

## VJEŽBA 3: RAD S PANDAS PYTHON BIBLIOTEKOM. EKSPLORATIVNA ANALIZA PODATAKA.

**I. Cilj vježbe:** *Upoznati se s načinom korištenja Pandas biblioteke za programski jezik Python. Upoznati se s eksplorativnom analizom podataka pomoću grafičkog prikaza podataka.*

### II. Opis vježbe:

U vježbi se studenti upoznaju s Pandas bibliotekom za programski jezik Python. Ova biblioteka omogućava relativno laganu manipulaciju podacima te, zajedno s grafičkom bibliotekom, omogućuje dobivanje uvida u karakteristike raspoloživih podataka (distribucije, srednje vrijednosti i sl.). Ovo je obično i prvi korak u problemima strojnog učenja, a poznat je i pod nazivom eksplorativna analiza podataka.

#### II.1. Pandas biblioteka

[Pandas](#) je *open source* Python biblioteka koja značajno olakšava učitavanje i analizu podataka u Pythonu. Osnovna struktura podataka u Pandas biblioteci su DataFrame i Series objekti zasnovani na Numpy koji omogućuje brzu i efikasnu manipulaciju pohranjenih podataka. U Pandas biblioteci su dostupni alati za učitavanje datoteka u kojima su pohranjeni podaci kao na primjer CSV i tekstualne datoteke, Excel, SQL baza i HDF5 datoteke. Učitani podaci spremaju se u DataFrame, a omogućen je ispis DataFrame u datoteke. Podržane su različite operacije nad DataFrameovima, kao izdvajanje, grupiranje i slično. Na taj način je moguće brzo dobiti uvid u karakteristike raspoloživih podataka.

##### II.1.1. Pandas Series

Series je jednodimenzionalni objekt sličan polju pri čemu je svakom elementu polja pridružen indeks (vrijednosti indeksa su 0 do N gdje je N duljina polja). Element polja može biti bilo koji tip podatka (cjelobrojne vrijednosti, realni brojevi, znakovni nizovi, Python objekti, itd.) kao što prikazuje primjer 3.1.

##### Primjer 3.1.

```
import pandas as pd
import numpy as np

s1 = pd.Series(['crvenkapica', 'baka', 'majka', 'lovac', 'vuk'])
print(s1)

s2 = pd.Series(5., index=['a', 'b', 'c', 'd', 'e'], name = 'ime_objekta')
print(s2)
print(s2['b'])

s3 = pd.Series(np.random.randn(5))
print(s3)
print(s3[3])
```

##### II.1.2. Pandas DataFrame

Struktura DataFrame je dvodimenzionalna označena struktura nalik tablici gdje su u stupcima pohranjeni podaci istog tipa (npr. liste, rječnici, numpy polja i sl.). DataFrame struktura može se shvatiti i kao grupa više Series struktura koje imaju isti indeks. DataFrame se najčešće definira pomoću rječnika koji sadrži liste. Pri tome ključ definira naziv stupca u DataFrameu (vidi primjer 3.2.).

##### Primjer 3.2.

```
import numpy as np
```

```
data = {'country': ['Italy', 'Spain', 'Greece', 'France', 'Portugal'],
        'population': [59, 47, 11, 68, 10],
        'code': [39, 34, 30, 33, 351]}

countries = pd.DataFrame(data, columns=['country', 'population', 'code'])
print(countries)
```

Vrlo čest problem je učitavanje podataka iz nekog vanjskog izvora. Ako je skup podataka zapisan u CSV datoteci (engl. *comma separated values*), podaci se mogu učitati u DataFrame pomoću naredbe `read_csv`. Jednom kada su podaci učitani, na raspolaganju su različite metode DataFramea s kojima je moguće dobiti informacije o podacima kako prikazuje primjer 3.3.:

- `.info()` – daje osnovne informacije o DataFrameu,
- `.head(n)` – vraća prvih `n` zapisa u DataFrameu,
- `.tail(n)` – vraća zadnjih `n` zapisa u DataFrameu,
- `.describe()` – vraća statistiku za svaku veličinu u DataFrameu.

