

# 4. Laboratorijska vježba: Multivarijantni financijski vremenski nizovi

*Prosinac 2024.*

## 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 `prices.csv`.

Vaš zadatak je u bilježnicu na odgovarajuća mjesta dopisati kod Vašeg rješenja,.

**Riješenu bilježnicu potrebno je predati kao izvještaj u .pdf formatu na Moodle najkasnije do 17.1.2024. u 23:59h. Datoteka koju predajete se mora zvati *PrezimelmeJMBAG.pdf*.**

## Uvod

U laboratorijskoj vježbi razmatra se dinamika cijena vrijednosnica na financijskim tržištima. Dane su povijesne dnevne cijene 24 ETF-a (eng. *exchange traded fund*) koji prate određene dioničke, obvezničke ili druge indekse.

Oznaka	Naziv	Klasa imovine
SPY	SPDR S&P 500 ETF Trust	Equity: U.S. - Large Cap
IEFA	iShares Core MSCI EAFE ETF	Equity: Developed Markets Ex-U.S. - Total Market
VWO	Vanguard FTSE Emerging Markets ETF	Equity: Emerging Markets - Total Market
EWJ	iShares MSCI Japan ETF	Equity: Japan - Total Market
XLF	Financial Select Sector SPDR Fund	Equity: U.S. Financials
XLK	Technology Select Sector SPDR Fund	Equity: U.S. Technology
XLV	Health Care Select Sector SPDR Fund	Equity: U.S. Health Care
XLY	Consumer Discretionary Select Sector SPDR Fund	Equity: U.S. Consumer Cyclicals
XLP	Consumer Staples Select Sector SPDR Fund	Equity: U.S. Consumer Non- cyclicals
XLU	Utilities Select Sector SPDR Fund	Equity: U.S. Utilities
XLI	Industrial Select Sector SPDR	Equity: U.S. Industrials

Oznaka	Naziv	Klasa imovine
	Fund	
XLE	Energy Select Sector SPDR Fund	Equity: U.S. Energy
XLC	Communication Services Select Sector SPDR Fund	Equity: U.S. Telecommunications
XLRE	Real Estate Select Sector SPDR Fund	Equity: U.S. Real Estate
XLB	Materials Select Sector SPDR Fund	Equity: U.S. Basic Materials
BND	Vanguard Total Bond Market ETF	Fixed Income: U.S. - Broad Market
LQD	iShares iBoxx USD Investment Grade Corporate Bond ETF	Fixed Income: U.S. - Corporate
BNDX	Vanguard Total International Bond ETF	Fixed Income: Global Ex-U.S. - Broad Market
SHV	iShares Short Treasury Bond ETF	Fixed Income: U.S. - Government
HYG	iShares iBoxx USD High Yield Corporate Bond ETF	Fixed Income: U.S. - Corporate
GLD	SPDR Gold Trust	Commodities: Precious Metals Gold
SLV	iShares Silver Trust	Commodities: Precious Metals Silver
PDBC	Invesco Optimum Yield Diversified Commodity Strategy No K-1 ETF	Commodities: Broad Market
USO	United States Oil Fund LP	Commodities: Energy Crude Oil

Pri modeliranju zajedničkog kretanja i rizika vrijednosnica, koristit ćemo aritmetičke povrate:

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

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

prices = pd.read_csv('prices.csv')
prices.set_index('Time', inplace=True)
```

```
prices.index = pd.to_datetime(prices.index) # bez ovog bi vrijendoit
bili obicni stringovi ovkao pretvarmao bas u datetime object
```

