Programski jezik Python u analizi podataka i rješavanju problema u inženjerstvu okoliša

Ricijaš, Viktor

Master's thesis / Diplomski rad

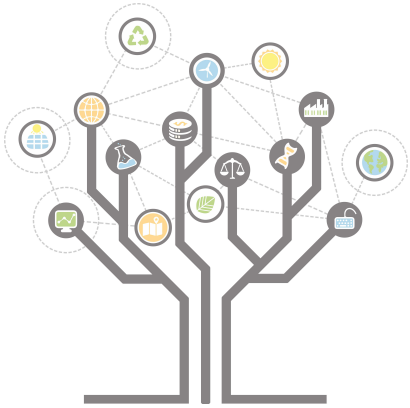
2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Geotechnical Engineering / Sveučilište u Zagrebu, Geotehnički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:130:112491

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**



Repository / Repozitorij:

Repository of Faculty of Geotechnical Engineering - Theses and Dissertations

zir.nsk.hr

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U ZAGREBU

GEOTEHNIČKI FAKULTET

VIKTOR RICIJAŠ

PROGRAMSKI JEZIK PYTHON U ANALIZI PODATAKA I RJEŠAVANJU PROBLEMA U INŽENJERSTVU OKOLIŠA

DIPLOMSKI RAD

VARAŽDIN, 2019.

SVEUČILIŠTE U ZAGREBU

GEOTEHNIČKI FAKULTET

PROGRAMSKI JEZIK PYTHON U ANALIZI PODATAKA I RJEŠAVANJU PROBLEMA U INŽENJERSTVU OKOLIŠA

DIPLOMSKI RAD

KANDIDAT:

VIKTOR RICIJAŠ

MENTOR:

Doc. dr. sc.MARIO GAZDEK

KOMENTOR:

Doc.dr.sc BOJAN ĐURIN

NEPOSREDNI VODITELJ:

Dr.sc. DAVOR STANKO

VARAŽDIN, 2019.

Sveučilište u Zagrebu
Geotehnički fakultet

ZADATAK ZA DIPLOMSKI RAD

Pristupnik: VIKTOR RICIJAŠ

Matični broj: 212 - 2017./2018.

Smjer: GEOINŽENJERSTVO OKOLIŠA

NASLOV DIPLOMSKOG RADA:

PROGRAMSKI JEZIK PYTHON U ANALIZI PODATAKA I RJEŠAVANJU PROBLEMA U INŽENJERSTVU OKOLIŠA

Rad treba sadržati: 1. Uvod
2. Općenito o programskom jeziku Python
3. Primjeri iz inženjerstva okoliša
4. Prikaz rješavanja problema iz inženjerstva okoliša pomoću Pythona
5. Diskusija
6. Zaključak

Pristupnik je dužan predati mentoru jedan uvezen primjerak diplomskog rada sa sažetkom. Vrijeme izrade diplomskog rada je od 45 do 90 dana.

Zadatak zadan: 01.04.2019. Rok predaje: 04.07.2019.

Mentor:	Drugi mentor/komentor:	Predsjednik Odbora za nastavu:
Doc.dr.sc. Mario Gazdek	Doc.dr.sc. Bojan Đurin	Izv.prof.dr.sc. Igor Petrović

Neposredni voditelj:

Dr.sc. Davor Stanko

REPUBLIKA HRVATSKA
GEOTEHNIČKI FAKULTET
SVEUČILIŠTE U ZAGREBU
VARAŽDIN

IZJAVA O AKADEMSKOJ ČESTITOSTI

Izjavljujem i svojim potpisom potvrđujem da je diplomski rad pod naslovom

PROGRAMSKI JEZIK PYTHON U ANALIZI PODATAKA I RJEŠAVANJU PROBLEMA U INŽENJERSKIM OKOLIŠA

(naslov diplomskog rada)

rezultat mog vlastitog rada koji se temelji na istraživanjima te objavljenoj i citiranoj literaturi te je izrađen pod mentorstvom **Doc. dr. sc. Mario Gazdek**. Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno da je prepisan iz necitiranog rada te da nijedan dio rada ne krši bilo čija autorska prava. Izjavljujem također, da nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

U Varaždinu, 4.7.2019.

VIKTOR RICIJAŠ
(Ime i prezime)

(Vlastoručni potpis)

Sažetak

Autor: Viktor Ricijaš

Naslov rada: Programski jezik Python u analizi podataka i rješavanju problema u inženjerstvu okoliša

Sažetak rada: U ovom radu nastojano je prikazati prednosti i nedostatke Python programskog jezika u analizi i prikazu podataka iz inženjerstva okoliša poput: a) mjerenja koncentracije radona u kućama, b) određivanje dominantne frekvencije vibracijske ploče, c) analiza zapisa potresa s raznih seizmoloških postaja, d) korelacije geofizičkih i in-situ istraživanja, e) analiza prosječne temperature, e) statistička analiza pomoću RAPS metode u praćenju kakvoće otpadne vode, i, f) analiza srednje temperature za vremensko razdoblje od godinu dana. RAPS (*Rescaled Adjusted Partial Sums*) metoda zasnovana je na analizi vremenske raspodjele otjecanja metodom sumarne krivulje odstupanja. Objašnjene su metode analize i koraci u pisanju koda kako bi olakšali i omogućili reproduciranje koda za gore navedene probleme.

Ključne riječi: Python, RAPS, Inženjerstvo okoliša, geofizička istraživanja

Sadržaj

1. UVOD 1
2. OPĆENITO O PROGRAMSKOM JEZIKU PYTHON 2
2.1. Instalacija Pythona na Windows operativnom sustavu 2
3. PRIMJERI IZ INŽENJERSTVA OKOLIŠA 6
3.1. Radon u okolišu 6
3.2. Uloga vibracija u okolišu i primjena Fourierove transformacije 6
3.3. Potresi u okolišu 7
3.4. Korelacije geofizičkih i in-situ istraživanja 8
3.5. RAPS metoda u praćenju kakvoće otpadne vode 9
3.6. Srednja temperatura za određeno vremensko razdoblje 10
4. PRIKAZ RJEŠAVANJA PROBLEMA IZ INŽENJERSTVA OKOLIŠA POMOĆU PYTHONA 11
4.1. Analiza mjerenja koncentracije radona 11
4.2. Određivanje dominantne frekvencije vibracijske ploče 14
4.3. Prikaz zapisa potresa sa seizmoloških stanica 16
4.4. Analiza podataka korelacije između v_s i N_{DPH} 20
4.5. Analiza i prikaz podataka RAPS metode u praćenju kakvoće otpadne vode 24
4.6. Analiza srednje temperature za vremensko razdoblje od godinu dana 25
5. DISKUSIJA 29
6. ZAKLJUČAK 31
LITERATURA 32
POPIS SLIKA 34
POPIS TABLICA 35
DODATAK 36

1. UVOD

U današnjem svijetu sve se više ističe važnost očuvanja i zaštite okoliša. Napretkom tehnologije postaju nam dostupni različiti alati i metode za prikupljanje i obradu podataka u inženjerstvu okoliša. Pošto inženjerstvo okoliša zahvaća sve sastavnice okoliša (voda, tlo i atmosfera) potrebne su razne i vrlo često specifične metode prilikom rješavanja problema vezanih uz određene sastavnice okoliša. Zbog složenosti i raznovrsnosti problema koje je potrebno analizirati primjenjuju se statistički i matematički modeli kojima pokušavamo što točnije prikazati i predvidjeti ponašanje sustava. Za to se najčešće koriste gotovi programi za statističku analizu kao što su: Microsoft Office Excel, Sigma Plot, Origin, Statistica ili složeniji programi poput Matlab-a. Tu svakako treba napomenuti da su to sve programi koji se plaćaju od strane korisnika.

Programski jezici posebice se ističu u prilagođavanju programa za izvršavanje različitih zadataka kao i to da treba razne analize i obrade ponavljati više puta. Također, omogućuju i obradu velikog broja podataka štedeći vrijeme. U zadnjih 40-ak godina programski jezici intenzivno se razvijaju kako bi njihova primjena bila što raznovrsnija i u današnje vrijeme postaju nezaobilazan i moćan alat u obradi podataka. Programski jezik je umjetan jezik stvoren za izdavanje naredbi računala koristi strogo definirana sintaktička pravila za izdavanje naredbi. Skupina naredaba koje obavljaju neku funkciju može se ujediniti u proceduru (niz naredaba koje se izvršavaju zadanim redoslijedom), koja se pokreće kao cjelina, pa su složeni programi sastavljeni od pozivanja unaprijed pripremljenih procedura. Programski jezik Python danas se koristi u svim sferama znanosti kao i ostalim strukama za rješavanja raznih problema, prvenstveno zato što je „open source“, te se intenzivno razvija od strane korisnika.

U ovom radu korišten je programski jezik Python za prikazivanje i obradu podataka iz različitih problema inženjerstva okoliša poput: a) mjerenja koncentracije radona u kućama, b) određivanje dominantne frekvencije vibracijske ploče, c) analiza zapisa potresa s raznih seizmoloških postaja, d) korelacije geofizičkih i in-situ istraživanja, e) analiza prosječne temperature, e) statistička analiza pomoću RAPS metode u praćenju kakvoće otpadne vode, i, f) analiza srednje temperature za vremensko razdoblje od godinu dana. Cilj rada je pokušati uočiti prednosti i moguće nedostatke u analizi podataka u Pythonu za rješavanje jednostavnih i složenih problema iz inženjerstva okoliša.

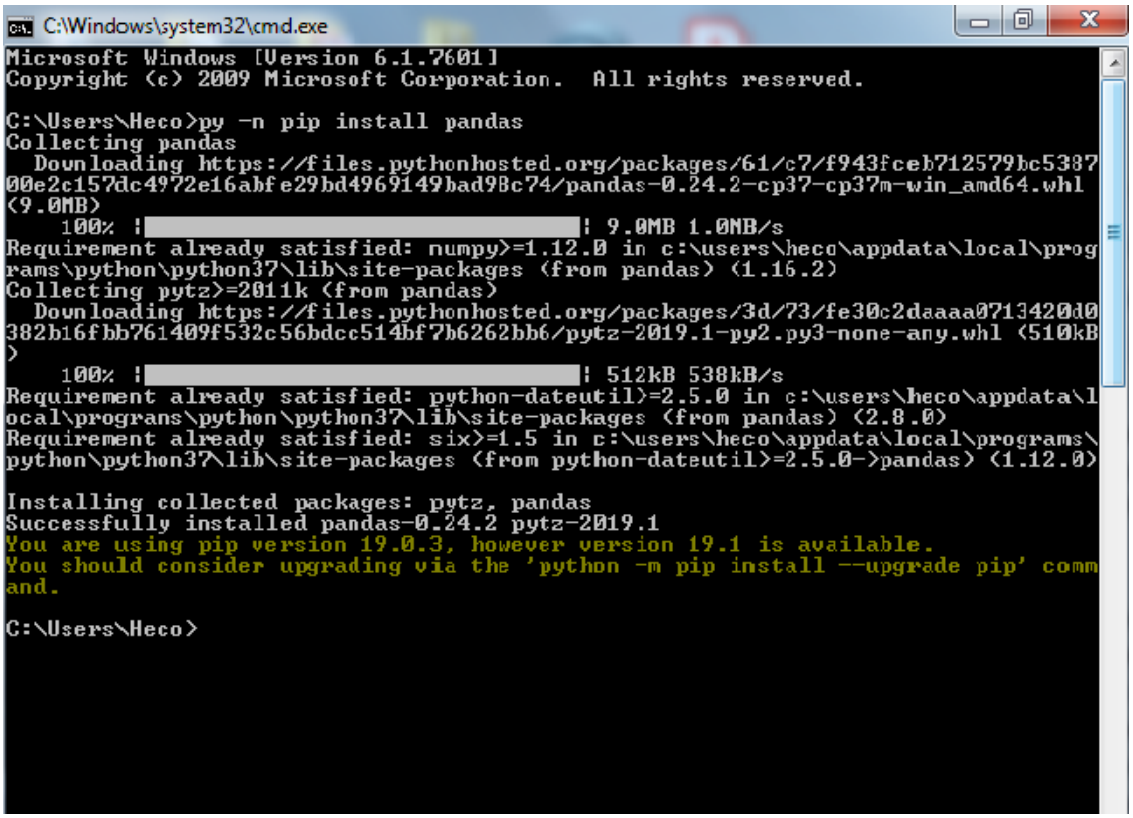
2. OPĆENITO O PROGRAMSKOM JEZIKU PYTHON

Python kao ideja započela je ranih 1980-ih, dok je implementacija započela 1989.godine, od strane Guida van Rossuma u Nizozemskoj. Python spada u grupu objektno-orijentiranih programskih jezika visoke razine čiji kod je jednostavan za čitanje, te ga odlikuje veoma jednostavna sintaksa. Uz to ima i ugrađene alate koji programeru olakšavaju neke zadaće poput automatskog upravljanja memorijom (Tulchak i Marchuk, 2016.). Najveća prednost programskog jezika Python u odnosu na druge programske jezike jest u tome što je „open source“, tako da je dopuštena distribucija i izmjena, što znači da ljudi diljem svijeta mogu sudjelovati u razvoju Python-a. Formirane su i besplatne biblioteke s raznim edukativnim materijalima, tutorijalima i primjerima koda. Moguće ga je pokrenuti na raznim platformama i operacijskim sustavima poput Windows, Linux i dr.

2.1. Instalacija Pythona na Windows operativnom sustavu

Instalacija programskog jezika Python može predstavljati izazov za osobu koja nema iskustva u programskim jezicima jer nakon instalacije je potrebno u Cmd-u (Command Prompt) izdati naredbu Pythonu da instalira potrebne pakete kako bi mogao izvršiti određene složene operacije. Za potrebe ovog rada korištena je verzija 3.7.3 dostupna na linku (https://www.python.org/downloads/release/python-373/).

Cmd se najjednostavnije pokreće upisom u tražilicu na početnom izborniku. Nakon što se pokrene Cmd treba se izdati naredba za instaliranje „pip“ alata. Pip je alat za instaliranje Python paketa te se pomoću njega instaliraju sljedeći alati: pandas (služi za statističku obradu podataka), matplotlib (ispis i crtanje grafova), te biblioteke numpy i scipy. Numpy i scipy biblioteke omogućuju Pythonu korištenje zahtjevnih i složenih matematičkih funkcija. Za instalaciju u Cmd-u je potrebno unijeti točnu lokaciju instalacije Pythona te pokrenuti naredbu „python get-pip.py“. Nakon toga program zaprima podatke s interneta i instalira alat. Drugi korak je instalacija pandas paketa, koja se izvršava naredbom „py –n pip install pandas“. Proces i konačan rezultat instalacije pandas paketa je prikazan na Slici 1.