Postoje gotove matematičke funkcije u obliku metoda:

- `.mean()` – srednja vrijednost svakog stupca (veličine)
- `.median()` – medijan vrijednost svakog stupca (veličine)
- `.max()` – maksimalna vrijednost svakog stupca (veličine)
- `.min()` – minimalna vrijednost svakog stupca (veličine)
- `.sort_values(by=['col'])` – sortiranje DataFrame prema željenom stupcu `col`

### Primjer 3.3.

```
import pandas as pd
import numpy as np

mtcars = pd.read_csv('mtcars.csv')
print(len(mtcars))
print(mtcars)

print(mtcars.head(5))
print(mtcars.tail(3))
print(mtcars.info())
print(mtcars.describe())
```

Izdvajanje pojedinog stupca iz DataFramea moguće je izvesti pomoću zagrada `[]` ili točke na način da se navede naziv stupca kako je prikazano u primjeru 3.4. Ako se želi izdvojiti više stupaca, potrebno je unutar zagrada predati listu koja sadrži nazive stupaca. Metoda `.iloc` izdvaja redove s određenim indeksima. Moguće je postaviti logičke uvjete na pojedine stupce – rezultat je DataFrame koji ima vrijednosti `True` ili `False` te se na taj način mogu izdvojiti samo redovi koji zadovoljavaju postavljeni uvjet.

### Primjer 3.4.

```
import pandas as pd
import numpy as np

print(mtcars['car'])
print(mtcars.cyl)

print(mtcars.cyl > 6)
print(mtcars[mtcars.cyl > 6])
print(mtcars[(mtcars.cyl == 4) & (mtcars.hp > 100)].car)
print(mtcars[['car', 'cyl']])
print(mtcars.cyl[2:4])
```

```
print(mtcars[5:12])
print(mtcars.mpg[3:5])

mtcars['jedinice'] = np.ones(len(mtcars))
mtcars['heavy'] = mtcars.wt > 4.5
print(mtcars[['car', 'heavy']])
print(mtcars.query('cyl == [4,6]').car)

print(mtcars.iloc[1:3, 5:10])
print(mtcars.iloc[:, 3:5])
print(mtcars.iloc[:, [0,4,7]])
print(mtcars.iloc[[1,29], :])
```

Grupiranje podataka u DataFrameu je brz način dobivanja karakterističnih vrijednosti raspoloživih podataka.

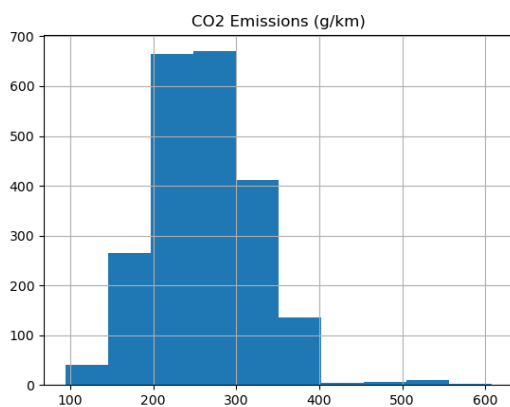
### Primjer 3.5.

```
import pandas as pd
import numpy as np

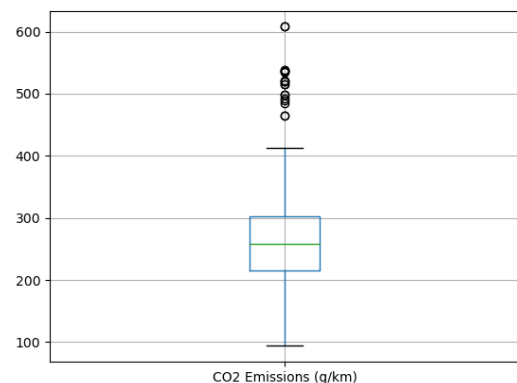
new_mtcars = mtcars.groupby('cyl')
print(new_mtcars.count())
print(new_mtcars.sum())
new_mtcars = mtcars.iloc[:, 1:]
new_mtcars = new_mtcars.groupby('cyl')
print(new_mtcars.mean())
```