```
prices.head()
```

	SPY	IEFA	VWO	EWJ	XLF
XLK \ Time					
2019-01-02 58.594608	228.403641	45.796803	32.461216	45.781605	21.425070
2019-01-03 55.637482	222.953278	45.495949	31.952288	45.521481	20.943409
2019-01-04 58.103325	230.421265	46.874874	32.961674	47.064281	21.639147
2019-01-07 58.622952	232.238052	47.100510	33.029510	47.180885	21.665901
2019-01-08 59.114239	234.420044	47.434795	33.216133	47.405128	21.683739
	XLV	XLV	XLP	XLU ...	XLB
\ Time					
2019-01-02	77.260376	94.483681	43.373131	43.356884 ...	45.167751
2019-01-03	75.691597	92.437935	43.123962	43.348557 ...	43.886906
2019-01-04	77.949570	95.497078	44.043327	43.990444 ...	45.612503
2019-01-07	78.248802	97.656494	43.983185	43.690346 ...	45.772594
2019-01-08	78.856377	98.736206	44.387012	44.232204 ...	46.252922
	BND	LQD	BNDX	SHV	HYG
GLD \ Time					
2019-01-02 121.330002	67.512207	92.559242	46.240116	95.842651	58.963661
2019-01-03 122.430000	67.775871	92.600182	46.240116	95.877365	58.992794
2019-01-04 121.440002	67.571701	92.542915	46.197544	95.903488	59.982925
2019-01-07 121.860001	67.469627	92.551117	46.086876	95.903488	60.565334
2019-01-08 121.529999	67.384583	92.755562	46.052830	95.894760	60.958515
	SLV	PDBC	USO		

```
Time
2019-01-02  14.56  8.201962  78.800003
2019-01-03  14.75  8.261633  79.599998
2019-01-04  14.73  8.380974  81.440002
2019-01-07  14.67  8.446068  82.320000
2019-01-08  14.69  8.516588  84.000000
```

```
[5 rows x 24 columns]
```

## 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 povrat pomnože s 252 (cca. broj trgovinskih dana u godini), a volatilnost s  $\sqrt{252}$ .

**Izračunajte anualizirane srednje povrate i volatilnosti. Sve ETF-ove prikažite u dijagramu raspršenja s volatilnošću na x-osi i srednjim povratom na y-osi.**

Razmislite - koji se ETF-ovi ističu po odnosu povrata i rizika (posebno dobri ili posebno loši kao investicije)?

```
from matplotlib import pyplot as plt

daily_returns = prices.pct_change().dropna()# aritmeticki povrat

mean_returns = daily_returns.mean()
#print(mean_returns)
std_returns = daily_returns.std()
#print(std_returns)

annualized_mean_returns = mean_returns * 252 # broj radnih dana u
godini
annualized_std_returns = std_returns * (252 ** 0.5)
print(annualized_mean_returns)
print(annualized_std_returns)

# Graf raspršenja
plt.figure(figsize=(10, 6))
plt.scatter(annualized_std_returns, annualized_mean_returns, c='blue',
edgecolors='k')
plt.title('Raspršenje: Annualizirani povrati vs volatilnost')
```

```
plt.xlabel('Volatilnost (annualizirana)')
plt.ylabel('Povrat (annualiziran)')
plt.grid(True)

# Dodavanje oznaka za svaku vrijednosnicu
for i, txt in enumerate(prices.columns):
    plt.annotate(txt, (annualized_std_returns[i],
annualized_mean_returns[i]))

plt.show()
# Posebno dobri ETF-ovi bi bilo oni koji imaju veliki povrat a
relativno malu varijabilnos npr. XLK a los bi bio onaj npr. koji ima
veliku volatilnost i lose povrat npr. USO
```

SPY	0.177849
IEFA	0.088117
VWO	0.072797
EWJ	0.077634
XLF	0.165866
XLK	0.265742
XLV	0.112391
XLY	0.177103
XLP	0.114491
XLU	0.116566
XLI	0.160515
XLE	0.164938
XLC	0.176782
XLRE	0.107505
XLB	0.131425
BND	0.013445
LQD	0.029531
BNDX	0.014820
SHV	0.023082
HYG	0.052390
GLD	0.126956
SLV	0.147122
PDBC	0.098013
USO	0.076122
dtype: float64	
SPY	0.198523
IEFA	0.188451
VWO	0.204551
EWJ	0.179911
XLF	0.249093
XLK	0.265948
XLV	0.176093
XLY	0.244746
XLP	0.160785
XLU	0.218113
XLI	0.221584

