

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

Sveučilišni diplomski studij računarstva

Vizualizacija podataka

Vizualizacija statistike Nobelovih nagrada

Projektni zadatak

Luka Kaučić

Osijek, 2023.

Sadržaj

1	DEFINIRANJE PROJEKTOG ZADATAKA	1
1.1	Projektni zadatak	1
1.2	Podaci i predobrada	1
1.3	Relevantne vrste prikaza za korištene podatke	2
2	DIZAJN VIZUALIZACIJE PODATAKA	3
2.1	Pitanja na koja vizualizacija daje odgovor	3
2.2	Skica vizualizacije podataka	4
2.3	Pregled postojećih rješenja	4
2.4	Boje i podaci	5
3	Izrada vizualizacije podataka	6
3.1	Jednostavne/složene funkcionalnosti i ponašanja	6
3.2	Implementacija programske podrške	8
	Literatura	16

1 DEFINIRANJE PROJEKTOG ZADATAKA

Ovaj projektni zadatak napravljen je u sklopu kolegija Vizualizacija podataka na Sveučilišnom diplomskom studiju računarstva. U nastavku poglavlja dat će se pregled projektnog zadatka, odabranog podatkovnog skupa, kao i postupaka predobrade istog. Na kraju će se po dostupnim podacima i sličnim projektima dostupnima na internetu dati prijedlog vrsta prikaza koje će se u ovom radu koristiti.

1.1 Projektni zadatak

Za temu projektnog zadatka odabrana je vizualizacija statistike Nobelovih nagrada. Potrebno je dati pregled osvojenih Nobelovih nagrada po državama svijeta, na način da se jasno može razlikovati količina osvojenih Nobelovih nagrada po zemljama. Nadalje, potrebno je dati i prikaz statistike po spolovima za odabranu zemlju i kategoriju. Izvorni kod za projekt nalazi se na sljedećoj poveznici: <https://github.com/lkaucic/nobelPrizesD3.git>

1.2 Podaci i predobrada

Podatkovni skup preuzet je s https://www.kaggle.com/datasets/rishidamarla/nobel-prize-winners-19002020?resource=download&select=nobel_prize_by_winner.csv u formatu CSV datoteke, a sadrži sljedeće stupce:

1. Id	8. BornCity	15. Share
2. Firstname	9. DiedCountry	16. Motivation
3. Surname	10. DiedCountryCode	17. Name
4. Born	11. Gender	18. City
5. Died	12. Year	19. Country
6. BornCountry	13. Category	
7. BornCountryCode	14. OverallMotivation	

Prilikom predobrade podataka uvideno je sljedeće:

- dobitnici su muškarci/žene i organizacije; radi jednostavnosti i smislenosti vizualizacije, organizacije se neće razmatrati
- na pojedinim unosima nisu navedeni godina, područje za koje je osoba dobila Nobelovu nagradu te većina ostalih podataka; s obzirom na to, takvi unosi neće se razmatrati
- većina vrijednosti stupca „overallMotivation“ nedostaju pa se taj stupac u cjelosti neće razmatrati
- cilj projekta je prikazati podatke u kontekstu zemalja, drugim riječima relevantni podaci vezani su uz zemlju, spol, godinu i područje dobivene nagrade; s obzirom na to, podaci pripadajućih institucija te određenih osobnih podataka neće se razmatrati
- Korišten standard za notaciju država je alpha-2 (dva slova, npr. GB) dok je u geojson datoteci korišten alpha-3 standard (tri slova, npr. GBR); prema tome, bilo je potrebno prepraviti standard u korištenom podatkovnom skupu (zato što je isto jednostavnije od promjene standarda u geojson datoteci)

Nakon predobrade podaci su svedeni na sljedeće stupce:

- | | |
|--------------------|---------------|
| 1. Firstname | 5. Gender |
| 2. Surname | 6. Year |
| 3. BornCountry | 7. Category |
| 4. BornCountryCode | 8. Motivation |

1.3 Relevantne vrste prikaza za korištene podatke

S obzirom da za svakog dobitnika postoje podaci po zemljama i područjima, moguće je prikazati podatke na karti svijeta. Definiranjem skale boje na istoj moguće je prikazati zemlje s manje dobivenih nagrada (slabiji intenzitet boje) i s više (jači intenzitet boje). Podaci o odnosu osvojenih nagrada po kategorijama ili spolu može se prikazati tortnim dijagramom.

