**SVEUČILIŠTE U RIJECI  
ODJEL ZA INFORMATIKU**

**Preddiplomski studij informatike**

**Seminarski rad iz kolegija  
OSNOVE INFORMATIKE**

**TEHNIČKA DOKUMENTACIJA  
PYTHON – ZADATAK 2**

Autor: Dino AhmičićMentori: prof. dr. sc. Sanda Martinčić-Ipšić  
 dr. sc. Slobodan Beliga

U Rijeci, Siječanj 2021.

Sadržaj

[1. Početak koda 2](#_Toc62393373)

[2. Učitavanje podataka u liste 3](#_Toc62393374)

[3. Funkcije operacijskih izračuna 6](#_Toc62393375)

[3.1. Suma i aritmetička sredina 6](#_Toc62393376)

[3.2. Mod i medijan 7](#_Toc62393377)

[3.3. Standardna devijacija 8](#_Toc62393378)

[3.4. Minimum i maksimum 9](#_Toc62393379)

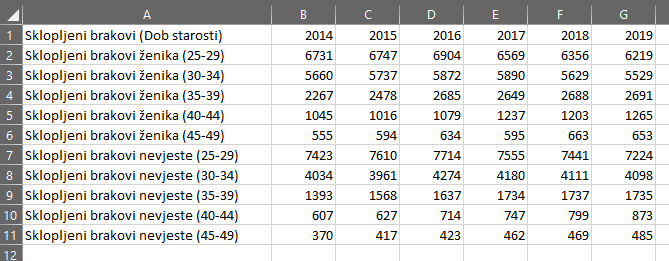
[4. Ispis rezultata pozivajući funkcije 10](#_Toc62393380)

[5. Usporedba rezultata sa excel-om 11](#_Toc62393381)

[6. Grafikon 12](#_Toc62393382)

# Početak koda

U ovom zadatku seminara potrebno je izračunati sumu, aritmetičku sredinu, mod, medijan, standardnu devijaciju, minimum i maksimum podataka koji smo preuzeli sa Državnog zavoda za statistiku (<https://www.dzs.hr/>) kao .csv datoteku, naša datoteka se sove “statistika2.csv”.



Slika 1. Korišteni podaci

Ovi podaci nam govore ukupan broj sklopljenih brakova ženika i nevjesta po dobi starosti u 2014., 2015., 2016., 2017., 2018., 2019., godine.

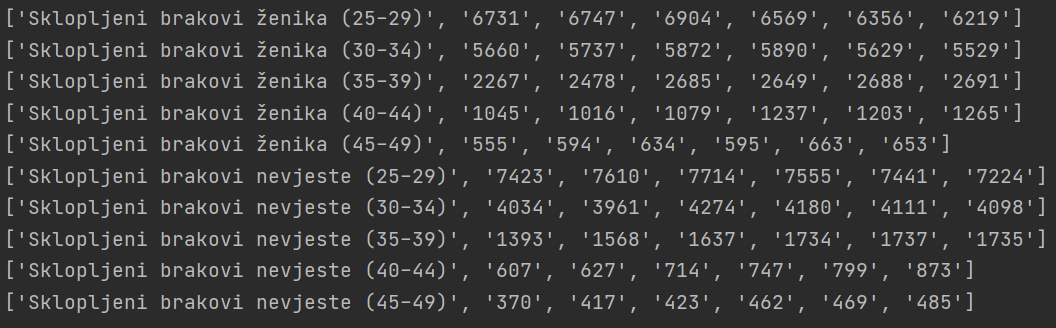
Naš zadatak je sad za te podatke napraviti prethodno spomenute izračune sa python-om.

Prvo korak jest učitati tu .csv datoteku u python, te ih odmah zapisati u listu, to radimo sljedećim kodom:



Slika 2. Učitavanje .csv datoteke

Linijama 11 i 12 ispisujemo tu listu sa podacima.

