Vizualizacija podataka u Pythonu

Nina Dobša

Prirodoslovno Matematički Fakultet, Matematički odsjek
Kolegij Poslovna inteligencija
Studeni, 2024.

Sadržaj

Uvod..... **1**
Matplotlib..... **2**
Seaborn..... **3**
Plotly..... **6**
GgPlot / Plotnine..... **9**
Ostale biblioteke..... **10**
Literatura..... **12**

Uvod

Vizualne reprezentacije omogućuju nam da uočimo bitne informacije već u prvim sekundama gledanja. One nam mogu ukazati na važne trendove, nepravilnosti ili povezanosti koje bi nam inače lako promakle. S obzirom na količinu informacija s kojima se svakodnevno susrećemo, brz pristup bitnim podacima od presudne je važnosti.

Python je već dugi niz godina jedan od najkorištenijih programskih jezika u području analize i vizualne reprezentacije podataka. Unutar Pythona razvijeno je mnogo biblioteka za vizualizaciju, a svaka od njih nudi specifične mogućnosti i prilagodbe. U ovom seminaru predstavit ćemo sedam odabranih biblioteka za vizualizaciju podataka – Matplotlib, Seaborn, Plotly, Ggplot, Bokeh, Altair i Geoplotlib. Sve ove biblioteke (osim Geoplotlib), omogućuju prikaz osnovnih grafova poput histograma, stupičastih i linijskih grafova. Međutim, svaka od njih ističe se specifičnim karakteristikama i tipovima vizualizacija, prilagođenim različitim profilima korisnika - od iskusnih programera do onih s manje tehničkog znanja.

Za izradu vizualizacija koristit ćemo poznati skup podataka *Life Expectancy* [1], koji obuhvaća ekonomske i zdravstvene podatke (kao što su: očekivani životni vijek, BDP po stanovniku, stopa konzumacije alkohola, smrtnost dojenčadi, itd…) za 179 država u razdoblju od 15 godina (2000.-2015.). Analiza i vizualizacije izrađene su u programskom jeziku Python, koristeći Jupyter bilježnicu unutar okruženja Visual Studio Code.

Matplotlib

Matplotlib razvio je 2003. godine neurobiolog John D. Hunter kao open-source biblioteku za vizualizaciju podataka u Pythonu. Biblioteka je nastala po uzoru na MATLAB, no za razliku od MATLAB–a, Matplotlib se koristi na objektno-orijentirani način. Jedna od osnovnih filozofija Matplotliba je da korisnici mogu kreirati jednostavne grafove s minimalnim brojem naredbi. Na primjer, za prikaz histograma ili linijskog grafa korisnik ne mora definirati objekte i metode, već se omogućava njihovo brzo kreiranje pozivanjem jednostavnih funkcija i definiranjem x odnosno y osi. Osim jednostavnih grafova, Matplotlib nudi bogat izbor grafova kao što su i 3D prikazi te animacije.

Matplotlib ima mnoge prednosti koje ga čine omiljenim alatom za vizualizaciju podataka u znanstvenim i istraživačkim krugovima. Jedna od njih je prilagođenost drugim bibliotekama u Pythonu kao što su NumPy i Pandas što omogućava rad s velikom količinom podataka u matricama odnosno dataframe-ovima. Osim toga, njegova opsežna dokumentacija i podrška zajednice olakšavaju rješavanje problema kroz mnogo primjera i vodiča. Zbog mogućnostima prilagodbe elemenata poput fonta, boja i veličine slike, ova je biblioteka pogodna za publikacije.

Ipak, Matplotlib ima i nekoliko nedostataka. Kompleksne prilagodbe zahtijevaju više kodiranja i vremena, što bi moglo predstavljati problem početnicima. Što se tiče interaktivnosti, iako biblioteka nudi osnovnu interaktivnost, ona nije interaktivna poput nekih drugih biblioteka koje ćemo spomenuti u nastavku.

```
# !pip install matplotlib
import matplotlib.pyplot as plt
```

✓ 0.0s

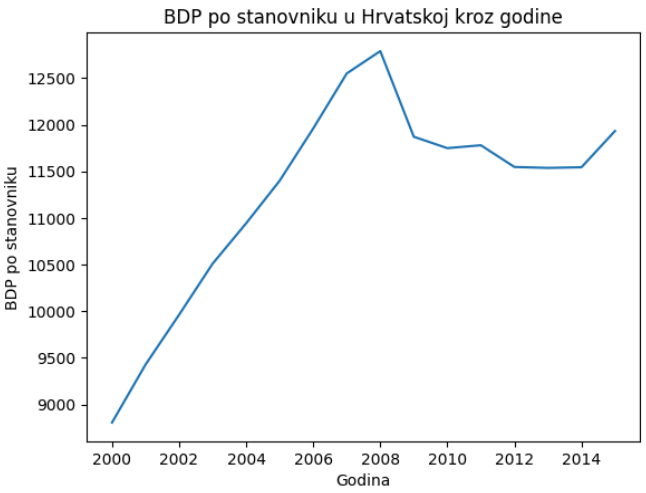
Linijski graf

```
plt.plot(data_CRO['Year'], data_CRO['GDP_per_capita'])

plt.xlabel('Godina')
plt.ylabel('BDP po stanovniku')
plt.title('BDP po stanovniku u Hrvatskoj kroz godine')

plt.show()
```

