Usporedba biblioteka za vizualizaciju podataka u Pythonu

Jurić, Simona

Sveučilište Jurja Dobrile u Puli

Fakultet Informatike u Puli

SIMONA JURIĆ

USPOREDBA BIBLIOTEKA ZA VIZUALIZACIJU PODATAKA U PYTHONU

Završni rad

Pula, 2019.

Sveučilište Jurja Dobrile u Puli

Fakultet Informatike u Puli

SIMONA JURIĆ

USPOREDBA BIBLIOTEKA ZA VIZUALIZACIJU PODATAKA U PYTHONU

Završni rad

JMBAG: 030308604, redovita studentica

Kolegij: Modeliranje i simulacije

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentor: doc. Dr. sc. Darko Etinger

Pula, 2019.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani _____, kandidat za prvostupnika _____ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____

U Puli, _____, _____ godine

IZJAVA
o korištenju autorskog djela

Ja, _____ dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom _____

_____

_____ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____ (datum)

Potpis

_____

SADRŽAJ

1. SAŽETAK I KLJUČNE RIJEČI

SAŽETAK:

Svakodnevno se susrećemo sa raznim podacima i statistikama, međutim ponekad ti podaci sami po sebi nisu sasvim jasni. Kako bi se olakšalo osobama koje rade s tim podacima odnosno kako bi uopće ti podaci mogli biti razumljivi, poželjno bi ih bilo vizualizirati. Vizualizacija podataka je bitan čimbenik u radu sa podacima te ju je moguće ostvariti kroz razne programe i alate. U ovom slučaju podaci će biti vizualizirani pomoću programskog jezika Python te njegovih biblioteka. On sadrži mnoštvo biblioteka od kojih je jedan dio namijenjen simulaciji i vizualizaciji podataka koji na lagan i interaktivan način približava korisnicima rad sa podacima.

KLJUČNE RIJEČI: Python, Vizualizacija podataka, biblioteke, usporedba biblioteka, podaci, Matplotlib, Seaborn, Ggplot, Bokeh, Plotly

ABSTRACT:

We encounter various data and statistics on a daily basis, but sometimes these data alone will not be quite clear. In order to make it easier for people who work with this information, or to make the data understandable at all, it is advisable to visualize it. Data visualization is an important factor in working with data and can be achieved through various programs and tools. In this case, the data will be visualized using the Python programming language and its libraries. It contains many libraries, one of which is intended for simulation and visualization of data, that brings users closer to working with data in a light and interactive way.

KEYWORDS: Python, Data Visualization, Libraries, Library Comparison, Data, Matplotlib, Seaborn, Ggplot, Bokeh, Plotly

2. UVOD

Kroz vrijeme razvijala se potreba za novim metodama prikaza podataka koliko u poslovanju toliko i u opće važnim temama. Nekada su se koristili ručno nacrtanim vizualizacijama, raznolikim grafikonima, dijagramima te ostalim. Samo neki od poznatijih primjera jesu vizualizacija Johna Snowa s prikazom Broad Street-a i izbijanja kolere u Londonu 1854. godine. On se koristio mapom točaka kako bi ilustrirao klaster slučaja oko žarišta izbijanja. Uz to se koristio i statistikom kako bi mogao odrediti povezanost između kvalitete vode sa uzrokom kolere. Tu možemo povući paralele i s današnjim izvođenjem vizualizacija koje se često koriste statistikom.
Osim navedenog primjera, tu je i slučaj s Florence Nightingale koja je izradila dijagram polarnog područja koji prikazuje uzroke smrtnosti u vojsci tijekom godina dok je ona djelovala kao medicinska sestra. Njena vizualizacija prati nekoliko atributa te nije bilo jednostavno izrada ,ali je vrlo jasna. Ovi primjeri u svakom slučaju mogu poslužiti kao motivacija za rad sa vizualizacijom i analizom podataka kako bi se na jednostavan i interaktivan način mogli iščitavati podaci koje posjedujemo.

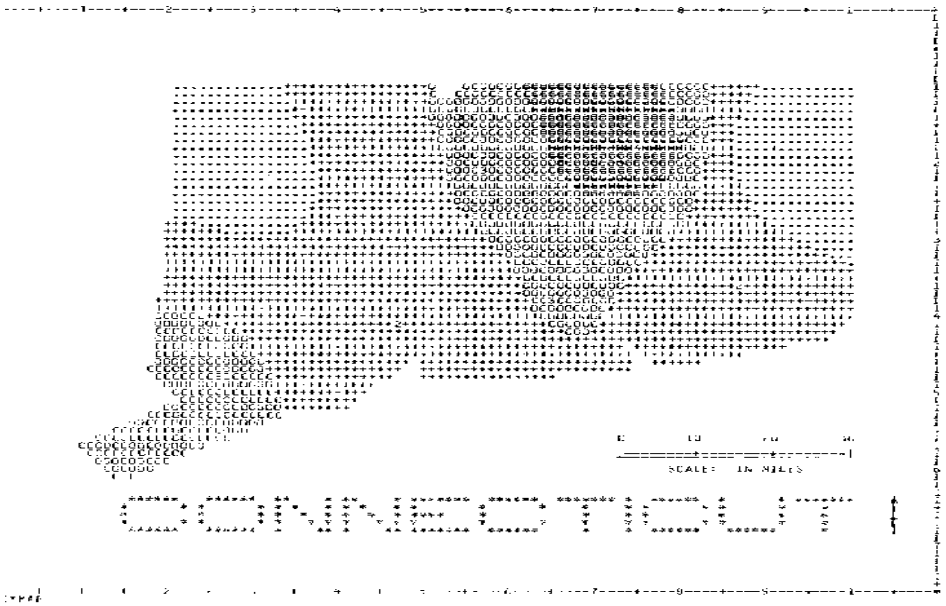
Postoje razni programi i programski jezici te alati koji omogućuju vizualizaciju podataka međutim među poznatijima su R i Python. R je napravljen sa ciljem i osnovnom zadaćom za analizu i prikaz podataka dok se je Python približio vizualizaciji izgradnjom i uporabom biblioteka i alata za vizualizaciju podataka. Iako je sad već veliki broj biblioteka, ili drugim riječima knjižnica koje služe za vizualizaciju kao što su Matplotlib, Ggplot, Bokeh, Seaborn, Plotly, Leather, Geoplotlib, Pygal, Missingo te ostale, u nastavku će biti obrađene samo prvih pet nabrojanih. One su najkorištenije i najpoznatije stoga će biti analizirane i uspoređene jedna s drugom.

3. VIZUALIZACIJA PODATAKA

3.1. Vizualizacija

Kao što je u samom uvodu spomenuto, potreba za vizualizacijom podataka postoji dugi niz godina. U prošlosti se vizualizacija podataka uglavnom odnosila na crtanje zemljopisnih karata, grafikona, dijagrama itd. Tako prikupljeni i sustavno prikazani podaci služili su za jasniji i detaljniji prikaz. Naše oči naviknute su na boje i oblike zbog čega su vizualizirani podaci bolji izbor od brojnih stupaca i redaka podataka. Vizualizacija je oblik vizualne umjetnosti koji povećava i zadržava zainteresiranost promatrača. Dobrom vizualizacijom podataka u mogućnosti smo prikazati priču, ističući korisne, a zanemarujući nebitne informacije. Međutim nije ju uvijek lako izvesti, vrlo je bitno dobro baratati znanjem kako bi podaci i tip vizualizacije bili kompatibilni te kako bi u krajnjem prikazu bili jasni.(https://www.tableau.com/learn/articles/data-visualization)

Jedna od prvih računalnih vizualizacija podataka potječe još iz šezdesetih godina prošlog stoljeća, a predstavlja kartu savezne države Connecticut. Generirana je MIT-ovim (Massachusetts Institute of Technology) SYMAP programom kojeg je osmislio Howard Fisher. (https://towardsdatascience.com/the-first-computer-visualization-3d00dc8c9aea)



Vizualizacije podataka često su najbolji pristup jer omogućuju razmjenu velikih količina podataka u malim prostorima. Dok je na analizu velikih skupova podataka potrebno potrošiti sate, dane ili tjedne, vizualizacije omogućuju brzo i učinkovito tumačenje podataka. Zbog toga je vizualizacija potrebna za gotovo svaku djelatnost koja se koristi podacima.

“Jedan od uobičajenih izazova u sklopu poslovnog obavještavanja jest količina. Morate dubinski razumjeti podatke da biste vidjeli da vizualizacije vode do odluka. Bez konteksta vizualizacije nisu pretjerano učinkovite.
No rješenje je prilično jednostavno: neka alati rade svoj posao, a zaposlenici svoj. Ako koristite odgovarajuće alate i ako osobe koje analiziraju podatke dubinski razumiju odakle podaci potječu, tko će ih koristiti i kako će se tumačiti, bit će očitije kako vizualizacija podataka pojednostavnjuje donošenje velikih odluka.
Svakodnevno sve više tvrtki otkriva važnost vizualizacije podataka u poslovnom obavještavanju. Analitički alati vrhunskih performansi omogućuju analizu podataka na bolji način i brže nego ikad prije, pa je za konkurentnost tvrtke osim mogućnosti prikaza podataka na smislen način nužno i znati što poduzeti na temelju njih.” (products.office.com, 2019.)

3.2. Podaci

Kako bi se neobrađene podatke pretvorilo u odluke, najprije mora biti ispunjen uvjet razumijevanja tih podataka. Vizualizacija podataka je jedan od najkorisnijih alata za razumijevanje poslovnog obavještavanja.

Razvitkom tehnologije omogućena je obrada sve većih količina podataka još većom brzinom. Svi oni podaci koji bi mogli ostati neotkriveni u tekstu, sada su u mogućnosti biti brzo otkriveni i obrađeni pomoću programa za vizualizaciju podataka.

Podatak je jednostavna neobrađena činjenica koja ima značenje i svrhu. Podaci predstavljaju činjenice i pojmove kroz znakovni prikaz. Podatak je uvijek nematerijalne prirode,

ne postoji u stvarnosti već samo u našim mislima te se uvijek pridružuje značenju kojim opisujemo svojstva objekata.

Postoji više oblika koje podaci mogu poprimiti ,ali za svrhu vizualizacije podatka uglavnom se koriste numerički podaci. Oni su pogodni za obradu ili interpretaciju, a mogu biti diskretni ili kontinuirani. Diskretni su podaci oni koji mogu poprimiti samo određene vrijednosti iz nekog skupa i to su cijeli brojevi. S druge strane su kontinuirani podaci koji mogu poprimiti bilo koju vrijednost iz nekog skupa odnosno intervala vrijednosti.

Podaci su glavni resurs koji čini vizualizaciju, određeni skup podataka mora postojati ukoliko korisnik ili analitičar želi primijeniti analizu i vizualizaciju,.

(MathIsFun, n.d.)

3.3. Estetska obilježja

Estetska obilježja su način prikaza podataka na grafu. Estetska obilježja se razlikuju ovisno o vrsti ili tipu grafa koji se odabire za prikaz podataka, te ovisno o biblioteci koju koristimo. Iako različite vrste grafova ili slojeva dijele ista ili slična obilježja, ona svejednp imaju određene karakteristike koje pripadaju samo tom određenom grafu.

Najčešća estetska obilježja :

- x – podaci prikazani na x osi
- y – podaci prikazanih na y osi
- color ili fill – boja, niz boja ili diskretnog skupa podataka za vanjsko ispunjenje određenih točaka/linija
- size – veličina određenih točaka ili linija
- alpha – razine transparentnosti određenih točaka ili linija
- linetype – tip linije, npr. ravna, isprekidana…

- labels – prikaz točnom brojkom veličine podatka

  (http://rstudio-pubs-static.s3.amazo-naws.com/487485_306c20fd1f434b11b492a651fbd24921.html)

4. PYTHON BIBLIOTEKE

4.1. Povijest Pythona

Python je programski jezik kojeg je 1990. godine razvio Guido van Rossum uz doprinos mnogih drugih kolega. Nastao je u istraživačkom institutu u Amsterdamu – „Centrum Wiskunde & Informatica“. Napisan je u programskom jeziku C no zapravo je sljedbenik jezika „ABC“ nastalog u istom istraživačkom centru. Od njegovog osnutka tim za razvoj je djelovao unutar nekoliko nekoliko timova te organizacija ,a od 2001. sve do sada djeluje po vodstvom organizacije „The Python Software Foundation“. Inspiracija za naziv ovog programskog jezika je proizašla kada je njen tvorac čitao objavljene skripte engleske humoristične serije „Monty Python's Flying Circus“.