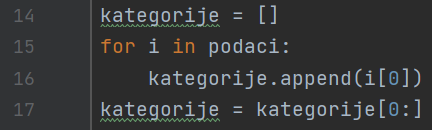


Slika 3. Ispisani podaci u terminalu

# Učitavanje podataka u liste

Sljedeće što moramo napraviti jest napraviti posebne liste kategorija, te sklopljenih brakova za posebne godine. Što znači da ćemo imati ukupno 7 listi za kategorije, te pojedine godine od 2014. – 2019.

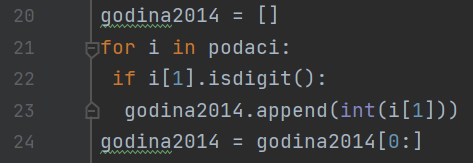
Prvu listu koju ćemo napraviti jesu za kategorije (npr. Sklopljeni brakovi ženika (25-29)). To radimo sljedećim kodom:



Slika 4. Lista za kategorije

Nakon što smo definirali listu koja će nam se zvati kategorije, for petljom u nju spremamo sve nulte vrijednosti glavne liste, to su nama sve kategorije, odnosno svi sklopljeni brakovi. Nakon što smo to napravili, zadnjom linijom koda smo definirali “kategorije[0:]” da bi nam se u listu spremale znamenke, odnosno slova od početka do kraja stringa.

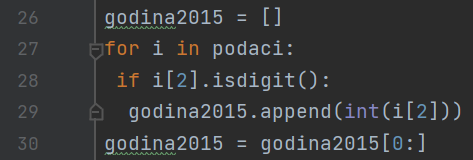
Kada smo definirali prvi listu sa našim kategorijama, možemo početi izrađivati liste naših podataka za sve posebne godine.



Slika 5. Lista za 2014. Godinu

Identično kao i za kategorije, prvo moramo definirati listu, to radimo prvom linijom koda. Kada smo definirali listu, for petljom prolazimo kroz glavnu listu, te funkcijom “.isdigit()” u poziciji 1 provjeravamo radi li se o broju. U našoj situaciji, na poziciji 1 upravo i jesu brojevi, te sve brojeve stavljamo u listu “godina2014”, a kao i u prethodnoj situaciji “godina2014[0:]” nam zapisuje sve znamenke od početka do kraja to integer-a.

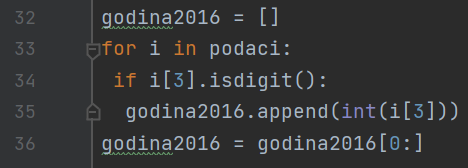
Isto kao i kod kreiranja liste za 2014. godinu, moramo napraviti liste i za preostale godine.



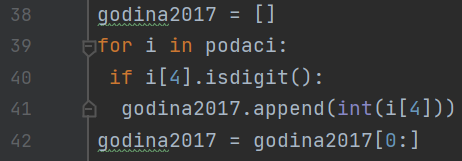
Slika 6. Lista za 2015. Godinu

Ovo nam je lista za podatke iz 2015. godine. Gotovo je identična kao i za 2014. no moramo paziti na koju se poziciju glavne liste stavljamo. Kako nam se podaci za 2015. godinu nalaze na poziciji 2 glavne liste, u if grani moramo postaviti “i[2].isdigit()”, kao i kod funkcije “append(int(i[2]))”.

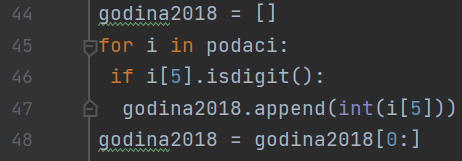
Na isti način kreiramo liste za sve ostale godine, pazeći na pozicije u glavnoj listi.



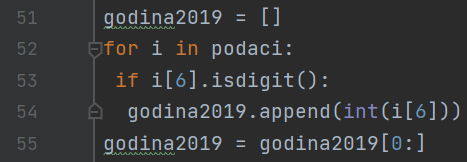
Slika 7. Lista za 2016. Godinu



Slika 8. Lista za 2017. Godinu



Slika 9. Lista za 2018. Godinu



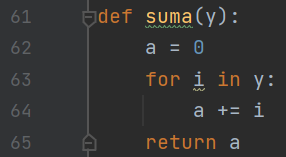
Slika 10. Lista za 2019. Godinu

# Funkcije operacijskih izračuna

Nakon što smo kreirali sve potrebne liste, moramo napraviti funkcije kojima ćemo izračunavati tražene operacije.