✓ 0.2s

Slika 1 - instalacija Matplotlib biblioteke te primjer koda

Na slici 1 prikazana je instalacija i uključivanje biblioteke, kao i kod za prikaz linijskog grafa. Ovaj primjer jasno pokazuje jednostavnost korištenja – za prikaz linijskog grafa potrebno je pozvati metodu plot() s argumentima za osi x i y, te koristiti naredbu show() za prikaz grafa u bilježnici. Uz dodatne naredbe za postavljanje naziva osi i naslova grafa, u svega par linija koda generirali smo jednostavnu vizualizaciju.



Slika 2 - Linijski graf (engl. Line plot)

Slika 3 - Tortni dijagram (engl. Pie chart)

Na slikama 2 i 3 prikazani su primjeri linijskog i tortnog grafa. Linijski graf prikazuje BDP po stanovniku u Hrvatskoj kroz godine, dok tortni dijagram prikazuje ekonomski status država u 2015. godini.

Seaborn

Seaborn je besplatna open-source biblioteka koju je razvio 2014. godine neuroznanstvenik Michael Waskom. Biblioteka je dizajnirana kako bi se prevladala ograničenja Matplotliba, koji nije primjeren za kreiranje složenih statističkih grafova. Seaborn nadograđuje mogućnosti Matplotliba i daje intuitivniji i sažetiji način pisanja koda te dodatne značajke koje su posebno osmišljene za vizualizaciju statističkih podataka.

Kako je biblioteka nadogradnja Matplotliba, ona zadržava prednosti Matplotliba kao što je pojednostavljeni rad sa ostalim bibliotekama poput Pandas i NumPy, bogat izbor vizualizacija te jednostavna kreacija grafova bez potrebe za poznavanjem koda. Jedna od ključnih prednosti biblioteke Seaborn, kao što smo spomenuli u prethodnom odlomku, je njegova prilagodba za rad s podacima koji zahtijevaju statističke vizualizacije. Ugrađene funkcije za prikaz distribucije, korelacija i trendova omogućuju korisnicima lakše razumijevanje statističkih svojstava podataka. Seaborn također nudi širu paletu tema i boja što odmah čini vizualizacije privlačnijima i podložnijima personaliziranju. Isto kao i Matplotlib, Seaborn ne nudi visoku razinu interaktivnosti, što ga čini manje pogodnim za vizualizacije koje zahtijevaju aktivnu interakciju s korisnikom.

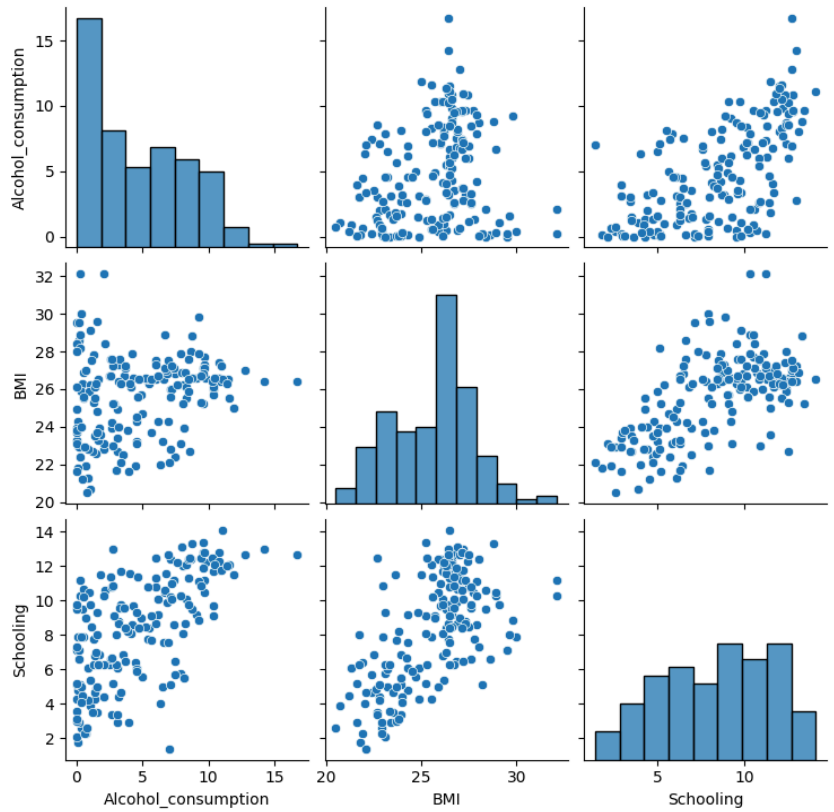
```
# !pip install seaborn
import seaborn as sns
```
✓ 0.0s