```

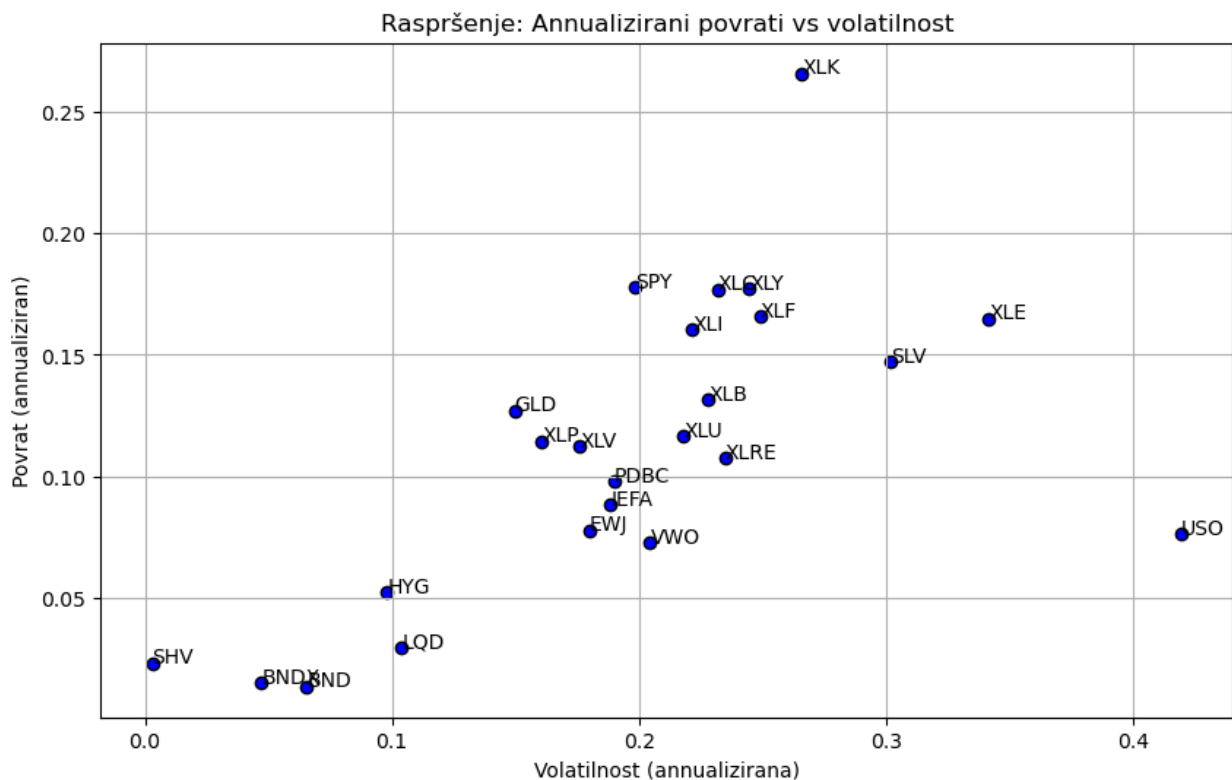
XLE      0.341693
XLC      0.232001
XLRE     0.234709
XLB      0.228101
BND      0.065573
LQD      0.104004
BNDX     0.047116
SHV      0.002958
HYG      0.098116
GLD      0.149683
SLV      0.301798
PDBC     0.190054
USO      0.419480
dtype: float64

```

```

C:\Users\Fran\AppData\Local\Temp\ipykernel_11424\755916217.py:26:
FutureWarning: Series.__getitem__ treating keys as positions is
deprecated. In a future version, integer keys will always be treated
as labels (consistent with DataFrame behavior). To access a value by
position, use `ser.iloc[pos]`
  plt.annotate(txt, (annualized_std_returns[i],
annualized_mean_returns[i]))

```



1.2. Kovarijancu i korelaciju moguće je iz podataka izračunati koristeći Pandas, ali i NumPy ili neke druge biblioteke.

Koristeći dnevne povrate, izračunajte matricu kovarijance  $\Sigma$  i matricu korelacije  $C$  povrata svih ETF-ova. Matrice ispišite u konzolu ili vizualizirajte.

Proučite strukturu matrice i razmislite o tome koje zajedničke komponente u podacima možete očekivati.