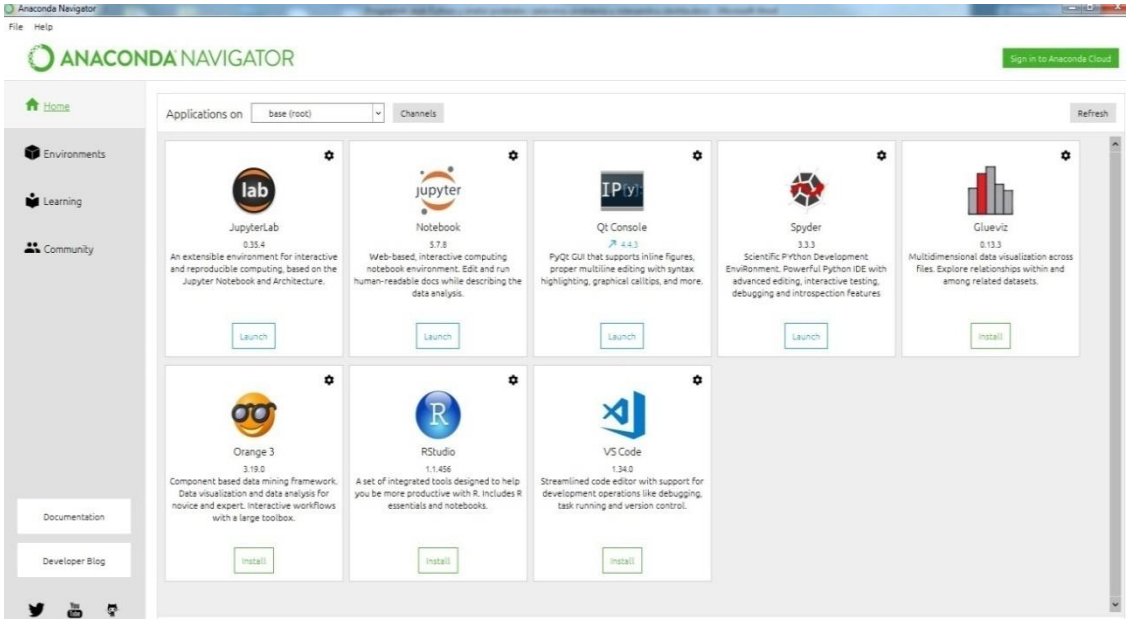
Slika 1. Prikaz instalacije pandas paketa za statističku obradu

Nakon instalacije pandas programa potrebno je instalirati matplotlib paket kako bi mogli crtati grafove i NumPy i SciPy biblioteke (Slika 2). Instalacija svih ovih paketa omogućuje nam da pomoću Python-a možemo analizirati složene i raznovrsne probleme iz inženjerstva okoliša. Python dolazi sa svojom verzijom radnog okruženja zvanom IDLE (Integrated DeveLopment Environment) te služi za testiranje jednostavnih kodova. Za složenije programe poželjno je kod pisati u drugim radnim okruženjima. U ovom radu koristio sam „Jupyter Notebook“ i „Spyder“ radno okruženje. Također, važno je napomenuti da su dostupna još i mnoga druga radna okruženja poput: PyCharm, Thonny, Atom, Visual Studio, itd..., specijalizirana za rješavanje određenih problema.

Slika 2. Prikaz instalacije NumPy, SciPy i matplotlib paketa pomoću pip-a

Sve ovo može djelovati dosta zbunjujuće nekome tko se ne bavi programiranjem. Postoji jednostavniji način kako instalirati sve gore navedene pakete, biblioteke i radna okruženja, a to je instalacija Anaconda platforme koja je dostupna na linku (https://www.anaconda.com/distribution/). Anaconda platforma dolazi sa svim potrebnim paketima te sadrži osam alata: JupyterLab, Jupyter Notebook, QTConsole, Spyder, Glueviz, Orange App, Vs Code i RStudio (Slika 3).

Spyder je aplikacija unutar Anaconda platforme koja pruža znanstveno-razvojno okruženje koje olakšava napredno uređivanje, interaktivno testiranje, otklanjanje pogrešaka i mogućnost introspekcije (Kadiyala i Kumar, 2017). Još jedna prednost Anaconda platforme je što sažima na jedno mjesto materijale za učenje, razne primjere i kodove, forume te mnogo korisnih poveznica kako bi što brže i jednostavnije koristili Python programski jezik. Također, dodatne informacije se mogu pronaći na poveznicama pod nazivom Learning i Community koje su ujedno prikazane na Slici 3.



Slika 3. Izgled sučelja Anaconda platforme i dostupnih alata

3. PRIMJERI IZ INŽENJERSTVA OKOLIŠA

3.1. Radon u okolišu

Radon (^{222}Ra) je radioaktivni element bez boje i mirisa kojeg u prirodi nalazimo u plinovitom stanju kao produkt raspadanja urana. Koncentracija radona u zraku koji se udiše prilično je niska, no može doseći visoku razinu unutar stambenog prostora posebice ako nema dobru ventilaciju. Najčešće prodire u stambeni prostor na kontaktu temeljnog tla i podruma u samu građevinu. Vrsta tla ispod građevine igra veliku ulogu koncentraciji radona: ako se radi o tlu visoke propusnosti, poput pjeskovitog tla, radon se lako može kretati i time uzrokovati visoku koncentraciju unutar zatvorenih prostorija.

Radon se smatra jednim od najčešćih uzročnika karcinoma pluća (Cothern i Smith, 1987). Uređaji za mjerenje produkata raspada radona u zraku rade na principu usisavanja zraka pomoću usisne pumpe preko aerosolnog filtera na kojemu se zadržavaju produkti raspada radona, te se filter uzima iz uređaja i pomoću detektora alfa čestica mjeri aktivnost raspada. Rezultat mjerenja je krivulja koja prikazuje smanjivanje koncentracije radona u vremenu (Trešnjo i Adrović, 2005).

Pomoću Python-a prikazat će se analiza mjerenih podataka koncentracije radona te veza s temperaturom i vlagom u stambenom prostoru.

3.2. Uloga vibracija u okolišu i primjena Fourierove transformacije

Prilikom raznih istraživanja u inženjerstvu okoliša primjenjuju se brojni instrumenti i metode za dobivanje određene vrste podataka koji se obrađuju različitim statističkim metodama kako bi uočili povezanost. Vibracijska ploča može se primijeniti u geotehnici za sijanje materijala i određivanje granulometrijskog sastava, te se koristi za određivanje drugih parametara materijala poput maksimalne gustoće. Prema Barnesu (1993) dominantna frekvencija obično se procjenjuje brojanjem relativnih maksimuma unutar nekog intervala, tj. frekvencija koja prenosi najviše energije. Jedan od primjera je određivanje dominantne frekvencije vibracijske ploče korištenjem brze Fourierove transformacije. Brza Fourierova transformacija (FFT) ima široku primjenu u obradi signala i konceptualnim frekvencijskim analizama. Omogućuje pretvaranje vremenskog signala u frekvencijski signal. Matematička formula Fourierove transformacije glasi:

$$S(f) = \int_{-\infty}^{\infty} s(t)e^{-j2\pi ft}dt \quad (1)$$

gdje je *s(t)* valni oblik koji se razlaže u zbroj sinusoida, *S(f)* je Fourierova transformacija vremenskog signala s(t).

FFT izračunava diskretnu Fourierovu transformaciju (DFT) koristeći smanjeni broj aritmetičkih operacija u usporedbi s procjenom DFT-a. Metoda je učinkovita, jer eliminira redundancije koje nastaju dodavanjem određenih vrijednosti sekvenci podataka nakon što su pomnožene istim faktorima fiksnih kompleksnih konstanti tijekom procjene različitih DFT koeficijenata transformacije (Elliott, 1987). Česta područja primjene su konvencionalni radari, aplikacije za obradu signala, sonari, spektroskopija, metalurška analiza, analiza nelinearnih sustava i slično.

Bit Fourierove transformacije valnog oblika je razložiti ili odvojiti valni oblik u zbroj sinusoida različitih frekvencija.Ako se suma sinusoidi poklapa s izvornim valnim oblikom, tada je određena Fourierova transformacija tog valnog oblika (Oran i Brigham, 1988).

Pomoću Python-a prikazat će se spektralna analiza određivanje dominantne frekvencije vibracijske ploče. Također, FFT se može primijeniti i za spektralnu obradu kod mjerenja buke i vibracija kao štetnih (a ponekad i poželjnih) utjecaja u okolišu.

3.3. Potresi u okolišu

Potresi mogu imati znatan utjecaj na okoliš i čovjeka te kao se je njihovo istraživanje i analiza od velike važnosti. Kod potresa dolazi do naglog oslobađanja energije u unutrašnjosti Zemlje te ih dijelimo prema uzroku nastajanja i izvoru energije na tektonske, vulkanske i urušne. Izvor tektonskih potresa je naglo oslobađanje akumulirane elastične energije te je oko 90% svih potresa tog tipa. Vulkanski potresi nastaju naglim oslobađanjem termokemijske energije iz magme pri prodoru na površinu. Urušni potresi nastaju prilikom urušavanja šupljina u Zemljinoj kori, koje nastaju djelovanjem vode na materijale topive u vodi poput kalcita (Markušić, 2003).

Potresi antropogenog uzroka (Strelec i Jug, 2017) nastaju prilikom građevinskih i eksploatacijskih miniranja, te mogu u urbanim sredinama uznemiriti stanovništvo i oštetiti postojeće objekte koji se nalaze u blizini. Stoga je poželjno provesti prethodna

istraživanja, u vidu mjerenja seizmičkih efekata probnih miniranja, kako bi se što preciznije odredio režim sigurnih miniranja i intenzitet oscilacija tla.

Seizmograf je uređaj koji bilježi potres tako da registrira pomak, brzinu ili akceleraciju tla. Obično se za prikaz zapisa potresa sa seizmoloških postaja koriste gotovi programski sustavu od strane proizvođača seizmografa. Pomoću Pythona prikazat će se zapis potresa u Novom Marofu (30.6.2016) s više seizmoloških postaja.

3.4. Korelacije geofizičkih i in-situ istraživanja

Geofizička istraživanja su vrlo bitan alat u istraživanju geomedija a samim time i u inženjerstvu okoliša zbog svoje široke primjene. Istraživanja su jeftina, brza i neinvazivna a kao rezultat daju bitne parametre tla poput sastava i položaja slojeva tla, razinu podzemne vode, dinamička svojstva tla, prisutnost nekog onečišćivala itd. Najčešće korištene metode u geofizici su gravimetrijske, magnetometrijske, geoelektrične, seizmičke, radiometrijske i georadar (Milsom, 2003). MASW (Multi-channel Analysis of Surface Waves) višekanalno snimanje omogućuje učinkovitu identifikaciju i izolaciju smetnji koristeći 24 ili više geofona, radi se disperzijska analiza da bi se dobila brzina posmičnih valova (v_s) po dubini (Park, Miller i Xia, 1999). Ova metoda omogućava mjerenje i određivanje brzine posmičnih valova v_s koja se smatra bitnom fizikalnom veličinom za procjenu dinamičkih svojstava tla (Strelec i sur. 2016). Kod SPT pokusa se broje udarci potrebni za penetraciju sonde od 30 cm pri padu utega od 63 kg s visine 76 cm u određenim točkama sondažnog profila dok kod DPH pokusa se broje udarci potrebni za penetraciju sonde od 10 cm pri padu utega od 50 kg s visine 50 cm (Strelec i sur. 2016). Uz geofizičke metode koristila su se još in-situ ispitivanja standardnim penetracijskim pokusom (SPT) i teškom udarnom sondom (DPH) kako bi se uspostavila korelacijska veza između podataka pomoću:

$$V_S = A \cdot N_{SPT,DPH}^{B} \tag{2}$$

Pomoću Pythona prikazat će se usporedna analiza obrade podataka iz rada Strelec i sur. (2016) s obzirom da su podaci u tom radu analizirani u Excel-u.

3.5. RAPS metoda u praćenju kakvoće otpadne vode

Voda je vrlo bitna sastavnica okoliša jer omogućava sav život na zemlji. Zbog kruženja vode u prirodi ,voda prolazi kroz razne medije poput tla i atmosfere te često na svom putu dolazi u kontakt s raznim tipovima onečišćenja. U analizi kakvoće otpadnih voda gledaju se vrijednosti KPK, BPK_5 i količina suspendiranih tvari prije i nakon ulaska u jedinicu uređaj za pročišćavanje otpadnih voda.

Kemijska potrošnja kisika (KPK) je masena koncentracija kisika, ekvivalentna količini bikromata, potrebna pod određenim uvjetima za oksidaciju otopljene i suspendirane tvari ako se uzorak vode obrađuje takvim oksidansom (Tušar, 2009). Biokemijska potrošnja kisika (BPK5) je masena koncentracija otopljenog kisika potrošena unutar pet dana pod određenim uvjetima biološkom oksidacijom organske i/ili anorganske tvari u vodi (Tušar, 2004). Suspendirane tvari su tvari koje se pod određenim uvjetima uklanjaju filtracijom ili centrifugiranjem (Tušar, 2009). RAPS metoda nam pomaže u analizi i prikazu podataka u ključnim vremenskim intervalima.

RAPS (*Rescaled Adjusted Partial Sums*) metoda zasnovana je na analizi vremenske raspodjele otjecanja metodom sumarne krivulje odstupanja. Vizualni grafički prikaz zasnovan na RAPS transformaciji pogodan je jer omogućava prevladavanje malih sustavnih i slučajnih promjena, grešaka i varijabilnosti u analiziranom vremenskom nizu. Grafički prikaz RAPS-a upućuje na postojanje više podrazdoblja sa sličnim karakteristikama, većeg broja trendova, naglih skokova ili padova vrijednosti, neregularnih fluktuacija, postojanje periodičnosti u analiziranom vremenskom nizu (Bonacci, 2010).

Prema Kim i Ryu (2019) izraz za proračun RAPS-a definiran je pomoću izraza:

$$RAPS_k = \sum_{t=1}^{k} \frac{Y_t - \bar{Y}}{S_Y} \tag{3}$$

gdje je Y_t prosječna vrijednost cijelog skupa podataka, S_Y je standardna devijacija cijelog skupa podataka, k (k = 1, 2…, n) predstavlja brojač tijekom sumiranja, n je broj podataka u vremenskom nizu.

U ovom radu bit će prikazana statistička analiza korištenjem RAPS metode na setu podataka za praćenje kakvoće vode.

3.6. Srednja temperatura za određeno vremensko razdoblje

Klimatske promjene predstavljaju izrazite rizike za izvore ljudske prehrane kao što su vodni resursi i poljoprivreda. Tijekom protekla tri stoljeća ljudske aktivnosti su znatno utjecale na ponašanje Zemljinih sustava na globalnoj razini. Kumulativni učinci antropogenih aktivnosti utječu na strukturu i funkcioniranje zemaljskih sustava, te imaju znatan utjecaj na litosferu, atmosferu i obalna područja, ponajviše izgaranje fosilnih goriva bilo je najistaknutiji čimbenik klimatskih promjena u posljednjih par desetljeća (Mohorji i Almazroui, 2017).

IPCC (Intergovernmental Panel on Climate Change) ustanovio je da približan porast prosječne globalne temperature zraka na Zemlji, počevši od 1900. godine, iznosi oko 0,8±0,1 ºC (Bonacci, 2010). U posljednjih nekoliko desetljeća vođena je statistika globalne temperature, ugljičnog dioksida, nitrata, sumpora, metana i drugih kemijskih elemenata, stoga je sada moguće numeričke vrijednosti tretirati u obliku vremenskih serija različitim metodologijama kako bi se istražile unutarnje sustavne strukture kao što su trendovi, koji pružaju znanstvene informacije za bolje modeliranje, predviđanje i kontrolni mehanizam dotičnog fenomena (Mohorji i Almazroui, 2017). Programski jezik Python omogućuje razna numerička modeliranja s velikim setom vremenskih nizova, što će se pokazati na primjeru srednje dnevne temperature za razdoblje od godinu dana.