Pair Plot - grafovi raspršenosti za svake dvije varijable

```
sns.pairplot(data_2015_only3)
```
✓ 2.1s

Slika 4 - instalacija Seaborn biblioteke te primjer koda

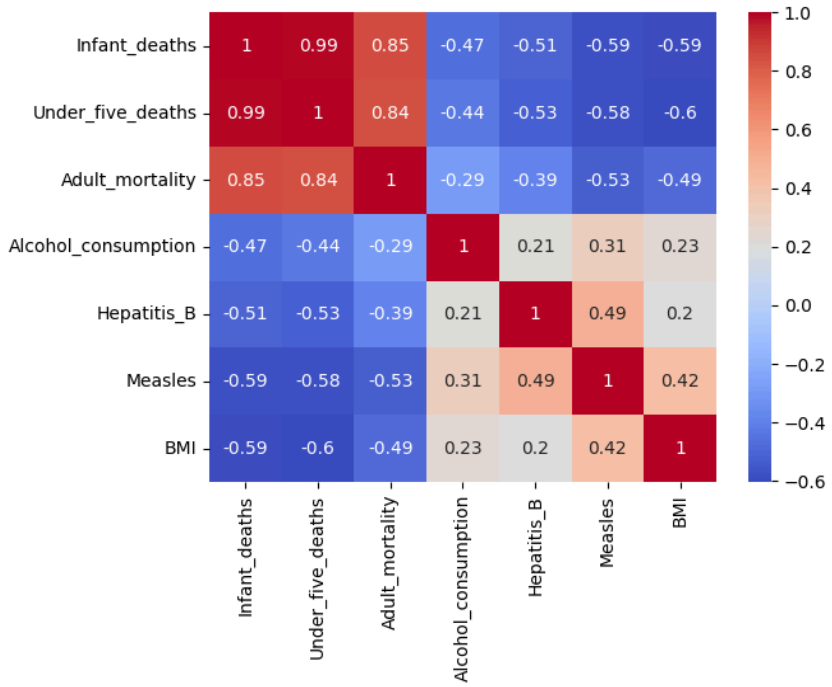
Na slici 4 prikazana je instalacija i uključivanje biblioteke, kao i kod za prikaz grafikona parova. Grafikon parova prikazuje međusobnu ovisnost svakih dviju varijabli u tablici na mjestima van glavne dijagonale te distribuciju pojedinačnih varijabli na glavnoj dijagonali. Za ovaj smo primjer filtrirali samo 3 varijable: stopa konzumacije alkohola, ITM (indeks tjelesne mase) i stopa školovanja za 2015. godinu. Kao i u Matplotlibu, ponovno možemo uočiti jednostavnost korištenja. U samo jednoj liniji koda dobili smo informativan graf na temelju kojeg možemo donositi daljnje odluke.



Slika 5 - Grafikon parova (engl. Pair plot)

Ništa teže nije bilo generirati i toplinski grafikon prikazan na slici 6. Za ovaj primjer filtrirali smo 6 varijabli za 2015. godinu, a graf prikazuje korelacije između različitih

varijabli s time da pozitivne korelacije poprimaju crvenu boju dok negativne korelacije poprimaju plavu boju.



Slika 6 - Toplinski graf (engl. Heatmap)

Plotly

Plotly je besplatna open-source biblioteka za interaktivno prikazivanje podataka koju je razvio Plotly Inc 2013. godine. Izrađena je na temelju Plotly JavaScript biblioteke (plotly.js) i omogućuje kreiranje vizualizacija podataka koje se mogu prikazivati unutar Jupyter bilježnica, web aplikacija pomoću Dash-a (Dash je također open-source python sučelje od Plotly Inc. koje se koristi za izgradnju web-aplikacija) ili se mogu spremati kao pojedinačne HTML datoteke. Plotly nudi više od 40 tipova grafova, a za razliku od većine drugih biblioteka, Plotly omogućuje i izradu kontur grafova (vrsta vizualizacije za prikazivanje trodimenzionalnih podataka u dvodimenzionalnom formatu). Jedna od najbitnijih značajki biblioteke Plotly su interaktivne vizualizacije. Interaktivnost grafova značajno poboljšava korisničko iskustvo i omogućuje dublje razumijevanje podataka što ovu biblioteku svrstava u najkorištenije python biblioteke za kreiranje vizualizacija.

