

Obrada informacija

4. Laboratorijska vježba: Multivarijantni financijski vremenski nizovi

Prosinac 2021.

Upute

U ovoj bilježnici dana je priprema sa svim uputama za 4. laboratorijsku vježbu iz predmeta Obrada informacija - uz bilježnicu su dostupni i podatci u datoteci `ETFprices.csv`.

Vaš zadatak je u bilježnicu na odgovarajuća mjesta dopisati kod Vašeg rješenja, te odgovore na zadana pitanja.

Riješenu bilježnicu potrebno je predati kao izvještaj u .pdf formatu na Moodle najkasnije do 9.1.2022. u 23:59h. Datoteka koju predajete se mora zvati *Prezimelme.pdf*.

▼ Uvod

U laboratorijskoj vježbi razmatra se dinamika cijena vrijednosnica na financijskim tržištima. Dane su povijesne dnevne cijene 8 ETF-ova (eng. *exchange traded fund*) koji prate određene dioničke, obvezničke ili druge indekse.

Ticker	Fond	Klasa imovine
SPY	SPDR S&P 500 ETF Trust	Equity: U.S. - Large Cap
VTI	Vanguard Total Stock Market ETF	Equity: U.S. - Total Market
QQQ	Invesco QQQ Trust	Equity: U.S. - Large Cap
VEA	Vanguard FTSE Developed Markets ETF	Equity: Developed Markets Ex-U.S. - Total Market
AGG	iShares Core U.S. Aggregate Bond ETF	Fixed Income: U.S. - Broad Market, Broad-based Investment G
BND	Vanguard Total Bond Market ETF	Fixed Income: U.S. - Broad Market, Broad-based Investment G
LQD	iShares iBoxx USD Investment Grade Corporate Bond ETF	Fixed Income: U.S. - Corporate, Broad-based Investment Grad
VCIT	Vanguard Intermediate-Term Corporate Bond ETF	Fixed Income: U.S. - Corporate, Broad-based Investment Grad

Pri modeliranju zajedničkog kretanja i rizika vrijednosnica, najčešće se koriste povrati:

$$R(t) = \frac{S(t) - S(t-1)}{S(t-1)},$$

gdje je $S(t)$ cijena vrijednosnice u danu t . U sklopu ove laboratorijske vježbe cilj je analizirati kretanje danih ETF-ova i izračunati glavne komponente (PCA) koje utječu na njihovu dinamiku. Laboratorijsku vježbu je potrebno riješiti unutar ove bilježnice i predati riješenu bilježnicu kao izvještaj.

```
import pandas as pd
import numpy as np
```

```
import math as m
import seaborn as sn
import matplotlib.pyplot as plt
from google.colab import drive

drive.mount('/content/drive')

prices = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/ETFprices.csv')
prices.set_index('Time', inplace=True)
prices.index = pd.to_datetime(prices.index)

prices.head()
```

Mounted at /content/drive

	SPY	VTI	QQQ	VEA	AGG	BND	LQD	VCIT
Time								
2019-01-02	245.5347	125.3143	153.6161	35.8827	103.6870	77.2074	109.3645	80.1500
2019-01-03	239.6755	122.4103	148.5974	35.5922	104.1151	77.5089	109.4129	80.3336
2019-01-04	247.7036	126.4622	154.9550	36.6866	103.8038	77.2755	109.3452	80.1597
2019-01-07	249.6567	127.7376	156.7999	36.8028	103.6286	77.1587	109.3549	80.0534
2019-01-08	252.0023	129.0522	158.2182	37.0643	103.5508	77.0615	109.5965	80.1211

Zadatak 1 - Računanje korelacijske matrice i matrice kovarijance povrata

1.1. U prvom zadatku ove laboratorijske vježbe potrebno je prvo iz danih cijena (gore učitanih u Pandas DataFrame) izračunati dnevne povrate za sve pojedine vrijednosnice (prateći formulu danu u uvodu).

Izračunajte srednje povrate i volatilnost (standardnu devijaciju povrata) za svaku pojedinu vrijednosnicu. Pri analizi srednjih povrata i volatilnosti, te se brojke često *anualiziraju* - to znači da se srednji povrati pomnože s 252 (cca. broj trgovinskih dana u godini), a volatilnost s $\sqrt{252}$. Izračunajte anualizirane srednje povrate i volatilnosti te rezultate ispišite u konzolu.