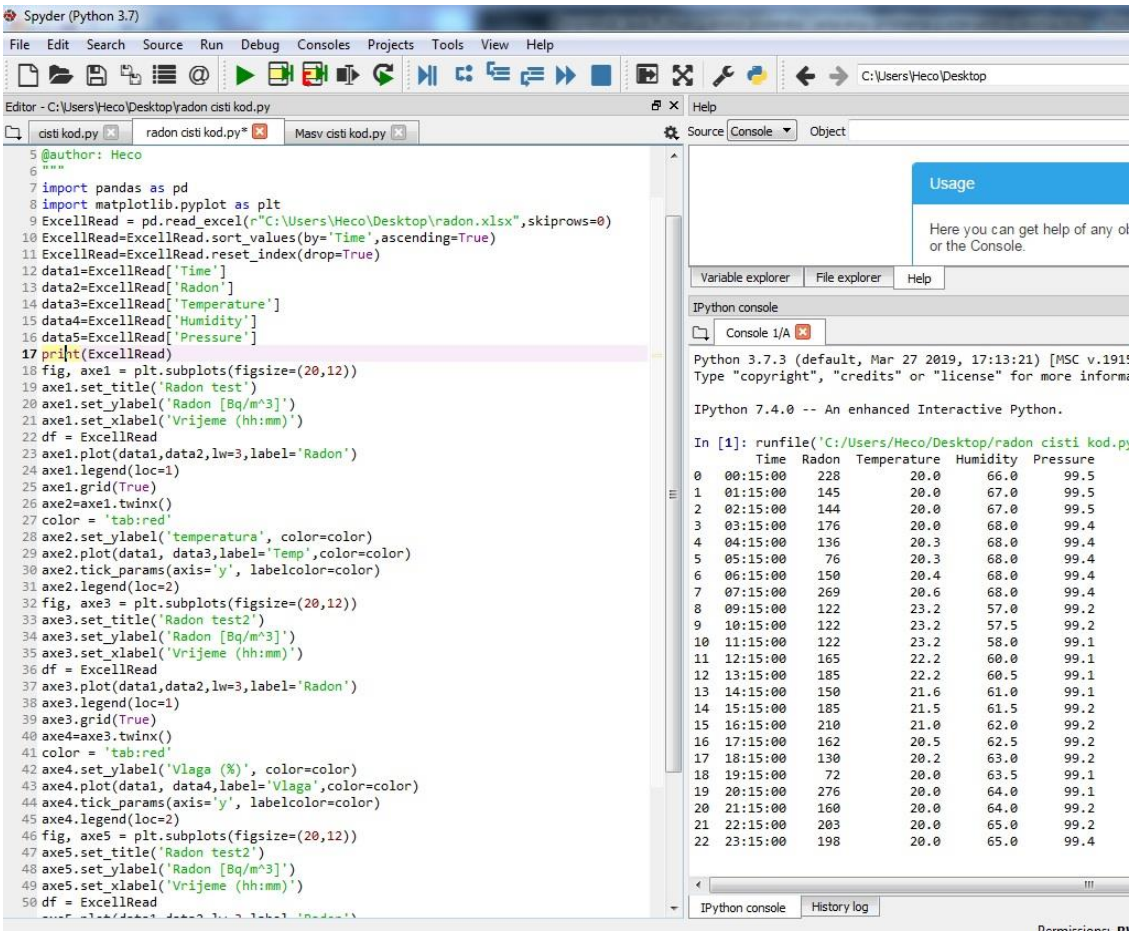
4. PRIKAZ RJEŠAVANJA PROBLEMA IZ INŽENJERSTVA OKOLIŠA POMOĆU PYTHONA

4.1. Analiza mjerenja koncentracije radona

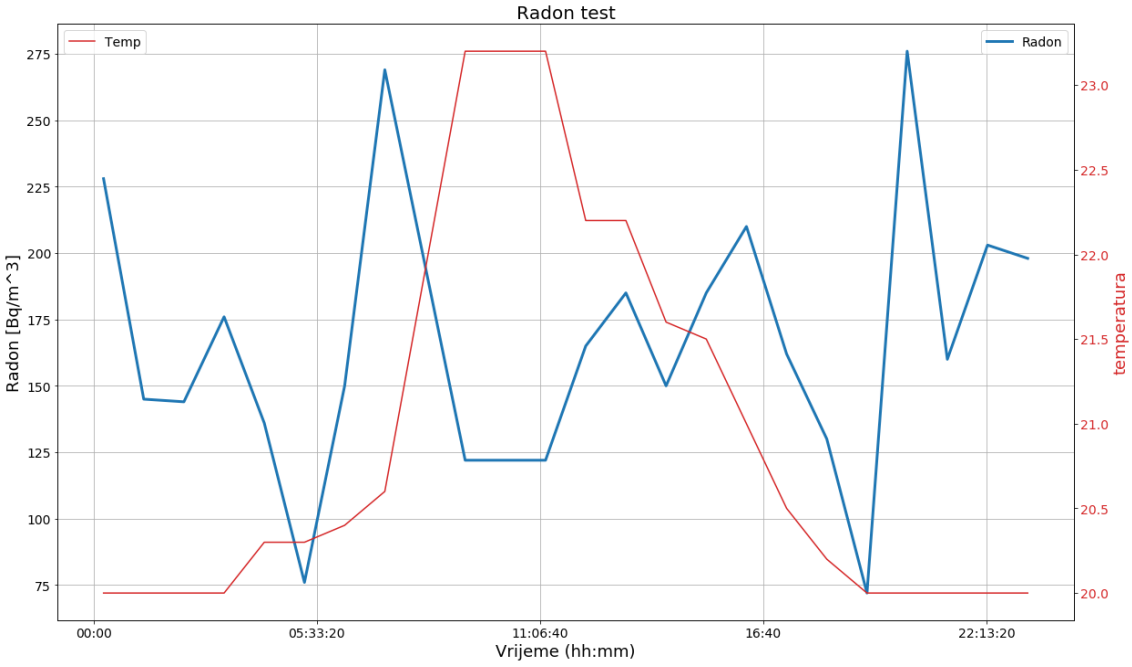
U ovom primjeru prikazana ja analiza podataka mjerenja koncentracije radona, temperature i vlage u stambenom prostoru obrađena pomoću Python programskog jezika, te su objašnjeni svi koraci u pisanju koda.

Nakon pokretanja radnog okruženja, naredbama (1 i 2) učitan je paket za statističku obradu pandas kao kratica „pd“ i paket matplotlib.pyplot kao kratica „plt“,naredbom (3) učitani su podaci iz Excel datoteke koja se nalazi na upisanoj lokaciji. Naredbama (4 i 5) poredane su vrijednosti u stupcu „Time“ od nižih prema višima i time su dobiveni podaci u vremenskom nizu. Naredbama (6-10) definiran je svaki pojedini stupac podataka određenim imenom, u ovom slučaju izabrano je ime „data od 1 do 5“. Ovaj korak nije nužan ali služi za lakše snalaženje u kodu i izbjegavanje mogućih grešaka u sintaksi. Važno je napomenuti da svaki izraz u kodu mora biti točno definiran inače programski jezik ispisuje pogrešku.

Naredba (11) ispisuje učitane podatke u program (Slika 4). Nakon unosa podatka iz Excel tablice slijedi grafički prikaz tih podataka (Slika 5). Naredbe (12-15) služe za određivanje oblika i izgleda područja u kojem će se prikazati podaci, naredba (16) služi za grafički prikaz dva seta podataka pod nazivima „data1 i 2“, naredbe (17 i 18) služe za prikaz legende i mreže (grid). Naredba (19) je bitna jer nam omogućuje dodavanje druge y-osi s desne strane, naredbe (20 i 21) određuju boju i naziv druge y-osi, naredbe (22 i 23) zadaju parametre za crtanje drugog grafa uzimajući u obzir drugu y-os, te naredba (24) prikazuje još jednu legendu.

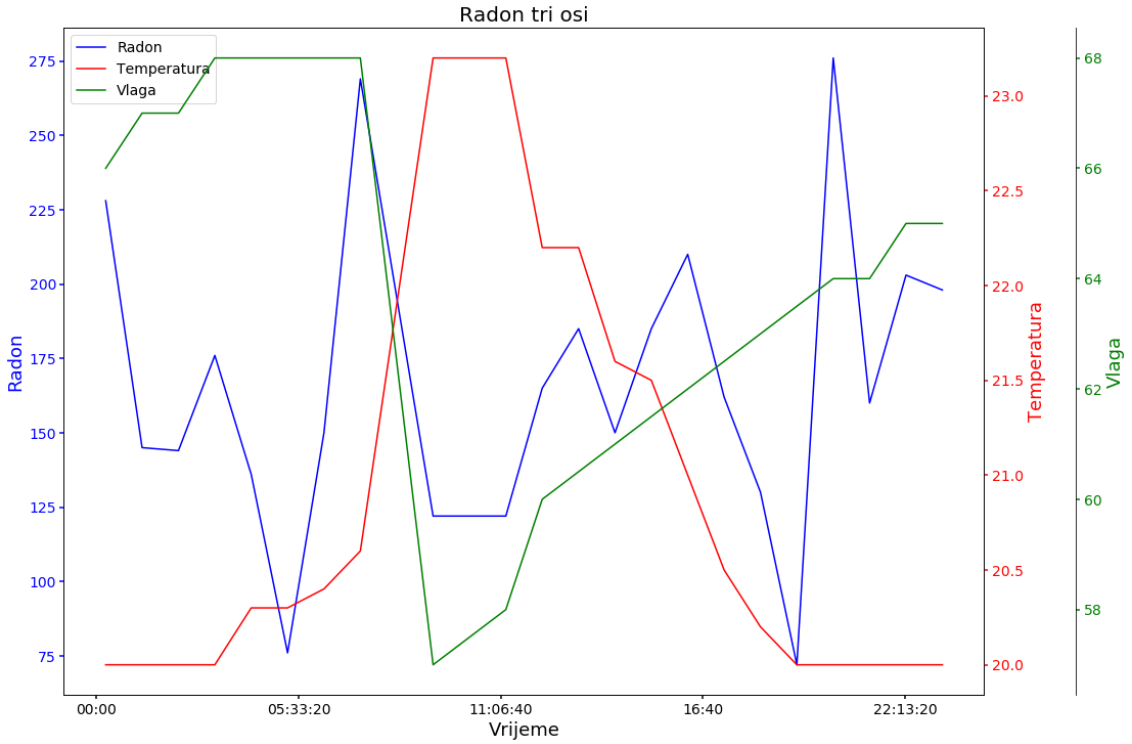


Slika 4. Prikaz Spyder radnog okruženja s ispisanim podacima iz Excel tablice



Slika 5. Grafički prikaz dva seta podataka (Radon i temperatura) u vremenu

Naredbe (25-50) ponavljaju ove prethodne korake još dva puta uzimajući druge podatke kako bi dobili prikaz podataka vezanih uz tlak i vlagu. Za razliku od Excel-a u Python programskom jeziku je moguće nacrtati graf s tri y-osi u različitim mjerilima, naredbe (51-73). Naredba (51) definira funkciju koja služi kako bi bilo izbjegnuto preklapanje dvije y-osi s desne strane, naredbe (52 i 53) prikazuju područje grafa i pomiču ga u desnu stranu, naredbe (54-57) definiraju y-osi i razmiču ih za određenu vrijednost, u ovom slučaju za 1.1, naredbe (58-61) određuju podatke koji će se prikazati u grafu, naredbe (62-75) određuju boju, natpise i legendu grafa.



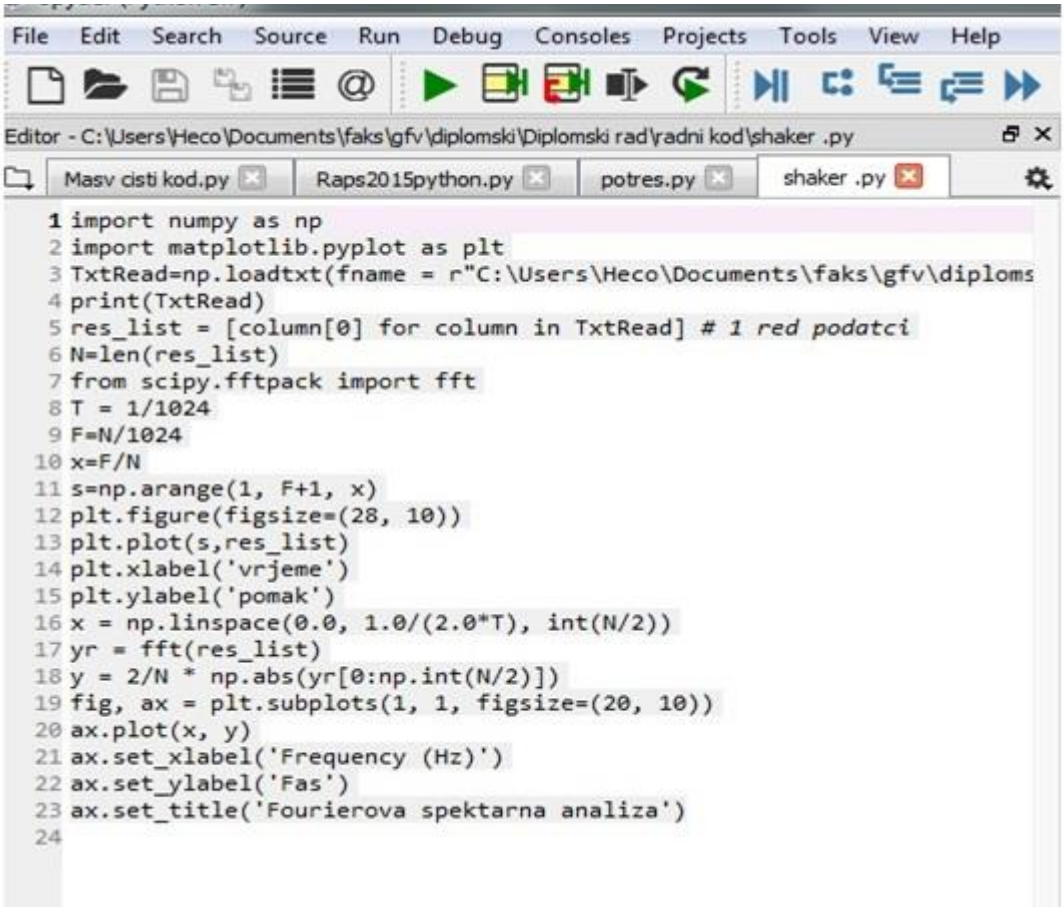
Slika 6. Prikaz grafikona s tri seta podataka i tri y-osi različitih mjerila

Prednost Pythona u ovom primjeru je mogućnost manipuliranja s podatcima a da se pritom ne mijenja izvorna datoteka, prikaz grafa s tri y-osi s različitim mjerilima što je nedostatak kad se podaci obrađuju u Excel-a. Također, Python je „open source“ što je prednost u odnosu na komercijalne programe za statističku obradu i prikaz podataka. Još je bitno napomenut da napisani kod se vrlo jednostavno može primijeniti na bilo koji novi set podataka uz par promjena, te se tako znatno ubrzava daljnja analiza podataka.