2 DIZAJN VIZUALIZACIJE PODATAKA

U ovom poglavlju istražuju se ključni aspekti dizajna vizualizacije statistike Nobelovih nagrada u D3 JavaScript okruženju. Fokus je usmjeren na postavljanje ciljeva vizualizacije, razvoj skice, pregled postojećih rješenja i odabir boja za optimalnu prezentaciju podataka. Cilj je stvoriti informacijski bogatu i vizualno privlačnu vizualizaciju. Prvo će se baviti pitanjima na koja želi pružiti odgovor putem vizualizacije, kako bi se usmjerila pažnja na relevantne aspekte podataka i korisničke potrebe. Zatim će se istražiti proces razvoja skice vizualizacije, koji omogućuje testiranje ideja i osiguravanje jasne komunikacije informacija o Nobelovim nagradama. Pregled postojećih rješenja pružit će inspiraciju i uvid u najbolje prakse koje se mogu primijeniti. Također će se istražiti važnost odabira boja za poboljšanje čitljivosti, naglašavanje ključnih podataka i stvaranje ugodnog korisničkog iskustva.

2.1 Pitanja na koja vizualizacija daje odgovor

U ovom potpoglavlju istražuju se ključna pitanja na koja se odgovor pruža putem vizualizacije statistike Nobelovih nagrada. Proučavaju se raspodjela dobitnika Nobelove nagrade prema različitim kriterijima te se analiziraju promjene kroz godine i među zemljama. Ovdje su iznesena pitanja koja se istražuju:

- Koja zemlja ima više, a koja ima manje dobitnika Nobelove nagrade po području?
- Kakav je omjer muškaraca i žena koji su dobitnici Nobelove nagrade?
- Kakva je statistika dobivenih Nobelovih nagrada po području za odabranu zemlju?
- Kako se raspodjela dobitnika Nobelovih nagrada mijenjala kroz godine po zemljama?
- Kako se raspodjela dobitnika Nobelovih nagrada mijenjala kroz godine po spolu?

Ova pitanja pružaju uvid u različite aspekte statistike Nobelovih nagrada i omogućuju da se razumije raspodjela nagrada na temelju različitih kriterija. Putem vizualizacije podataka, dobiva se cjelovita slika i bolje razumijevanje važnih trendova i karakteristika povezanih s Nobelovim nagradama.

2.2 Skica vizualizacije podataka

Glavni dio vizualizacije je interaktivna karta svijeta. Korisnik može odabrati za koju će se kategoriju, tj. Nobelovu nagradu prikazati podaci na karti. Prelaskom mišem preko određene zemlje prikazuje se *pop-up* s podatkom o imenu zemlje te broju osvojenih nagrada za odabranu kategoriju. Klikom na isu zemlju pored karte se prikazuje tortni dijagram koji prikazuje raspodjelu osvojenih nagrada po spolu. Uz navedeno radi estetike implementirat će se i tranzicije/animacije.