Dakle, kao što je spomenuto u prethodnom odlomku, glavne prednosti ove biblioteke su interaktivnost vizualizacije, moguća integracija s web-aplikacijom i veća ponuda vizualnih prikaza. Plotly također omogućava jednostavan rad sa Pandas i NumPy bibliotekama. Jedan od glavnih nedostataka Plotly biblioteke je njegova složenost pri instalaciji i konfiguraciji - i mi smo imali problema s instalacijom paketa koji bi

omogućio prikazivanje grafova u jupyter bilježnici, a kako bi zaobišli problem spremali smo grafove u obliku html datoteka. Isto tako, kod djeluje kompliciranije i nije toliko intuitivan kao u ranije spomenutim bibliotekama što bi moglo predstavljati problem neiskusnim korisnicima.

```
# !pip install plotly
import plotly.express as px
```

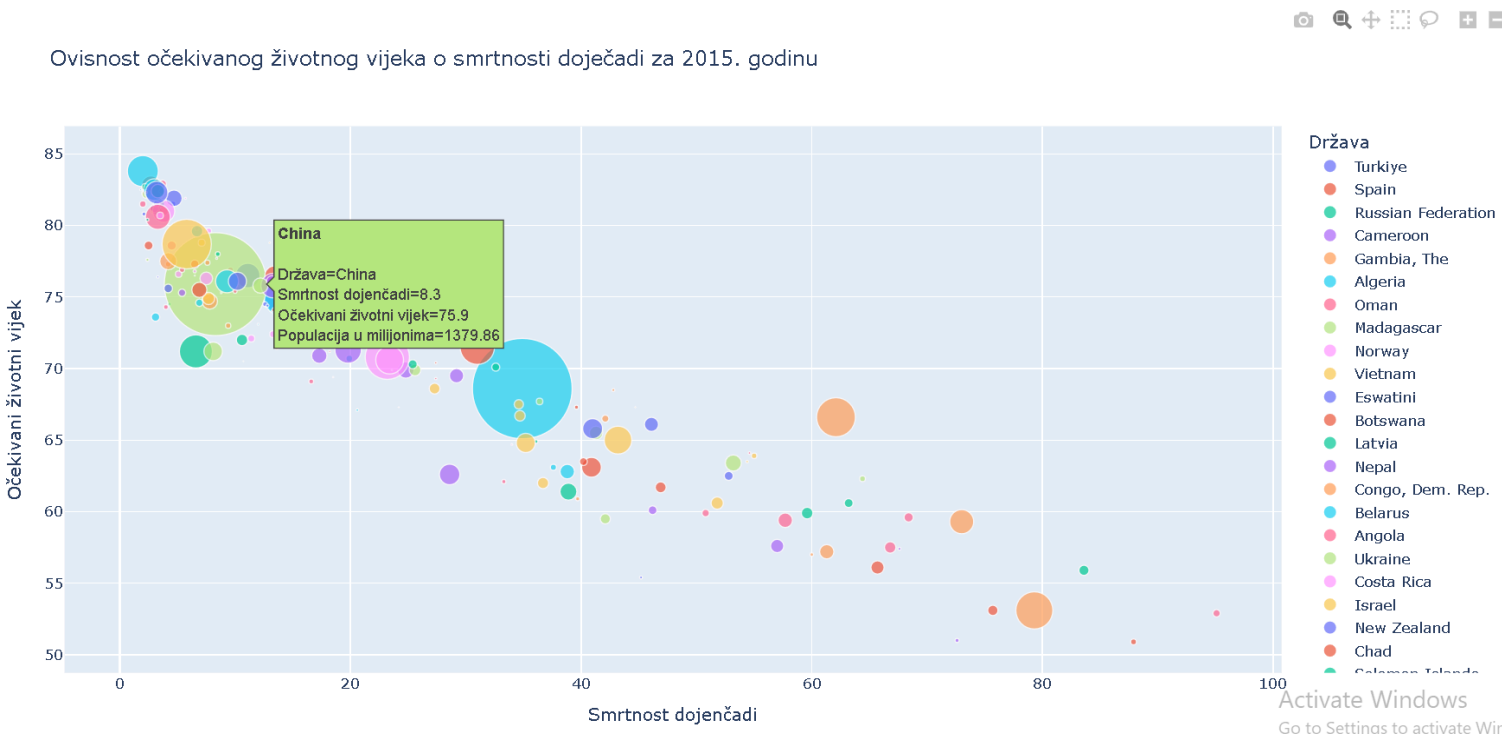
✓ 0.0s

Mjeurićasti dijagram

```
bubble_chart = px.scatter(
    data_2015,                  # skup podataka tipa dataframe
    x="Infant_deaths",          # pridruživanje podataka x-osi
    y="Life_expectancy",        # pridruživanje podataka y-osi
    size="Population_mln",      # veličina točke ovisi o populaciji
    size_max = 60,
    color="Country",            # boja točke prema državi
    hover_name="Country",       # prikaz naziva države pri prelasku mišem
    title="Ovisnost očekivanog životnog vijeka o smrtnosti doječadi za 2015. godinu",
    labels={"Infant_deaths":"Smrtnost dojenčadi", "Country":"Država",
            "Life_expectancy":"Očekivani životni vijek", "Population_mln":"Populacija u milijonima"} # mijenjanje natpisa
)

# dodavanje naziva osi
bubble_chart.update_layout(
    xaxis_title="Smrtnost dojenčadi",
    yaxis_title="Očekivani životni vijek"
)

bubble_chart.write_html("bubble_chart.html")
```