#Vaš kod ide ovdje

```
#srednji povrati svih ETF-ova
#volatilnost svih ETF-ova
```

```
#df = prices
data = {}
```

```

#fullArray = []
for (name, content) in prices.items():
    #print('Name: ' + name)
    helpArray = []
    helpArray.append(0)
    #first = True
    for i in range(1, content.size) :
        helpArray.append((content[i] - content[i - 1]) / content[i - 1])
    data[name] = helpArray
df = pd.DataFrame(data)

print('Data Frame: ')
print(df.to_string())
print('\n')
print('Annual average retrurns: ')
print((df.mean() * 252).to_string())
print('\n')
print('Annual std: ')
print((df.std() * m.sqrt(252)).to_string())

```

213	-0.001106	-0.000895	0.000549	-0.000232	-0.003728	-0.003452	-0.005753	-0.0
214	0.000228	-0.000448	-0.002247	-0.000930	0.001514	0.002150	0.005628	0.0
215	0.003517	0.003393	0.003354	0.004422	-0.004716	-0.005125	-0.007803	-0.0
216	0.002466	0.002807	0.003991	-0.002086	-0.000536	-0.000959	-0.001351	-0.0
217	-0.001910	-0.001972	-0.001342	-0.000929	0.000179	0.000600	0.000318	0.0
218	0.002108	0.001722	0.002886	0.000232	0.001073	0.000599	0.000954	0.0
219	0.000324	0.000445	0.000248	-0.003253	0.001787	0.001796	0.002383	0.0
220	0.001456	0.001018	-0.000794	-0.000699	0.002853	0.002870	0.005231	0.0
221	0.007236	0.007626	0.007347	0.006066	-0.000533	-0.000358	0.000158	-0.0
222	0.000738	0.000568	0.000789	0.001159	0.001867	0.001431	0.001104	0.0
223	-0.000289	0.000315	0.001477	-0.001853	0.000710	0.001072	0.001968	0.0
224	-0.003719	-0.003151	-0.005999	-0.006033	0.002396	0.002736	0.001415	0.0
225	-0.001609	-0.002529	-0.002226	-0.001401	-0.001416	-0.001661	-0.002354	-0.0
226	0.002224	0.002219	0.000645	0.001169	0.000532	0.000594	0.002753	0.0
227	0.007750	0.009485	0.011841	0.007706	0.000975	0.000950	0.003452	0.0
228	0.002266	0.002129	0.001910	0.000695	0.001415	0.001661	0.002580	0.0
229	0.004458	0.004751	0.006989	0.002779	-0.001237	-0.001540	-0.000078	-0.0
230	-0.003709	-0.004479	-0.004562	-0.007621	-0.001239	0.000000	-0.002028	-0.0
231	-0.008495	-0.009061	-0.010239	-0.004421	-0.001101	-0.002611	-0.004329	-0.0
232	-0.006706	-0.005928	-0.007832	-0.003740	0.004533	0.005364	0.007712	0.0
233	0.006170	0.006471	0.005064	0.008681	-0.001681	-0.002015	-0.001328	-0.0
234	0.001798	0.001450	0.001976	-0.000930	-0.001153	-0.001069	-0.001095	-0.0
235	0.009134	0.008875	0.010699	0.008614	-0.001421	-0.001309	-0.000548	-0.0
236	-0.003144	-0.003182	-0.004537	-0.005309	0.000533	0.000239	-0.000157	0.0
237	-0.001115	-0.000876	-0.000833	0.000928	-0.000355	-0.000715	0.000000	-0.0
238	0.002839	0.002380	0.005297	0.005564	0.002666	0.003336	0.003917	0.0
239	0.008619	0.008187	0.007464	0.007841	-0.004254	-0.004750	-0.004604	-0.0
240	0.000599	0.000000	0.003293	0.005719	0.004006	0.004533	0.006507	0.0
241	0.006870	0.007314	0.010039	0.010008	-0.002040	-0.002731	-0.002258	-0.0
242	0.000219	0.000369	0.000574	-0.003603	0.000089	0.000118	-0.000781	0.0
243	0.000063	0.000430	0.000764	-0.002486	-0.001954	-0.001309	-0.000703	-0.0
244	0.004099	0.004366	0.006252	0.000906	0.001072	0.000715	0.000987	0.0
245	0.004363	0.005142	0.004031	0.001585	0.000178	0.000239	-0.000313	-0.0
246	0.001528	0.000609	0.002634	0.001356	-0.000712	-0.000628	0.000470	-0.0
247	0.000031	0.000219	0.000519	-0.000912	0.001247	0.000956	0.001018	0.0
248	0.005323	0.004834	0.008824	0.004564	0.000890	0.000955	0.001799	0.0
249	-0.000248	-0.000913	-0.000842	0.002726	0.001334	0.001431	0.001405	0.0

```

250 -0.005513 -0.005424 -0.006554 -0.007250 0.000000 -0.000357 0.001793 0.0
251 0.002429 0.002635 0.001885 0.005477 -0.002308 -0.001072 -0.004125 -0.0