## II.2. Eksplorativna analiza podataka

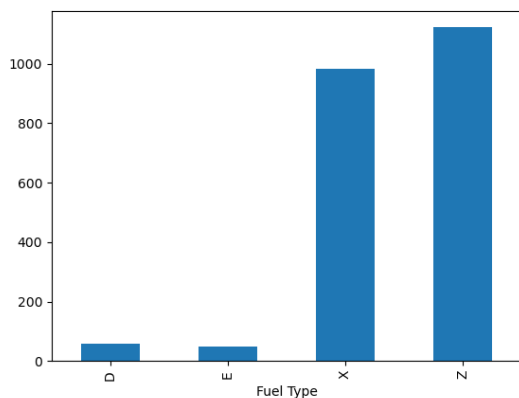
Eksplorativna analiza podataka je pristup analizi podataka na način da se sumiraju određene karakteristike podataka te prikazu grafički. Ovim postupkom se dobiva uvid u karakteristike podataka, u odnose među veličinama, eventualne probleme vezane za prikupljanje podataka i sl. Eksplorativna analiza podataka je važan korak prilikom rješavanja problema strojnog učenja jer na temelju nje se može usmjeriti postupak učenja te odlučiti koje hipoteze bi bilo smisleno istraživati. Grafički prikazi koji se mogu koristiti su različiti, ali najčešće se koriste: histogram, kutijasti dijagram (engl. *boxplot*), stupčasti dijagram (engl. *barplot*), dijagram raspršenja (engl. *scatterplot*). Primjeri dijagrama dani su na slici 3.1., a detalji se mogu potražiti na [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/visualization.html](https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html)



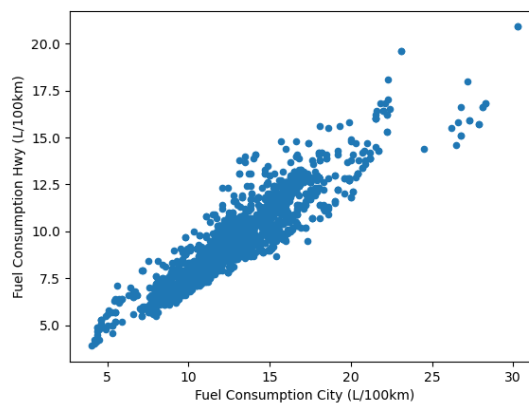
a) histogram



b) kutijasti dijagram



c) stupčasti dijagram



d) dijagram raspršenja

Sl. 3.1. Primjeri grafičkih prikaza.

### **III. Priprema za vježbu:**

Nema posebne pripreme za vježbu.

### **IV. Rad na vježbi:**

1. Isprobajte Python primjere iz II. Opis vježbe u Visual Studio Code IDE. Razmislite o svakoj liniji programskog koda i što je njen rezultat. Pokrenite primjere u *Debug* modu i pogledajte u *Explorer*-u kako izgleda svaka od varijabli u danim primjerima.
2. Riješite dane zadatke.

#### **Zadatak 1**

Za `mtcars` skup podataka napišite programski kod koji će odgovoriti na sljedeća pitanja:

1. Kojih 5 automobila ima najveću potrošnju? (koristite funkciju `sort`)
2. Koja tri automobila s 8 cilindara imaju najmanju potrošnju?
3. Kolika je srednja potrošnja automobila sa 6 cilindara?
4. Kolika je srednja potrošnja automobila s 4 cilindra mase između 2000 i 2200 lbs?
5. Koliko je automobila s ručnim, a koliko s automatskim mjenjačem u ovom skupu podataka?
6. Koliko je automobila s automatskim mjenjačem i snagom preko 100 konjskih snaga?
7. Kolika je masa svakog automobila u kilogramima?

#### **Zadatak 2**

Napišite programski kod koji će iscrtati sljedeće slike za `mtcars` skup podataka:

1. Pomoću `barplot`-a prikažite na istoj slici potrošnju automobila s 4, 6 i 8 cilindara.
2. Pomoću `boxplot`-a prikažite na istoj slici distribuciju težine automobila s 4, 6 i 8 cilindara.
3. Pomoću odgovarajućeg grafa pokušajte odgovoriti na pitanje imaju li automobili s ručnim mjenjačem veću potrošnju od automobila s automatskim mjenjačem?
4. Prikažite na istoj slici odnos ubrzanja i snage automobila za automobile s ručnim odnosno automatskim mjenjačem.

**Zadatak 3**

Na stranici <http://iszz.azo.hr/iskzl/exc.htm> moguće je dohvatiti podatke o kvaliteti zraka za Republiku Hrvatsku. Podaci se mogu preuzeti korištenjem RESTfull servisa u XML ili JSON obliku. Koristite skriptu `AirQualityRH.py` koja dohvaća podatke te ih pohranjuje u odgovarajući `DataFrame`. Prepravite/nadopunite skriptu s programskim kodom kako bi dobili sljedeće rezultate:

1. Dohvaćanje mjerenja dnevne koncentracije lebdećih čestica  $PM_{10}$  za 2017. godinu za grad Osijek.
2. Ispis tri datuma u godini kada je koncentracija  $PM_{10}$  bila najveća.

**V. Izvještaj s vježbe**

Kao izvještaj s vježbe prihvaća se web link na repozitorij pod nazivom PSU\_LV.