

Domaći zadatak broj 1

Ime i prezime studenta: Nastasija Stankovic Broj indeksa: 17955

Uputstvo: 1. Pre početka izrade promenite ime datoteke u Domaci1_ime_prezime. (ubacite svoje ime i prezime) 2. Popunite ćeliju ispod naslova odgovarajućim podacima. 3. Prvo rešite prvi zadatak kako bi naredbe modula bile dostupne. 4. Za rešavanje zadataka, ukoliko je potrebno, otvorite ispod teksta zadataka dodatne ćelije za upisivanje tekstualnog odgovora (Markdown) ili programskog koda (Code). 5. Nakon završetka izrade rešenja Notebook dokument sačuvati u pdf formatu i proslediti ga nastavniku. To možete da uradite ili kroz Teams ili na mail adresu jovana.dzunic@elfak.ni.ac.rs

Zadatak 1. Učitati NumPy modul i dati mu skraćeno ime np ili npy, ili neko drugo prema sopstvenom izboru. Tu skraćenicu koristiti u pozivu naredbi modula NumPy u ostalim zadacima.

```
In [3]: import numpy as np
```

Zadatak 2. Podeliti ovu ćeliju na delove (split cell), gde će svaki od delova počinjati pod-zadatkom, tj. počinjati sa a), b), itd. Zatim između ćelija sa pod-zadacima kreirati kodne ćelije za njihovo rešavanje. Napisati kod za:

a) Generisanje promenljive a koja je niz brojeva od 0,1,2,...,15. Brojeve ne unositi ručno već koristiti neku od naredbi NumPy modula za automatsko generisanje nizova.

```
In [6]: a=np.arange(16)
print(a)

[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]
```

b) Generisanje promenljive b, na osnovu postojećeg niza a, upotrebom operacija nad nizovima (+,-,*,/ i slično). Elementi niza b glase -3, -1, 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27. Uočiti pravilnost (linearna je) i iskoristiti je za kreiranje niza b.

```
In [7]: b=2*a-3
print (b)

[-3 -1  1  3  5  7  9 11 13 15 17 19 21 23 25 27]
```

c) Generisanje vektora cb centriranih vrednosti niza b.

```
In [8]: cb=b-np.mean(b)
print ("Centriran niz cb: ", cb)

Centriran niz cb: [-15. -13. -11.  -9.  -7.  -5.  -3.  -1.   1.   3.   5.   7.   9.  11.
 13.  15.]
```

d) Indeksiranjem promeniti sadržaj vektora a prema sledećem pravilu:

$$[0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13\ 14\ 15] \mapsto [0\ 1\ 2\ 3\ 4\ 5\ -1\ 2\ -1\ 2\ 10\ 11\ 0\ 0\ 0\ 0]$$

```
In [9]: a=np.arange(16)
print("Niz a pre izmene: ",a)
a[6:10]=2*[-1, 2]
a[12:]=[0]
print("Niz a posle izmene: ",a)

Niz a pre izmene: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]
Niz a posle izmene: [ 0  1  2  3  4  5 -1  2 -1  2 10 11  0  0  0  0]
```

Zadatak 3. U kodnoj ćeliji ispod teksta zameniti ime modula numpy odgovarajućom skraćenicom iz zadatka 1.

Iskoristiti kod za kreiranje promenljive pom, tj. aktivirati kodnu ćeliju kako bi pom bila promenljiva koju interpretator prepozna.

```
In [10]: pom=3*np.random.rand(10,)+2
print (pom)

[2.56611912  2.19417245  3.17743593  3.32414358  3.43713385  3.30752853
 4.36104523  3.86579127  3.17561614  3.13397718]
```

a) Napisati kod za izračunavanje raspona između maksimalne i minimalne vrednosti unutar niza pom, na sledeći način:

1. Odrediti minimalnu vrednost minpom niza pom koristeći odgovarajuću naredbu
2. Oduzeti minpom od svih elemenata niza pom i rezultat smestiti u niz c.
3. Odrediti maksimalnu vrednost raspon niza c.
4. Odštampati poruku o vrednosti traženog raspona.

```
In [11]: minpom=min(pom)
c=pom-minpom
raspon=max(c)
print("Raspon je",raspon)

Raspon je 2.166872772305686
```

b) Uporediti vrednost raspona sa standardnim odstupanjem sigma vektora pom, tj. kreirati promenljivu

$$x = raspon - sigma$$

```
In [12]: sv=np.mean(pom)
y=pom-sv
norma=np.linalg.norm(y)
sigma=norma/np.sqrt(pom.size)
x=raspon-sigma
print(x)
```

1.595559694205027

c) Koristeći Čebiševljevu nejednakost odrediti moguć procenat komponenti vektora pom koje imaju odstupanje od srednje vrednosti mu veće od $a = \frac{raspon - sigma}{2} = \frac{x}{2}$.

```
In [19]: a=x/2
print (a)
sknj=np.linalg.norm(pom)/np.sqrt(len(pom))
print ("Srednje-kvadratna norma vektora pom:", sknj)
brk=np.count_nonzero(pom>a)
print ("Broj komponenti vektora pom čije su apsolutne vrednosti veće od odstupanja:", brk)
alfa=a/sknj
print (alfa)
```

0.7977798471025135
Srednje-kvadratna norma vektora pom: 3.304064650607947
Broj komponenti vektora pom čije su apsolutne vrednosti veće od odstupanja: 10
0.24145406687357712

Kako je u ovom slučaju $a < ||v||_{sr}$, pomoću Čebiševljeve nejednakosti nećemo dobiti netrivialne informacije.