```

Annual average retrurns:

```

SPY      0.278633
VTI      0.274802
QQQ      0.338290
VEA      0.211612
AGG      0.080953
BND      0.083231
LQD      0.158252
VCIT     0.131246

```

Annual std:

```

SPY      0.125082
VTI      0.126120
QQQ      0.161893
VEA      0.111774
AGG      0.032298
BND      0.033657
LQD      0.048506

```

1.2 Koristeći dnevne povrate, potrebno je izračunati matricu kovarijance Σ i matricu korelacije C svih ETF-ova. Kovarijancu i korelaciju moguće je iz podataka izračunati koristeći Pandas, ali i NumPy ili neke druge biblioteke. Matrice ispišite u konzolu ili vizualizirajte.

#Vaš kod ide ovdje

```

# matrica korelacije C
corrMatrix = df.corr()
sn.heatmap(corrMatrix, annot = True)
print('Korelacijska matrica: ')
print(corrMatrix.to_string())
print('\n')
plt.show()

```

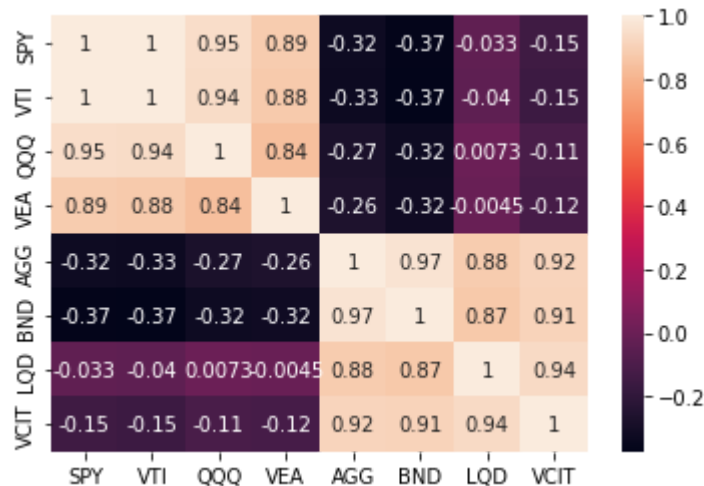
```

# matrica kovarijance Σ
covMatrix = df.cov()
sn.heatmap(covMatrix, annot = True)
print('\nMatrica kovarijance: ')
print(covMatrix.to_string())
print('\n')
plt.show()

```

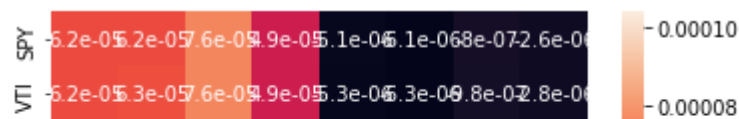
Korelacijska matrica:

	SPY	VTI	QQQ	VEA	AGG	BND	LQD	
SPY	1.000000	0.995610	0.946714	0.885594	-0.320296	-0.365559	-0.033173	-0.
VTI	0.995610	1.000000	0.941159	0.877538	-0.327880	-0.373639	-0.040173	-0.
QQQ	0.946714	0.941159	1.000000	0.839308	-0.272319	-0.320930	0.007319	-0.
VEA	0.885594	0.877538	0.839308	1.000000	-0.263013	-0.317639	-0.004507	-0.
AGG	-0.320296	-0.327880	-0.272319	-0.263013	1.000000	0.972636	0.884654	0.
BND	-0.365559	-0.373639	-0.320930	-0.317639	0.972636	1.000000	0.874278	0.
LQD	-0.033173	-0.040173	0.007319	-0.004507	0.884654	0.874278	1.000000	0.
VCIT	-0.147897	-0.154328	-0.108278	-0.120053	0.915354	0.912699	0.938561	1.