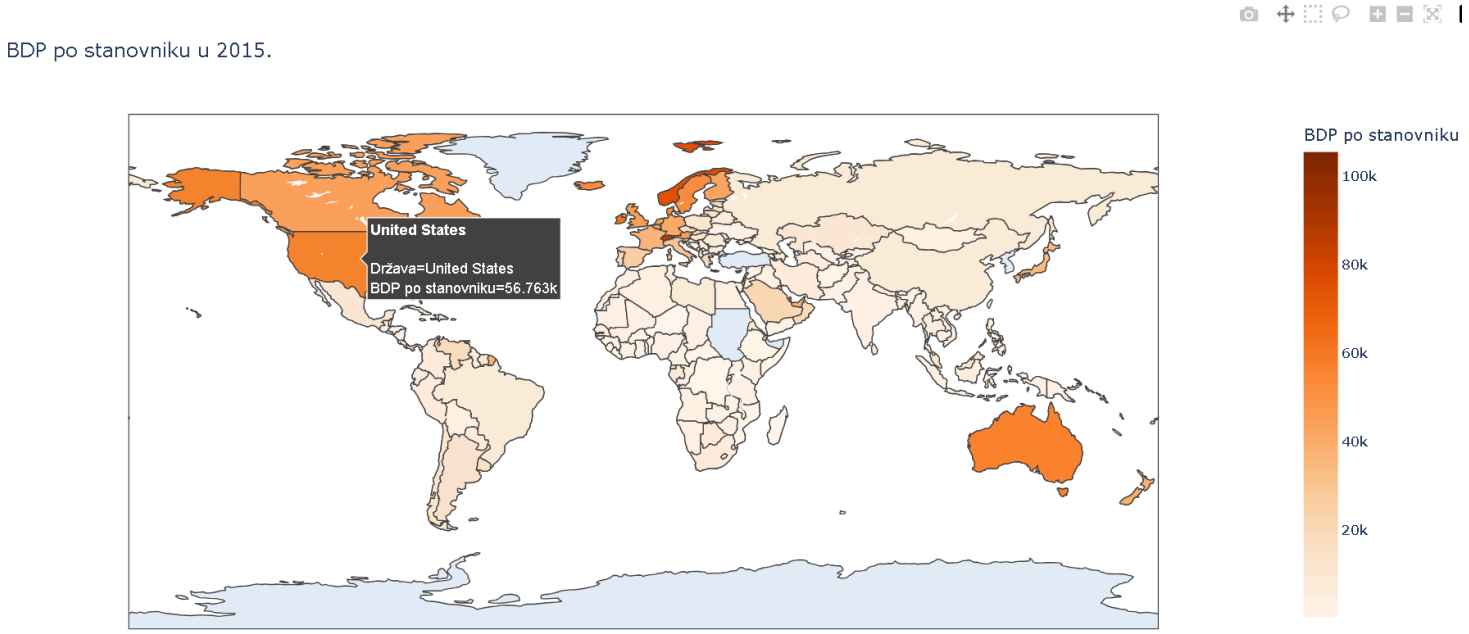
Slika 7 - instalacija Plotly biblioteke te primjer koda

Na slici 7 prikazana je instalacija i aktivacija biblioteke, kao i kod za generiranje mjehurićastog grafikona. Već na prvi pogled možemo uočiti da kod djeluje kompliciranije od kodova u prijašnjim bibliotekama, no isto tako moramo uzeti u obzir da je i graf kompleksniji od običnog linijskog grafa u primjeru biblioteke Matplotlib. Mjehurićasti grafikon, koji je prikazan na slici 8, predstavlja graf raspršenosti koji prikazuje povezanost između očekivanog životnog vijeka i smrtnosti dojenčadi u 2015. godini. Veličine točaka (koje predstavljaju države) na grafu ovise o broju stanovnika, zbog čega će Kina i Indija biti prikazane s najvećim mjehurićima, dok će Hrvatska biti prikazana manjim mjehurićem. Također, omogućena je interakcija, pa će se prilikom prelaska mišem preko određenih točaka (država) pojaviti detaljni podaci o toj državi: naziv države, smrtnost dojenčadi, očekivani životni vijek i broj stanovnika.