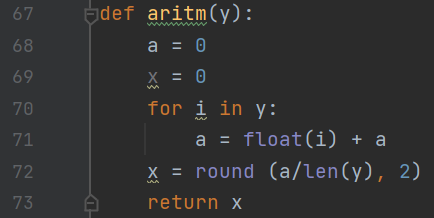
## Suma i aritmetička sredina

Kao prve dvije tražene operacije jesu suma i aritmetička sredina.



Slika 11. Funkcija za sumu

Prikazana je funkcija sume. U toj funkciji morali smo definirati varijablu “a = 0” kako bi u nju spremali zbrojeve naših podataka. For petljom se krećemo po argumentu y (nama će y biti godine2014, godine2015,…), te svi brojevi se zbrajaju u varijabli a.

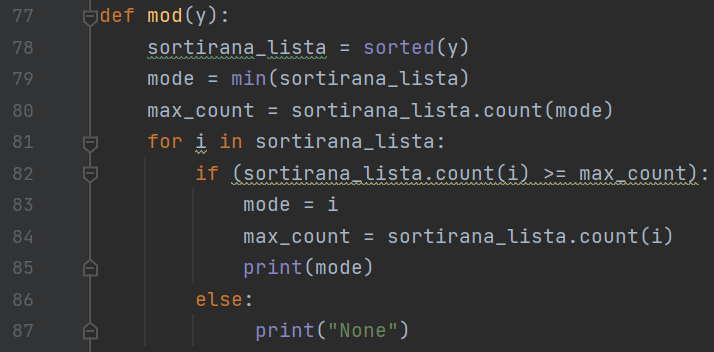


Slika 12. Funkcija za aritmetičku sredinu

Sljedeća je funkcija za aritmetičku sredinu. For petljom moramo proći kroz sva polja liste y (godine 2014, godine2015, …), te brojeve sumirati kao tip float. Pri kraju izračunamo prosjek, tako da podijelimo dobiveni zbroj sa količinom brojeva (len(y) funkcijom dobivamo koliko argument ima brojeva). Funkcija nam vraća float tog broja.

## Mod i medijan

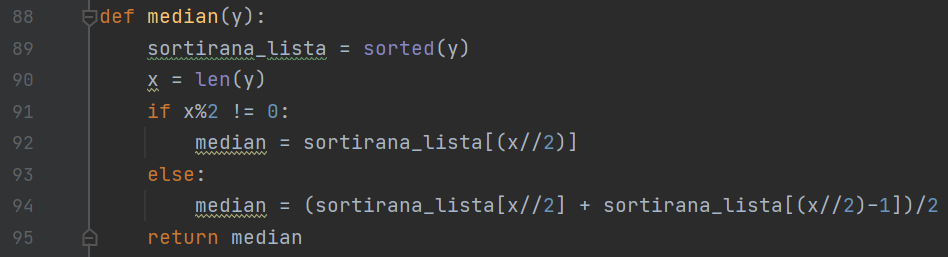
Sljedeće funkcije koje moramo napraviti jesu funkcije za izračun moda i medijana.



Slika 13. Funkcija za izračun mod-a

U ovoj funkciji preddefinirali smo 3 varijable. Prva varijabla nam je “sortirana\_lista” te smo u nju spremili listu koja je sortirana od manjeg prema većemu broju. U drugu varijablu “mode” spremili smo najmanji broj u toj sortiranoj listi. A kao treću “max\_count” varijabli spremili smo koliko puta se ponavlja najmanji broj u sortiranoj listi.

For petljom krećemo se po toj sortiranoj listi. If granom proveravamo da li je broj ponavljanja najmanjeg broja veći ili jednak prvom ponavljanju najmanjeg broja. Ako je to istinito, novo određeni broj ponavljanja, se zapisuje u varijablu “max\_count” a broj koji se ponavlja se zapisuje u varijablu “mode”. Ako se u listi ne ponavljaju brojevi, funkcija ispisuje da nema ponavljanja.