Od nastanka Pythona, pa sve do danas njegova popularnost sve više raste te se u 2019. godini nalazi na trećem mjestu najboljih programskih jezika. Navedenu informaciju iznosi TIOBE. TIOBE je organizacija specijalizirana za procjenu i praćenje kvalitete softvera, a u nastavku se nalazi njihova tablica najboljih programskih jezika kroz niz godina.

„Nakon prvih verzija 2000. godine izdan je je Python 2.0 čija je posljednja podverzija 2.7 objavljena 2010. godine. Usporedno s verzijom 2, krajem 2008. godine objavljen je Python 3.0. on je unio značajne promjene u sintaksi jezika, tako da se u njemu često ne mogu pokretati programi pisani za starije verzije. Za potrebe lakšeg prelaska razvijen je alat 2to3 koji automatski prevodi kod pisan u verziji 2.x u kod za verziju 3.x.“ (Kalafatić, Pošćić, Šegvić i Šribar, 2016., Predgovor)

Programming Language	2019	2014	2009	2004	1999	1994	1989
Java	1	2	1	1	13	-	-
C	2	1	2	2	1	1	1
Python	3	7	5	7	24	21	-
C++	4	4	3	3	2	2	2
Visual Basic .NET	5	10	-	-	-	-	-
C#	6	5	6	6	19	-	-
JavaScript	7	8	8	8	17	-	-
PHP	8	6	4	5	-	-	-
SQL	9	-	-	89	-	-	-
Objective-C	10	3	32	39	-	-	-
Perl	16	11	7	4	3	10	22
Lisp	32	13	20	13	11	5	3
Pascal	218	16	14	88	6	3	20

SLIKA 1 TABLICA NAJPOPULARNIJIH PROGRAMSKIH JEZIKA (TIOBE, 2019.)

4.1.1. Značajke Pythona

On je objektno orijentirani, interaktivni, skriptni jezik koji je gotovo najkorišteniji programski jezik današnjice. Ne uvjetuje korištenje isključivo objektno-orijentirane paradigme kao kod drugih jezika iste karakteristike. Osobito je korišten kada su u pitanju simulacije, izrada modela te ostale statističke obrade, analize i vizualizacije. „Python je platformski neovisan jezik: većina programa može se izvoditi na Windowsima, n Linuxu, na Mac OS X-u, kao i na brojim manje popularnim platformama. U praksi , ono je prvenstveno moguće zbog vrlo opsežne standardne biblioteke koja pruža platfomski specifičnu funkcionalnost iza transparentnog platformski neovisnog sučelja. Standardna biblioteka omogućava obavljanje složenih zadataka poput preuzimanja mrežne stranice ili komprimiranja i enkripcije podataka, i sve to u nekoliko redaka izvornog koda. Dodatno, postoje tisuće nezavisnih biblioteka, koje u odnosu na standardnu biblioteku nude specijalizirane mogućnosti poput matričnih operacija ili detekcije lica u slikama[5].“ (Kalafatić, Pošćić, Šegvić i Šribar, 2016., str. 5.)

[5] *Važnije nezavisne biblioteke predstavljene su na mrežnim stranicama http://pypi.python.org.*

Važno bi još bilo naglasiti kako se za Python smatra da znatno bolje skalira u odnosu na druge programske jezike visoke razine. Što bi značilo da se jednako lijepo se pišu i skripte od kratkog koda i složeni programski sustavi sastavljeni od više od tisuću linija izvornog koda

4.1.2. Zajednice i konferencije

PyData pruža forum za međunarodnu zajednicu korisnika i razvojnih alata za analizu podataka za razmjenu ideja i učenje jednih od drugih. Globalna PyData mreža promiče raspravu o najboljim praksama, novim pristupima i novim tehnologijama za upravljanje podacima, obradu, analitiku i vizualizaciju. PyData zajednice pristupaju znanosti o podacima pomoću više jezika, uključujući (ali ne ograničavajući se na) Python, Julia i R.

Godišnja konferencija SciPy omogućuje sudionicima iz akademskih, komercijalnih i vladinih organizacija predstaviti svoje najnovije znanstvene projekte Python, učiti od iskusnih korisnika i razvojnih programera i surađivati na razvoju koda. Konferencije se uglavnom sastoje od višednevnih tutorijala nakon kojih slijede dva do tri dana prezentacija, a završava se 1-2 dana, a programer ubrzava projekte zainteresirane za polaznike

PyData je obrazovni program NumFOCUS-a, neprofitne dobrotvorne organizacije koja promiče otvorene prakse u istraživanju, podacima i znanstvenom računalstvu. On služi kao posrednička organizacija, odnosno kao posrednik koji financijski potpomaže manje developere dajući im priliku za razvoj.

NumFOCUS kao organizacija nudi mogućnost usavršavanja te nudi stipedije i potpomaže razvoj dokumentacije svima koji su željni napredovati u ovom području. NumFOCUS je bitan za ovu temu budući da su iz njegovog rada iznikli mnogi alati i biblioteke poput NumPy, SciPy, Matplotlib, IPython i drugi.

4.2. Matplotlib biblioteka

Matplotlib je osmislio John Hunter (1968.-2012.). On je zajedno sa svojim brojnim kolegama, radio na realizaciji ove biblioteke za koju je utrošena velika količina vremena i truda. Njegova kvaliteta je vidljiva u tome da ga sada koriste tisuće znanstvenika širom svijeta.

Ova biblioteka iliti knjižnica je svojim dizajnom namijenjena za laku ,ali vrlo moćnu vizualizaciju podataka. Namijenjena je vizualizaciji 2D sadržaja, i prva je od knjižnica sa tom svrhom. Ona postoji više od 14 godina te je najraširenija knjižnica za planiranje u Python zajednici. Budući da je jednostavna za korištenje, nudi dovoljno mogućnosti za podešavanje načina prikaza podataka. Na temelju njega izgrađene su mnoge biblioteke, a jedna od njih je i Seaborn. One su osmišljene da rade zajedno s analizom, a knjižnice poput Pandas-a su omotači koji omogućuju pristup brojnim Matplotlibovim metodama uz korištenje manje količine koda.

Sadrži širok raspon grafova, kao što su histogrami koji se mogu iscrtati pomoću nje. Periodično izlaze nove verzije ,a posljednja stabilna verzija je 3.1.1. iz 2019. godine. Matplotlib ima vrlo sličnu sintaksu alatu Matlab te njemu kao i Mathematici čini veliku konkurenciju.

Matplotlib se može koristiti u Python skriptama, Python i IPython ljuskama, Jupyter prijenosnom računalu, web aplikacijskim poslužiteljima i četiri grafička korisnička sučelja.

Raznovrsnost Matplotliba može se koristiti za izradu mnogih vrsta vizualizacije, a samo neke od njih su:

- Stupčasti grafikoni i histogrami
- Linijski crteži
- Okrugli grafikoni
- Dijagrami pogrešaka
- Spektri snage
- Spektrogrami

Mana na koju se može naići koristeći Matplotlib je ta što je potrebno napisati više linija koda, a time i više truda kako bi se došlo da naprednijih simulacija i vizualizacija.

Vrlina koju posjeduje je ta da ima opsežnu dokumentaciju što olakšava učenje i rad iskusnim korisnicima ,ali i početnicima u području analize, obrade i vizualizacije podataka.

Najviša razina predstavljena je funkcionalnim sučeljem nazvanim **matplotlib.pyplot** , koje omogućuje korisnicima stvaranje složenih infografika sa samo nekoliko linija koda odabirom gotovih rješenja iz funkcija koje nudi sučelje.

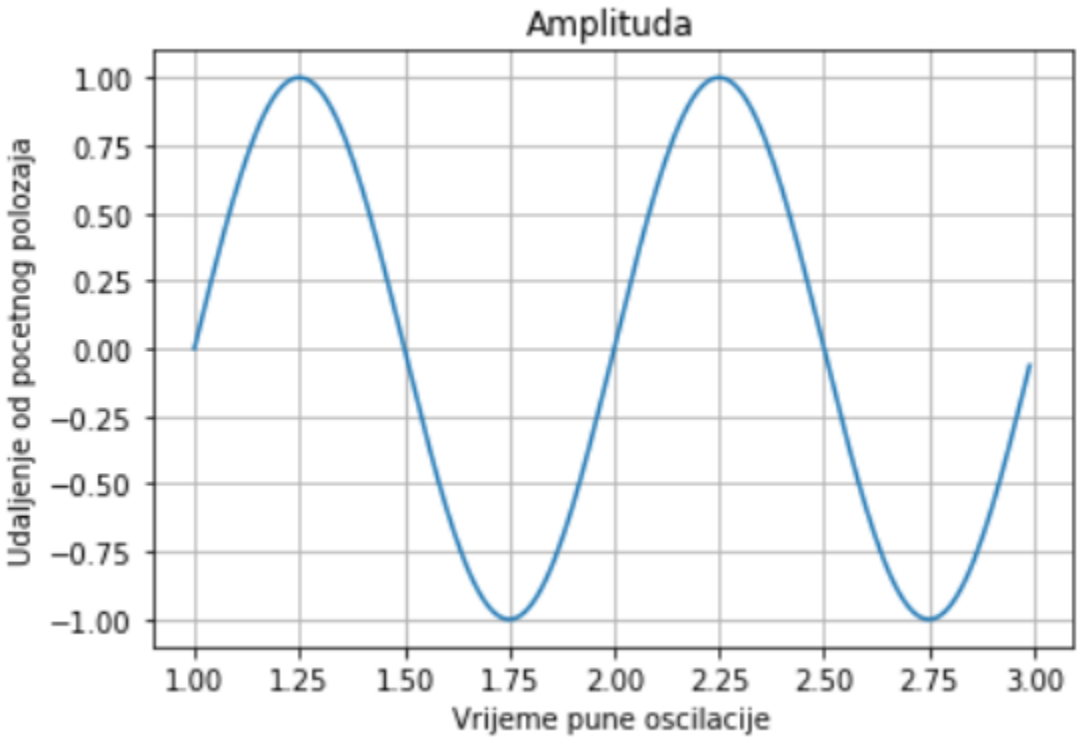
U nastavku će biti prikazana dva primjera. Jedan je jednostavniji te nije preuzet ni od koga, a drugi je preuzet sa službenih stranica Matplotliba te sadrži kompleksniji kod.

```
import matplotlib.pyplot as Plt

import numpy as np

t = np.arange(1.0,3.0,0.01)
s = 0 + np.sin(2*np.pi*t)
plt.plot(t, s)

plt.xlabel('Vrijeme pune oscilacije')
plt.ylabel('Udaljenje od pocetnog polozaja')
plt.title('Amplituda')
plt.grid(True)
plt.legend("test.png")
plt.show()
```

SLIKA 2 PRIMJER KODA ZA MATPLOTLIB

Izvor: vlastiti rad

Na početku koda uvozimo pakete koji su nam važni za potrebe zadatka. Zatim smo inicirali varijablu t, ona označava raspon brojki na y traci. Varijabla s označava izgled amplitude. Plt.xlabel i plt.ylabel služe samo kako bismo unijeli nazive osi, kao što je plt.title za naziv cijelog grafikona. Otvorena nam je opcija postavljanja pozadinskih rešetki te naziva slike odnosno grafikona koji će se prikazati. Na kraju biramo opciju plt.show() kako bi se prikazao željeni graf.