4.2. Određivanje dominantne frekvencije vibracijske ploče

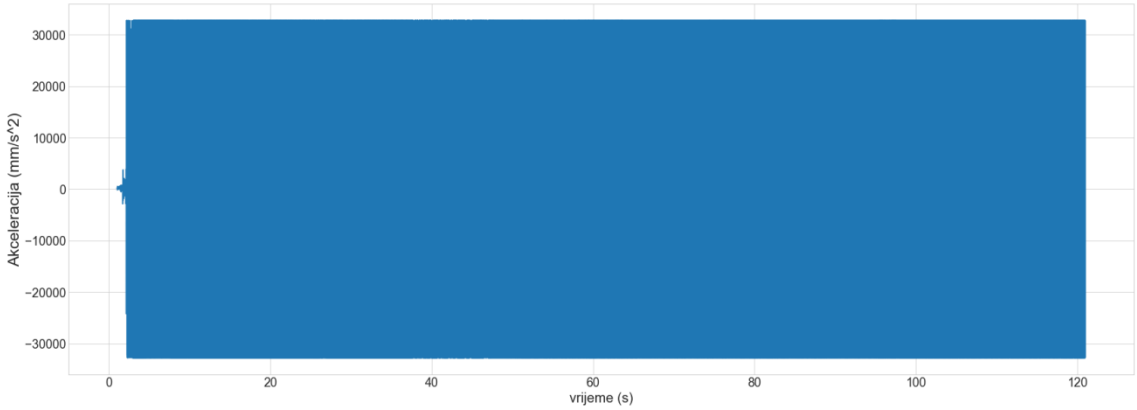
U ovom primjeru prikazani su podaci dobiveni mjerenjem oscilacija vibracijske ploče u trajanju 2 min., te analiza dobivenog signala pomoću brze Fourierove transformacije.

Slika 7 prikazuje radno okruženje Spyder i kod za analizu signala koji je objašnjen u daljnjem tekstu. Naredbama (1 i 2) učitani su paket numpy kao kratica „np“ i paket matplotlib.pyplot kao kratica „plt“, naredba (3) učitava tekstualnu „txt“ datoteku s upisane lokacije, naredba (4) ispisuje zadanu datoteku, nije nužna ali služi za provjeru kako se ne bi učitala pogrešna datoteka. Naredba (5) uzima u obzir samo podatke iz prvog reda datoteke te mijenjanjem broja u uglatoj zagradi se mijenja stupac s kojim se računa. Važno je napomenuti da Python prvi stupac definira s nulom [0], i onda redom dalje.

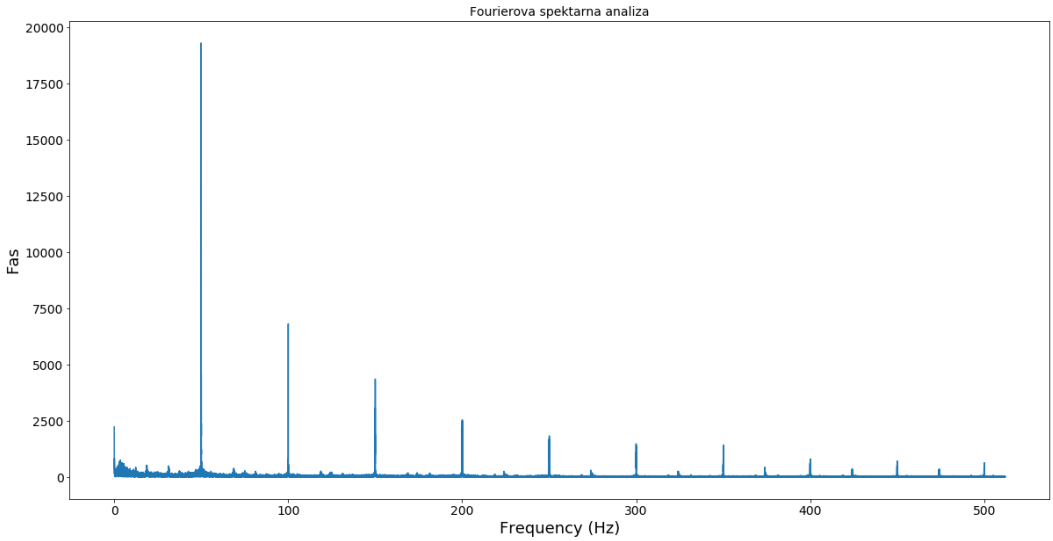


Slika 7. Prikaz koda za analizu signala pomoću Fourierove transformacije u radnom okruženju Spyder

Naredba (6) prebrojava broj podataka u stupcu, naredba (7) učitava brzu Fourierovu transformaciju „fft“, naredba (8) uvjetuje frekvenciju uzorkovanja, naredba (9) izračunava broj podataka u vremenskom intervalu, naredba (10) određuje razmak između vrijednosti u zadanom vremenskom intervalu. Naredba (11) stvara set podataka poredanih po veličini sa zadanim inkrementom, naredbe (12-15) definiraju vrijednosti i naziv x i y-osi grafa. Na Slici 8 je prikazan je vremenski vibracijski signal, te se lako može zaključiti da takav zapis ne prikazuje neku povezanost niti se iz njega mogu izvući konkretni zaključci. Naredba (16) definira podatke x-osi grafa, naredbe (17 i 18) definiraju brzu Fourierovu transformaciju za prvi stupac podataka, dok naredbe (19-23) ispisuju graf i definiraju mu x i y-osi. Slika 9 prikazuje signal dobiven FFT transformacijom te jasno prikazuje dominantnu frekvenciju uređaja od 50 Hz.



Slika 8. Prikaz vibracijskog vremenskog signala akceleracije, x-os pokazuje vrijeme, y-os pokazuje akceleraciju



Slika 9. Prikaz signala obrađenog brzom Fourierovom transformacijom s jasno istaknutom dominantnom frekvencijom od 50 Hz.

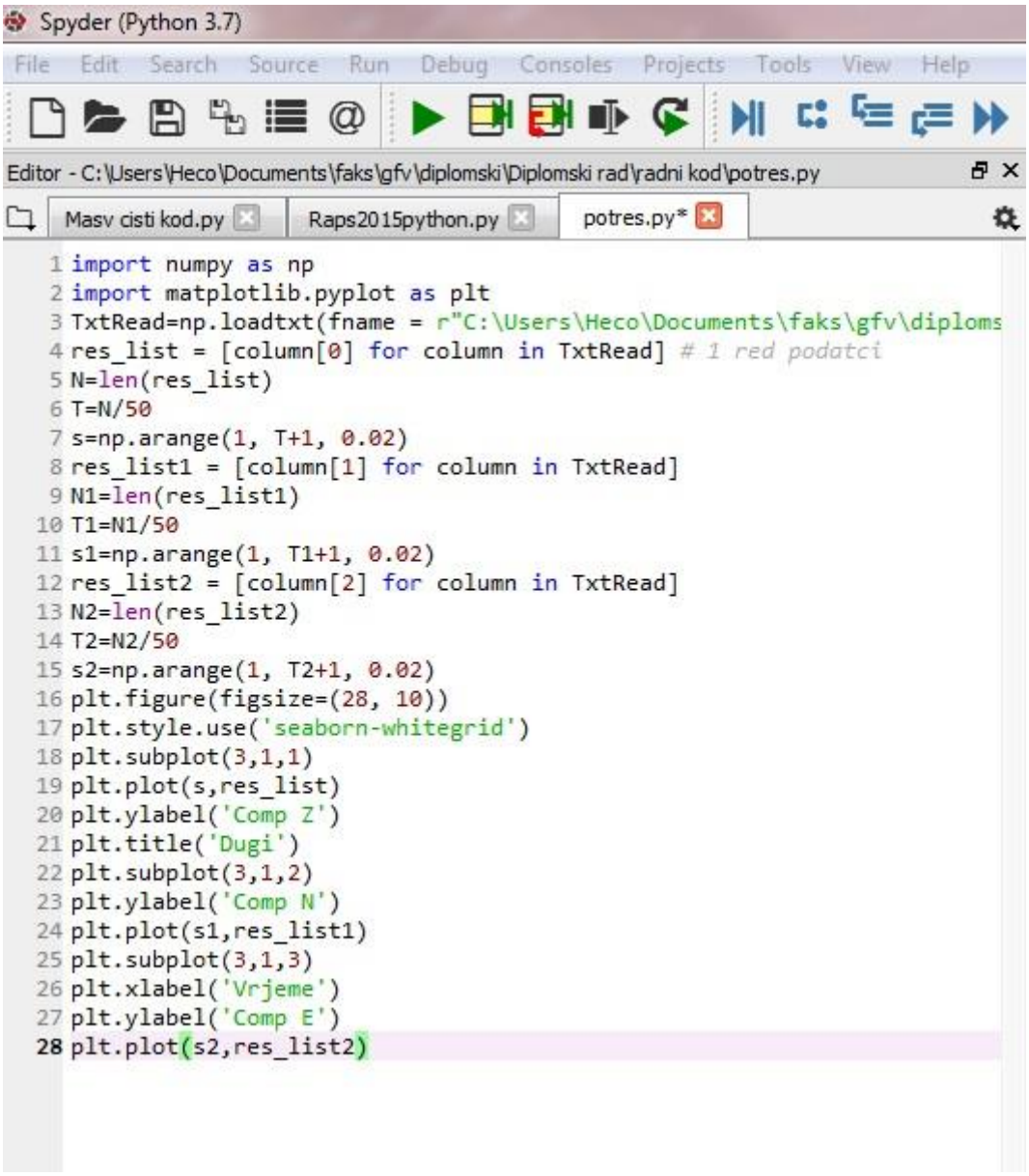
Ovaj primjer ukazuje na prednost Python programskog jezika kada je riječ o korištenju složenih matematičkih funkcija kao što je brza Fourierova transformacija u analizi signala, te mogućnost učitavanja raznih oblika datoteka poput već prikazanih Excel datoteka, tekstualnih (txt) datoteka, datoteka odvojenih sa zarezom (Comma separated values), arhiviranih ZIP datoteka, JavaScript Object Notation (JSON) datoteka, XML i HTML datoteka koje služe za izradu web stranica, Hierarchical Data Format (HDF) datoteka, PDF datoteka, Microsoft Word (DOCX) datoteka, te MP3 i MP4 datoteka (Gupta, 2017).

4.3. Prikaz zapisa potresa sa seizmoloških stanica

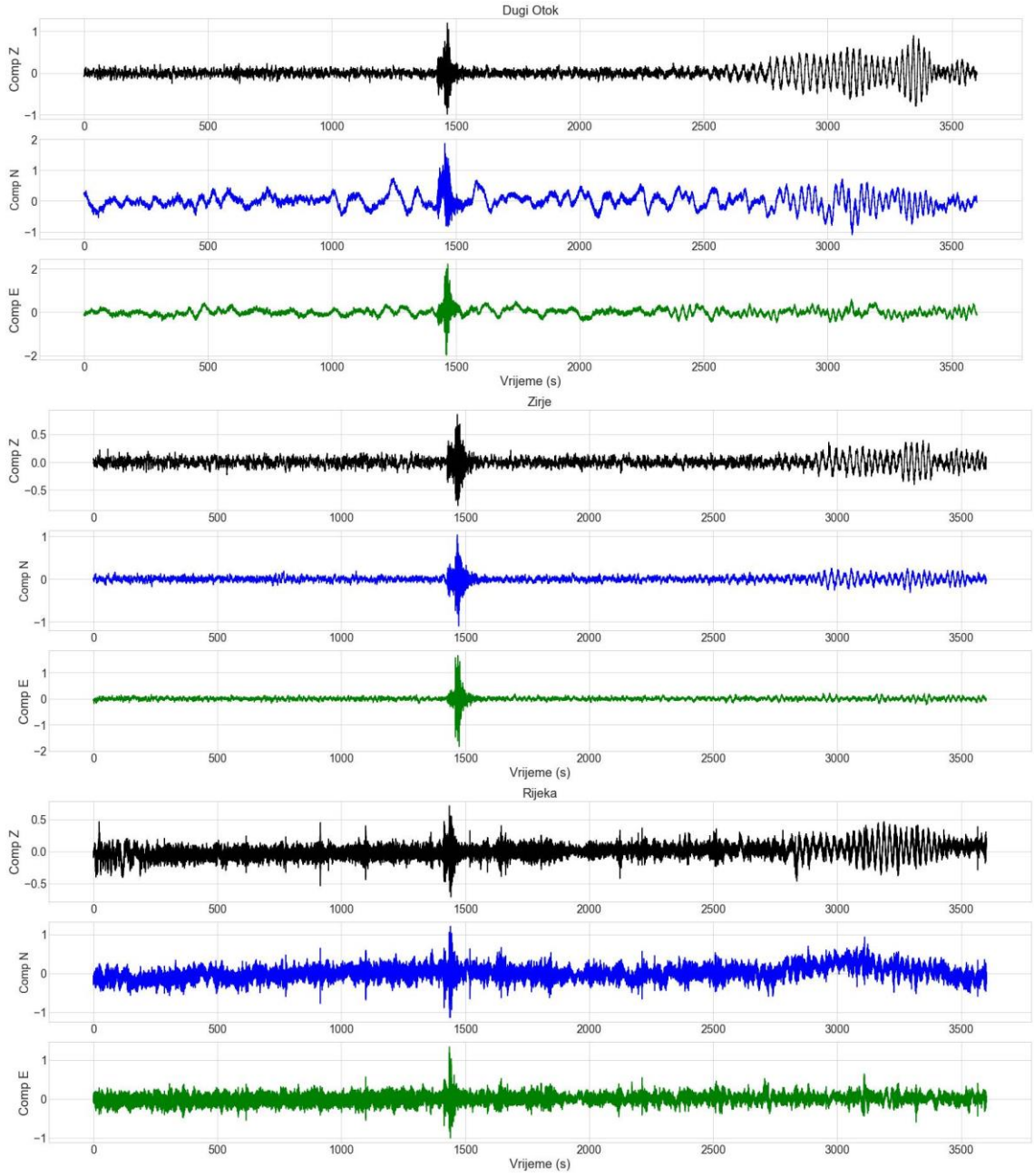
U ovom primjeru prikazani su podaci zabilježeni s 9 seizmoloških stanica prilikom potresa. Slika 10 prikazuje radno okruženje Spyder i kod za prikaz zapisa potresa. Naredbama (1 i 2) učitani su paket numpy kao kratica „np“ i paket matplotlib.pyplot kao kratica „plt“, naredba (3) učitava tekstualnu „txt“ datoteku s upisane lokacije,naredba (4) uzima u obzir samo podatke iz prvog reda datoteke. Naredba (5) prebrojava broj podataka u stupcu, naredba (6) izračunava broj podataka u vremenskom intervalu, naredba (7) stvara set podataka poredanih po veličini sa zadanim inkrementom (frekvencija mjerenja), naredbe (9-15) ponavljaju te iste radnje još 2 puta kako bi prikazali podatke za sva tri smjera širenja vala.