Matrica kovarijance:

	SPY	VTI	QQQ	VEA	AGG	BND	LQD	VCIT
SPY	6.208503e-05	6.232559e-05	7.607481e-05	4.913253e-05	-0.000005	-0.000005	-0.000005	-0.000005
VTI	6.232559e-05	6.312012e-05	7.625631e-05	4.908976e-05	-0.000005	-0.000005	-0.000005	-0.000005
QQQ	7.607481e-05	7.625631e-05	1.040057e-04	6.026862e-05	-0.000006	-0.000006	-0.000006	-0.000006
VEA	4.913253e-05	4.908976e-05	6.026862e-05	4.957725e-05	-0.000004	-0.000004	-0.000004	-0.000004
AGG	-5.134707e-06	-5.299927e-06	-5.650382e-06	-3.767810e-06	0.000004	0.000004	0.000004	0.000004
BND	-6.106974e-06	-6.293774e-06	-6.939258e-06	-4.741865e-06	0.000004	0.000004	0.000004	0.000004
LQD	-7.986782e-07	-9.752625e-07	2.280766e-07	-9.695734e-08	0.000005	0.000005	0.000005	0.000005
VCIT	-2.617702e-06	-2.754200e-06	-2.480484e-06	-1.898803e-06	0.000004	0.000004	0.000004	0.000004



▼ Zadatak 2 - Analiza glavnih komponenti

2.1. Izračunajte svojstvene vektore i pripadajuće svojstvene vrijednosti matrice kovarijance povrata Σ (svojstvenu dekompoziciju možete pronaći u sklopu biblioteke NumPy <https://numpy.org/doc/stable/reference/generated/numpy.linalg.eig.html>). Poredajte komponente padajući po svojstvenim vrijednostima i prikažite svojstvene vrijednosti grafički.

```
#Vaš kod ide ovdje
from numpy import linalg as LA

eigvalues, eigvectors = LA.eig(covMatrix)
eigvaluesorted = -np.sort(-eigvalues)
```

```

print('Svojstveni vektori: ')
print(eigvectors)
print('\n')
print('Svojstvene vrijednosti: ')
print(eigvaluessorted)
plt.plot(eigvaluessorted, marker='.')
print('\nGraficki prikaz sortiranih svojstvenih vrijednosti: ')
plt.show()

```

Svojstveni vektori:

```

[[ 4.80141562e-01  1.98999052e-02 -3.28459435e-03 -4.74179062e-01
  -9.02516784e-02 -6.43383587e-02  7.25875702e-01 -7.09308011e-02]
 [ 4.82465797e-01  3.59744692e-02 -1.81422564e-02 -5.56115395e-01
   2.37895667e-02  5.28889782e-02 -6.71700258e-01  4.23638342e-02]
 [ 6.13048969e-01 -1.07258473e-01 -5.28100570e-01  5.76827241e-01
  -1.07136556e-02  9.13090425e-03 -2.78281072e-02  8.61069665e-03]
 [ 3.95671118e-01 -2.60513549e-02  8.48221831e-01  3.49226851e-01
  -9.94326419e-03  1.55511499e-02 -2.73985724e-02  1.53178349e-02]
 [-3.96872474e-02 -4.06139885e-01  1.95344469e-02 -1.71531431e-02
  -5.40036163e-01 -6.86936914e-01 -1.24427725e-01 -2.31939327e-01]
 [-4.78191384e-02 -4.16710298e-01  8.77893650e-03 -3.68551328e-02
  -5.18053998e-01  7.20638888e-01  2.17883884e-04 -1.86895928e-01]
 [-5.40209547e-03 -6.53162882e-01  2.72226008e-02 -7.20513862e-02
   6.56381929e-01 -5.91171261e-03  1.95646455e-02 -3.69000859e-01]
 [-2.03772771e-02 -4.70008456e-01  9.47882461e-03 -6.60067611e-02
   1.47919288e-02 -3.87355265e-02  6.73280117e-02  8.76343411e-01]]