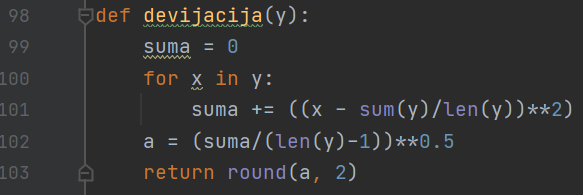


Slika 14. Funkcija za medijan

U funkciji median preddefinirali smo 2 varijable. Prva je “sortirana\_lista” u koju spremamo sortiranu listu argumenta, te “x” kao broj pozicija u tom argumentu. Nakon toga if grananjem ispitujemo da li je x neparan, ako je medijan izračunavamo kao cijelobrojno dijeljenje varijable x u toj sortiranoj listi. Ako je paran, medijan izračunavamo kao cjelobrojno dijeljenje varijable x te liste, dodajemo joj [(x//2)-1] te liste, te sve to podijelimo sa 2.

## Standardna devijacija

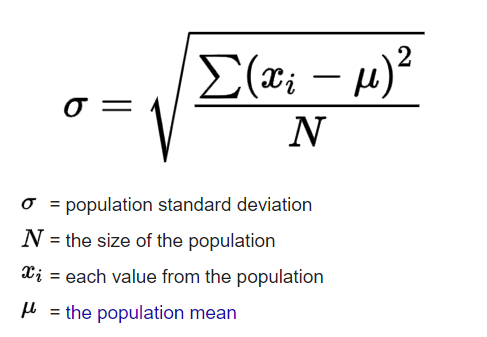
Sljedeća funkcija kojuu moramo napraviti jest funckija za standardnu devijaciju.



Slika 15. Funkcija za standardnu devijaciju

Standardna devijacija jest formula kojom izračunavamo mjeru raspršenosti podataka u listi.

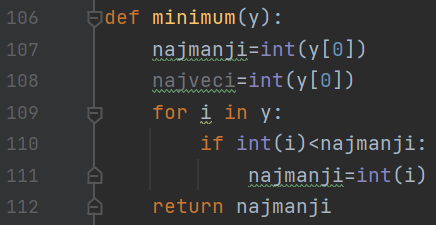
Za početak peddefinirati ćemo “suma = 0”. For petljom prolazimo kroz listu, te u varijablu suma svakim korakom pribrojimo (x – “suma čitavog argumenta” / ”duljina čitavog argumenta”)2. U novu varijablu koju smo nazvali “a” spremimo izraz .



Slika 16. Matematička formula standardne devijacije

## Minimum i maksimum

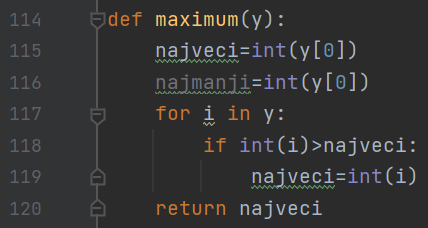
Zadnje funkcije koje ćemo napraviti jesu funkcije za izračun minimuma i maksimuma.



Slika 17. Funkcija za minimum

Kod funkcije za minimum preddefinirali smo dvije varijable, “najmanji” u koji ćemo spremati najmanji element kao integer i “najveci” u koji ćemo spremati najveći element isto kao integer.

For petljom kao i do sada, šetamo kroz listu. Prvim elementom ispitujemo da li je manji od preddefiniranog (u početku kao 0). Ako je, taj element se zapisuje u varijablu “najmanji”. Funkcija vraća varijablu “najmanji”.

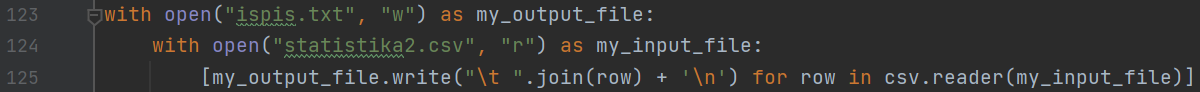


Slika 18. Funkcija za maksimum

Funkcija za izračun maksimuma će nam biti zapravo vrlo slična funkciji minimuma. Jedina razlika jest da ćemo svakim korakom ispitati da li je testni element veći od preddefiniranog (u početku kao 0). Ako je, taj novi element se zapisuje u varijablu “najveci”.   
Funkcija vraća varijablu “najveci”.