Slika 8 - Mjehurićasti grafikon (engl. Bubble chart)

Na slici 9 možete vidjeti primjer koropletne karte, to je vrsta tematske karte na kojoj su različiti dijelovi geografskog područja obojeni prema vrijednostima podataka, u našem slučaju prema vrijednosti BDP-a po stanovniku u 2015. godini. Također smo pridružili interaktivno svojstvo prilikom prelaska mišem kao i u mjehurićastom grafikonu. U desnom gornjem kutu kod oba grafa možete primjetiti opcije zumiranja kao i opcije fokusiranja na određeni dio grafa (engl. focus and context).



Slika 9 - Koropletna karta (engl. Choropleth map)

GgPlot / Plotnine

Ggplot je Python implementacija gramatike grafova temeljena na poznatoj biblioteci ggplot2 u R-u. Vizualizacije se kreiraju po načelu gramatike grafova - gramatika omogućuje strukturiran i intuitivan način izrade vizualizacija specificiranjem različitih slojeva grafa i njihovih estetskih preslikavanja.

Jedna od prednosti ove biblioteke je izrada složenijih grafova s višestrukim slojevima, dodajući geometrije i mapiranja za prikazivanje podataka. Dobro se integrira s ostalim bibliotekama kao što su Pandas i Numpy, a također nudi mnogo paleta i tema koji omogućuju prilagodbu grafa specifičnim zahtjevima.

Iako je gramatika grafova vrlo moćna, njeno učenje i primjena može biti izazovna za početnike jer zahtijeva razumijevanje koncepta slojeva i mapiranja. Drugi nedostatak su performanse, za jako velike skupove podataka Ggplot može biti sporiji u usporedbi s drugim alatima poput Matplotliba ili Plotlyja. Također, baš kao i Matplotlib te Seaborn, Ggplot ne nudi visoku razinu interaktivnosti. No najveći nedostatak ove biblioteke jest što ne nudi podršku kao ranije spomenute biblioteke. Osim što je dokumentacija mnogo oskudnija, sama biblioteka nije bila ažurirana za novije verzije Pythona. Iz tog razloga umjesto biblioteke Ggplot koristit ćemo se bibliotekom Plotnine koja se također temelji na gramatici grafova, a postala je popularna kao zamjena za biblioteku Ggplot jer je bolje održavana i usklađena s novim verzijama Pythona.

```
# !pip install ggplot
# !pip install plotnine
from plotnine import ggplot, aes, geom_line, labs, theme_minimal, ggsave, guides, guide_legend
```

✓ 0.0s

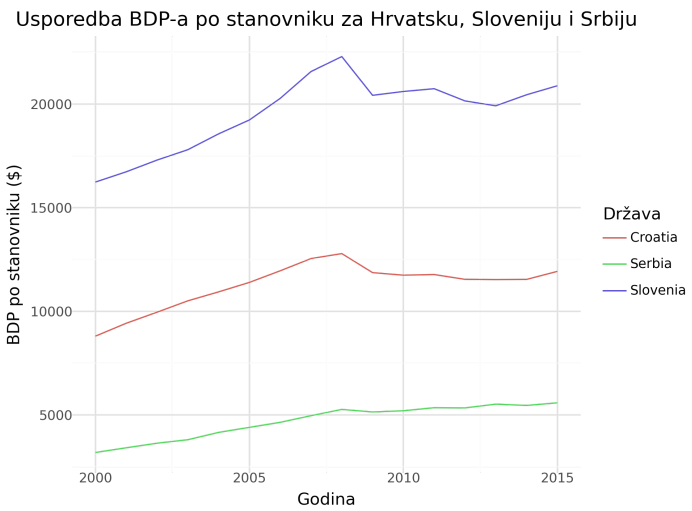
Linijski graf

```
data_jugo = data[data['Country'].isin(['Croatia', 'Slovenia', 'Serbia'])]

plot = ggplot(data_jugo, aes(x='Year', y='GDP_per_capita', color='Country')) + \
       geom_line() + \
       labs(title='Usporedba BDP-a po stanovniku za Hrvatsku, Sloveniju i Srbiju',
            x='Godina',
            y='BDP po stanovniku ($)') + \
       theme_minimal() + \
       guides(color=guide_legend(title='Država'))

ggsave(plot, filename="bdp_jugo.png", dpi=300)
```