```
import seaborn as sns

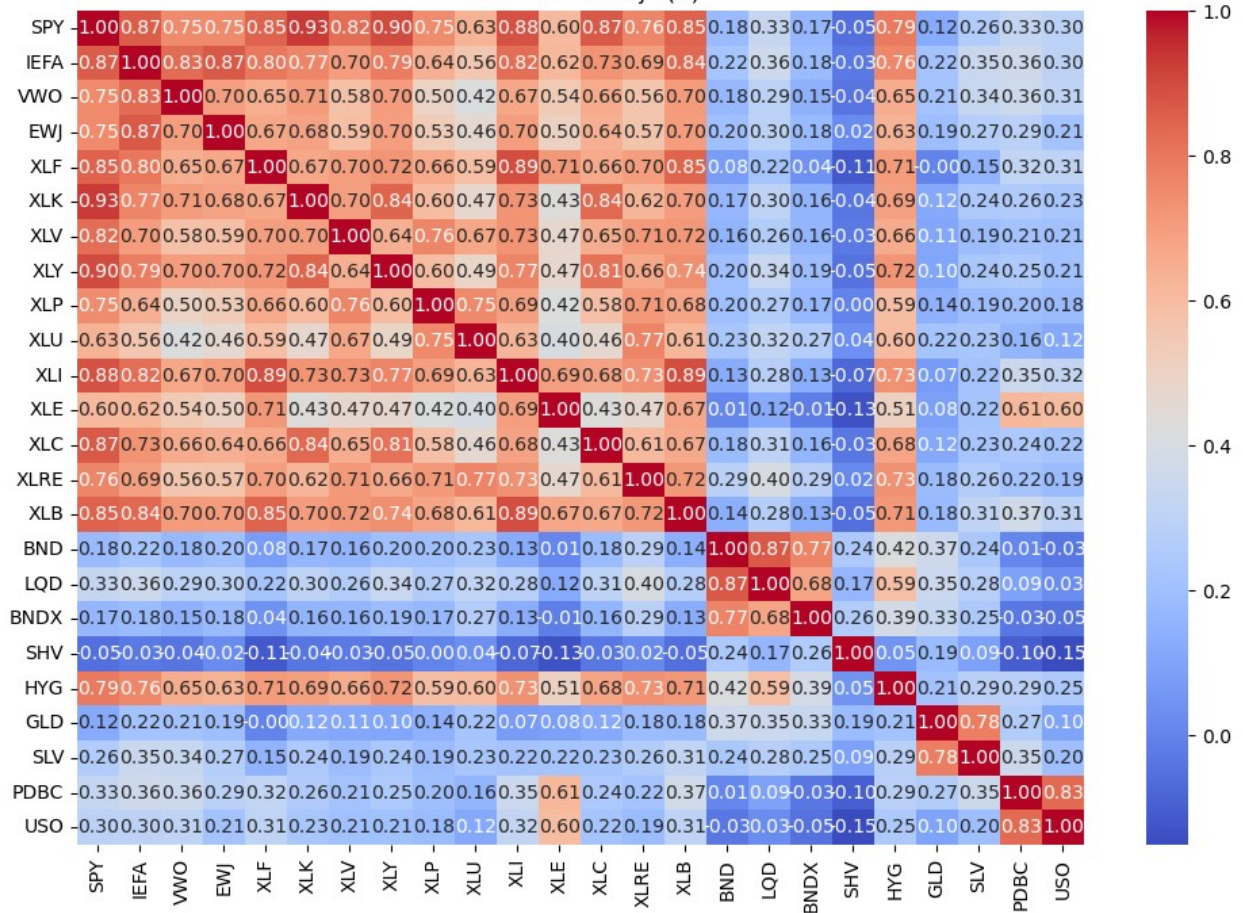
covariance_matrix = daily_returns.cov()
correlation_matrix = daily_returns.corr()

# Vizualizacija matrice korelacije
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f",
            cmap="coolwarm", cbar=True)
plt.title("Matrica korelacije (C)")
plt.show()