SLIKA 3 PRIKAZ GRAFA ZA MATPLOTLIB

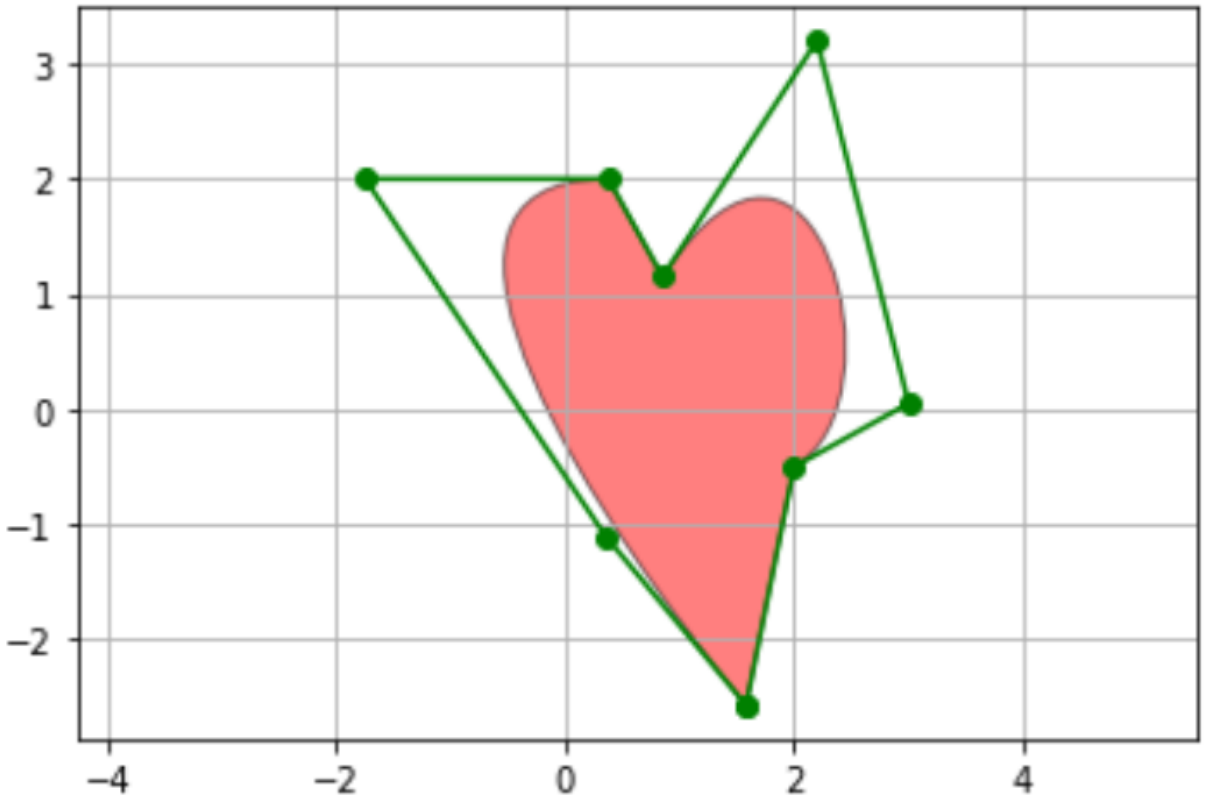
Izvor: vlastiti rad

U sljedećem primjeru, također je korišten matplotlib, ali sada s modulima koji su omogućili crtanje poput zelene linije (path modul) ili ružičastog ispunjenja (patches modul). Unutar koda potrebno je bilo unijeti podatke vezane uz kretanje ravnih linija, koje su uz pomoć funkcije Path.CURVE4 automatski vizualizirali zaobljene dijelova unutarnjeg dijela nacrta.

```
import matplotlib.path as mpath

import matplotlib.patches as mpatches
import matplotlib.pyplot as plt

fig, ax = plt.subplots()

Path = mpath.Path
path_data = [
    (Path.MOVETO, (1.58, -2.57)),
    (Path.CURVE4, (0.35, -1.1)),
    (Path.CURVE4, (-1.75, 2.0)),
    (Path.CURVE4, (0.375, 2.0)),
    (Path.LINETO, (0.85, 1.15)),
    (Path.CURVE4, (2.2, 3.2)),
    (Path.CURVE4, (3, 0.05)),
    (Path.CURVE4, (2.0, -0.5)),
    (Path.CLOSEPOLY, (1.58, -2.57)),
    ]
codes, verts = zip(*path_data)
path = mpath.Path(verts, codes)
patch = mpatches.PathPatch(path, facecolor='r', alpha=0.5)
ax.add_patch(patch)

# plot control points and connecting lines
x, y = zip(*path.vertices)
line, = ax.plot(x, y, 'go-')

ax.grid()
ax.axis('equal')
plt.show()

import matplotlib
matplotlib.path
matplotlib.path.Path
matplotlib.patches
matplotlib.patches.PathPatch
matplotlib.axes.Axes.add_patch
```

SLIKA 4 PRIMJER PREUZETOG KODA ZA MATPLOTLIB

Izvor: matplotlib.org, 2019.



SLIKA 5 PRIKAZ GRAFA PREUZETOG KODA ZA MATPLOTLIB

Izvor: [matplotlib.org, 2019.]

4.3. Seaborn biblioteka

Seaborn, kao što je već spomenuto, je knjižnica koja se bazira na knjižnici Matplotlib te je ovisna o njoj. Nedostatak Matplotliba je u tome što ima API niske razine stoga ga Seaborn nadopunjuje i čini svojevrsni omotač nad njime. Umjesto pisanja mnogo linija koda u Matplotlibu, kako bi se generirale složenije vizualizacije, poželjno je prebaciti rješavanje problema u Seaborn. Ona je fokusirana na izradu atraktivnih vizualizacija statičkih grafikona. Primjeri tih grafikona mogu biti toplinske mape, distribucije podataka te ostalo. Seaborn sadrži razne, unaprijed definirane oblike, stilove i palete boja koje stvaraju ljepši estetski dizajn u odnosnu na matičnu knjižnicu Matplotlib. Seaborn drži fokus na statističkoj vizualizaciji i modeliranju. Pomoću njega se pokušava vizualizaciju učiniti središnjim dijelom istraživanja i razumijevanja podataka. (http://seaborn.pydata.org/index.html)

Unutar Seaborn knjižnice postoje dataset-ovi ili tablice podataka koje su već ugrađene u sam program te se pomoću njih mogu vršiti analize i vizualizacije.

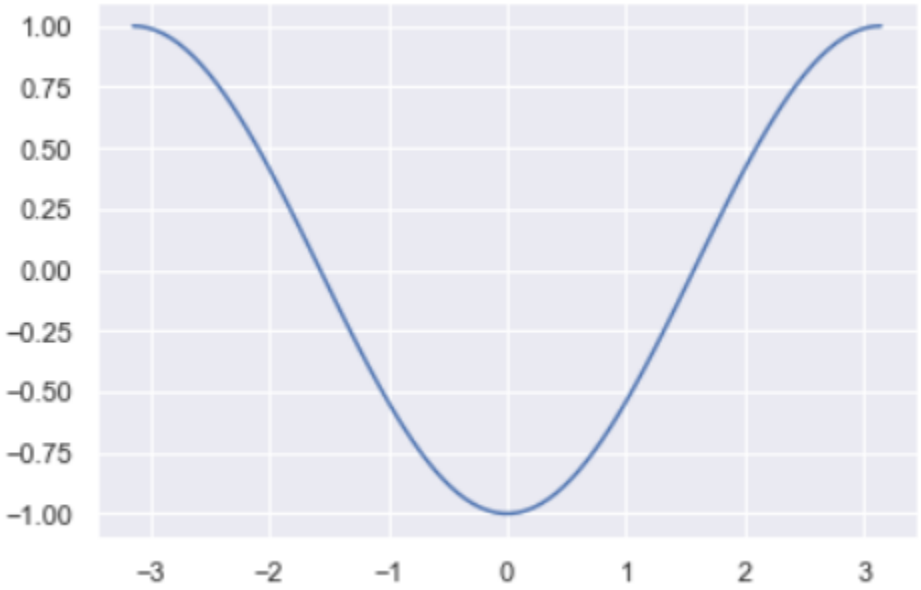
U sljedećem primjeru nije korištena nikakva tablica podataka budući da je vrlo jednostavan primjer u kojem je prikazana kosinusoida.

```
import seaborn as sns

import matplotlib.pyplot as plt
import numpy as np

def sinplot():
    x=np.linspace(-np.pi,np.pi,200)
plt.plot(x,-np.cos(x))
sns.set_style('darkgrid')
sinplot()
```

SLIKA 6 PRIMJER KODA ZA SEABORN

Izvor: vlastiti rad

X prikazuje koliki će biti raspon amplitude, odnosno njezin izgled, a u redu ispod je naglašeno da će to biti kosinusoida s negativnim predznakom. Za style je stavljeno darkgrid i iz tog razloga grafikon ima plave pozadinske rešetke. Taj style je unaprijed postavljen, a korisnik bira između nekoliko definiranih odnosno ponuđenih.



SLIKA 7 PRIKAZ GRAFA ZA SEABORN

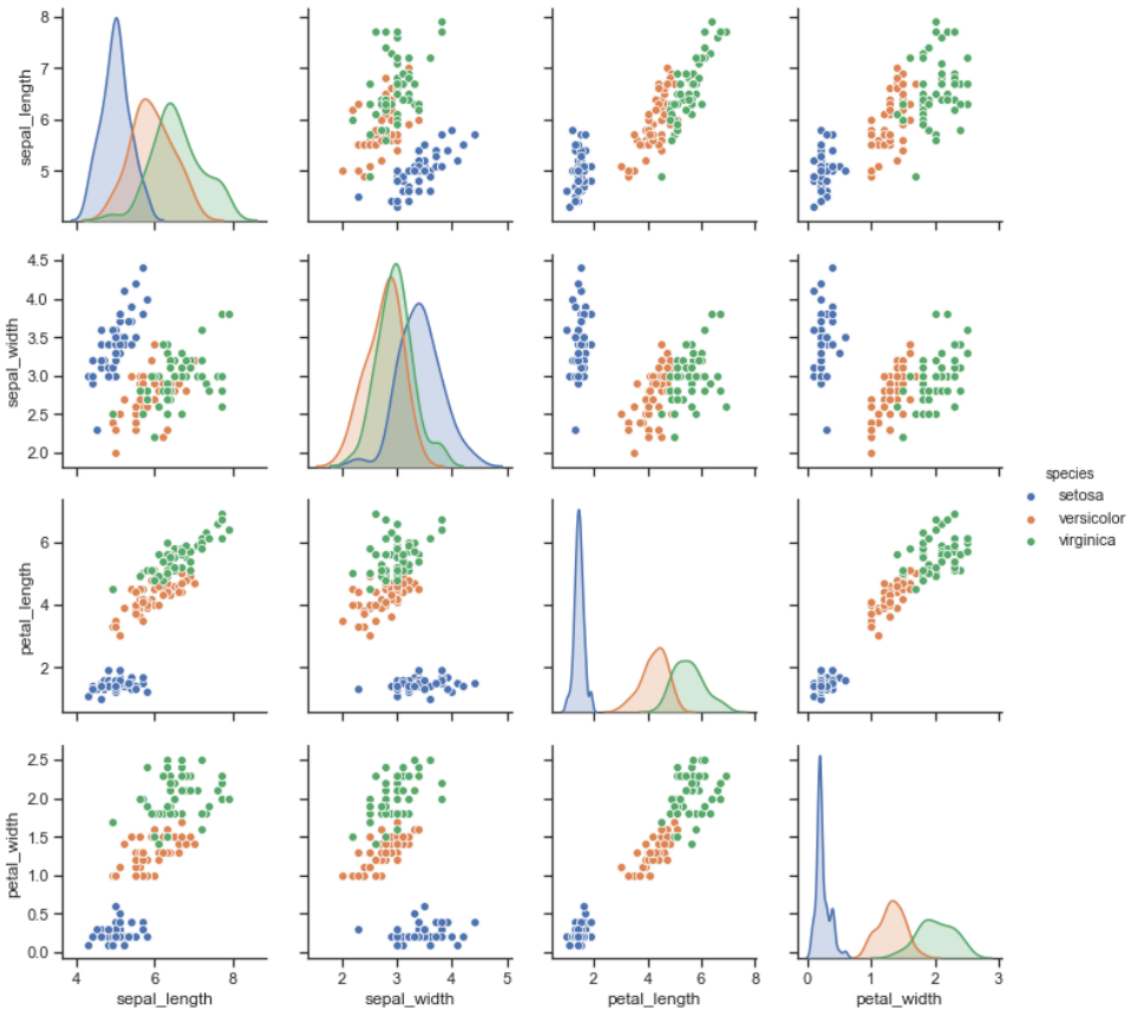
Izvor: vlastiti rad

Kao što je već napisano, moguće je koristiti već ugrađene tablice podataka. Uz samo četiri linije koda, za tablicu podataka „iris“, bilo je moguće dobiti usporedbu svih atributa međusobno.

```
import seaborn as sns

sns.set_style(style='ticks')
df = sns.load_dataset("iris")
sns.pairplot(df, hue="species")
```

SLIKA 8 PRIKAZ PRIMJERA PREUZETOG KODA ZA SEABORN

Izvor: seaborn.pydata.org, 2019



SLIKA 9 PRIKAZ GRAFA PREUZETOG KODA ZA SEABORN

Izvor: seaborn.pydata.org, 2019

4.4. Ggplot biblioteka

Korisnici koji imaju iskustva s R programskim jezikom, uglavnom su upoznati sa snažnim alatom za vizualizaciju ggplot2, u okruženju spomenutog R-a. Ovaj paket je prepoznat kao jedan od najboljih alata za grafičko prikazivanje informacija i statističko računanje. Zbog toga osim u R-u opsežne mogućnosti ove knjižnice dostupne su i u Python okruženju. Pomoću prijenosa paketa, on je dostupan za instaliranje pod imenom ggplot. Bitno je naglasiti da

iako je ggplot bio osmišljen po uzoru na ggplot2, on nije stvoren da bude identična preslika toj biblioteci .