Naredba (16) definira veličinu prikaza ispisa, prva vrijednost određuje širinu grafa, dok druga vrijednost visinu. Naredba (17) određuje stil mreže, naredba (18) određuje količinu i poziciju grafa, dok naredba (19) učitava zadani set podataka i ispisuje graf, naredba (20) ispisuje naziv y-osi, naredba (21) ispisuje naslov grafa, naredbe (22-25) ponavljaju prethodne tri naredbe 2 puta, naredbe (26 i 27) ispisuju x i y-osi, te naredba (28) učitava zadnji set podataka za graf. općenito naredbe (16-28) služe za unos i grafički prikaz podataka (Slika 11).

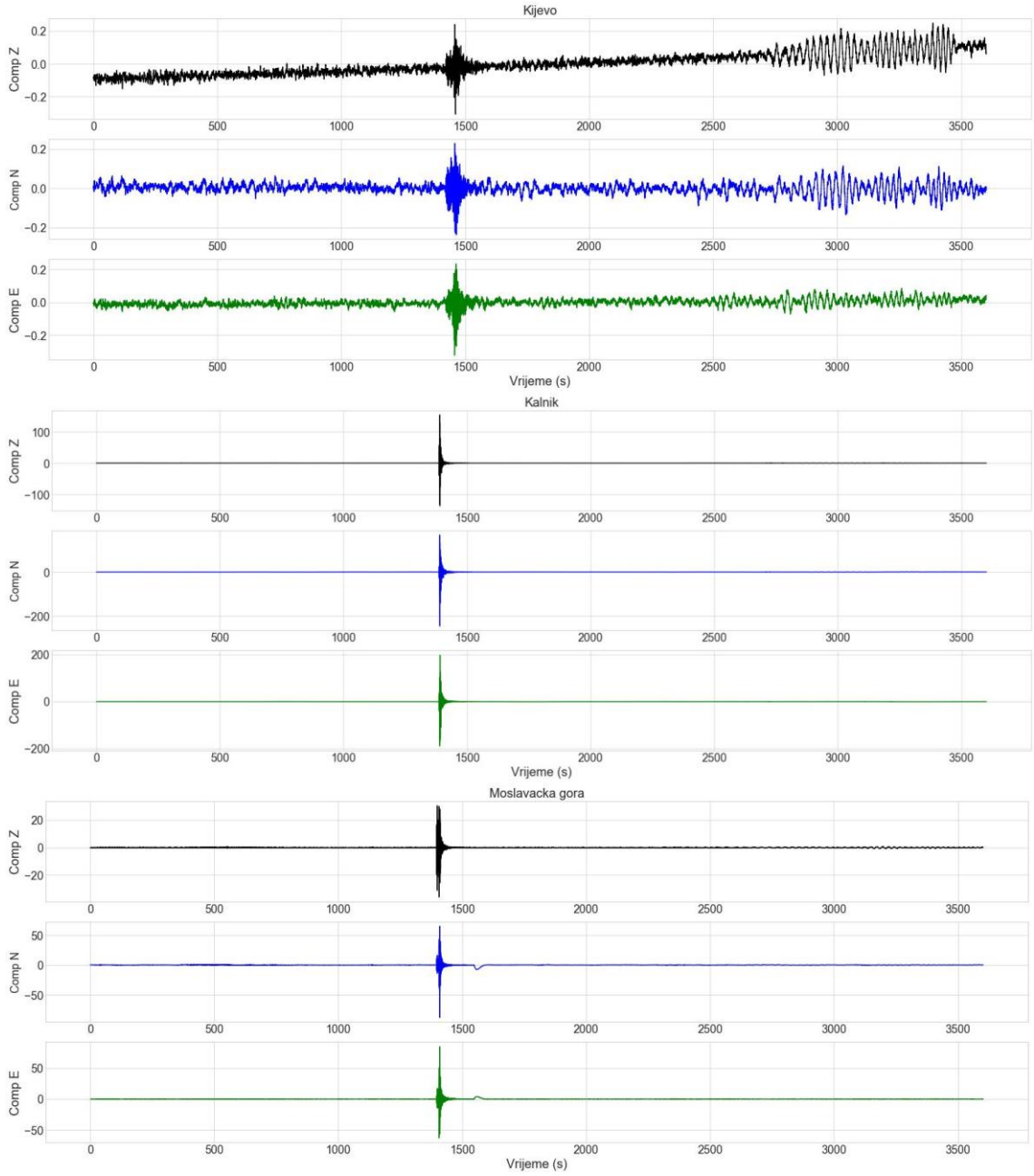
Prednost korištenja Python programskog jezika u rješavanju i analiziranju seizmičkih valova je taj što se jednostavno može učitati velik broj podataka, u ovom slučaju 180000 podataka i grafički ih prikazati dok to nije moguće kod nekih jednostavnih programa kao što je Excel, dok kod drugih programa postoji problem komercijalne prirode.



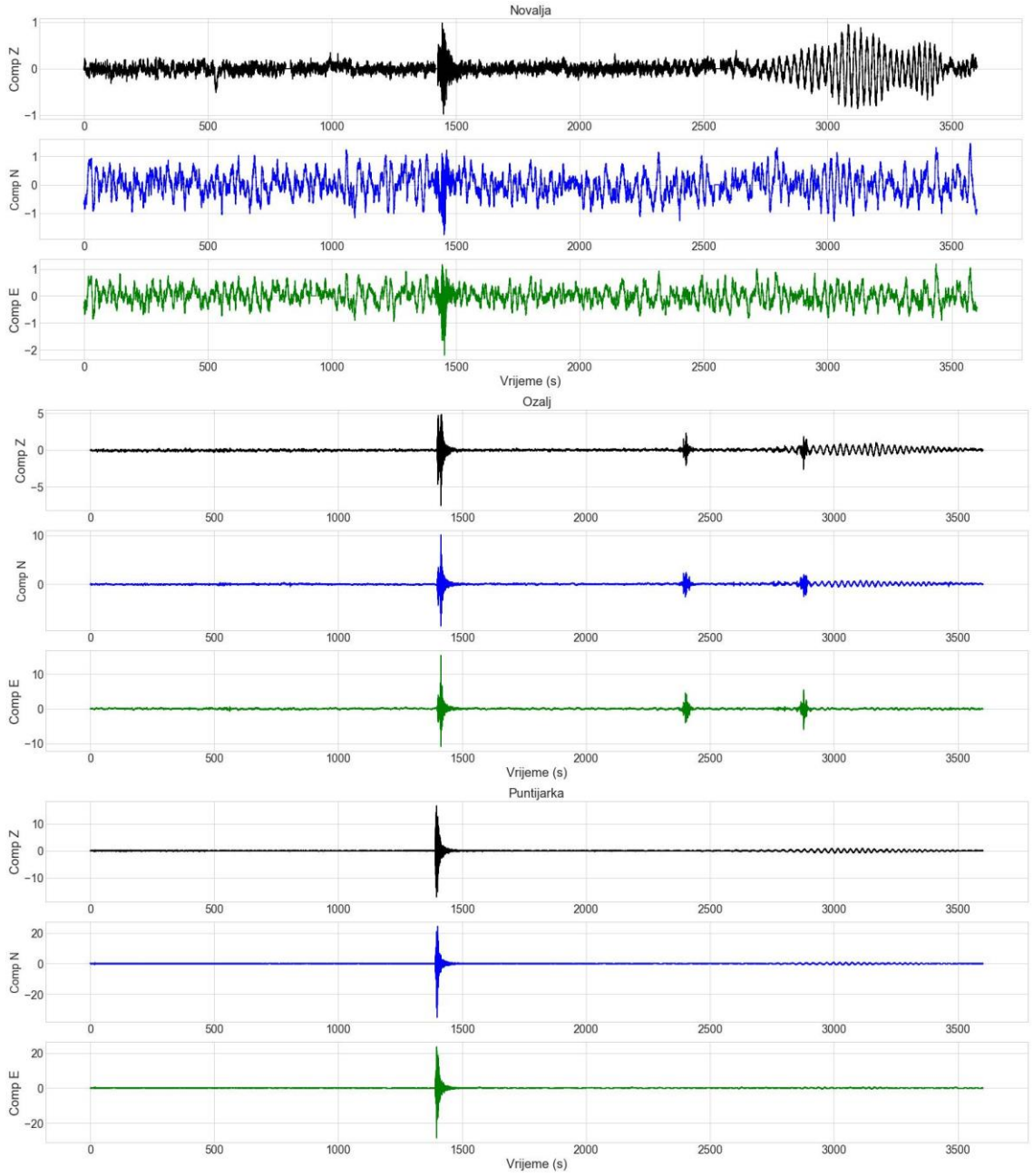
Slika 10. Prikaz koda za analizu zapisa potresa



Slika11. Grafički prikaz tri komponente seizmičkog vala s tri seizmološke postaje na lokacijama Dugi Otok, Žirje i Rijeka



Slika12. Grafički prikaz tri komponente seizmičkog vala s tri seizmološke postaje na lokacijama Kijevo, Kalnik i Moslavačka gora



Slika13. Grafički prikaz tri komponente seizmičkog vala s tri seizmološke postaje na lokacijama Novalja, Ozalj, i Puntijarka

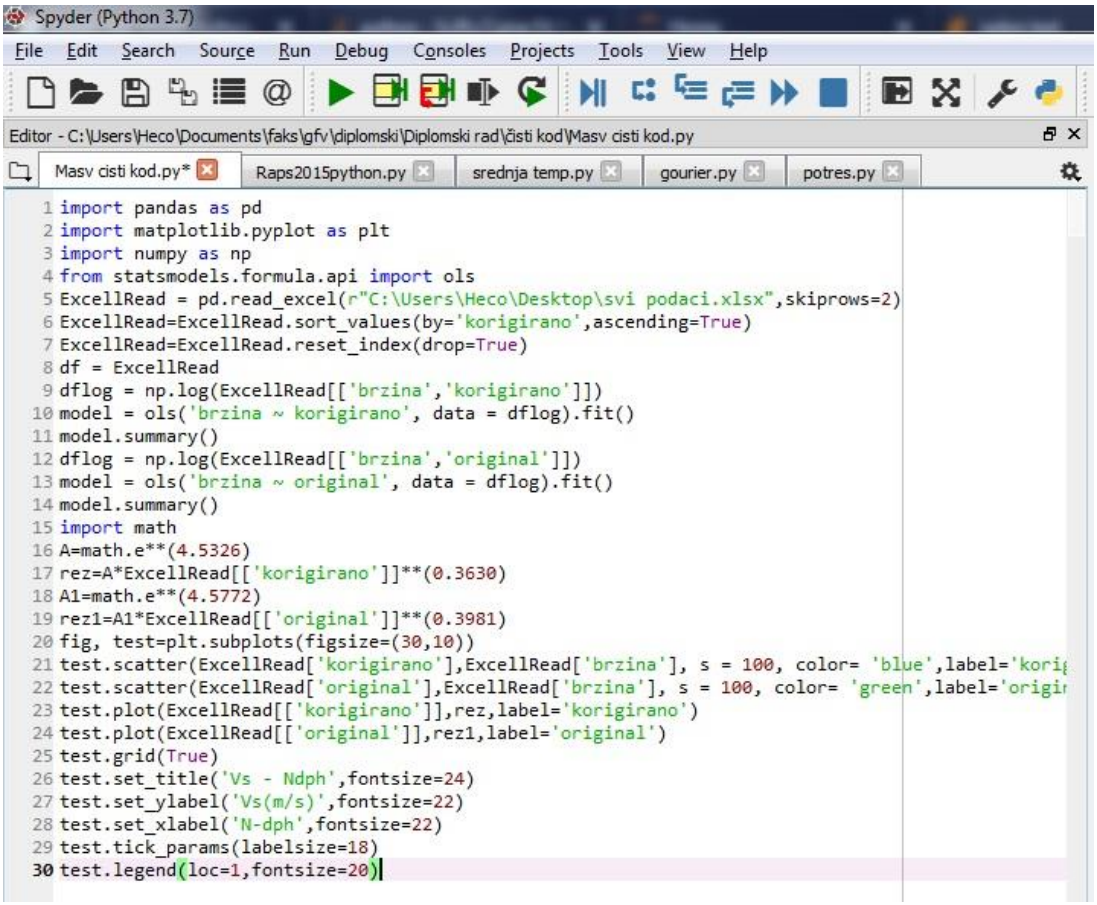
4.4. Analiza podataka korelacije između v_s i N_{DPH}

Programskim paketom Python napravljena je statistička obrada i određena pogodna krivulja regresije između broja udaraca teške sonde (N_{DPH}) i posmične brzine (V_s).

U ovom primjeru prikazana je statistička analiza i prilagođavanje regresijskih krivulja za podatke na temelju izraza (2). Slika 14 prikazuje radno okruženje Spyder i kod za statističku analizu. Naredbama (1 i 2) učitan je paket za statističku obradu pandas kao kraticu „pd“ i paket matplotlib.pyplot kao kraticu „plt“, naredba (3) učitava paket numpy kao kraticu „np“, naredba (4) učitava statistički model,naredba (5) učitava Excel datoteku s upisane lokacije. Naredbe (6 i 7) poredaju vrijednosti stupca „korigirano“ kako bi kasnije mogli nacrtati graf. Naredbe (8 i 9) daju logaritamski zapis redaka „korigirano“ i „brzina“ u Excel datoteci koje koristi naredba (10) koja učitava podatke u statistički model, koje zatim prikazujemo naredbom (11). Naredbe (12-14) ponavljaju iste korake za drugu skupinu podataka iz Excel datoteke. Naredba (15) učitava paket za izvršavanje složenijih matematičkih operacija. Iz podataka prikazanih naredbom (11) (Slika 15) i podataka prikazanih naredbom (14) (Slika 16). S vrijednostima intercepta i brzine se izvedu funkcije (16-19) pomoću kojih dobivamo vrijednosti A i B prikazanih u Tablici 1, iz izraza (2) koji su potrebni kako bi prikazali krivulju regresije za zadane podatke. Tablica 1. prikazuje zanemarive razlike nastale najvjerojatnije zbog zaokruživanja vrijednosti i potvrđuje podatke dobivene u Excel-u. Naredba (20) određuje visinu i širinu grafičkog prikaza, a naredbe (21 i 22) crtaju raspršene grafikone za zadane podatke i naredbe (23 i 24) crtaju regresijske krivulje. Naredbe (25-30) definiraju x i y-osi ,mrežu i legendu grafa.

Tablica 1: Usporedba koeficijenata regresije i koeficijenata A i B dobivenih u programskom jeziku Python s koeficijentima dobivenim u Excel-u (Strelec i sur., 2016).