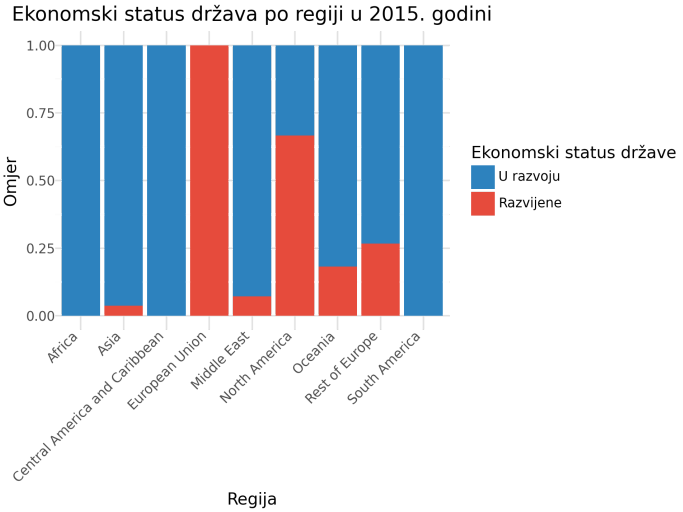
Slika 10 - instalacija Plotnine biblioteke te primjer koda

Na slici 10 prikazana je instalacija i aktivacija biblioteke Plotnine, kao i kod za generiranje linijskog grafa. Svaka od naredbi koja se koristi pri kreiranju grafa

zahtijeva prethodni import iz biblioteke što definitivno povećava kompleksnost rada i zahtijeva dodatnu pažnju kako bi se izbjegle potencijalne greške.



Slika 11 - Linijski graf (engl. Line plot)



Slika 12 -Složeni stupčasti dijagram (engl. stacked bar chart)

Na slici 11 prikazan je linijski graf iz kojeg možemo usporediti kretanje BDP-a po stanovniku u Hrvatskoj, Sloveniji i Srbiji u razdoblju od 2000. do 2015. godine. S druge strane, na slici 12, prikazan je takozvani složeni stupčasti dijagram iz kojeg se lako može iščitati omjer razvijenih država i država u razvoju u svakoj od regija u 2015. godini.

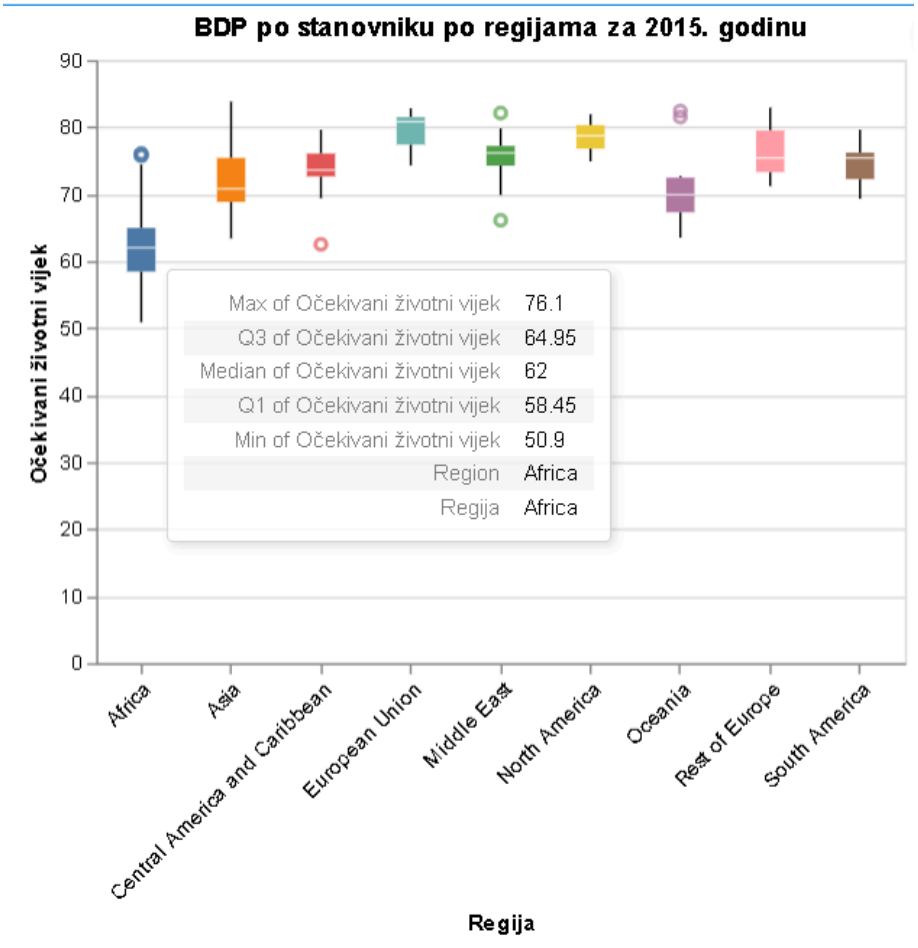
Ostale biblioteke

Neke od bilioteka koje se također često koriste u krugovima analize i vizualizacije podataka su biblioteke Bokeh, Altair i Geoplotlib, ukratko ćemo predstaviti svaku.