Razvijen je od strane tima YHAT iz SAD-a. Njihov primarni cilj je bio razvitak profesionalnih grafova uz minimum programskog koda. Obje su navedne biblioteke zamišljene kao realizacija knjige „Grammar of Graphics“, autora Lelanda Wilkinsona. (https://monashdatafluency.github.io/python-workshop-base/modules/plotting_with_ggplot/#making-plots-with-plotnine-aka-ggplot)

On daje definciju statističke grafike: "Statistička grafika je ili treba biti trans-disciplinarno područje koje u obzir uzima znanstvene, statističke, računalne, estetske, psihološke i druge čimbenike… Statistička grafika je preslikavanje podataka s estetskim atributima (boja, oblik, veličina) geometrijskih objekata (točke, linije, trake)“ (Wilkinson, Leland, 2005.,)
Ggplot implementira izražajno sučelje za programiranje aplikacija visoke razine, iz tog razloga je moguće izraditi graf u kratkom vremenskom roku bez ponovnog pisanja jednog te istog kôda ukoliko želimo dodati nove individualne komponente grafa.

Slojevi u ggplot-u omogućuju kombiniranje i dodavanje različitih vrsta vizualizacijskih komponenti (ili slojeva) zajedno. Ta mogućnost primarno služi stvaranju jasnog grafičkog prikaza podataka i komponenata.

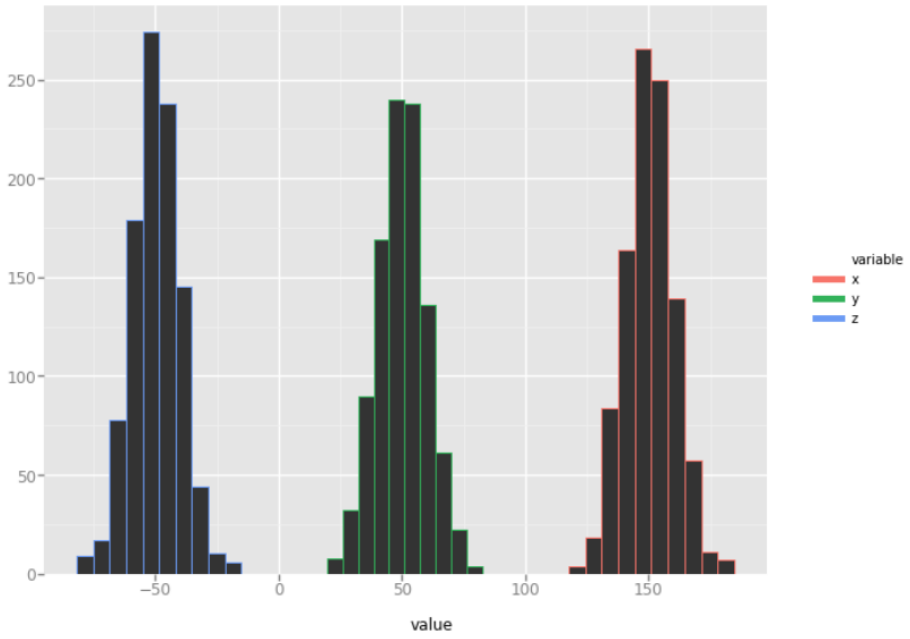
Proces vizualizacije podataka ima duboku unutarnju strukturu. Drugim riječima, proces stvaranja vizualizacije je jasno strukturiran sustav, koji uvelike utječe na način razmišljanja u procesu stvaranja infografike. A ggplot uči korisnika da razmišlja u tako strukturiranom pristupu, da razmišlja u skladu s ovim sustavom, tako da u procesu dosljedne izgradnje naredbi korisnik automatski započinje otkrivanje uzoraka u podacima.

U sljedećem primjeru funkcija 'geom_histogram' crta, odnosno stvara histogram, a osnovna estetska obilježja za ovu funkciju jesu 'x', 'alpha', 'color', 'fill', 'linetype', 'size'. Sadrži i dva specifična estetska obilježja. Uvezena su dva paketa 'pandas' i 'numpy'. U primjeru su navedene tri varijable (x, y i z) koje imaju određenu središnju poziciju, širinu i visinu.

```
geom_histogram(position = 'stack', stat = 'bin')

import pandas as pd
import numpy as np

df = pd.DataFrame({
    "x": np.random.normal(150, 10, 1000),
    "y": np.random.normal(50, 10, 1000),
    "z": np.random.normal(-50, 10, 1000)
})
df = pd.melt(df)

ggplot(aes(x='value', color='variable'), data=df) + \
    geom_histogram()
```

SLIKA 10 PRIMJER KODA ZA GGPLOT

Izvor: vlastiti rad



SLIKA 11 PRIKAZ GRAFA ZA GGPLOT

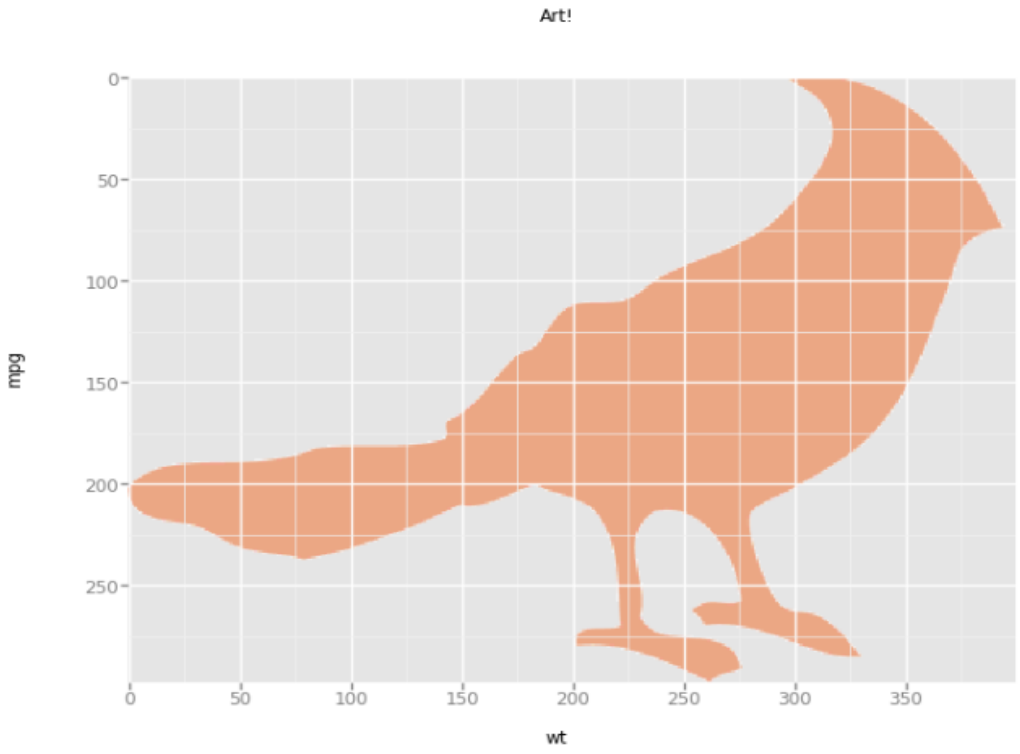
Izvor: vlastiti rad

Funkcija 'geom_point' prikazuje točke na grafu pomoću x i y koordinata. Pritom stvara graf raspršenosti za koji su nam potrebni podaci podijeljeni u stupce i retke, njih dodijeljujemo x i y osima. Estetska obilježja funkcije 'geom_point' jesu 'x','y','color','alpha','shape' te ostali.

```
from ggplot import *

geom_point(position = 'identity', stat = 'identity')
ggplot(mtcars, aes('wt', 'mpg')) + \
    geom_now_its_art() + \
    ggtitle("Art!")
```

SLIKA 12 PRIKAZ PRIMJERA PREUZETOG KODA ZA GGPLOT

Izvor: ggplot.yhathq.com, 2019.



SLIKA 13 PRIKAZ GRAFA PREUZETOG KODA ZA SEABORN

Izvor: ggplot.yhatq.com

4.5. Bokeh biblioteka

Bokeh je porijeklom iz Pythona. On se poput ggplot-a, također temelji na grafičkoj gramatici. Njegova jedinstvena tvrdnja je sposobnost stvaranja interaktivnih, web-spremnih čestica, koje se lako mogu ispisivati kao JSON objekti, HTML dokumenti ili interaktivne web aplikacije.

Bokeh je financijski sponzoriran projekt NumFOCUS-a , neprofitne organizacije koja je posvećena podršci znanstvene računske zajednice otvorenog koda.

Glavni fokus ove knjižnice je usmjeren na interakciju i prezentaciju vizualizacije kroz moderne web pretraživače. Njegov je cilj pružiti elegantnu, jezgrovitu konstrukciju svestrane grafike te to proširiti interaktivnošću visokih performansi na vrlo velikim skupovima podataka. Bokeh može brzo i jednostavno stvoriti interaktivne nacrte, nadzorne ploče i podatkovne aplikacije.

Bokeh nudi tri sučelja s različitim razinama kontrole za smještaj različitih tipova korisnika.

- Najviša razina je za brzo stvaranje karata odnosno grafikona. To uključuje metode za stvaranje uobičajenih dijagrama kao što su crtice na grafikonima, dijagrami okvira i histogrami.
- Srednja razina ima istu specifičnost kao matplotlib i omogućuje kontrolu osnovnih gradivnih blokova svake karte, primjerice točkice u dijagramu raspršenja.
- Najniža razina usmjerena je prema programerima i softverskim inženjerima. Nema unaprijed postavljenih zadanih postavki i zahtijeva da se definira svaki element karte.
-

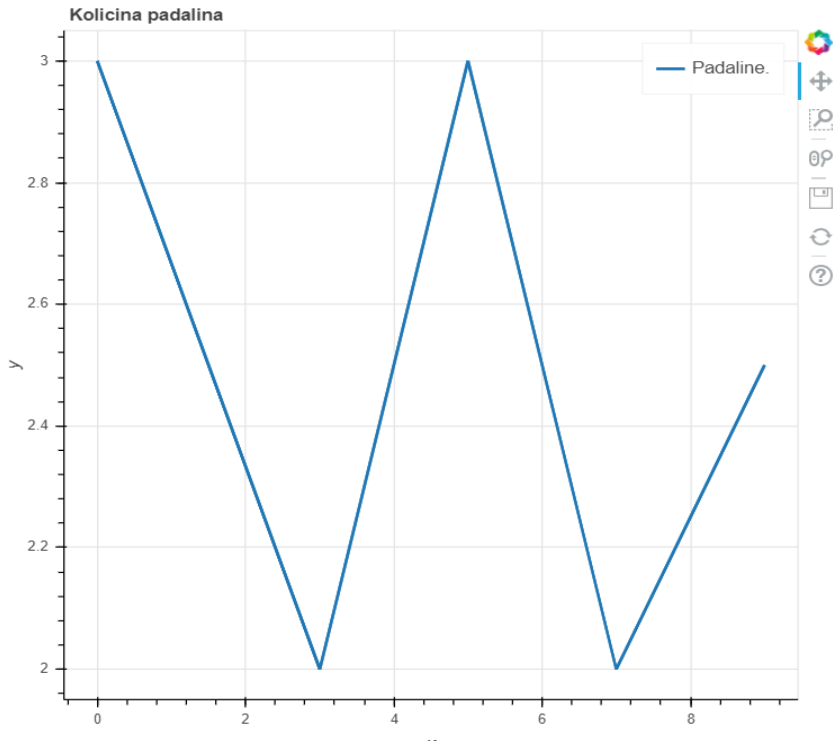
U sljedećem primjeru kao i u prethodnima, potrebno je bilo uvesti paket. Zatim je bilo potrebno popuniti podatke za koordinate kretanja, međutim tu je vidljiva razlika upisivanja koordinata za x i y osi u usporedbi s matplotlibom. Također za istu svrhu, ali na različit

način su unijeti nazivi osi i naziv grafa, te naziv atributa koji se prati. U ovom primjeru prikaz je ostvaren u HTML formatu.

```
from bokeh.plotting import figure, output_file, show


x = [0, 3, 5, 7, 9]
y = [3, 2, 3, 2, 2.5]

output_file("primjer.html")

p = figure(title="Kolicina padalina", x_axis_label='x',
y_axis_label='y')
p.line(x, y, legend="Padaline.", line_width=2)

show(p)
```