Koeficijenti	Python	Excel
R^2 (original)	0.725	0.723
A (original)	97.242	97.839
B (original)	0.398	0.395
R^2 (korigirano)	0.815	0.815
A (korigirano)	93.000	92.998
B (korigirano)	0.363	0.363



Slika 14. Prikaz koda za statističku analizu za parametre Vs i Ndph

OLS Regression Results

Dep. Variable:	brzina	**R-squared:**	0.815
Model:	OLS	**Adj. R-squared:**	0.814
Method:	Least Squares	**F-statistic:**	651.3
Date:	Sat, 29 Jun 2019	**Prob (F-statistic):**	4.55e-56
Time:	13:39:47	**Log-Likelihood:**	74.762
No. Observations:	150	**AIC:**	-145.5
Df Residuals:	148	**BIC:**	-139.5
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	4.5326	0.042	106.868	0.000	4.449	4.616
korigirano	0.3630	0.014	25.521	0.000	0.335	0.391

Omnibus:	0.813	**Durbin-Watson:**	1.665
Prob(Omnibus):	0.666	**Jarque-Bera (JB):**	0.443
Skew:	-0.042	**Prob(JB):**	0.801
Kurtosis:	3.253	**Cond. No.**	11.6

Slika 15. Prikaz statističke analize korigiranih podataka za prilagođavanje krivulje

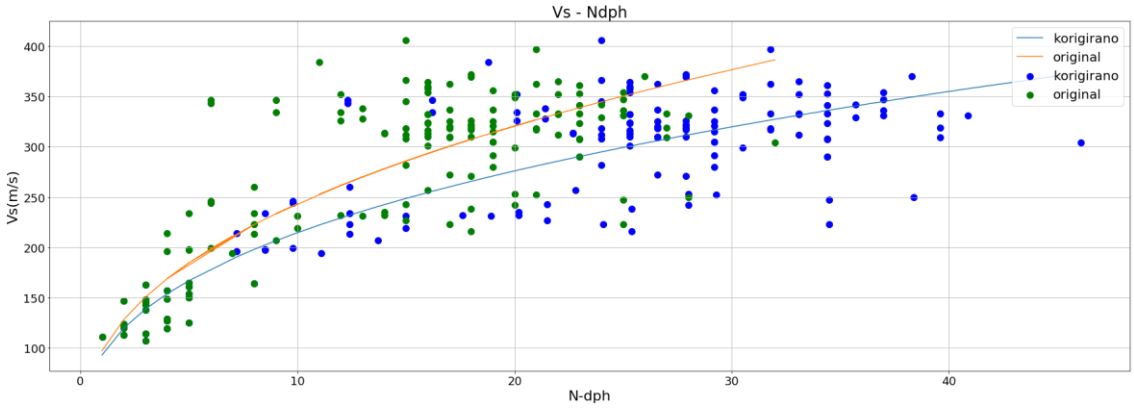
OLS Regression Results

Dep. Variable:	brzina	**R-squared:**	0.725
Model:	OLS	**Adj. R-squared:**	0.723
Method:	Least Squares	**F-statistic:**	390.8
Date:	Sat, 29 Jun 2019	**Prob (F-statistic):**	2.29e-43
Time:	13:39:50	**Log-Likelihood:**	45.176
No. Observations:	150	**AIC:**	-86.35
Df Residuals:	148	**BIC:**	-80.33
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	4.5772	0.052	87.448	0.000	4.474	4.681
original	0.3981	0.020	19.768	0.000	0.358	0.438

Omnibus:	2.352	**Durbin-Watson:**	1.449
Prob(Omnibus):	0.308	**Jarque-Bera (JB):**	2.041
Skew:	0.125	**Prob(JB):**	0.360
Kurtosis:	3.514	**Cond. No.**	10.5

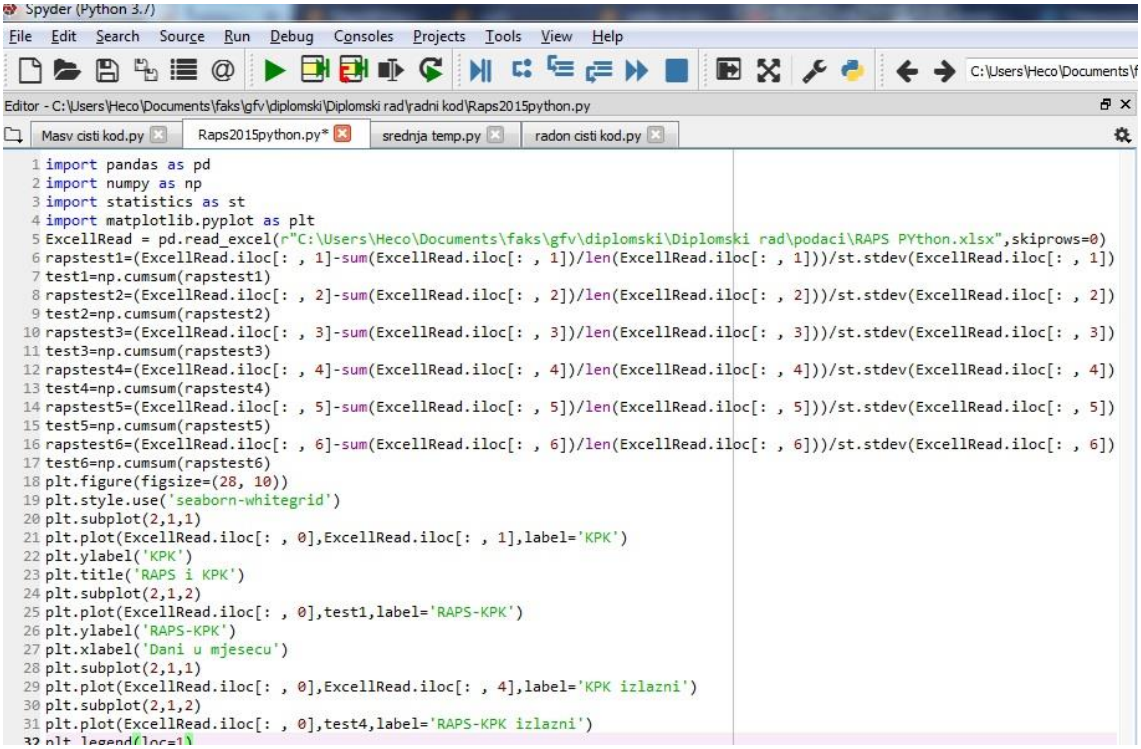
Slika 16. Prikaz statističke analize originalnih podataka za prilagođavanje krivulje



Slika 17. Prikaz podataka i regresijskih krivulja za korelaciju Vs i Ndph podataka

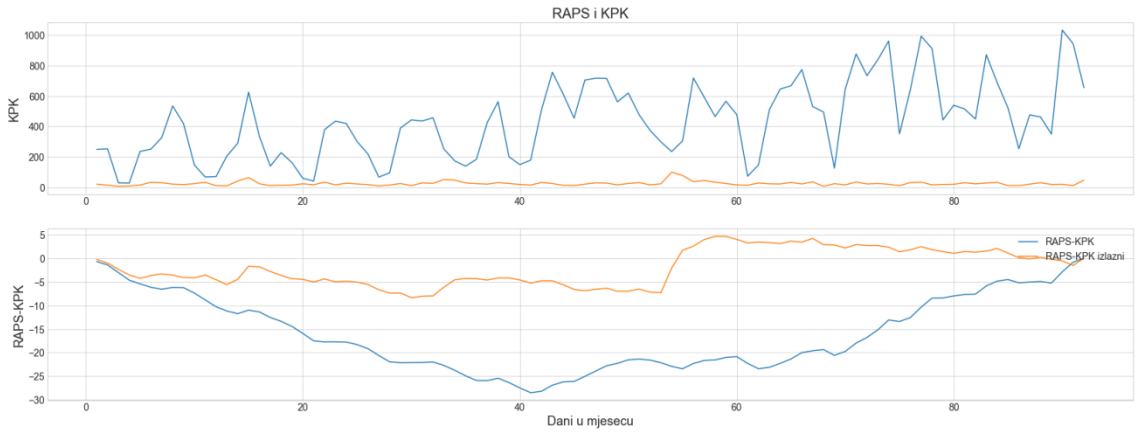
4.5. Analiza i prikaz podataka RAPS metode u praćenju kakvoće otpadne vode

U ovom primjeru prikazana je statistička analiza korištenjem RAPS metode na setu podataka za praćenje kakvoće vode. Slika 18 prikazuje radno okruženje Spyder i kod za statističku analizu. Naredbe (1 i 2) učitavaju je paket za statističku obradu pandas kao kraticu „pd“ i paket numpy kao kraticu „np“, naredba (3) učitava statističke funkcije za standardnu devijaciju, naredba (4) učitava paket matplotlib.pyplot kao kraticu „plt“.



Slika 18. Prikaz koda za RAPS analizu podataka

Naredba (5) učitava Excel datoteku s upisane lokacije, naredbe (6 i 7) izvršavaju RAPS metodu tako da funkcija (6) uzima podatke iz drugog reda Excel tablice i uvrštava ih u formulu (2), a funkcija (7) sumira svaku iduću vrijednost. Naredbe (8-17) ponavljaju prethodne dvije operacije za podatke iz preostalih redova. Naredba (18) definira veličinu prikaza ispisa, prva vrijednost određuje širinu grafa, dok druga vrijednost visinu.



Slika 19. Prikaz podataka na ulazu i izlazu iz jedinice za obradu otpadnih voda (gore), i prikaz RAPS podataka na ulazu i izlazu (dolje)

Naredba (19) određuje stil mreže, naredba (20) određuje količinu i poziciju grafa, dok naredba (21) učitava zadani set podataka i ispisuje graf, naredba (22) ispisuje naziv y-osi, naredba (23) ispisuje naslov grafa, naredbe (24-26) ponavljaju prethodne tri naredbe, naredbe (27) ispisuje x-os. Cilj je usporediti podatke na ulazu i izlazu iz jedinice za obradu otpadnih voda, stoga naredbe (20-25) prikazuju set podataka na ulazu dok naredbe (28-31) prikazuju set podataka na izlazu (Slika 19). Naredba (32) prikazuje legendu grafa.

Podnizovi dobiveni RAPS metodom omogućuju kompletniji uvid u nepravilnosti, poremećaje i diskontinuitete vremenskih nizova pokazatelja kakvoće otpadnih voda.

4.6. Analiza srednje temperature za vremensko razdoblje od godinu dana

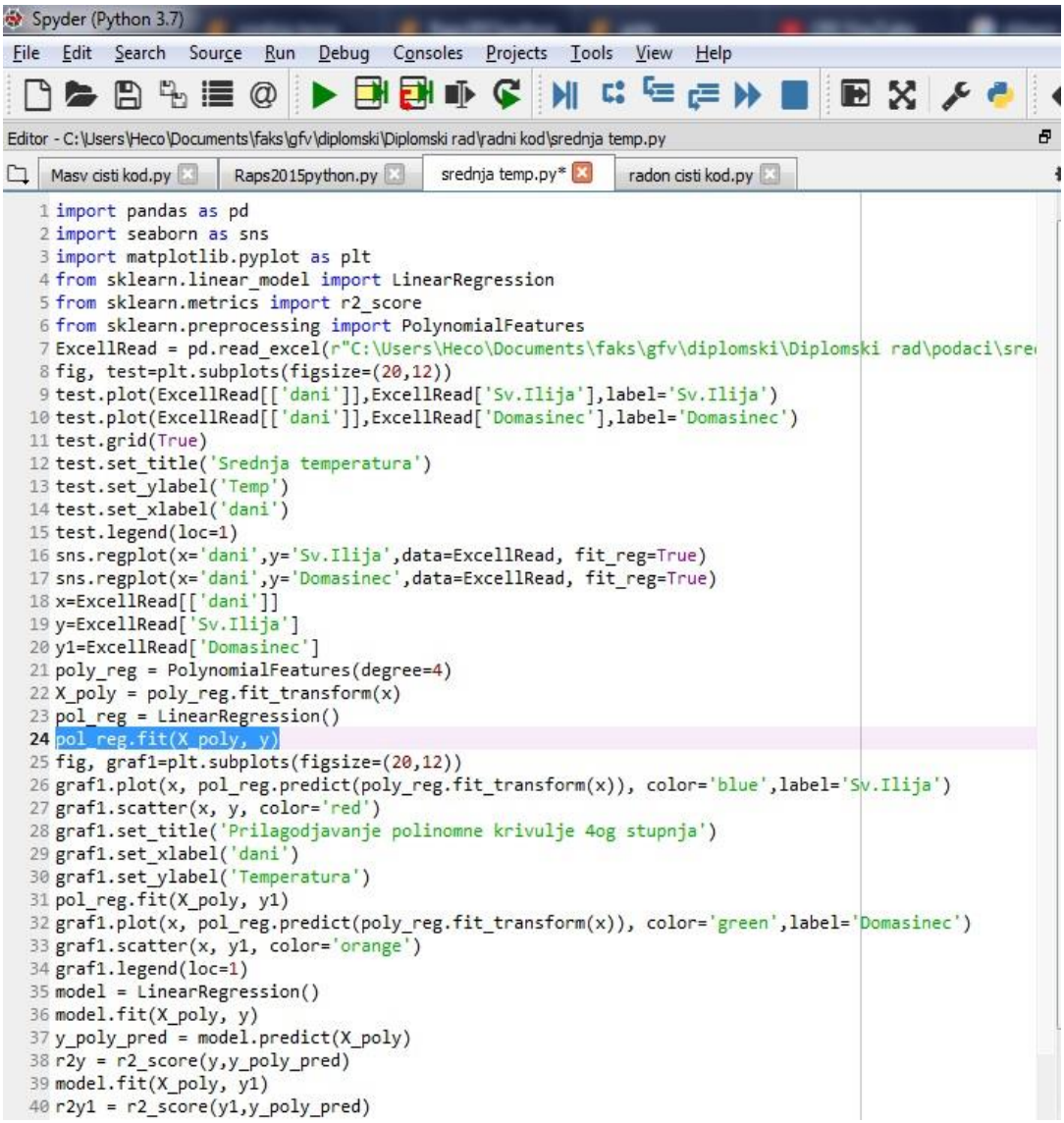
U ovom primjeru je prikazan ispis podataka srednje temperature za dva područja te su prilagođene polinomne regresijske krivulje četvrtog stupnja.

$$Y = b_1X^1 + b_2X^2 + b_3X^3 + b_4X^4. \quad (4)$$

gdje je Y ovisna varijabla, b_n su regresijski koeficijenti, i X je nezavisna varijabla koja u ovom slučaju predstavlja vrijeme u danima.