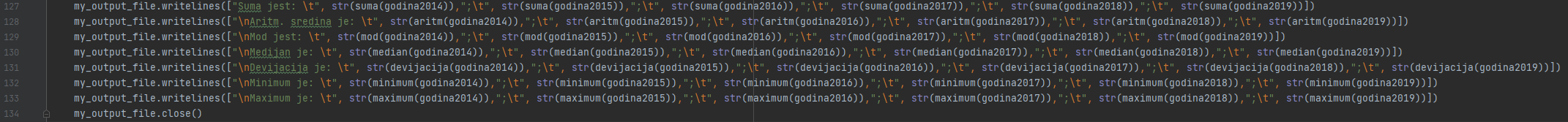
# Ispis rezultata pozivajući funkcije

Kod ispisa rezultata, morali smo kreirati novu .txt datoteku, kroz kod ispisati sve što je bilo u .csv datoteci ponovo u tu .txt datoteku te nakon toga ispisati sve rezultate.



Slika 19. Kod za ispis u novu .txt datoteku

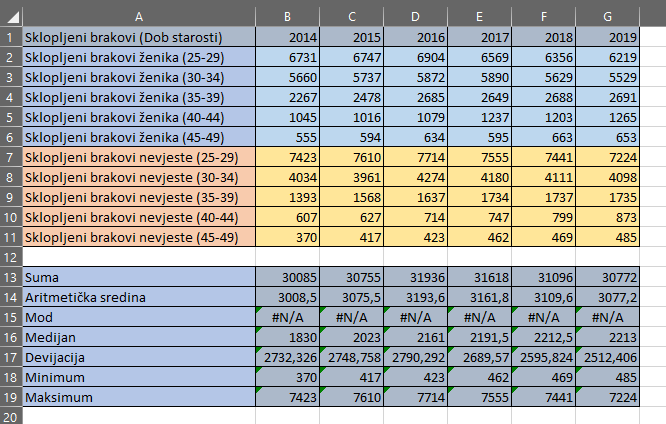
U ovome djelu koda, u početku smo kreirali te otvorili novu datoteku u koju zapisujemo, te smo ju nazvali “ispis.txt”. Nakon toga, otvorili smo .csv datoteku kao naš “input file”. Zadnjim djelom koda smo sve iz naše .csv datoteke ispisali u “ispis.txt” red po red.



Slika 20. Ispis rezultata u .txt datoteku

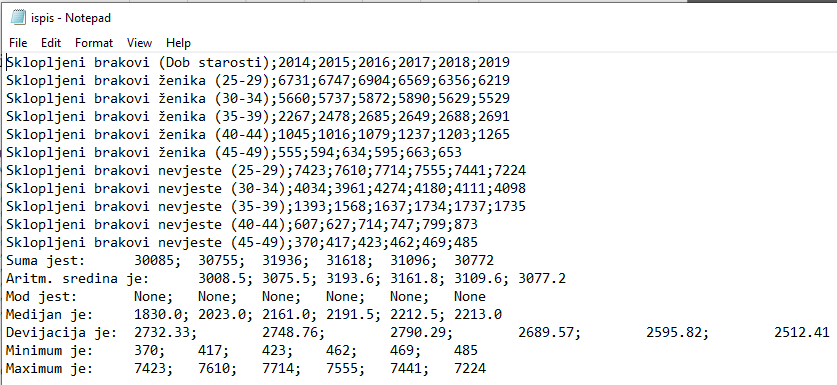
Na kraju našega koda, funkcijama “my\_output\_file.writelines()” zapisivali smo redove rezultata, pozivajući funkcije, te pazeći na tabelarno prikazivanje (\t).

# Usporedba rezultata sa excel-om



Slika 21. Svi podaci i rezultati u excel-u

Na slici 21. možemo vidjeti sve rezultate napravljene definiranim formulama u excel-u.