Bokeh je Python biblioteka, nastala 2013.godine, za stvaranje kvalitetnih interaktivnih vizualizacija. Grafovi generirani Bokeh-om mogu biti prikazani u web preglednicima, što je njegova glavna prednost, ali mogu biti i spremljeni u html ili json datoteku. Baš kao i Ggplot, Bokeh je također temeljen na gramatici grafova. Biblioteka podržava različite razine sučelja, za jednostavne i složene vizualizacije, a podijeljeno je u dvije razine: bokeh.plotting i bokeh.models. Prva razina, bokeh.plotting, se fokusira na brzo generiranje grafova, a često ju uspoređuju sa funkcionalnostima Matplotliba. S druge strane, napredniji korisnici i stručnjaci mogu izraditi složenije prikaze koristeći bokeh.models za potpunu kontrolu nad grafovima. Nedostaci ove biblioteke su potrebno veće znanje za naprednije prilagodbe i manja korisnička podrška u odnosu na popularnije biblioteke kao što su Matplotlib i Seaborn.

Altair je Python biblioteka za statističke vizualizacije, temeljena na Vega i Vega-Lite deklarativnim jezicima za vizualizaciju podataka. Dizajnirana je za jednostavno i učinkovito stvaranje složenih statističkih grafova. Korištenjem deklarativne sintakse, Altair omogućava korisnicima definiranje izgleda vizualizacije bez potrebe za detaljnim opisom konstrukcije, čime se olakšava čitljivost i održavanje koda. Prednosti Altaira su dobar rad sa složenim transformacijama podataka, mogućnost stvaranja informativnih i estetski privlačnih grafova te jednostavna integracija sa bibliotekom Pandas i njezinim tipovima podataka.

Geoplotlib je Python biblioteka specijalizirana za vizualizaciju geografskih podataka, a nastala je zbog ograničene podršku za karte u većini drugih biblioteka. Omogućuje jednostavno kreiranje različitih vrsta geografskih vizualizacija, kao što su karta gustoće točaka (engl. dot-density), koropletna karta (engl. choropleth maps) i simboličke karte (engl. symbol maps), čime postaje ključan alat za prikaz prostornih podataka. Geoplotlib ima nekoliko nedostataka. Podrška za različite tipove podataka nije toliko široka kao kod drugih biblioteka, zajednica korisnika je manja što znači da je i korisnička podrška slabija te ne radi najbolje sa vrlo velikih skupova podataka.



slika 13 - Dijagram pravokutnika (engl. Box plot)

Na slici 13 možete vidjeti graf generiran bibliotekom Altair. Višestruki dijagrami pravokutnika prikazuju očekivani životni vijek u 2015. godini za svaku od regija. Ovakav prikaz nam omogućuje lakšu usporedbu među regijama i dobra je osnova da daljnju analizu. Također, bez ikakvih dodavanja funkcionalnosti, grafu je pridružena i interaktivna komponenta - prozor s informacijama za svaku od regija prilikom prelaska mišem preko određenog dijagrama pravokutnika.

Zaključak

Ne postoji univerzalna Python biblioteka za koju bismo mogli reći da je najbolja i koja će zadovoljiti potrebe svakog korisnika. Svaka se biblioteka ističe na drugačiji način. Po popularnosti i visokoj razini potpore istaknuli bismo Matplotlib, Seaborn i Plotly. Matplotlib je često prvi izbor za jednostavne grafove zbog svoje jednostavnosti koja omogućuje brzu izradu osnovnih vizualizacija. Za atraktivniji dizajn grafova prednost ćemo dati bibliotekama poput Seaborn ili Plotly, one nude velik broj tema i paleta boja. Ako je cilj dublja statistička vizualizacija, Seaborn i Altair nude odlične mogućnosti za analizu i prikaz podataka u svega par linija koda. Za interaktivne grafikone, najbolji izbor su Plotly i Bokeh jer omogućuju dinamičnu interakciju s podacima. S obzirom na širok raspon alata, zaključujemo da je Python izvrstan odabir za vizualizaciju podataka.

Literatura

1. Life expectancy dataset (WHO), https://www.kaggle.com/datasets/lashagoch/life-expectancy-who-updated, pristupljeno u studenom 2024.
2. Top python libraries for data visualization https://www.geeksforgeeks.org/top-python-libraries-for-data-visualization/, pristupljeno u studenom 2024.
3. Top python libraries for data visualization https://www.analyticsvidhya.com/blog/2024/05/top-python-libraries-for-data-visualization/, pristupljeno u studenom 2024.
4. Matplolib, https://matplotlib.org/, pristupljeno u studenom 2024.
5. Seaborn, https://seaborn.pydata.org/index.html, pristupljeno u studenom 2024.
6. Plotly, https://plotly.com/, pristupljeno u studenom 2024.