Zadatak 4. U kodnoj ćeliji ispod teksta zameniti ime modula numpy odgovarajućom skraćenicom iz zadatka 1. Iskoristiti kod za kreiranje promenljivih u i v, tj. aktivirati kodnu ćeliju.

```
In [14]: ku=np.random.randint(-5,5)
kv=np.random.randint(-5,5)
u=np.random.rand(20,)-ku*np.random.rand(20,)
v=kv*np.random.rand(20,)+3
print(v)
```

[1.01543569 0.87040005 0.46876363 0.74298156 -0.17371921 2.49735427
-0.0869934 1.95563606 0.93139466 -0.71111883 1.66781211 2.86081063
 0.73508989 -0.905867 1.68958754 -0.01891249 -0.53933714 2.49890125
 2.18932049 1.54583721]

Izračunati L_1 i L_2 normu vektora v.

```
In [15]: L1=(np.linalg.norm(v,1))
L2=(np.linalg.norm(v))
print ("L1 norma vektora v je ",L1)
print ("L2 norma vektora v je ",L2)
```

L1 norma vektora v je 24.10527312911946
L2 norma vektora v je 6.552302597816771

Unutar markdown ćelije napisati matematički izraz po kom se računa L_∞ norma.

$L_\infty = \max\{|x_1|, |x_2|, \dots, |x_n|\}$

Unutar markdown ćelije napisati matematičke izraze po kojima se računaju kovarijansa i koeficijent korelacije dva vektora.

Kovarijanska se racuna po formuli:

$$cov(v,u) = \frac{\dot{v} \cdot \dot{u}}{n}$$

Koeficijent korelacije se racuna po formuli:

$$p(v,u) = \frac{\dot{v} \cdot \dot{u}}{||\dot{v}|| ||\dot{u}||}$$

Koristeći kod, izračunati kovarijansu i koeficijent korelacije vektora u i v.

```
In [16]: L=v-np.mean(v)
M=u-np.mean(u)
c=(np.dot(L,M))/(len(v))
print("Kovarijansa vektora je ", c)
k=(np.dot(L,M))/(np.linalg.norm(L)*np.linalg.norm(M))
print("Koeficijent korelacije vektora je ", k)
```

Kovarijansa vektora je -0.45586314130337324
Koeficijent korelacije vektora je -0.4434141773735409

Zadatak 5. Koristeći naredbe za dijagonalne matrice i osnovne aritmetičke operacije kreirati matricu A :

$$\begin{bmatrix} -1 & 2 & 1 & 0 & 0 \\ 3 & -1 & 2 & 1 & 0 \\ 0 & 3 & -1 & 2 & 1 \\ 0 & 0 & 3 & -1 & 2 \\ 0 & 0 & 0 & 3 & -1 \end{bmatrix}$$

```
In [17]: A=np.diag(5*[-1])+np.diag(4*[2], 1)+np.diag(3*[1], 2)+np.diag(4*[3], -1)
print (A)
```

[[-1 2 1 0 0]
 [3 -1 2 1 0]
 [0 3 -1 2 1]
 [0 0 3 -1 2]
 [0 0 0 3 -1]]

Ako je

$$A = \begin{bmatrix} -1 & 2 & 1 & 0 & 0 \\ 3 & -1 & 2 & 1 & 0 \\ 0 & 3 & -1 & 2 & 1 \end{bmatrix}$$

u narednim pod-zadacima označiti polje odgovora koje smatrate tačnim.

a) Šta će biti rezultat sledećeg koda? (Za opisivanje rezultata koda koristiti standardno prebrojavanje vrsta i kolona, tj. od 1,2,...)

A[:,2,:]

- ☐ Izdvajanje druge vrste matrice A
- ☐ Izdvajanje druge kolone matrice A
- ☒ Izdvajanje prve i treće vrste matrice A
- ☐ Izdvajanje prve i treće kolone matrice A

b) Šta će biti rezultat sledećeg koda? (Za opisivanje rezultata koda koristiti standardno prebrojavanje vrsta i kolona, tj. od 1,2,...)

A[3,1]

- ☐ Izdvajanje elementa iz treće vrste i prve kolone
- ☐ Izdvajanje elementa iz prve vrste i treće kolone
- ☒ Poruka greške
- ☐ Izdvajanje treće i prve vrste matrice A

c) Šta će biti rezultat sledećeg koda? (Za opisivanje rezultata koda koristiti standardno prebrojavanje vrsta i kolona, tj. od 1,2,...)

A[:,[0,4]] = A[:,[1,2]]

- ☐ Zamena mesta prve i poslednje vrste.
- ☐ Zamena mesta druge i treće kolone.
- ☒ Kopiranje sadržaja druge i treće kolone u prvu i poslednju kolonu, redom.
- ☐ Poruka greške