```

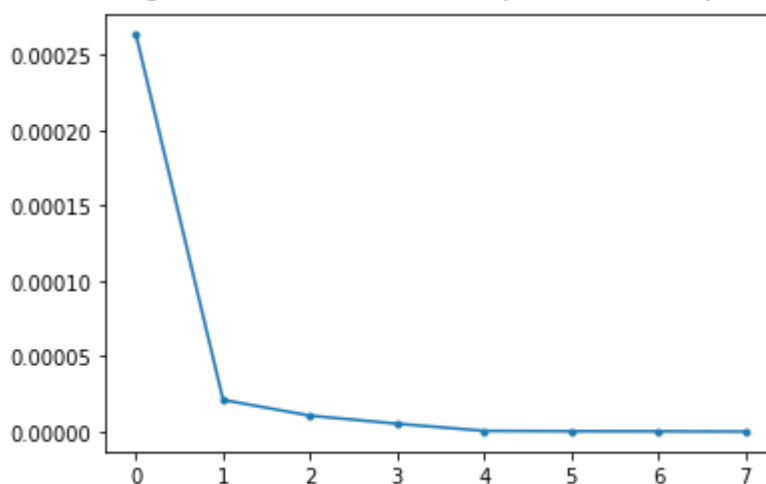
Svojstvene vrijednosti:

```

[2.63486759e-04 2.11174934e-05 1.06537725e-05 5.30545849e-06
 5.58942269e-07 3.20205407e-07 2.54064905e-07 1.08751662e-07]

```

Graficki prikaz sortiranih svojstvenih vrijednosti:



2.2. Izračunajte koliki udio varijance objašnjavaju prve dvije komponente?

```

#Vaš kod ide ovdje
varFirstLast = eigvaluessorted[0] + eigvaluessorted[1]
print('Varijanca prve i druge: ', varFirstLast)

```

Varijanca prve i druge: 0.0002846042522331656

2.3. Komponente PCA će u financijama često opisivati neke zajedničke faktore u podacima, što je moguće analizirati promatranjem pojedinih elemenata svojstvenih vektora. Ako je neki element određenog svojstvenog vektora velik po magnitudi (pozitivan ili negativan), to znači da ta komponenta opisuje odgovarajuću vrijednosnicu i objašnjava njenu varijancu, za razliku od slučaja kad je element blizu 0, što znači da razmatrana vrijednosnica ne ovisi previše o toj komponenti. Ispišite (ili grafički prikažite) elemente prva dva svojstvena vektora. Pritom pripazite na to što vraća funkcija koju koristite i u kojoj se dimenziji (stupac ili red) nalaze svojstveni vektori.

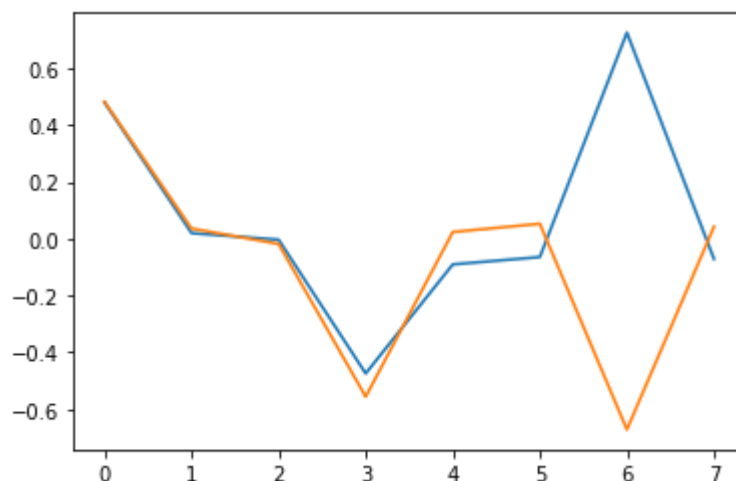
S obzirom na to koje vrijednosnice opisuju prve dvije komponente, možete li zaključiti koju klasu imovine opisuje prva komponenta (koja odgovara prvom svojstvenom vektoru), a koju klasu druga komponenta? (odgovor napišite u nastavku)