2.3 Pregled postojećih rješenja

U ovom poglavlju će se pružiti pregled različitih izvora i resursa koji su se bavili vizualizacijom statistike Nobelovih nagrada. Različiti pristupi i perspektive u vizualizaciji podataka o Nobelovim nagradama bit će obuhvaćeni. Specifičnosti će biti obrađene u svakom izvoru. Uključeni su izvori koji sadrže programski kod kako bi omogućili korisnicima da sami implementiraju vizualizacije i istražuju podatke. Na primjer, izvor LAV30 [1] uključuje programski kod koji može biti koristan za implementaciju vizualizacija Nobelovih nagrada u određenom programskom jeziku ili okviru. Primjeri i tehnike vizualizacije podataka o Nobelovim nagradama koristeći Python pruženi su u izvoru abigailchen na RPub [2]. Analiza određenih aspekata statistike Nobelovih nagrada obrađena je u nekim izvorima. Na primjer, raspodjela dobitnika Nobelove nagrade prema spolu i kategorijama istražena je u izvoru stats.areppim.com [3]. Slične informacije pružene su u izvoru statista.com [4] gdje se pružaju vizualni prikazi i statistička analiza zastupljenosti spolova među laureatima Nobelove nagrade. Interaktivne vizualizacije i alati za istraživanje podataka vezanih uz Nobelovu nagradu dostupni su u nekim izvorima. Na primjer, interaktivne vizualizacije koje se fokusiraju na strukturu i povijest svemira pružene su u izvoru nobelprize.org [5], dok izvor acanimal na GitHubu [6] pruža interaktivne vizualizacije i alate za istraživanje podataka o Nobelovim laureatima koristeći d3.js, JavaScript knjižnicu za vizualizaciju podataka. Također, neki izvori se bave specifičnim pitanjima ili temama vezanim uz Nobelovu nagradu. Na primjer, izvor ARomoH na GitHubu [7] pruža vizualizacije podataka o Nobelovim laureatima, dok se izvor Jagadish Katam na LinkedInu [8] bavi vizualizacijom statistike dobitnika Nobelove nagrade prema dobi koristeći R programski jezik. Također, izvor DataCamp [9] pruža vodiče za analizu dobitnika Nobelove nagrade koristeći R. Pregled izvora također uključuje analizu povijesti Nobelove nagrade i njezinih dobitnika. Na primjer, izvor acuriousanimal.com [10]

pruža vizualno istraživanje povijesti Nobelove nagrade. Važno je napomenuti da ovi izvori pružaju širok spektar perspektiva i pristupa u vizualizaciji podataka o Nobelovim nagradama te mogu pružiti korisne uvide u statistiku i trendove povezane s Nobelovom nagradom. Navedeni izvori koriste tipove grafova i vizualizacije slične predloženima u prethodnom poglavlju.

2.4 Boje i podaci

Prilikom vizualizacije koristit će se jedna odabrana paleta boja, kako bi sve vizualizacije bile ujednačene i kako ne bi bilo previše boja, što bi uzrokovalo dodatne nejasnoće (promatrači će bojama pridavati određeno značenje i kontekst, a previše boja bi stvorilo previše nejasnoća i dodanog šuma u informaciji koju želimo prenijeti). Konkretno, s obzirom da je riječ o prepoznatljivom brendu, koristi će se paleta boja koja je tipična za sve vizuale koji sadrže nobelovu nagradu.

3 Izrada vizualizacije podataka

Funkcionalnost se odnosi na mogućnosti ili akcije koje mogu biti izvršene pomoću softverske biblioteke ili okvira kao što je D3.js. Opisuje se što je biblioteka sposobna napraviti ili zadaci koje mogu biti obavljeni. U kontekstu D3.js, funkcionalnost može uključivati učitavanje podataka, manipulaciju podacima, stvaranje vizualizacija, obradu događaja i druge zadatke koji mogu biti postignuti korištenjem mogućnosti i metoda D3.js. S druge strane, ponašanje (behavior) u D3.js odnosi se na konkretne akcije ili odgovore koji su iskazani od strane stvorenih vizualnih elemenata ili komponenti pomoću D3.js-a. Opisuje se kako se elementi ili komponente ponašaju ili interagiraju s korisničkim unosima, događajima ili promjenama podataka. Ponašanje u D3.js odnosi se na to kako vizualizacije reagiraju na korisničke interakcije kao što su događaji mouseover, klikovi, prijelazi ili animacije. U nastavku su izlistane osnovne i napredne funkcionalnosti i ponašanja koje rješenje nudi, kao i implementacija osnovnih funkcionalnosti koje su potrebne za prototip vizualizacije.