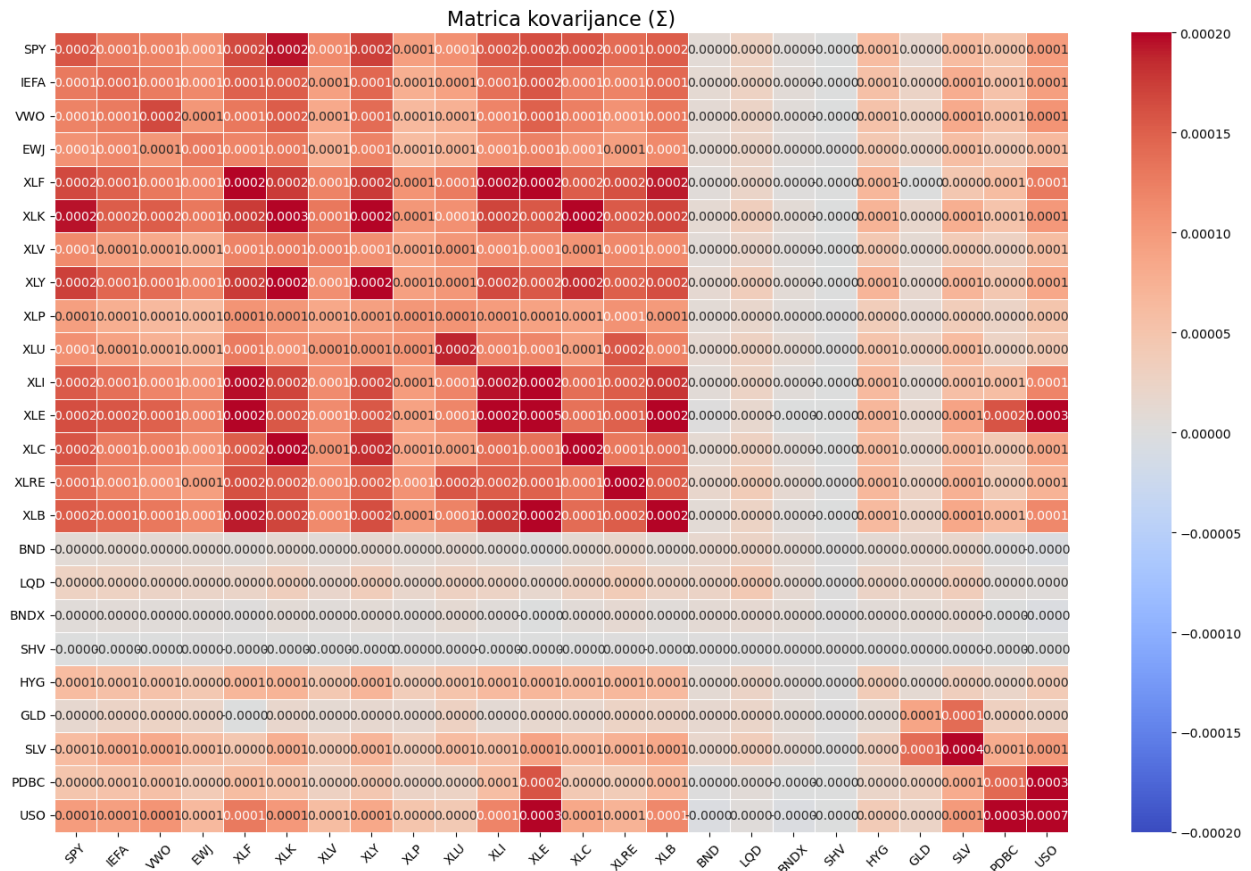
# Vizualizacija matrice kovarijance s poboljšanjima
plt.figure(figsize=(15, 10))
sns.heatmap(covariance_matrix, annot=True, fmt=".4f", cmap="coolwarm",
            cbar=True, linewidths=0.5, vmin=-0.0002, vmax=0.0002)
plt.title("Matrica kovarijance ( $\Sigma$ )", fontsize=16)
plt.xticks(rotation=45, fontsize=10)
plt.yticks(rotation=0, fontsize=10)
plt.tight_layout()
plt.show()