Slika 20 prikazuje radno okruženje Spyder i kod za statističku analizu. Naredba (1) učitava paket za statističku obradu pandas kao kraticu „pd“, naredba (2) učitava paket za statističku obradu seaborn kao kraticu „sns“, naredba (3) učitava paket matplotlib.pyplot

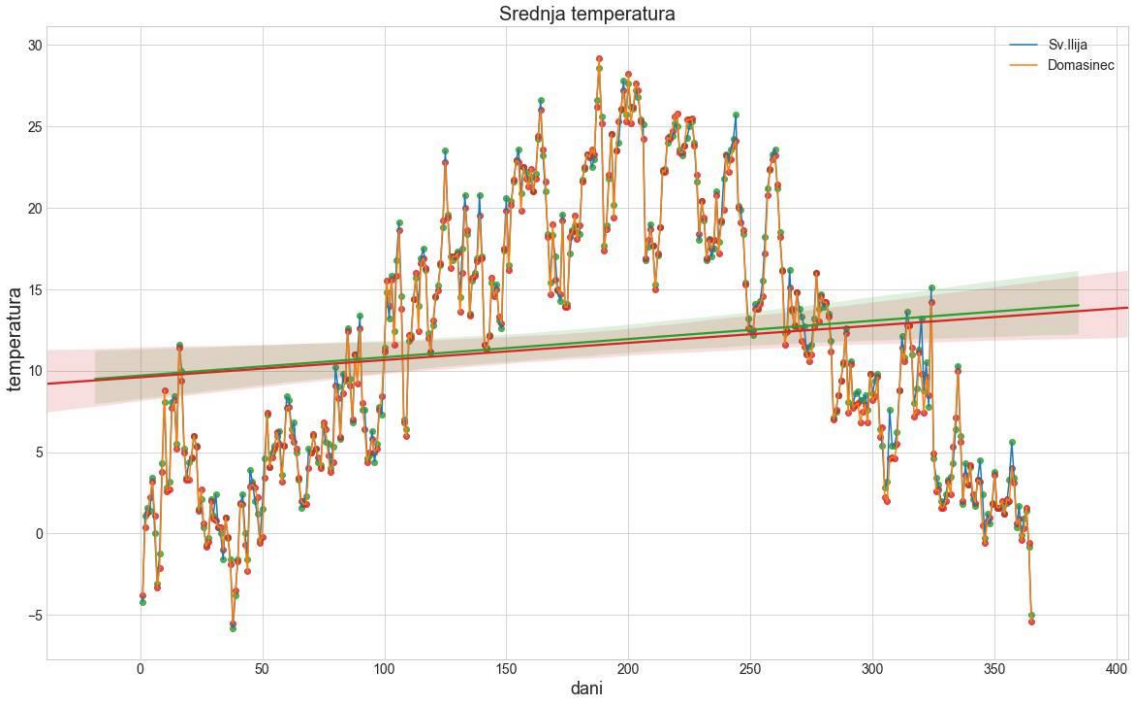
kao kraticu „plt“. Naredbe (4-6) učitavaju iz paketa scikit-learn, koji služi za obradu podataka, funkcije za linearnu regresiju, koeficijent korelacije i funkciju za polinome. Naredba (7) učitava Excel datoteku s upisane lokacije, naredbe (8-15) određuju parametre za prikaz grafa, dok naredbe (16 i 17) ispisuju linearnu regresiju za zadane podatke.



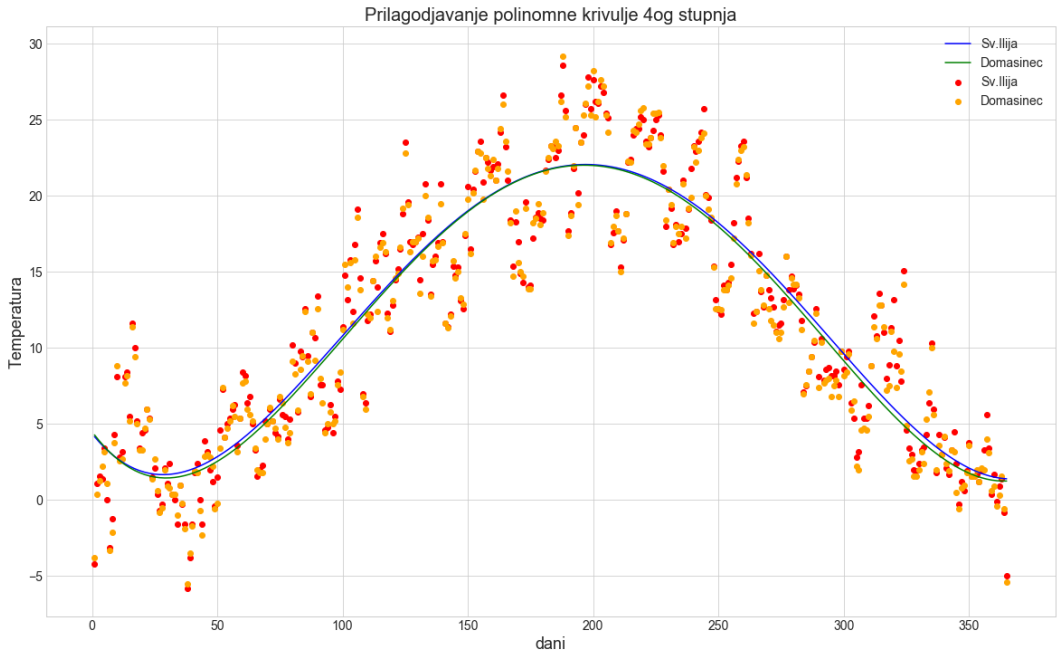
Slika 20. Prikaz radnog okruženja Spyder i koda za statističku analizu

Iz Slike 21 je vidljivo da linearna regresija ne prikazuje dobru povezanost, stoga je potrebno upotrijebiti polinomne regresijske krivulje. Naredbe (18-20) mijenjaju naziv funkcije za prizivanje podataka, ovi koraci nisu nužni ali ubrzavaju daljnju uporabu koda i smanjuju mogućnost pogreške. Naredba (21) definira polinomsku funkciju četvrtog stupnja, naredba (22) pretvara vrijednosti u polinome, naredba (23) stvara

prazan skup linearne regresije u koji naredba (24) postavlja podatke, naredba (25) definira veličinu prikaza ispisa, prva vrijednost određuje širinu grafa, dok druga vrijednost visinu, naredba (26) prikazuje polinomnu regresijsku krivulju za prvi set podataka, a naredba (27) prikazuje točkasti graf za prvi set podataka. Naredbe (28-30) definiraju naslov i nazive x i y-osi, naredba (31) definira drugi set podataka polinomne regresije, te ih naredbe (32 i 33) grafički prikazuju (Slika 22). Naredba (34) prikazuje legendu, dok naredbe (35-40) definiraju funkcije za određivanje koeficijenata korelacije za obije krivulje, dok ih naredbe (41 i 42) ispisuju.



Slika 21. Grafički prikaz linearne regresije za setove podataka



Slika 22. Grafički prikaz polinomne regresije četvrtog stupnja za setove podataka

Tablica 2 prikazuje koeficijente determinacije R^2 za slučaj linearne regresije i polinomne regresije četvrtog stupnja. Koeficijent determinacije R^2 nam zapravo daje informaciju o tome koliko rasipanja izlaznih podataka potječe od funkcijske ovisnosti, a koliko otpada na tzv. rezidualno ili neobjašnjeno rasipanje (tu informaciju očitavamo iz $1- R^2$). Drugim riječima daje informaciju o tome koliko je jaka funkcijska veza između x i y podataka. Što je vrijednost koeficijenta R^2 bliža 1, zavisnost je jača što se vidi iz razlike na Slikama 21 i 22.

Ovaj primjer dobro prikazuje prilagodljivost Python programskog jezika za raznovrsne zadatke poput rješavanja polinomne regresijske krivulje četvrtog stupnja i mogućnost ponavljanja složenih operacija s novim skupovima podataka u odnosu na jednostavne poput linearne regresije.

Tablica 2: Koeficijenti determinacije za slučaj linearne regresije i polinomne regresije četvrtog stupnja.

Vrsta regresijske krivulje	Lokacija Sv. Ilija	Lokacija Domašinec
Linearna	R^2=0,021	R^2=0,018
Polinomna četvrtog stupnja	R^2=0.833	R^2=0.837

5. DISKUSIJA

Programski jezik Python pruža brojne mogućnosti u obradi podataka i njihovom prikazu kao što je vidljivo iz prethodnih primjera. Treba uzeti u obzir prednosti i nedostatke prilikom korištenja Pythona naspram nekog drugog jednostavnijeg programa poput Excel-a.

Prva i očita prednost Pythona je to što je „open source“ tj. besplatan te omogućuje besplatno i jednostavno dijeljenje i stvaranje sadržaja. Druga bitna prednost je mogućnost korištenja raznovrsnih statističkih metoda i matematičkih funkcija u primjerima: npr. određivanja dominantne frekvencije vibracijske ploče gdje je korištena brza Fourierova transformacija i analiza srednje temperature gdje je korištena polinomna regresijska krivulja četvrtog stupnja. Također, primjer s analizom zapisa potresa ukazuje na prednost Pythona u rukovanju s velikim brojem podataka, te iznimno ubrzava analizu novih podataka s obzirom na to da je potrebno samo učitati novu datoteku, a i ovisno o kodu, promijeniti nekoliko vrijednosti. Primjer mjerenja koncentracije radona prikazuje mogućnost izrade grafičkih prikaza s tri osi različitih mjerila, a može i više što ponekad može biti jako korisno.

Općenito Python pruža velik broj mogućnosti za grafički prikaz različitih vrsta podataka iz različitih vrsta datoteka poput Excel datoteka, tekstualnih (txt) datoteka, datoteka odvojenih sa zarezom (Comma separated values), arhiviranih ZIP datoteka, JavaScript Object Notation (JSON) datoteka, XML i HTML datoteka koje služe za izradu web stranica, Hierarchical Data Format (HDF) datoteka, PDF datoteka itd...

U toku rada i obrade podataka primijetio sam da prilikom izrade koda i unosa podataka s manje podataka u datotekama, Excel se pokazao bržim i jednostavnijim. No međutim, prilikom izrade sličnih programa, zbog mogućnosti kopiranja dijelova koda daljnja uporaba Pythona postaje sve efikasnija, pogotovo kad brzo treba obraditi i prikazati veliki set raznovrsnih podataka. Bitna činjenica koja ide u prilog Pythonu je mogućnost stvaranja i istraživanja novih načina i metoda za rješavanje postojećih problema.

Neki od nedostataka Pythona su: potrebna je određena količina znanja i poznavanja principa rada programskih jezika, prilikom rada s kompliciranijim operacijama i velikim brojem podataka potrebna je veća snaga procesora, gubitak vremena pri pronalaženju i uklanjanju grešaka u kodu. No, jednom kada se savladaju početni problemi i krenu rješavati konkretni problemi kao što je prikazano u ovom radu, Python se zaista

pokazao kao jednostavan i moćan alat za analizu podataka i rješavanje raznih problema iz područja inženjerstva okoliša.

6. ZAKLJUČAK

U primjeni rješavanja problema vezanih za inženjerstvo okoliša, programski jezik Python se pokazao kao efikasan alat zbog svoje prilagodljivosti, načina rada, dostupnosti literature i materijala za učenje i usavršavanje, mogućnostima za obradu i prikaz raznih tipova podataka. Pokazao se kao dobra podloga za stvaranje i istraživanje novih načina i metoda za rješavanje postojećih problema u inženjerstvu okoliša. Povećanjem količine podataka i raznovrsnosti problema sve više dolazi do izražaja automatizacija ponavljajućih radnji učitavanja podataka koje omogućava Python u odnosu na Excel i slične komercijalne programe. U današnjem svijetu razvitka tehnologije, poznavanje rada na računalu i programskog jezika nedvojbeno otvara mogućnosti u analizi i rješavanju problema u inženjerstvu okoliša.

LITERATURA

Barnes A. (1993). Instantaneous spectral bandwidth and dominant frequency with applications to seismic reflection data. GEOPHYSICS, Vol. 58, No. 3 (1993), str. 419-428

Bonacci O. (2010). Analiza nizova srednje godišnje temperature zraka u Hrvatskoj.*Građevinar*. 62 (9).

Cothern C. R.,Smith, J. E. J. (1987). Enviromental Radon. New York: Springer Science+Buisness Media,LLC.

Definicija programskog jezika http://www.enciklopedija.hr/natuknica.aspx?ID=50558 (18.05.2019.)

Elliott D. F. (1987). Handbook of Digital Signal Processing: Engineering Applications. Academic press,inc. San Diego,California.

Gupta, A. (2017). How to read most commonly used file formats in Data Science (using Python) dostupno na:https://www.analyticsvidhya.com/blog/2017/03/read-commonly-used-formats-using-python/ (15.6.2019)

Kadiyala A., Kumar A. (2017). 'Applications of Python to Evaluate Environmental Data Science Problems', Wiley Online Library (wileyonlinelibrary.com)

Markušić S. (2003). 'Seizmologija i istraživanje unutrašnjosti zemlje', Hrvatsko fizikalno društvo str. 60-68

Matlab, dostupno na https://www.mathworks.com/products/matlab.html (2.7.2019)

Milsom J. (2003). Field Geophysics. Third edit. John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, PO19 8SQ, England.

Mohorji A. M., Almazroui M. (2017). 'Trend Analyses Revision and Global Monthly Temperature Innovative Multi-Duration Analysis', str. 1–13.

Nola I. i sur. (2013). 'Potresi – povijesni pregled, okolišni i zdravstveni učinci i mjere zdravstvene skrbi', (3), str. 327–337.

Oran Brigham E. (1988). The fast fourier transform and its applications. Englewood Cliffs, New Jersey: Prentice-Hall, Inc.A Division of Simon & Schuster.

Origin, dostupno na https://www.originlab.com/index.aspx?go=Products/Origin/Statistics (28.6.2019)

Park C. B, Miller R. D, Xia, J. (1999). 'Multichannel analysis of surface waves', str. 800–808.

Radon, dostupno na http://radon.dzrns.hr/o-radonu/ (28.6.2019)

Statistica, dostupno na http://www.statsoft.com/Products/STATISTICA-Features (2.7.2019)

Sigma Plot, dostupno na https://systatsoftware.com/products/sigmaplot/ (28.6.2019)

Strelec S, Gazdek M,Jeđud B. (2016). 'Procjena krutosti tla in-situ ispitivanjima i korelacije između vs, Nspt i N_{10H}', str. 5–17.

Strelec S, Jug J. (2017). Planiranje sigurnih miniranja u urbanim sredinama ,GEO-EXPO 2017. Sarajevo

Trešnjo Z, Adrović F. (2005). 'Tehnike mjerenja produkata raspada radona u zraku', str. 363–370.

Tulchak L, Marchuk A. (2016). 'History of Python', str. 1–3.

Tušar, B. (2004). Ispuštanje i pročišćavanje otpadne vode. Zagreb: Croatia Knjiga.

Tušar, B. (2009). Pročišćavanje otpadnih voda. Zagreb: Kigen.

POPIS SLIKA

Slika 1. Prikaz instalacije pandas paketa za statističku obradu

Slika 2. Prikaz instalacije NumPy, SciPy i matplotlib paketa pomoću pip-a

Slika 3. Izgled sučelja Anaconda platforme i dostupnih alata

Slika 4. Prikaz Spyder radnog okruženja s ispisanim podacima iz Excel tablice

Slika 5. Grafički prikaz dva seta podataka(Radon i temperatura) u vremenu

Slika 6. Prikaz grafikona s tri seta podataka i tri y-osi različitih mjerila

Slika 7. Prikaz koda za analizu signala pomoću Fourierove transformacije u radnom okruženju Spyder

Slika 8. Prikaz vibracijskogvremenskog signala akceleracije, x-os pokazuje vrijeme, y-os pokazuje akceleraciju

Slika 9. Prikaz signala obrađenog brzom Fourierovom transformacijom s jasno istaknutom dominantnom frekvencijom od 50 Hz.