3.1 Jednostavne/složene funkcionalnosti i ponašanja

Kroz praćenje kolegija, kao i kroz konzultacije s OpenAI modelom [11], sastavljen je popis osnovnih i naprednih funkcionalnosti i ponašanja implementiranih u ovom radu:

1. Jednostavne (osnovne) funkcionalnosti

- Definiraju se širina i visina kontejnera karte.
- Definira se putanja do GeoJSON datoteke.
- GeoJSON datoteka se učitava.
- CSV podaci se učitavaju i parsiraju.
- Postavlja se kontejner karte.
- Definira se projekcija i generator putanje.
- Stvara se SVG element za kartu.
- Stvara se element za prikazivanje tooltipa.
- Definira se trenutna kategorija.
- Ažuriraju se boje karte na temelju podataka.
- Inicijalno ažuriranje boja karte se poziva.

- Rukuje se događajem promjene odabira kategorije.
- Rukuje se događajem zumiranja.

2. Kompleksne funkcionalnosti

- Podaci se filtriraju na temelju odabrane kategorije.
- Podaci se grupiraju po zemlji i računaju se statistike.
- Stvara se skala boja za kartu.
- Crtaju se karte uz prijelaze.
- Putanja se ažurira uz prijelaz.
- Boja putanje se ažurira uz prijelaz.
- Rukuje se događajem klika za generiranje kružnog dijagrama za odabranu zemlju.
- Generira se kružni dijagram za odabranu zemlju.
- Briše se postojeći kružni dijagram.
- Podaci se filtriraju za odabranu zemlju i kategoriju.
- Podaci se grupiraju po spolu i računaju se statistike.
- Definiraju se dimenzije kružnog dijagrama.
- Stvara se SVG element za kružni dijagram.
- Definira se skala boja za kružni dijagram.
- Generiraju se podaci za kružni dijagram.
- Definira se raspored kružnog dijagrama.
- Generiraju se lukovi kružnog dijagrama.
- Crtaju se dijelovi kružnog dijagrama uz prijelaze.
- Dodaju se oznake na dijelove kružnog dijagrama.
- Dodaje se naslov kružnom dijagramu.

3. Jednostavna (osnovna) ponašanja

- Prikazuje se tooltip pri pomicanju miša iznad putanje.
- Tooltip se skriva kada miš napusti putanju.

4. Kompleksna ponašanja

- Lukovi kružnog dijagrama se interpoliraju radi glatkog prijelaza.

3.2 Implementacija programske podrške

```
1 // Define the width and height of the map container
2 var mapWidth = 1040;
3 var mapHeight = 650;
4 // Define the path to your GeoJSON file
5 var geojsonPath = "world_map.geojson";
6 // Load the GeoJSON file
7 d3.json(geojsonPath).then(function (geojson) {
8     // Load and parse the CSV data
9     d3.csv("manipulated_data.csv").then(function (data) {
10         // Set up the map container
11         var mapContainer = d3.select("#map-container");
12         // Define the projection and path generator
13         var projection = d3.geoMercator()
14             .fitSize([mapWidth, mapHeight], geojson);
15         var pathGenerator = d3.geoPath().projection(projection);
16         // Create SVG element for the map
17         var svg = mapContainer.append("svg")
18             .attr("width", mapWidth)
19             .attr("height", mapHeight);
20         // Create tooltip element
21         var tooltip = d3.select("body").append("div")
22             .attr("class", "tooltip")
23             .style("opacity", 0);
24         // Define the current category
25         var currentCategory = "Category1"; // Set the initial
           category
26         // Update the map colors based on the data
27         function updateMapColors() {
28             // Data manipulation and processing here
```

```
29     var filterCategory = currentCategory;
30
31     // Filter the data based on the chosen category
32     var filteredData = data.filter(function (d) {
33         return d.category === filterCategory;
34     });
35     // Group data by country and calculate statistics (e.g.,
36     count)
37     var countryData = d3.group(filteredData, function (d) {
38         return d.bornCountryCode; });
39     countryData = Array.from(countryData, function ([key,
40         values]) {
41         return {
42             bornCountryCode: key,
43             count: values.length,
44             // Calculate other statistics as needed
45         };
46     });
47
48     // Example color scale
49     var colorScale = d3.scaleSequential(d3.interpolateBlues)
50         .domain([0, d3.max(countryData, function (d) { return d.
51             count; })]);
52
53     // Draw the map
54     svg.selectAll("path")
55         .data(geojson.features)
56         .join(
57             enter => enter.append("path")
58                 .attr("d", pathGenerator)
59                 .style("fill", "gray")
60                 .call(enter => enter.transition()
61                     .duration(800)
62                     .style("fill", function (d) {
```

```

59         var countryCode = d.properties.ISO_A3;
60         var country = countryCode ? countryData.find(
61             function (c) {
62                 return c.bornCountryCode === countryCode;
63             }) : null;
64         var count = country ? country.count : 0;
65         return count > 0 ? colorScale(count) : "gray";
66     })),
67     update => update
68     .call(updatePath)
69     .call(updateColor)
70 )
71 .on("mouseover", function (d) {
72     var countryCode = d.properties.ISO_A3;
73     var country = countryCode ? countryData.find(function (
74         c) {
75             return c.bornCountryCode === countryCode;
76         }) : null;
77     var count = country ? country.count : 0;
78     tooltip.html("Nobel Prizes Won by " + d.properties.
79         ADMIN + ": " + count)
80     .style("left", (d3.event.pageX + 10) + "px")
81     .style("top", (d3.event.pageY + 10) + "px")
82     .style("opacity", 1);
83 })
84 .on("mouseout", function (d) {
85     tooltip.style("opacity", 0);
86 })
87 .on("click", function (d) {
88     var countryCode = d.properties.ISO_A3;
89     var country = countryCode ? countryData.find(function (
90         c) {
91             return c.bornCountryCode === countryCode;
92         }) : null;