# Na grafovima je lako uočljivo nekoliko mogućih grupacija podataka
# ovinso koji imaju visoku korelaciju. Ove komponente se temelje na
# sektorima, regijama, vrstama imovine i tržišnim uvjetima.
```

Matrica korelacije (C)







## Zadatak 2 - Analiza glavnih komponenti

2.1. Za analizu glavnih komponenti potrebno je izračunati svojstvenu dekompoziciju, koju možete pronaći u sklopu biblioteke NumPy

<https://numpy.org/doc/stable/reference/generated/numpy.linalg.eig.html>.

**Izračunajte svojstvene vektore i pripadajuće svojstvene vrijednosti matrice kovarijance povrata  $\Sigma$ . Poredajte komponente padajući po svojstvenim vrijednostima i prikazite svojstvene vrijednosti grafički (scree plot).**

```
import numpy as np

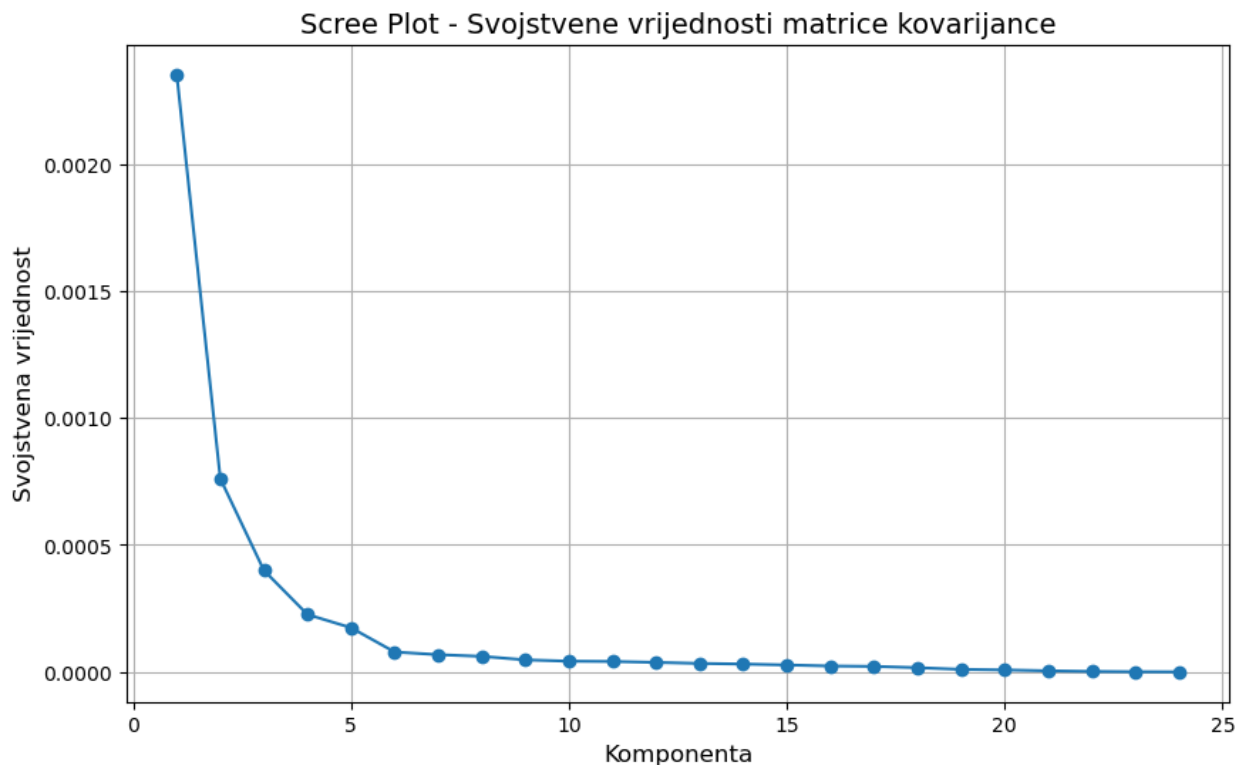
# Izračun svojstvenih vrijednosti i svojstvenih vektora
eigenvalues, eigenvectors = np.linalg.eig(covariance_matrix)

