function(d) { return y(d[1]); })
    .attr("height", function(d) { return y(d[0]) - y(d[1]); })
    .attr("width", x.bandwidth())
    .attr("stroke", "grey")
    .on("mouseover", mouseover)
    .on("mousemove", mousemove)
    .on("mouseleave", mouseleave)
})
});

```

Sl. 2.12. Skripta appearances.js



Sl. 2.13. Grafički prikaz nastupa po igračima u sezoni 2019./20.

LITERATURA

- [1] D3.js, *Data-Driven Documents*, <https://d3js.org/>, pristupljeno 9. kolovoza 2020.
- [2] D3.js Graph Gallery, *An introduction to d3.js in 10 basic examples*,
https://www.d3-graph-gallery.com/intro_d3js.html, pristupljeno 9. kolovoza 2020.
- [3] freeCodeCamp, *Learn D3.js in 5 minutes*,
<https://www.freecodecamp.org/news/learn-d3-js-in-5-minutes-c5ec29fb0725/>,
pristupljeno 9. kolovoza 2020.
- [4] D3.js Graph Gallery, *Pie chart with annotation*,
https://www.d3-graph-gallery.com/graph/pie_annotation.html,
pristupljeno 9. kolovoza 2020.
- [5] D3.js Graph Gallery, *Stacked barplot with tooltip*,
https://www.d3-graph-gallery.com/graph/barplot_stacked_hover.html,
pristupljeno 9. kolovoza 2020.