```

```
89
90     // Generate pie chart for the clicked country
91     if (country) {
92         generatePieChart(country, filterCategory);
93     }
94     });
95
96     // Path update function with transition
97     function updatePath(path) {
98         path.transition()
99             .duration(800)
100             .attr("d", pathGenerator);
101     }
102
103     // Path color update function with transition
104     function updateColor(path) {
105         path.transition()
106             .duration(800)
107             .style("fill", function (d) {
108                 var countryCode = d.properties.ISO_A3;
109                 var country = countryCode ? countryData.find(function
110                     (c) {
111                         return c.bornCountryCode === countryCode;
112                     }) : null;
113                 var count = country ? country.count : 0;
114                 return count > 0 ? colorScale(count) : "gray";
115             });
116     }
117
118     // Call the initial map colors update
119     updateMapColors();
120
121     // Handle category picker change event
```

```
22     d3.select("#category-picker").on("change", function () {
23         currentCategory = d3.select(this).property("value");
24         updateMapColors();
25     });
26     // Handle zoom event
27     var zoom = d3.zoom()
28         .scaleExtent([1, 8])
29         .on("zoom", zoomed);
30
31     svg.call(zoom);
32     // Zoom event handler function
33     function zoomed() {
34         mapGroup.attr("transform", d3.event.transform);
35     }
36     // Generate pie chart for the selected country
37     function generatePieChart(country, category) {
38         // Clear any existing pie chart
39         d3.select("#popup-content").html("");
40         // Filter the data for the selected country and category
41         var filteredData = data.filter(function (d) {
42             return d.bornCountryCode === country.bornCountryCode && d
43                 .category === category;
44         });
45         // Group data by gender and calculate statistics (e.g.,
46             count)
47         var genderData = d3.group(filteredData, function (d) {
48             return d.gender; });
49         genderData = Array.from(genderData, function ([key, values
50             ] ) {
51             return {
52                 gender: key,
53                 count: values.length,
54                 // Calculate other statistics as needed
55             };
56         });
```

```
52     });
53     // Define pie chart dimensions
54     var pieWidth = 400;
55     var pieHeight = 400;
56     var pieRadius = Math.min(pieWidth, pieHeight) / 2;
57     // Create SVG element for the pie chart
58     var pieSvg = d3.select("#popup-content")
59         .append("svg")
60         .attr("width", pieWidth)
61         .attr("height", pieHeight)
62         .append("g")
63         .attr("transform", "translate(" + pieWidth / 2 + "," +
64             pieHeight / 2 + ")");
65     // Define pie chart color scale
66     var pieColorScale = d3.scaleOrdinal(d3.schemeCategory10);
67
68     // Generate pie chart data
69     var pieData = genderData;
70
71     // Define pie chart layout
72     var pie = d3.pie()
73         .value(function (d) { return d.count; });
74
75     // Generate pie chart arcs
76     var arc = d3.arc()
77         .innerRadius(0)
78         .outerRadius(pieRadius);
79     // Draw pie chart slices
80     var slices = pieSvg.selectAll("path")
81         .data(pie(pieData))
82         .enter()
83         .append("path")
84         .attr("d", arc)
```

```
184     .attr("fill", function (d) { return pieColorScale(d.data.  
185         gender); })  
186     .attr("stroke", "white")  
187     .style("stroke-width", "2px")  
188     .each(function (d) { this._current = d; })  
189     .call(enter => enter.transition()  
190         .duration(800)  
191         .attrTween("d", function (d) {  
192             var interpolate = d3.interpolate({ startAngle: 0,  
193                 endAngle: 0 }, d);  
194             return function (t) {  
195                 return arc(interpolate(t));  
196             };  
197         }));  
198  
199     // Add labels to the pie chart slices  
200     var labels = pieSvg.selectAll("text")  
201     .data(pie(pieData))  
202     .enter()  
203     .append("text")  
204     .attr("transform", function (d) {  
205         var centroid = arc.centroid(d);  
206         return "translate(" + centroid[0] + "," + centroid[1] +  
207             ")";  
208     })  
209     .attr("text-anchor", "middle")  
210     .text(function (d) { return d.data.gender + " (" + d.data  
211         .count + ")"; })  
212     .style("fill", "white")  
213     .style("font-size", "12px")  
214     .style("opacity", 0)  
215     .transition()  
216     .duration(800)  
217     .style("opacity", 1);  
218  
219     // Add a title to the pie chart
```



```
214     pieSvg.append("text")
215         .attr("text-anchor", "middle")
216         .attr("y", -pieHeight / 2)
217         .text("Gender Distribution")
218         .style("font-size", "16px");
219     }
220
221     }).catch(function (error) {
222         console.log("Error loading data:", error);
223     });
224
225     }).catch(function (error) {
226         console.log("Error loading GeoJSON:", error);
227     });
```