# 2. Sortiranje svojstvenih vrijednosti u opadajućem redoslijedu
sorted_indices = np.argsort(eigenvalues)[::-1] # nadi mi one indekse
# koji bi sortirali elemente uzlazno i onda to obrni da bude silazno
sorted_eigenvalues = eigenvalues[sorted_indices]
sorted_eigenvectors = eigenvectors[:, sorted_indices]

# 3. Prikaz svojstvenih vrijednosti (scree plot)
```

```
plt.figure(figsize=(10, 6))
plt.plot(range(1, len(sorted_eigenvalues) + 1), sorted_eigenvalues,
marker='o', linestyle='-')
plt.title('Scree Plot - Svojstvene vrijednosti matrice kovarijance',
fontSize=14)
plt.xlabel('Komponenta', fontSize=12)
plt.ylabel('Svojstvena vrijednost', fontSize=12)
plt.grid(True)
plt.show()
```

```
# 4. Ispis svojstvenih vrijednosti i vektora
print("Svojstvene vrijednosti (poredane opadajuće):")
print(sorted_eigenvalues)
print("\nSvojstveni vektori (poredani prema svojstvenim
vrijednostima):")
print(sorted_eigenvectors)
```



```
Svojstvene vrijednosti (poredane opadajuće):
[2.35244403e-03 7.60687740e-04 3.99747228e-04 2.26012206e-04
1.73911679e-04 7.83695474e-05 6.79520823e-05 6.12661410e-05
4.74491307e-05 4.22362893e-05 4.10997512e-05 3.74837308e-05
3.34208466e-05 3.10065716e-05 2.73517811e-05 2.33055220e-05
2.17177198e-05 1.66668660e-05 1.00251299e-05 7.79431310e-06
4.23066992e-06 1.82519871e-06 5.71175049e-07 3.02242803e-08]
```

Svojstveni vektori (poredani prema svojstvenim vrijednostima):