SLIKA 14 PRIMJER KODA ZA BOKEH

Izvor: vlastiti rad



SLIKA 15 PRIKAZ GRAFA ZA BOKEH

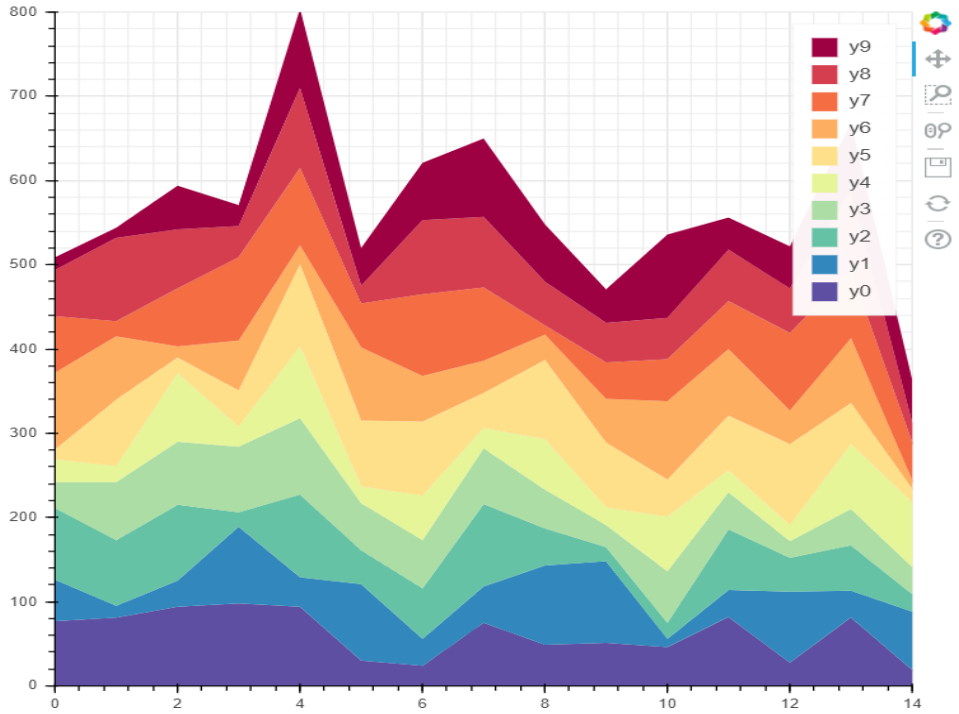
Izvor: vlastiti rad

Primjer preuzetog koda predstavlja prikaz složenog grafa kojem je određen broj N te boja svakog od njih. Definirana je veličina grafa i rang.

```
import numpy as np

import pandas as pd

from bokeh.core.properties import value
from bokeh.plotting import figure, show, output_file
from bokeh.palettes import brewer

output_file('stacked_area.html')

N = 10
df = pd.DataFrame(np.random.randint(10, 100, size=(15, N))).add_prefix('y')

p = figure(x_range=(0, len(df)-1), y_range=(0, 800))
p.grid.minor_grid_line_color = '#eeeeee'

names = ["y%d" % i for i in range(N)]
p.varea_stack(stackers=names, x='index', color=brewer['Spectral'][N], leg-
end=[value(x) for x in names], source=df)

# reverse the legend entries to match the stacked order
p.legend[0].items.reverse()

show(p)
```

SLIKA 16 PRIKAZ PRIMJERA PREUZETOG KODA ZA BOKEH

Izvor: bokeh.pydata.org, 2019.



SLIKA 17 PRIKAZ GRAFA PREUZETOG KODA ZA BOKEH

Izvor: [bokeh.pydata.org, 2019.]

4.6. PLOTLY BIBLIOTEKA

Pythonova Plotly grafička knjižnica stvara interaktivne grafikone kvalitete objavljivanja. Sadrži primjere izrade linijskih grafikona, dijagrama raspršivanja, grafikona područja, stupčastih grafikona, stupaca pogrešaka, dijagrama okvira, histograma, termalnih mapa, podsklopova, višestrukih osi, polarnih karata i dijagrama mjehurića.

Vrlo je dobra knjižnica ukoliko su potrebni interaktivni dijagrami, budući da daje odlične rezultate. Pozicioniran je prvenstveno kao internetska platforma na kojoj korisnici mogu kreirati i objavljivati vlastite vizualizacije. Međutim, knjižnica se također može koristiti izvanmrežno bez učitavanja vizualizacije na poslužitelj. Točnije njezinim mogućnostima može se pristupiti putem Python prijenosnika. Kao i Bokeh, Plotly stvara interaktivne grafove, ali nudi neke grafikone koje nije moguće pronaći u većini knjižnica poput dendograma, konturnih crteža i 3D grafikona..

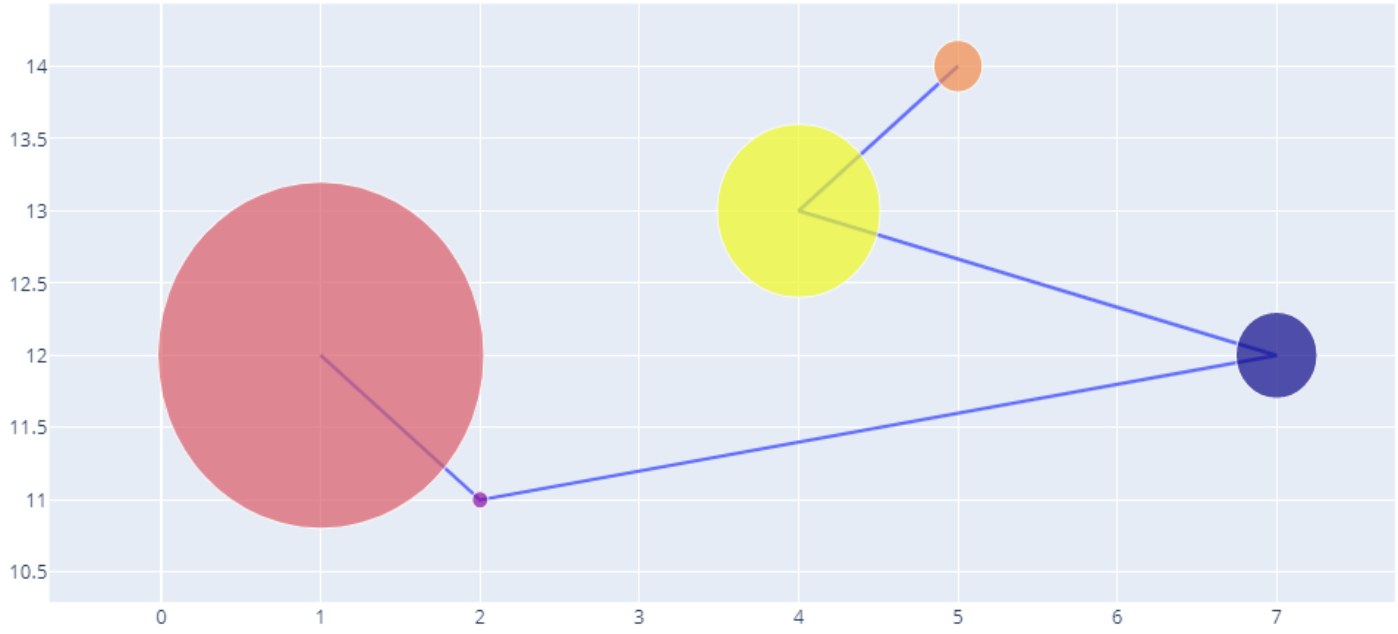
S obzirom na to da je ova knjižnica razvijena uglavnom kao autonomni proizvod, stalno se usavršava i proširuje. Dakle, korisnicima pruža doista neograničene mogućnosti za vizualizaciju podataka, bilo da se radi o interaktivnoj grafici ili konturama.

U nadolazećem primjeru, osim što je uvezen paket, upisane su i koordinate središta krugova koji će biti nacrtani. Nova mogućnost je 'mode' u kojem se mogu odabrati samo linije koje će povezivati točke koje su zadane ili se mogu odabrati markeri koji stvaraju krug određenog radijusa oko središnje točke. Kada je izabran 'mode' 'lines+markers', prikazat će se oboje. U dodatnoj liniji koda potrebno je definirati veličinu i boju za simbol kruga koji će se stvoriti oko točke.

```
import plotly.grapf_objects as go


fig = go.Figure(data=go.Scater(
    x=[1, 2, 7, 4, 5]
    y=[12, 11, 12, 13, 14]
    mode='lines+markers',
    marker=dict(size=[200, 10, 50, 100, 30],
                color=[4, 2, 0, 7, 5])
))

fig.show()
```

SLIKA 18 PRIMJER VLASTITOG KODA ZA PLOTLY

Izvor: vlastiti rad



SLIKA 19 PRIKAZ GRAFA VLASTITOG PRIMJERA ZA PLOTLY

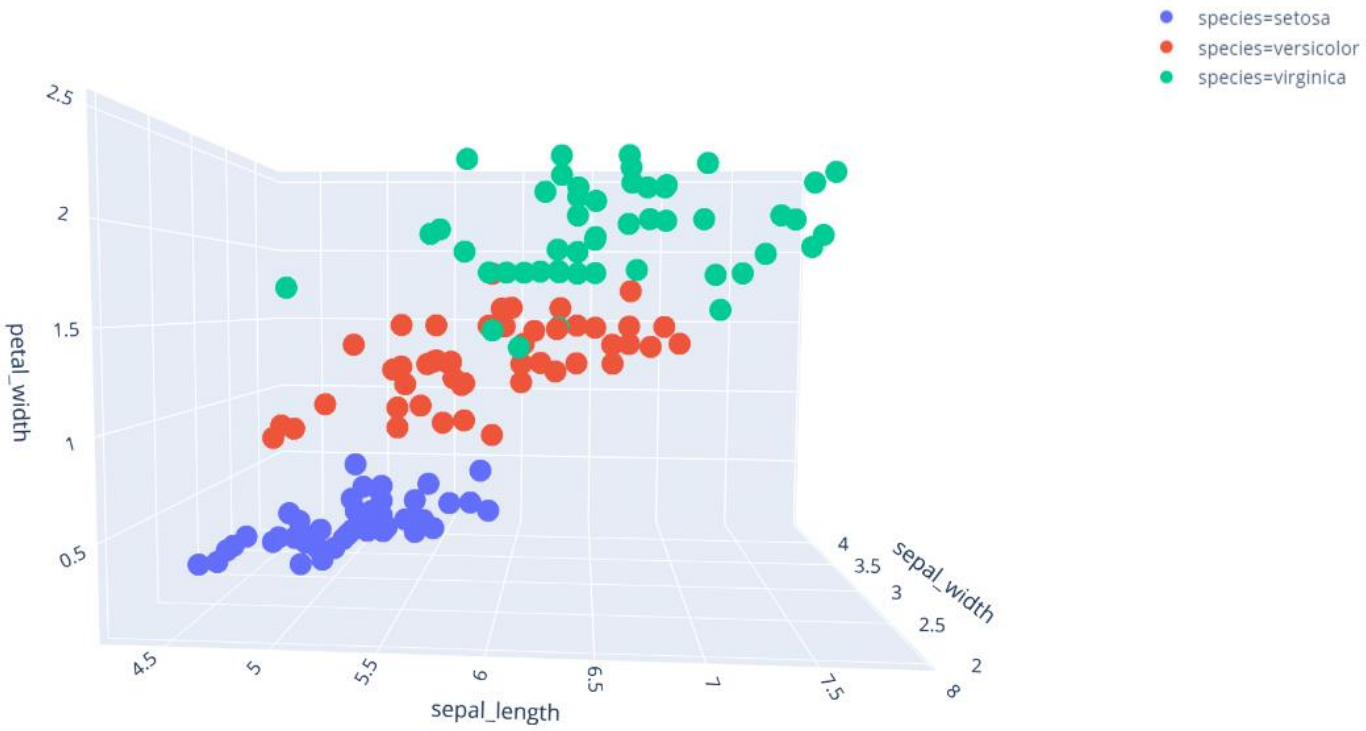
Izvor: vlastiti rad

Posljednji primjer, ponovno, uz mali broj linija koda, ali uz pogodan paket i odabran skup podataka vizualizira podatke. Prikaz je u 3D obliku, stoga se otvara mogućnost približavanja svakom podatku kako bi se moglo točno znati koje su njegove specifikacije.

```
import plotly.express as px

iris = px.data.iris()
fig = px.scatter_3d(iris, x='sepal_length', y='sepal_width',
z='petal_width',
              color='species')
fig.show()
```