Izlistanje 1: JavaScript code snippet

Literatura

- [1] LAV30, *Nobel Prize Data Visualization*, <https://lav30.github.io/nobelprizes/>, Accessed: June 8, 2023.
- [2] abigailchen, *Nobel Prize Data Exploration*, <https://rpubs.com/abigailchen/nobelprize>, Accessed: June 8, 2023.
- [3] areppim, *Nobel Prize Winners by Sex and Category*, https://stats.areppim.com/stats/stats_nobel_sexcat.htm, Accessed: June 8, 2023.
- [4] Statista, *Nobel Prize Winners by Gender*, <https://www.statista.com/chart/2805/nobel-prize-winners-by-gender/>, Accessed: June 8, 2023.
- [5] T. N. Prize, *Interactive Visualisations: The Universe's Structure and History*, <https://www.nobelprize.org/interactive-visualisations-the-universes-structure-and-history/>, Accessed: June 8, 2023.
- [6] acanimal, *Nobel Prize Explorer*, <https://github.com/acanimal/nobel-prize-explorer>, Accessed: June 8, 2023.
- [7] ARomoH, *Data Visualization Nobel Laureates*, <https://github.com/ARomoH/Data-Visualization-Nobel-Laureates>, Accessed: June 8, 2023.
- [8] J. Katam, *Data Visualization of Nobel Prizes Age using R*, <https://www.linkedin.com/pulse/data-visualization-nobel-prizes-age-using-r-kaggle-jagadish-katam/>, Accessed: June 8, 2023.
- [9] DataCamp, *Analyzing Nobel Prize Winners with R*, <https://www.datacamp.com/projects/441>, Accessed: June 8, 2023.
- [10] acuriousanimal, *A Visual Exploration of the Nobel Prize History*, <https://www.acuriousanimal.com/blog/20130815/a-visual-exploration-of-the-nobel-prize-history>, Accessed: June 8, 2023.
- [11] OpenAI, *ChatGPT: OpenAI's Large Language Model*, <https://openai.com/>, 2021.