#Vaš kod ide ovdje

```
# Ispišite (ili grafički prikažite) elemente prva dva svojstvena vektora
print('Elementi prvog vektora: ')
print(eigvectors[0])
print('Elementi drugog vektora: ')
print(eigvectors[1])
print('\n')
print('Graficki prikaz prvog i drugog vektora: ')
plt.plot(eigvectors[0])
plt.plot(eigvectors[1])
plt.show()
```

```
Elementi prvog vektora:
[ 0.48014156  0.01989991 -0.00328459 -0.47417906 -0.09025168 -0.06433836
  0.7258757  -0.0709308 ]
Elementi drugog vektora:
[ 0.4824658  0.03597447 -0.01814226 -0.55611539  0.02378957  0.05288898
 -0.67170026  0.04236383]
```

Graficki prikaz prvog i drugog vektora:



Prva komponenta opisuje klasu imovine: Fixed Income: U.S. - Corporate, Broad-based Investment Grade
Druga komponenta opisuje klasu imovine: Fixed Income: U.S. - Corporate, Broad-based Investment Grade

2.4. Ponovite prethodnu analizu za matricu korelacije povrata C . Koliki udio varijance u tom slučaju objašnjavaju prve dvije komponente? Usporedite elemente prva dva svojstvena vektora u ovom slučaju i u prethodnom slučaju - razlikuje li se interpretacija i kako? (*odgovor napišite u nastavku*)

#Vaš kod ide ovdje

```
eigvaluesCorr, eigvectorsCorr = LA.eig(corrMatrix)
eigvaluesCorrSorted = -np.sort(-eigvaluesCorr)

print('Svojstveni vektori: ')
print(eigvectorsCorr)
print('\n')
print('Svojstvene vrijednosti: ')
print(eigvaluesCorrSorted)
plt.plot(eigvaluesCorrSorted, marker='.')
print('\nGraficki prikaz sortiranih svojstvenih vrijednosti: ')
plt.show()

print('Elementi prvog vektora: ')
print(eigvectorsCorr[0])
print('Elementi drugog vektora: ')
print(eigvectorsCorr[1])
print('\n')
print('Graficki prikaz prvog i drugog vektora: ')
plt.plot(eigvectorsCorr[0])
plt.plot(eigvectorsCorr[1])
plt.show()
```



```

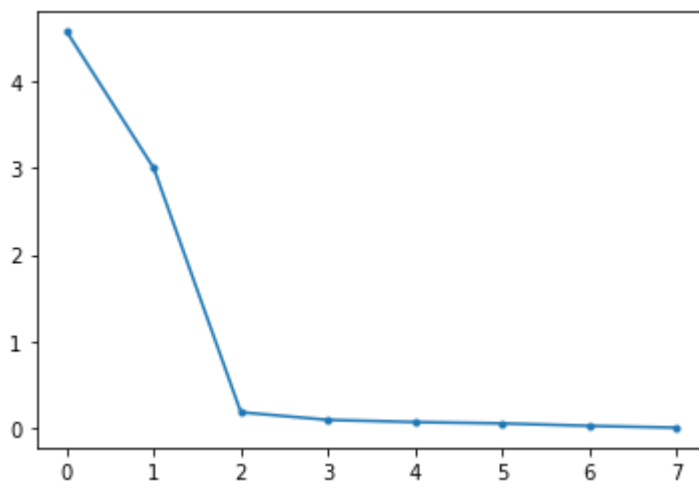
-0.09194669 -0.04965729]
[-0.34964798  0.35123903  0.40291042 -0.03845442 -0.22274576 -0.7168311
  0.15744739  0.04805941]
[-0.33968035  0.33422815 -0.86360342 -0.02731805 -0.00105767 -0.13403079
  0.07608587  0.05096295]
[ 0.37991095  0.31584872 -0.07713342  0.01534888 -0.54541171  0.03814473
 -0.05344505 -0.66928079]
[ 0.39433073  0.29167166 -0.01919693 -0.03371082 -0.4464489  0.12187307
 -0.0356369  0.73653671]
[ 0.28338945  0.44075125  0.03251849  0.00821958  0.51190248 -0.20654725
 -0.64765421 -0.01190982]
[ 0.32582822  0.39445536  0.07059609  0.0052388  0.4358444  0.09552946
  0.7294709  -0.04489637]]