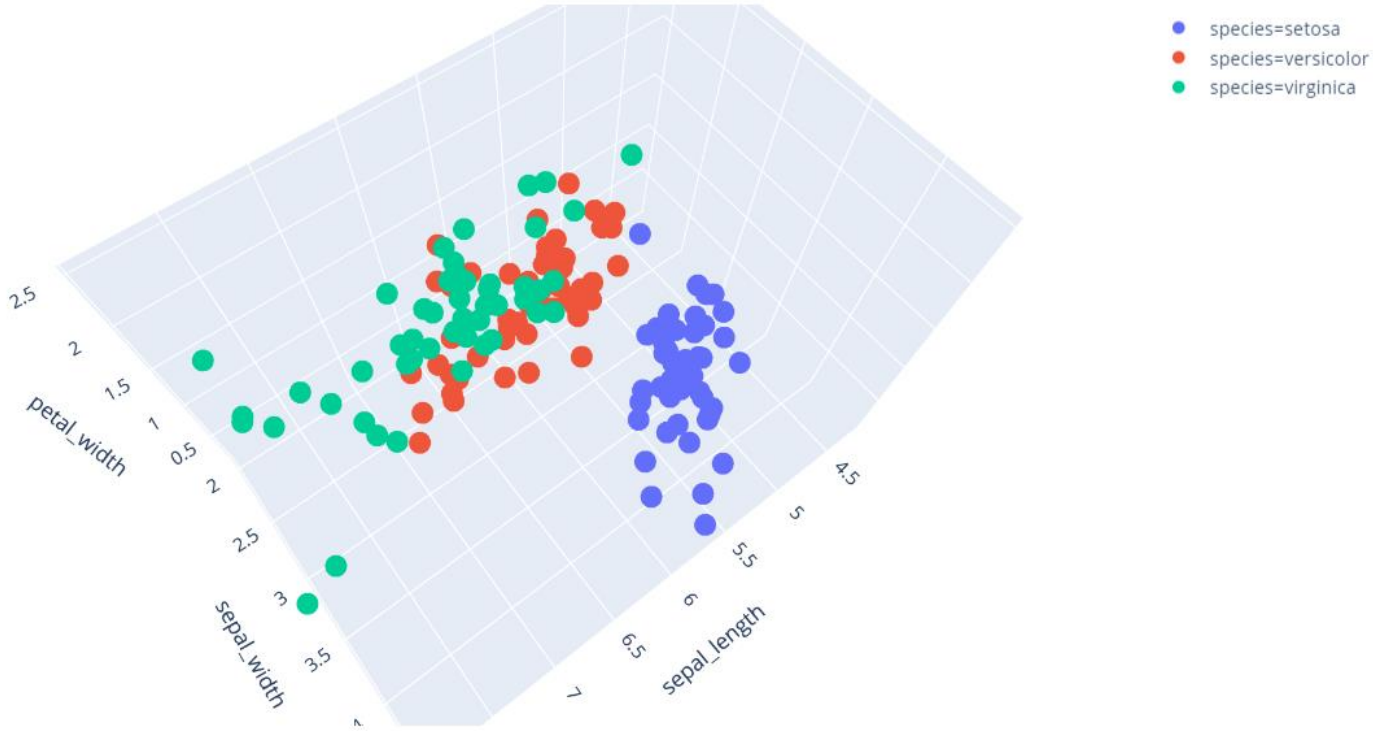
SLIKA 20 PRIKAZ PRIMJERA PREUZETOG KODA ZA PLOTLY

Izvor: plot.ly, 2019.



SLIKA 21 PRIKAZ GRAFA PREUZETOG KODA ZA PLOTLY

Izvor: plot.ly, 2019.



SLIKA 22 PRIKAZ GRAFA PREUZETOG KODA ZA PLOTLY

Izvor: plot.ly, 2019.

5. USPOREDBA BIBLIOTEKA

5.1. Primjer usporedbe

Kako bi spomenute biblioteke bile objašnjene još bolje te kako bi se njihove razlike i sličnosti lakše uočile, u nastavku slijedi prikaz jednakog grafa kroz svih pet biblioteka. Točnije graf prikazuje neka od naseljenih mjesta u Hrvatskoj sa više od 5000 stanovnika. U svaku biblioteku su uneseni jednaki opći podaci, koji su se prilagodili sintaksi i pravilima pojedine biblioteke.

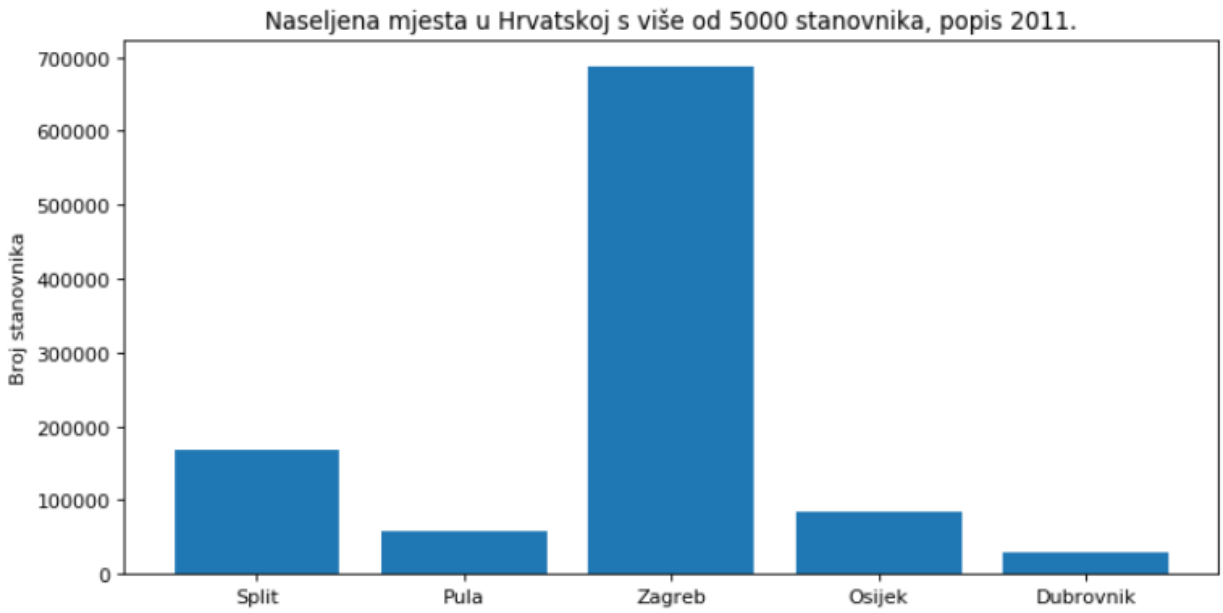
5.1.1. Matplotlib

Najprije su uvezeni *matplotlib.pyplot* i *numpy* kao osnovne kolekcije i naredbe pomoću kojih će biti generiran graf. *Plt.figure* regulira veličinu slike koja će nam se prikazati nakon što pokrenemo kod. *Plt.title* označava naslov grafa, a *plt.ylabel* označava naziv ipsilon osi. Zatim su definirane dvije varijable kojima su dodijeljene odgovarajuće vrijednosti. *Plt.bar* i *plt.show* omogućavaju davanje naredbe da se generira i prikaže stupičasti grafikon koji se nalazi na slici 24.

```
import matplotlib.pyplot as plt
import numpy as np

plt.figure(figsize=(10,5))
plt.title("Naseljena mjesta u Hrvatskoj s više od 5000 stanovnika,,
          popis 2011.")
plt.ylabel("Broj stanovnika")

gradovi = ["Split", "Pula", "Zagreb", "Osijek", "Dubrovnik"]
broj_stanovika = [167121, 57460, 688163, 84104, 28434]

plt.bar(gradovi, broj_stanovnika)
plt.show()
```

SLIKA 23 PRIKAZ VLASTITOG KODA USPOREDBE ZA MATPLOTLIB

Izvor: vlastiti rad



SLIKA 24 PRIKAZ GRAFA USPOREDBE ZA MATPLOTLIB

Izvor: vlastiti rad

5.1.2. Seaborn

Budući da je Seaborn samo nadopuna, a ne zamjena za Matplotlib, kod sadrži većinu istih segmenata. Međutim kada se uključi Seaborn biblioteka, otvara se niz novih mogućnosti. U ovom primjeru uključena je *sns.set_style* kojim se ostvaruje prikaz pozadinskih rešetki za koju je odabrana 'darkgrid' rešetka. Zatim se pomoću *sns.set_context* regulira izgled grafa, veličina slova i brojki, debljina rešetki.

Nakon pokretanja ovog koda, dobivamo graf koji se nalazi na slici 26. Korištenjem Seaborn biblioteke graf postaje jasniji i vizualno ljepši.

```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

plt.figure(figsize=(10,5))
plt.title("Naseljena mjesta u Hrvatskoj s više od 5000 stanovnika,
          popis 2011.")
plt.ylabel("Broj stanovnika")
gradovi = ["Split", "Pula", "Zagreb", "Osijek", "Dubrovnik"]
broj_stanovnika = [167121, 57460, 688163, 84104, 28434]

plt.bar(gradovi, broj_stanovnika)

sns.set_style('darkgrid')
sns.set_context("poster", font_scale=0.5, rc={"grid.linewidth": 3})

plt.show()
```