Slika 10. Prikaz koda za analizu zapisa potresa

Slika11. Grafički prikaz tri komponente seizmičkog vala s tri seizmološke postaje

Slika12. Grafički prikaz tri komponente seizmičkog vala s tri seizmološke postaje

Slika13. Grafički prikaz tri komponente seizmičkog vala s tri seizmološke postaje

Slika 14. Prikaz koda za statističku analizu za parametre Vs i Ndph

Slika 15. Prikaz statističke analize korigiranih podataka za prilagođavanje krivulje

Slika 16. Prikaz statističke analize originalnih podataka za prilagođavanje krivulje

Slika 17. Prikaz podataka i regresijskih krivulja za korelaciju Vs i Ndph podataka

Slika 18. Prikaz koda za RAPS analizu podataka

Slika 19. Prikaz podataka na ulazu i izlazu iz jedinice za obradu otpadnih voda (gore), i prikaz RAPS podataka na ulazu i izlazu (dolje)

Slika 20. Prikaz radnog okruženja Spyder i koda za statističku analizu

Slika 21. Grafički prikaz linearne regresije za setove podataka

Slika 22. Grafički prikaz polinomne regresije četvrtog stupnja za setove podataka

POPIS TABLICA

Tablica 1: Usporedba koeficjenata regresije i koeficijenata A i B dobivenih u programskom jeziku Python s koeficijentima dobivenim u Excel-u (Strelec i sur.,2016).

Tablica 2: Koeficijenti determinacije za slučaj linearne regresije i polinomne regresije četvrtog stupnja.

DODATAK

4.1. KOD ZA PRIKAZ MJERENJA KONCENTRACIJE RADONA

```
import pandas as pd
import matplotlib.pyplot as plt
ExcelRead = pd.read_excel(r"C:\Users\Heco\Desktop\radon.xlsx",skiprows=0)
ExcelRead=ExcelRead.sort_values(by='Time',ascending=True)
ExcelRead=ExcelRead.reset_index(drop=True)
data1=ExcelRead['Time']
data2=ExcelRead['Radon']
data3=ExcelRead['Temperature']
data4=ExcelRead['Humidity']
data5=ExcelRead['Pressure']
print(ExcelRead)
fig, axe1 = plt.subplots(figsize=(20,12))
axe1.set_title('Radon test')
axe1.set_ylabel('Radon [Bq/m^3]')
axe1.set_xlabel('Vrijeme (hh:mm)')
axe1.plot(data1,data2,lw=3,label='Radon')
axe1.legend(loc=1)
axe1.grid(True)
axe2=axe1.twinx()
color = 'tab:red'
axe2.set_ylabel('temperatura', color=color)
axe2.plot(data1, data3,label='Temp',color=color)
axe2.tick_params(axis='y', labelcolor=color)
axe2.legend(loc=2)
fig, axe3 = plt.subplots(figsize=(20,12))
axe3.set_title('Radon test2')
axe3.set_ylabel('Radon [Bq/m^3]')
axe3.set_xlabel('Vrijeme (hh:mm)')
axe3.plot(data1,data2,lw=3,label='Radon')
axe3.legend(loc=1)
axe3.grid(True)
axe4=axe3.twinx()
color = 'tab:red'
axe4.set_ylabel('Vlaga (%)', color=color)
axe4.plot(data1, data4,label='Vlaga',color=color)
axe4.tick_params(axis='y', labelcolor=color)
axe4.legend(loc=2)
fig, axe5 = plt.subplots(figsize=(20,12))
axe5.set_title('Radon test2')
axe5.set_ylabel('Radon [Bq/m^3]')
axe5.set_xlabel('Vrijeme (hh:mm)')
axe5.plot(data1,data2,lw=3,label='Radon')
axe5.legend(loc=1)
axe5.grid(True)
axe6=axe5.twinx()
color = 'tab:red'
axe6.set_ylabel('Tlak (kPa)', color=color)
axe6.plot(data1, data5,label='Tlak',color=color)
axe6.tick_params(axis='y', labelcolor=color)
axe6.legend(loc=2)
def make_patch_spines_invisible(ax):
    ax.set_frame_on(True)
    ax.patch.set_visible(False)
    for sp in ax.spines.values():
        sp.set_visible(False)
fig, host = plt.subplots(figsize=(20,12))
fig.subplots_adjust(right=0.75)
par1 = host.twinx()
par2 = host.twinx()
make_patch_spines_invisible(par2)
par2.spines["right"].set_position(("axes", 1.1))
par2.spines["right"].set_visible(True)
p1, = host.plot(data1,data2, "b-", label="Radon")
p2, = par1.plot(data1,data3, "r-", label="Temperatura")
p3, = par2.plot(data1,data4, "g-", label="Vlaga")
host.set_xlabel("Vrijeme")
host.set_ylabel("Radon")
par1.set_ylabel("Temperatura")
par2.set_ylabel("Vlaga")
host.yaxis.label.set_color(p1.get_color())
```

```
par1.yaxis.label.set_color(p2.get_color())
par2.yaxis.label.set_color(p3.get_color())
tkw = dict(size=4, width=1.5)
host.tick_params(axis='y', colors=p1.get_color(), **tkw)
par1.tick_params(axis='y', colors=p2.get_color(), **tkw)
par2.tick_params(axis='y', colors=p3.get_color(), **tkw)
host.tick_params(axis='x', **tkw)
lines = [p1, p2, p3]
host.legend(lines, [l.get_label() for l in lines])
```

4.2. KOD ZA ODREĐIVANJE DOMINANTNE FREKVENCIJE VIBRACIJSKE PLOČE

```
import numpy as np
import matplotlib.pyplot as plt
TxtRead=np.loadtxt(fname = r"C:\Users\Heco\Documents\faks\gfv\diplomski\Diplomski rad\podaci\shaker\Test 1_0.1.txt",skiprows=26)
print(TxtRead)
res_list = [column[0] for column in TxtRead] # 1 red podatci
N=len(res_list)
from scipy.fftpack import fft
T = 1/1024
F=N/1024
x=F/N
s=np.arange(1, F+1, x)
plt.figure(figsize=(28, 10))
plt.plot(s,res_list)
plt.xlabel('vrijeme (s)',fontsize=20)
plt.ylabel('Akceleracija (mm/s^2)',fontsize=23)
x = np.linspace(0.0, 1.0/(2.0*T), int(N/2))
yr = fft(res_list)
y = 2/N * np.abs(yr[0:np.int(N/2)])
fig, ax = plt.subplots(1, 1, figsize=(20, 10))
ax.plot(x, y)
ax.set_xlabel('Frequency (Hz)',fontsize=18)
ax.set_ylabel('Fas',fontsize=18)
ax.set_title('Fourierova spektarna analiza',fontsize=14)
```

4.3. KOD ZA PRIKAZ ZAPISA POTRESA SA SEIZMOLOŠKIH STANICA

```
import numpy as np
import matplotlib.pyplot as plt
TxtRead=np.loadtxt(fname = r"C:\Users\Heco\Documents\faks\gfv\diplomski\Diplomski rad\podaci\potresi\ZIRJ__20160630_1200.txt",skiprows=8)
res_list = [column[0] for column in TxtRead] # 1 red podatci
N=len(res_list)
T=N/50
s=np.arange(1, T+1, 0.02)
res_list1 = [column[1] for column in TxtRead]
N1=len(res_list1)
T1=N1/50
s1=np.arange(1, T1+1, 0.02)
res_list2 = [column[2] for column in TxtRead]
N2=len(res_list2)
T2=N2/50
s2=np.arange(1, T2+1, 0.02)
plt.figure(figsize=(28, 10))
plt.style.use('seaborn-whitegrid')
plt.subplot(3,1,1)
plt.plot(s,res_list, color='black')
plt.ylabel('Comp Z',fontsize=20)
plt.title('Zirje',fontsize=20)
plt.subplot(3,1,2)
plt.ylabel('Comp N',fontsize=18)
plt.plot(s1,res_list1, color='blue')
plt.subplot(3,1,3)
plt.xlabel('Vrijeme (s)',fontsize=20)
plt.ylabel('Comp E',fontsize=20)
plt.plot(s2,res_list2, color='green')
```

4.4. KOD ZA ANALIZU PODATAKA KORELACIJE IZMEĐU v_s, N_{SPT} i N_{10H}

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from statsmodels.formula.api import ols
ExcellRead = pd.read_excel(r"C:\Users\Heco\Desktop\svi podaci.xlsx",skiprows=2)
ExcellRead=ExcellRead.sort_values(by='korigirano',ascending=True)
ExcellRead=ExcellRead.reset_index(drop=True)
df = ExcellRead
dflog = np.log(ExcellRead[['brzina','korigirano']])
model = ols('brzina ~ korigirano', data = dflog).fit()
model.summary()
dflog = np.log(ExcellRead[['brzina','original']])
model = ols('brzina ~ original', data = dflog).fit()
model.summary()
import math
A=math.e**(4.5326)
rez=A*ExcellRead[['korigirano']]**(0.3630)
A1=math.e**(4.5772)
rez1=A1*ExcellRead[['original']]**(0.3981)
fig, test=plt.subplots(figsize=(30,10))
test.scatter(ExcellRead['korigirano'],ExcellRead['brzina'], s = 100, color= 'blue',label='korigi
rano')
test.scatter(ExcellRead['original'],ExcellRead['brzina'], s = 100, color= 'green',label='origina
l')
test.plot(ExcellRead[['korigirano']],rez,label='korigirano')
test.plot(ExcellRead[['original']],rez1,label='original')
test.grid(True)
test.set_title('Vs - Ndph',fontsize=24)
test.set_ylabel('Vs(m/s)',fontsize=22)
test.set_xlabel('N-dph',fontsize=22)
test.tick_params(labelsize=18)
test.legend(loc=1,fontsize=20)
```

4.5. KOD ZA ANALIZU I PRIKAZ PODATAKA RAPS METODE U PRAĆENJU KAKVOĆE OTPADNE VODE

```
import pandas as pd
import numpy as np
import statistics as st
import matplotlib.pyplot as plt
ExcellRead = pd.read_excel(r"C:\Users\Heco\Documents\faks\gfv\diplomski\Diplomski rad\podaci\RAP
S PYthon.xlsx",skiprows=0)
rapstest1=(ExcellRead.iloc[: , 1]-
sum(ExcellRead.iloc[: , 1])/len(ExcellRead.iloc[: , 1]))/st.stdev(ExcellRead.iloc[: , 1])
test1=np.cumsum(rapstest1)
rapstest2=(ExcellRead.iloc[: , 2]-
sum(ExcellRead.iloc[: , 2])/len(ExcellRead.iloc[: , 2]))/st.stdev(ExcellRead.iloc[: , 2])
test2=np.cumsum(rapstest2)
rapstest3=(ExcellRead.iloc[: , 3]-
sum(ExcellRead.iloc[: , 3])/len(ExcellRead.iloc[: , 3]))/st.stdev(ExcellRead.iloc[: , 3])
test3=np.cumsum(rapstest3)
rapstest4=(ExcellRead.iloc[: , 4]-
sum(ExcellRead.iloc[: , 4])/len(ExcellRead.iloc[: , 4]))/st.stdev(ExcellRead.iloc[: , 4])
test4=np.cumsum(rapstest4)
rapstest5=(ExcellRead.iloc[: , 5]-
sum(ExcellRead.iloc[: , 5])/len(ExcellRead.iloc[: , 5]))/st.stdev(ExcellRead.iloc[: , 5])
test5=np.cumsum(rapstest5)
rapstest6=(ExcellRead.iloc[: , 6]-
sum(ExcellRead.iloc[: , 6])/len(ExcellRead.iloc[: , 6]))/st.stdev(ExcellRead.iloc[: , 6])
test6=np.cumsum(rapstest6)
plt.figure(figsize=(28, 10))
plt.style.use('seaborn-whitegrid')
plt.subplot(2,1,1)
plt.plot(ExcellRead.iloc[: , 0],ExcellRead.iloc[: , 1],label='KPK')
plt.ylabel('KPK',fontsize=18)
plt.title('RAPS i KPK')
plt.subplot(2,1,2)
plt.plot(ExcellRead.iloc[: , 0],test1,label='RAPS-KPK')
plt.ylabel('RAPS-KPK',fontsize=18)
plt.xlabel('Dani u mjesecu',fontsize=18)
plt.subplot(2,1,1)
plt.plot(ExcellRead.iloc[: , 0],ExcellRead.iloc[: , 4],label='KPK izlazni')
plt.subplot(2,1,2)
plt.plot(ExcellRead.iloc[: , 0],test4,label='RAPS-KPK izlazni')
plt.legend(loc=1,fontsize=14)
```

4.6. KOD ZA ANALIZU SREDNJE TEMPERATURE ZA VREMENSKO RAZDOBLJE OD GODINU DANA

```
1.  import pandas as pd
2.  import seaborn as sns
3.  import matplotlib.pyplot as plt
4.  from sklearn.linear_model import LinearRegression
5.  from sklearn.metrics import r2_score
6.  from sklearn.preprocessing import PolynomialFeatures
7.  ExcellRead = pd.read_excel(r"C:\Users\Heco\Documents\faks\gfv\diplomski\Diplomski rad\podaci\sre
    dnja_temp.xlsx",skiprows=0)
8.  fig, test=plt.subplots(figsize=(20,12))
9.  test.plot(ExcellRead[['dani']],ExcellRead['Sv.Ilija'],label='Sv.Ilija')
10. test.plot(ExcellRead[['dani']],ExcellRead['Domasinec'],label='Domasinec')
11. test.grid(True)
12. test.set_title('Srednja temperatura',fontsize=20)
13. test.set_ylabel('Temp',fontsize=18)
14. test.set_xlabel('dani',fontsize=18)
15. test.legend(loc=1,fontsize=14)
16. sns.regplot(x='dani',y='Sv.Ilija',data=ExcellRead, fit_reg=True)
17. sns.regplot(x='dani',y='Domasinec',data=ExcellRead, fit_reg=True)
18. x=ExcellRead[['dani']]
19. y=ExcellRead['Sv.Ilija']
20. y1=ExcellRead['Domasinec']
21. poly_reg = PolynomialFeatures(degree=4)
22. X_poly = poly_reg.fit_transform(x)
23. pol_reg = LinearRegression()
24. pol_reg.fit(X_poly, y)
25. fig, graf1=plt.subplots(figsize=(20,12))
26. graf1.plot(x, pol_reg.predict(poly_reg.fit_transform(x)), color='blue',label='Sv.Ilija')
27. graf1.scatter(x, y, color='red')
28. graf1.set_title('Prilagodjavanje polinomne krivulje 4og stupnja',fontsize=20)
29. graf1.set_xlabel('dani',fontsize=18)
30. graf1.set_ylabel('Temperatura',fontsize=18)
31. pol_reg.fit(X_poly, y1)
32. graf1.plot(x, pol_reg.predict(poly_reg.fit_transform(x)), color='green',label='Domasinec')
33. graf1.scatter(x, y1, color='orange')
34. graf1.legend(loc=1,fontsize=14)
35. model = LinearRegression()
36. model.fit(X_poly, y)
37. y_poly_pred = model.predict(X_poly)
38. r2y = r2_score(y,y_poly_pred)
39. model.fit(X_poly, y1)
40. r2y1 = r2_score(y1,y_poly_pred)
41. print(r2y)
42. print(r2y1)
```