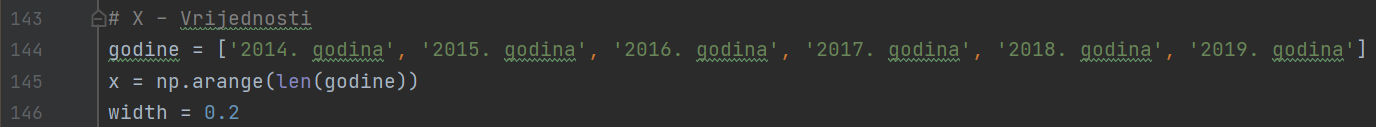
Slika 22. Svi podaci i rezultati u ispis.txt

Ovo su podaci zapisani u ispis.txt datoteci. Vidimo da su oni identični rezultatima koji smo provjerili u excel-u.

# Grafikon

Za izradu grafikona koristili smo stupčasti grafikon (bar). U tom grafikonu slikovno ćemo prikazati podatke vezane za naš prijašnji zadatak. Podatke koje ćemo prikazati jesu aritmetička sredina, devijacija, minimum i maksimum za sve pojedine godine.

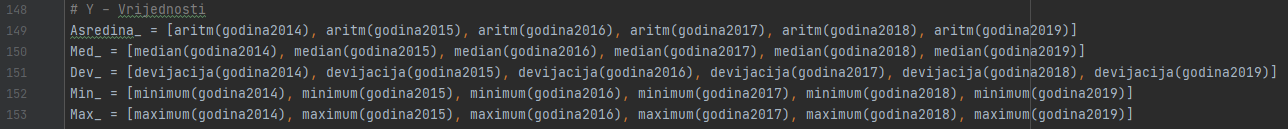
Za početak, definirati ćemo vrijednosti x – osi.



Slika . Vrijednosti x – osi

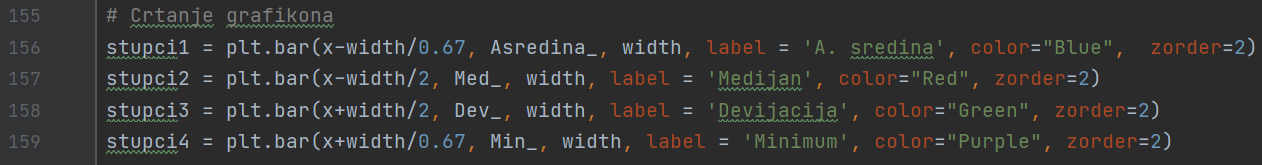
U varijablu “godine” zapisati ćemo listu stringova koji nam govore o kojoj godini je riječ. Kao varijablu “x” spremili smo duljinu te liste, odnosno koliko godina imamo za prikazati, u ovom slučaju je to 6. Kao “width” smo definirali širinu naših stupaca u grafikonu.

Zatim smo definirali vrijednosti y – osi.



Na y – osi će se nalaziti vrijednosti prethodno spomenutih podataka (aritm. sredina, devijacija, itd.). Za svaki podatak, definirali smo liste u koje spremamo vrijednosti pozivajući funkcije koje smo ranije definirali.

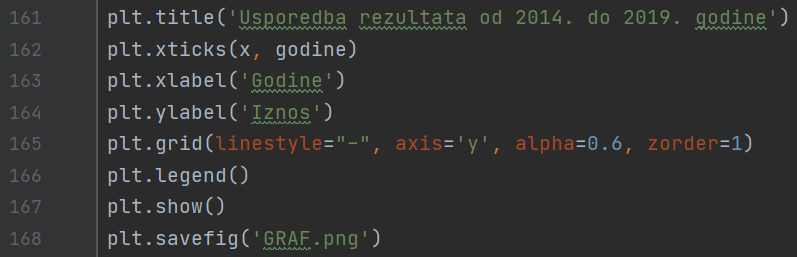
Kada smo definirali sve potrebne liste I varijable, krećemo sa crtanjem grafikona.