SLIKA 25 PRIKAZ VLASTITOG KODA USPOREDBE ZA SEABORN

Izvor: vlastiti rad

SLIKA 26 GRAFA USPOREDBE ZA SEABORN

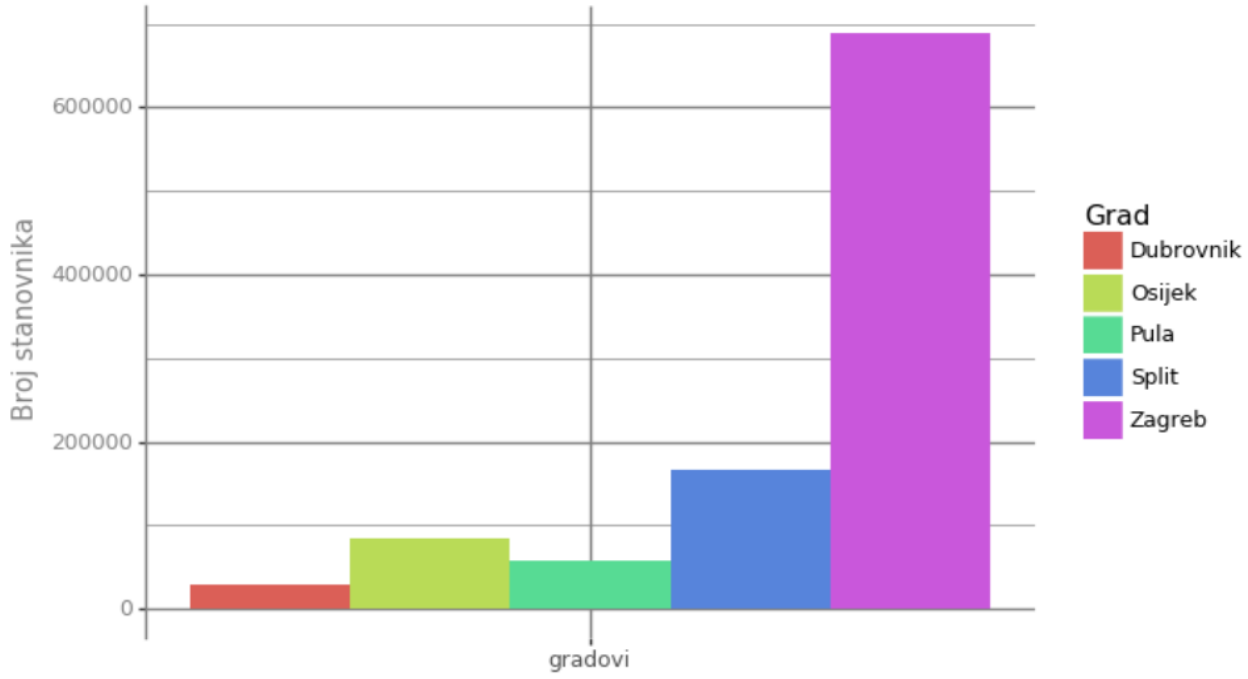
5.1.3. Ggplot

Najprije su uvedeni paketi pandas i numpy, a zatim i plotnine koji je napisan po uzoru na ggplo2 za R, a sinonim je za ggplot unutra Pythona. Nakon toga su definirane varijable i unesene vrijednosti. U ovom kodu je sintaksa za unos kompliciranija nego u prethodna dva primjera, ali zato pruža još više mogućnosti nego prethodna dva. Ponuđen je niz mogućnosti pomoću kojih se može izmijeniti izgled grafa po želji. Sa strane je vidljiva legenda gradova, a niz gradova je automatski poredan abecedno. Moguće je prilagoditi izgled, boju i veličinu naziva, vanjskih linija, unutarnjih rešetki te ostalog.

```
import pandas as pd
import numpy as np
from plotnine import *

df = pd.DataFrame({
    'Varijabla': ['gradovi', 'gradovi', 'gradovi', 'gradovi', 'gradovi'],
    'Grad': ['Split', 'Pula', "Zagreb", "Osijek", "Dubrovnik"],
    'Broj stanovnika': [167121, 57460, 688163, 84104, 28434],
})
df['Varijabla'] = pd.Categorical(df['Varijabla'], categories=['gradovi'])

df.title = ("Naseljena mjesta u Hrvatskoj s više od 5000 stanovnika,
             popis 2011.")

(ggplot(df, aes( x='Varijabla', y='Broj stanovnika', fill='Grad'))
 + geom_bar(stat='identity', position='dodge')
 + theme(panel_background=element_rect(fill='white'),
         axis_title_y=element_line(color='grey'),
         axis_title_x=element_blank(),
         axis_line_x=element_blank(),
         axis_line_y=element_line(color='grey'),
         axis_text_y=element_line(color='grey'),
         panel_grid=element_line(color='grey'),
         panel_border=element_blank()
))
```

SLIKA 27 PRIKAZ VLASTITOG KODA USPOREDBE ZA GGPLOT

Izvor: vlastiti rad



SLIKA 28 GRAFA USPOREDBE ZA GGPLOT

Izvor: vlastiti rad

5.1.4. Bokeh

Bokeh pruža mogućnost izvoza grafa u html formatu, što ga razlikuje od prethodnih biblioteka. Osim toga i pojedinih razlika u sintaksi, Bokeh za grafove ne nudi izrazito atraktivna vizualna rješenja. Neke od interesantnih mogućnosti su regulacija debljine stupaca pomoću *p.vbar(width=0.9) ,* postavljanje alatne trake sa *toolbar_*location ili kreiranje linije koja se nastavlja iz stupaca u grafikonu pomoću *p.xgrid.grid-line_color.* To je vidljivo i na prikazu samog grafa na slici .

```
from bokeh.io import show, output_file

from bokeh.plotting import figure

output_file("gradovi.html")

gradovi = ["Split", "Pula", "Zagreb", "Osijek", "Dubrovnik"]
broj_stanovnika = [167121, 57460, 688163, 84104, 28434]

p = figure(x_range=gradovi, plot_height=250,
           title="Naseljena mjesta u Hrvatskoj s više od 5000 stanovnika,
                  popis 2011.",
           y_axis_label='Broj stanovnika',
           toolbar_location=right, tools="")

p.vbar(x=gradovi, top=broj_stanovnika, width=0.9)

p.xgrid.grid_line_color = 'Blue'
p.y_range.start = 0

show(p)
```

SLIKA 29 PRIKAZ VLASTITOG KODA USPOREDBE ZA BOKEH

Izvor: vlastiti rad

SLIKA 30 GRAFA USPOREDBE ZA BOKEH

Izvor: vlastiti rad

5.1.5. Plotly

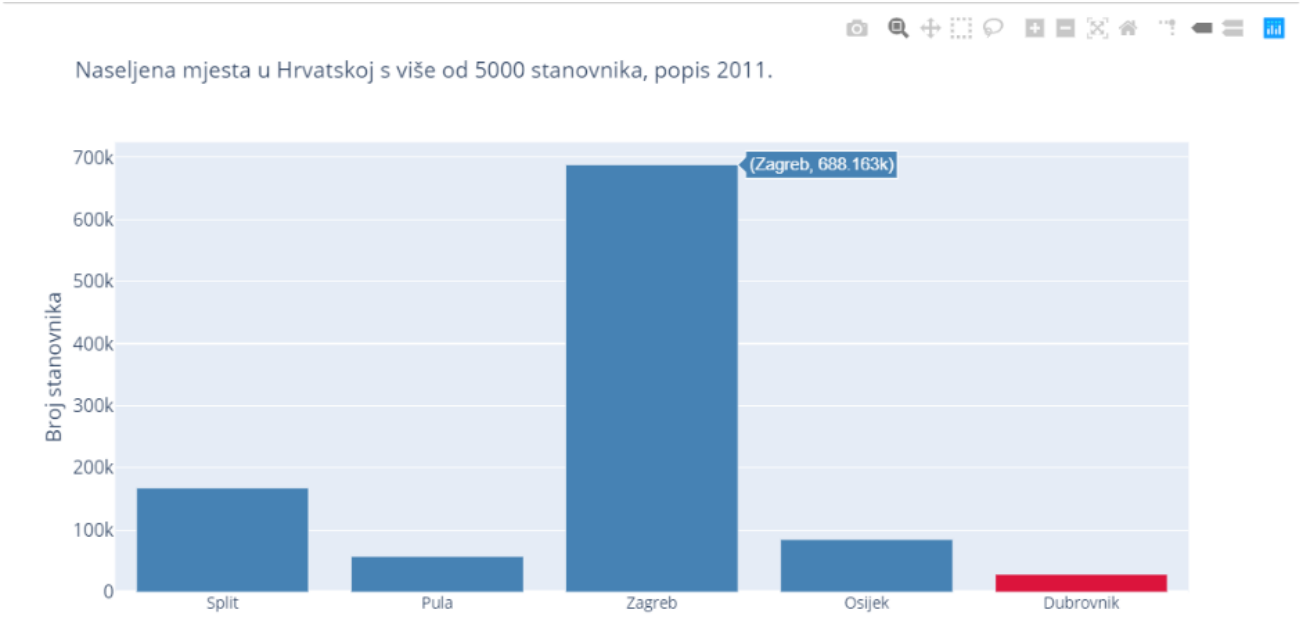
Plotly je zasigurno najatraktivnija biblioteka od pet navedenih koja nudi razne mogućnosti koje do sada niti jedna nije imala. Pomoću *colors[4]='crimson'* prema indexu je prilagođena boja stupca, dok ostali ostaju kako smo ih iz prve odredili. . Sintaksa je vrlo jasna te je moguće koristiti već formirane tablice podataka koje će se uvesti u kodu ili je moguće, kao što je to u ovom slučaju, unijeti direktno podatke koji će biti obrađeni. Koristeći fig.update.layout određuju se naslovi i njihova veličina i boja, te isto i za nazive ispsilon i iks osi.

Kada pokrenemo kod dobivamo alatnu traku u pomoću koje se može približavati sliku ili je udaljavati, moguće je napraviti fotografiju grafa, izrezivati određene dijelove grafa te ostalo. Nakon što pređemo preko pojedinog stupca, prikaže nam se njegov naziv sa točnim vrijednostima. Graf se nalazi na slici .

```
import plotly.graph_objects as go


colors = ['steelblue',] * 5
colors[4] = 'crimson'

fig = go.Figure(data=[go.Bar(
    x=['Split', 'Pula', 'Zagreb', 'Osijek', 'Dubrovnik'],
    y=[167121, 57460, 688163, 84104, 28434],
    marker_color=colors
)])
fig.update_layout(title_text='Naseljena mjesta u Hrvatskoj s više od 5000
stanovnika, popis 2011.')
fig.update_layout(
    yaxis=dict(
        title='Broj stanovnika',
        titlefont_size=16,
        tickfont_size=14,
    ))
```

SLIKA 31 PRIKAZ VLASTITOG KODA USPOREDBE ZA PLOTLY

Izvor: vlastiti rad



SLIKA 25 PRIKAZ USPOREDBE SEABORNA I MATPLOTLIBA

SLIKA 32 GRAFA USPOREDBE ZA PLOTLY

Izvor: vlastiti rad

5.2. Zaključak usporedbe

Iako su već navedene razne značajke za svih pet obrađenih knjižnica, kako bi bilo jasnije koje su prednosti, a koje mane te u kojem slučaju je poželjno koristiti pojedinu, u nastavku će one biti uspoređene.

Matplolib zahtijeva malo kodiranja za vizualizaciju i može se koristiti na bilo kojem operativnom sustavu. Može nositi velike numeričke i matrične izračune. Izuzetno je snažan, stvarajući koherentan spektar vizualizacije, a uz to može nositi velike numeričke izračune. Kao i on, Seaborn također gradi kompleksne vizualizacije i stvara grafikone od samo nekoliko redaka koda, no on za razliku od njega nudi ugrađene i regresijske nacrte. Seaborn osim toga nudi estetski zadovoljavajuće stilove za nacrte s mogućnošću korištenja palete

boja koje se lako mogu prilagoditi. Ukratko Seaborn koristi manje sintakse i ima zadivljujuće zadane teme, a Matplotlib je prilagodljivii. Dok su Matplotlib i Seaborn statični, knjižnica Bokeh stvara interaktivne i skalabilne vizualizacije u pregledniku pomoću JavaScript widgeta. Ako su podaci toliko složeni (ili im još nije pronađena "poruka"), poželjno bi bilo upotrijebiti Bokeh za izradu interaktivnih vizualizacija koje će omogućiti gledateljima da sami istražuju podatke.