```

Svojstvene vrijednosti:

```
[ 4.57416269e+00  2.99849003e+00  1.81886290e-01  9.41452118e-02
  6.86729858e-02  5.33146211e-02  2.52910456e-02  4.03712780e-03]
```

Graficki prikaz sortiranih svojstvenih vrijednosti:



Elementi prvog vektora:

$$\begin{bmatrix} -0.3709257 & 0.34178702 & 0.18545261 & 0.73238826 & -0.03866292 & 0.40989288 \\ -0.07250474 & 0.00682597 & & & & \end{bmatrix}$$

Elementi drugog vektora:

```
[ -0.37212232  0.33677121  0.21236831 -0.67817063 -0.00966157  0.48174791
 -0.09194669 -0.04965729]
```

Vidimo po grafovima da su prva dva svojstvena vektora obrnuto proporcionalni

▼ Zadatak 3 - Svojstveni portfelji

U primjeni PCA i svojstvenoj dekompoziciji kovarijance u financijama, svojstveni vektori se često zovu i tzv. svojstveni portfelji.

Općenito, portfelj je vektor $w = [w_1, \dots, w_N]$ u kojem svaki element predstavlja težinu ili udio kapitala u određenoj vrijednosnici. Često je dobro pomnožiti njihove težine s predznakom njihove sume - na taj način zapravo samo "okrećemo" predznak svojstvenog vektora tako da mu je suma pozitivna (konačni PCA rastav je i dalje isti ako svojstveni vektor pomnožimo s -1). Također, dobro je i skalirati svojstvene portfelje sa sumom njihovih apsolutnih vrijednosti:

$$\tilde{w}_i = \frac{w_i}{\sum_j^N |w_j|}.$$

Na taj način se osigurava da visoke magnitude pojedinih elemenata ne uzrokuju velike razlike u volatilitetima svojstvenih portfelja.

Ukoliko znamo povrate $R \in \mathbb{R}^{T \times N}$ (gdje je $R_i \in \mathbb{R}^T$ vektor povrata za vrijednosnicu i) za N vrijednosnica u nekom vremenskom periodu od T dana, povrate portfelja w u tom istom periodu možemo izračunati kao:

$$R_p = \sum R_i w_i = R \cdot w.$$

Izračunajte skalirane svojstvene portfelje \tilde{w} koji proizlaze iz prve dvije glavne komponente dobivene iz matrice kovarijance Σ . Za ta dva svojstvena portfelja izračunajte povijesne povrate kroz razmatrani period. Grafički prikažite vremensko kretanje njihovih vrijednosti (njihove povrate "vratite" natrag u cijene, s tim da početna cijena bude jednak za oba portfelja, npr. 100). Također izračunajte anualizirane srednje vrijednosti i volatiliteti svojstvenih portfelja.

#Vaš kod ide ovdje

```
df1 = np.array(df)
eigvector1 = eigvectors[0].reshape(8,1)
eigvector2 = eigvectors[1].reshape(8,1)
first = df1.dot(eigvector1).sum()
print('Skalirani svojstveni portfelj iz prve komponente: ')
print(first)
df2 = np.array(df)
second = df2.dot(eigvector2).sum()
print('Skalirani svojstveni portfelj iz druge komponente: ')
print(second)
```

```
Skalirani svojstveni portfelj iz prve komponente:
0.1306997783391659
Skalirani svojstveni portfelj iz druge komponente:
-0.07391123355475324
```

Ako usporedite dobivene rezultate s kretanjem cijena originalnih vrijednosnica, vidjet ćete sličnosti između vrijednosnica koje pripadaju određenim klasama imovina i pojedinih svojstvenih portfelja. Svojstveni portfelji dakle predstavljaju niže-dimenzionalan prostor tzv. sintetičkih vrijednosnica (u našem slučaju 2 umjesto originalnih 8) koje najbolje opisuju cijeli razmatrani skup podataka. Dobra procjena tih komponenti je ključna u razumijevanju zajedničkog kretanja većih skupova dionica i upravljanju financijskim rizikom.

✓ 0s completed at 7:46 PM



Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.