Slika . Crtanje grafikona

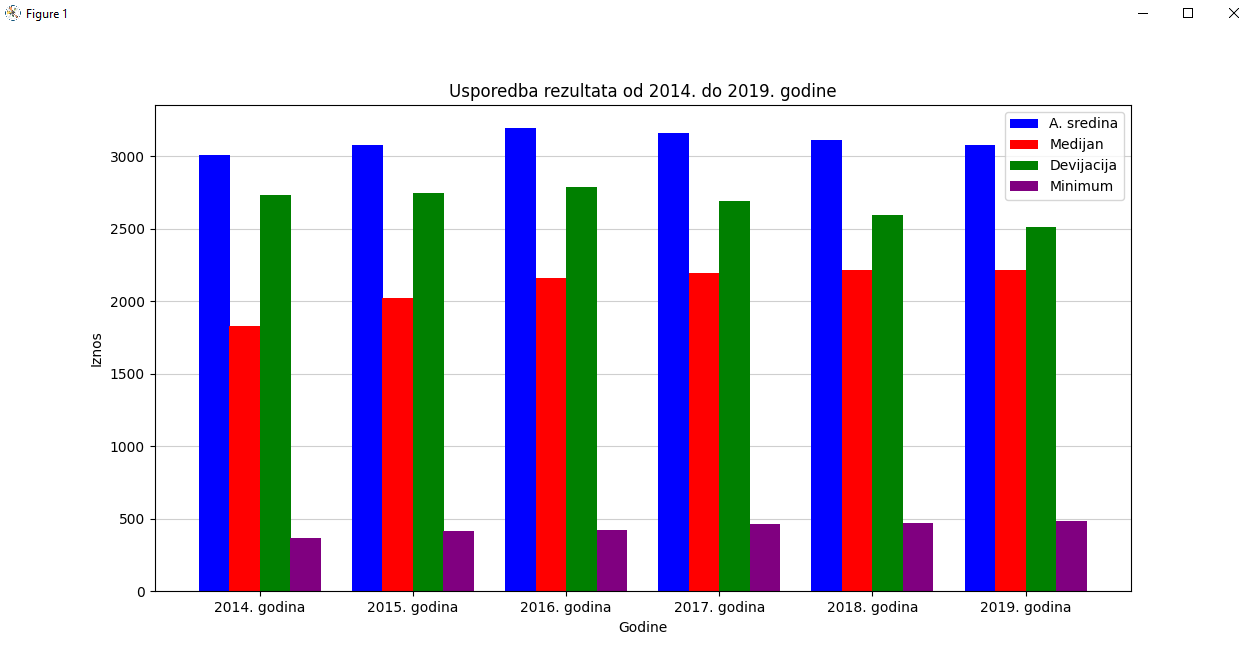
Na slici vidimo funkciju “plt.bar()” kojom crtamo pojedine stupce u grafikonu. Kao prvi argument stavljamo “x” koji nam pozicionira stupac na pojedine vrijednosti x-a. Drugi argument stavljamo vrijednosti podataka koje želimo prikazati (aritm. sredina, devijacija, itd.), treći argument jest širina stupaca koju smo ranije definiral. Zatim smo morali imenovati pojedine stupce, te smo im dodali različite boje. Kao zadnji argument koristili smo “zorder” koji možemo interpretirati kao prioritet prikazivanja na grafu (koristimo kako bi kasnije postavljali grid ispod stupaca).

Na kraju ćemo oblikovati prikaz našega grafikona.



Slika . Oblikovanje grafikona

Prvo ćemo dodati naziv našemu grafikonu, neka to nama bude “Usporedba rezultata od 2014. do 2019. godine”. Funkcijom “xticks” na x – osi zapisujemo parametre, te naziv tih parametara. Fukcijama “xlabel” i “ylabel” dodajemo nazive našim koordinatama. “grid” nam stavlja grid na naš koordinatni sustav, u ovom slučaju stil linije koju stavljamo jest puna, ona se prostire kao y vrijednost, argumentom “alpha” manipuliramo transparentnost linije, te i ovdje koristimo argument “zorder” kojemu postavljamo vrijednost 1 kako bi se taj grid crtao prvi na naš grafikon. Funkcijom “legend()” nam se izradi i legenda, te na kraju funkcijom “show()” crtamo naš grafikon. “savefig()” nam sprema grafikon u vanjsku datoteku.



Slika . Grafikon