Bokeh naime izvodi Python, proizvodi JSON, a stvara Javascript. On je za razliku od Plotly-a, manje integriran sa Pandas-om. S druge strane, slično kao u Plotly-ju, u Bokeh-u je lako staviti više manjih nacrta u jednu figuru no Plotly se smatra profesionalnijim nego Bokeh. Ova knjižnica je za razliku od Seaborna nezavisna o Matplotlib-u. U usporedbi sa matplotlibom, Bokeh je mlađi stoga i treba raditi a njegovoj funkcionalnosti, no već ima jako dobru perspektivu

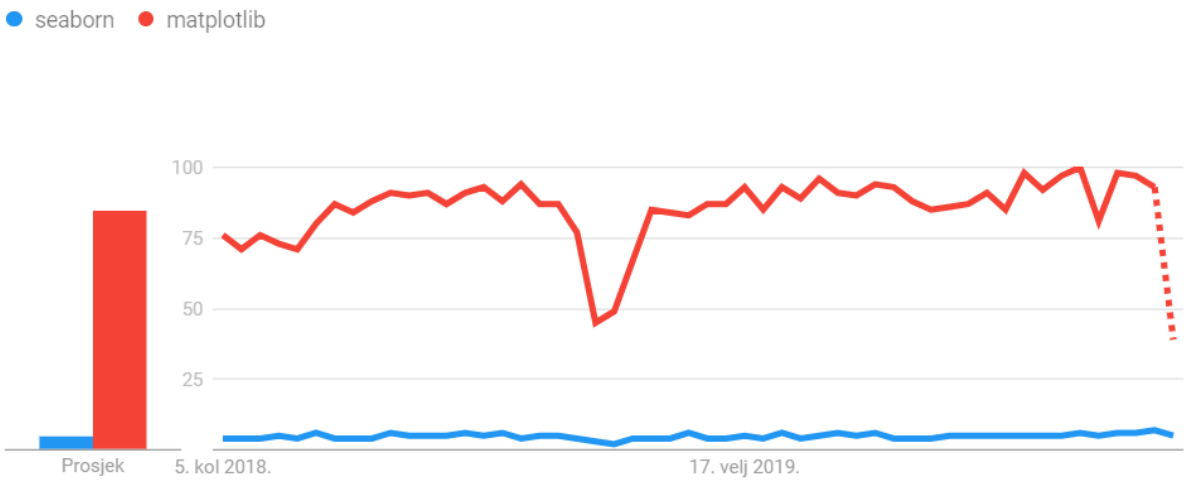
Plotly je također interaktivan i pozicioniran je prije svega kao internetska platforma na kojoj korisnici mogu stvarati i objavljivati vlastite vizualizacije. No, knjižnica se može koristiti i izvan mreže bez prijenosa vizualizacije na poslužiteljski web mjesto. Omogućuje stvaranje interaktivnih vizualizacija izgrađenih pomoću D3.js-a, a da međutim uopće nije nužno poznavanje D3.js-a. Plotly je izuzetno moćan, ali i postavljanje i stvaranje figure oduzimaju puno vremena, a niti jedno nije intuitivno. Sintaksa za stvaranje vizualizacije je vrlo jednostavna. To je jedna od najstabilnijih knjižnica s jednostavnim API-jem. Alat prihvaća mnoge formate, kao što su .xls, .xlsx ili .csv datoteke. Kompatibilan je s brojnim jezicima i alatima kao što su R, Python, MATLAB, Perl i drugi., Lako se može dijeliti s više osoba, ali nacrti su obično javni i mogu ih gledati svi, ne ostavljajući prostor za privatnost podataka .

Unatoč velikom potencijalu ggplot knjižnice, nedostaje joj interaktivnost. Stoga je za one kojima je potrebna interaktivna vizualizacija podataka stvorena Bokehova knjižnica. Bez obzira na to, ggplot ima i svojih prednosti poput mogućnosti da nudi sučelje koje je snažno, zabavno i jednostavno za učenje. Osim toga kombinira više skupova podataka u jedan graf, nudi varijaciju estetskih vrijednosti za izradu lijepih i razrađenih grafova. Jedna od velikih mana u odnosu na ostale je to što je on čvrsto integriran sa Pandasom, međutim

nije kompatibilan sa njenom posljednjom verzijom što stvara komplikacije pri samom pokretanju koda.

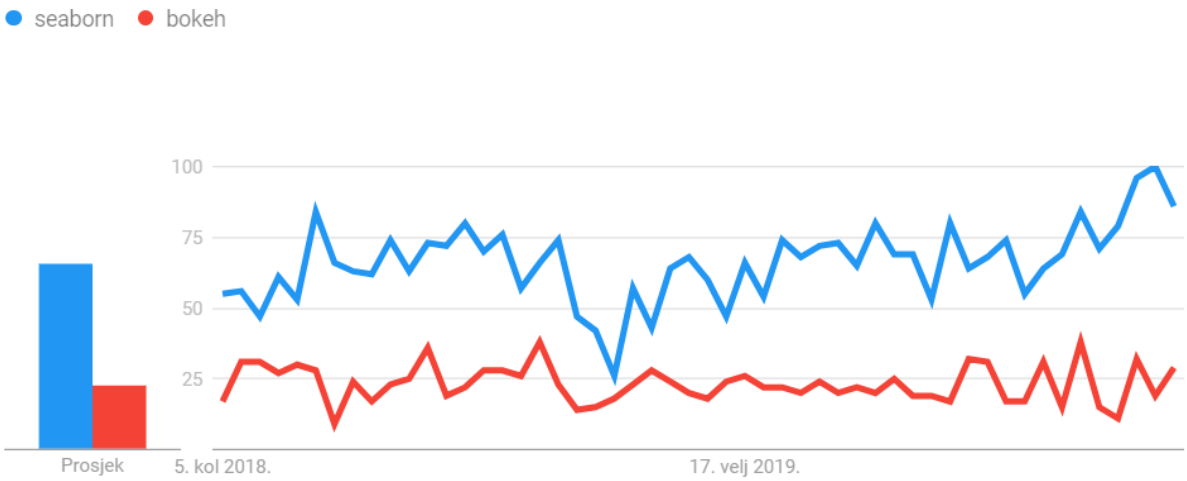
Python daje širok spektar opcija za vizualizaciju podataka, važno je identificirati metodu koja najbolje odgovara potrebama projekta. To može biti od osnovnih crtanja do sofisticiranih i kompliciranih statističkih karata i drugih.

Prateći Googlove trendove sa službenih stranica, u nastavku slijedi usporedba popularnosti navedenih knjižnica za vizualizaciju.



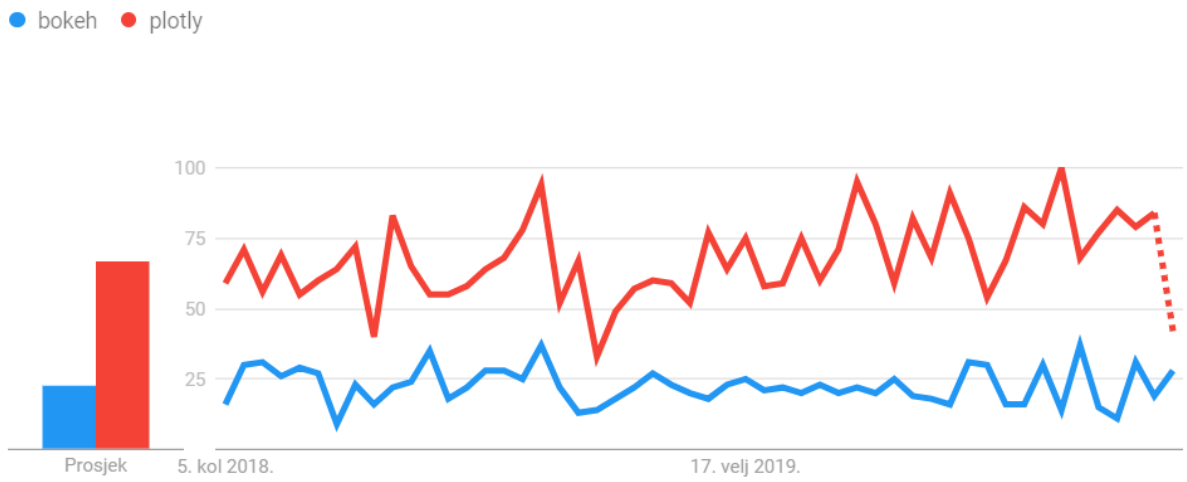
SLIKA 26 PRIKAZ USPOREDBE SEABORNA I MATPLOTLIBA

Izvor: python.libhunt.com, 2019.



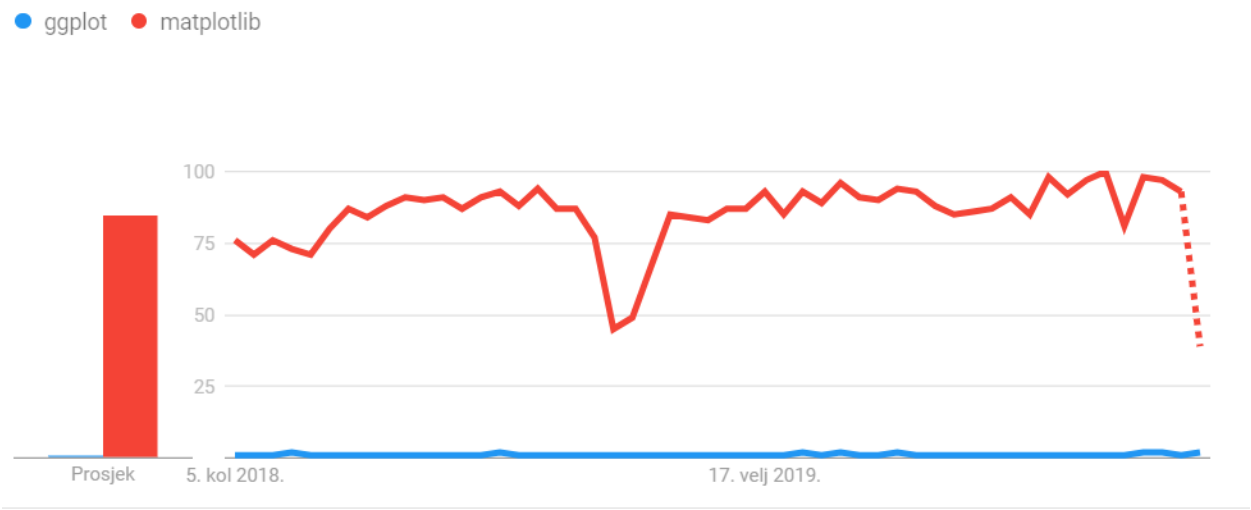
SLIKA 27 PRIKAZ USPOREDBE SEABORNA I BOKEHA

Izvor: python.libhunt.com, 2019.



SLIKA 28 PRIKAZ USPOREDBE BOKEHA I PLOTLYA

Izvor: python.libhunt.com, 2019.



SLIKA 29 PRIKAZ USPOREDBE GGPLOTA I MATPLOTLIBA

Izvor: python.libhunt.com, 2019.

6. ZAKLJUČAK

Vizualizacija podataka disciplina je koja pokušava pridonijeti razumijevanju podataka postavljanjem u vizualni kontekst. Koristi se kako bi se mogli izložiti obrasci, trendovi i korelacije koje inače ne bi bilo moguće otkriti. Vizualizacija podataka je umjetnost i znanost Ona zajedno s analizom, je oduvijek bila jedna od najvažnijih faza bilo kojeg cjevovoda za znanost o podacima bez obzira na složenost podataka ili projekta. Kako bi vizualizacija bila korisna te ispravno odrađena, najbitnije je prvenstveno dubinski razumjeti podatke koje se koristi

Vizualizacija u Pythonu nudi širok spektar mogućnosti. Python sadrži mnoštvo korisnih knjižnica koje se mogu kombinirati sa paketima koje dolaze unutar njih. Njihove značajke su raznovrsne, stoga je moguće koristiti statičke ili pak interaktivne knjižnice. Osim toga

moguće je razraditi učinkovite strategije vizualizacije podataka u više dimenzija, u rasponu od 1-D do 6-D

Neke od osnovih značajki opisanog Matplotliba jesu da je namijenjen niskoj razini te da pruža puno slobode korisniku. Seaborn s druge strane posjeduje sučelje na visokoj razini te ima sjajne zadane stilove. Ggplot je osmišljen na temelju R-ovog ggplot2 te se koristi grafičkom gramatikom. Za stvaranje interaktivnih grafikona poželjno je koristiti Bokeh, a iz istog razloga je koristan i Plotly.

Kako je vidljivo, svaka knjižnica ima puno prednosti ,ali naravno imaju i poneku manu. Za izradu vlastite vizualizacije najprije je potrebno skupiti svoje podatke, a potom i razraditi ideju kako bi se trebali ti podaci prezentirati. Na osnovu toga lako će se odabrati neku od knjižnica koja će biti pogodna.

7. LITERATURA

1. Kalafatić Z., Pošćić A., Šegvić S. i Šribar J. (2016.) Python za znatiželjne
2. MICROSOFT (2019.), Zašto je vizualizacija podataka nužna za razumijevanje poslovnog obavještavanja
   Dostupno na: https://products.office.com/hr-hr/business/articles/why-data-visualization-is-a-must-for-understanding-business-intelligence
   [Pristupljeno 28.07.2019.]
3. Wilkinson, Leland, (2005.) Gramatika grafike

8. POPIS SLIKA