```
[[-2.43080116e-01  1.16839662e-01 -4.17742916e-02  1.22366216e-01
-1.87520039e-02 -1.38940956e-01 -2.85677784e-02 -4.92634796e-02
-2.79270211e-02  8.98191727e-02 -4.76147275e-02  3.30514056e-02
-2.78417752e-02 -2.84333252e-02  5.20241771e-02 -1.08614434e-02
-1.90840893e-02 -2.68297559e-03 -1.21408716e-02 -5.16406458e-03
-1.73400413e-02  2.20564059e-02 -9.32561515e-01 -4.90491258e-03]
[[-2.19615263e-01  7.76246830e-02  3.02817003e-02  3.33362431e-02
 1.16802875e-01  3.70149925e-01 -3.93820510e-02 -5.90479463e-02
-8.78545429e-02 -9.31762195e-02  2.82902981e-02  1.12589720e-01
-4.64499396e-02  2.91481108e-02 -2.79572263e-02  4.62388945e-02
 7.53902799e-02 -6.09592620e-02  8.15239815e-01 -2.63086513e-01
 9.41852160e-02  1.08314220e-02 -6.05563503e-04  1.08275656e-03]
[[-2.09782221e-01  5.06032695e-02  5.81351024e-02  1.44515211e-01
 1.92822553e-01  5.13627825e-01  9.42521613e-02 -2.64027192e-01
 5.81354226e-01  1.00300524e-01  2.02345716e-01 -2.97226273e-01
-1.21641671e-02 -8.05284548e-03  8.88928054e-02 -6.80612976e-02
-6.84358094e-03  1.19866288e-01 -2.09282968e-01  2.84691880e-02
-1.85734167e-02  3.28739234e-03 -1.14326128e-02  3.47444952e-04]
[[-1.79053817e-01  8.51476589e-02  2.13463621e-02  7.01849961e-02
 1.50724422e-01  4.72475119e-01 -1.35411344e-02 -1.46022308e-01
-4.42000581e-01 -3.46873717e-01 -2.33683096e-01  3.53976884e-01
-7.97579291e-02  1.25484743e-02 -8.36313153e-02  3.14467813e-02
 9.67369877e-03  2.69127888e-02 -3.95293644e-01  1.26663870e-01
-6.75829808e-02  3.88026729e-03  7.09805565e-03 -2.18190916e-03]
[[-2.83003193e-01  7.98449228e-02 -1.70284995e-01 -1.93936931e-01
 1.27509735e-01 -2.72193134e-02 -3.27865284e-01  2.70176021e-01
-4.75986642e-02 -9.67762959e-02  3.21265466e-01  4.85184167e-02
-2.28836796e-01 -1.95647160e-01  5.29671382e-01 -1.73796600e-01
-1.80266406e-01 -2.65330361e-01 -1.13815508e-01 -7.05308624e-02
 3.28990591e-02  2.43581560e-02  9.12830907e-02  2.03860454e-03]
[[-2.82565240e-01  1.77564748e-01 -1.20689146e-02  4.35103662e-01
 1.29045106e-02 -2.40399933e-01  3.50814806e-02 -2.12557521e-01
 4.26046728e-02  3.43849872e-01 -3.52224736e-01  1.78702388e-01
-3.08827648e-02 -4.53880632e-01  4.02708483e-02 -1.39970565e-01
 4.59098234e-02 -1.11062123e-01  1.35926489e-02 -3.29919438e-02
 1.73636809e-02 -1.07123836e-02  2.72841901e-01  1.56819709e-03]
[[-1.76676091e-01  1.08492629e-01 -3.80060136e-02 -4.03938755e-02
-1.98069868e-01 -6.90250255e-02 -1.65842631e-01 -3.55491911e-01
-3.00499135e-02  3.62096179e-01  1.64978596e-01  2.74418215e-01
 1.26643376e-02  4.71658871e-01  1.22882431e-01  4.35654818e-01
-2.70955049e-01  1.40749105e-02 -5.63745930e-02 -3.90819497e-02
-8.10243236e-03 -8.89139771e-03  1.24663821e-01  8.66092688e-04]
[[-2.63396136e-01  1.70005870e-01 -2.51915868e-02  2.90356770e-01
 7.59083302e-02 -1.11491105e-01  1.01774353e-01  3.75134494e-01
 2.86557233e-02 -1.36340364e-01 -3.49918085e-01 -3.75765631e-01
-2.78408812e-01  4.86829223e-01 -2.65545570e-02  8.25153939e-02
-1.53162967e-01 -8.04656379e-02 -1.36943940e-02 -2.89433857e-02
-8.21642925e-03 -2.70742261e-03  1.06703268e-01  1.68793478e-03]
```

[-1.50493378e-01 9.80830585e-02 -2.05657227e-02 -1.05964447e-01  
-2.78834058e-01 -4.89378542e-02 -1.57671447e-01 -2.59189448e-01  
-1.58455595e-01 6.62236635e-02 6.50693837e-02 -1.38160319e-01  
-1.18468635e-01 3.53327445e-01 -2.07542829e-02 -5.93986747e-01  
4.71997161e-01 5.00249222e-03 2.06835083e-03 1.03910376e-01  
-3.18487210e-03 3.85138768e-02 6.20774985e-02 -1.77343058e-04]  
[-1.83775850e-01 1.39064664e-01 3.65871723e-02 -3.48294004e-01  
-5.10384149e-01 8.33517438e-03 9.87450211e-02 -2.70629084e-01  
-3.84694017e-02 -2.60763747e-01 -1.57025504e-01 -3.90855861e-01  
-1.72398492e-01 -3.38416370e-01 7.10180725e-03 2.78428531e-01  
-7.66375380e-02 -3.53411177e-02 4.82028325e-03 -5.84204239e-02  
-1.76719544e-02 -2.39802077e-02 2.27881499e-02 3.86175636e-04]  
[-2.60391399e-01 7.90086120e-02 -9.77329564e-02 -1.28330869e-01  
5.11716852e-02 -2.29291636e-02 -2.66742796e-01 2.45542442e-01  
-9.73585610e-02 1.45824370e-01 -2.56709690e-02 -7.16555906e-02  
3.84396993e-02 -1.36942602e-01 -1.03761739e-01 3.36258891e-02  
4.73336981e-03 8.30302331e-01 2.17370095e-02 1.83988523e-03  
-1.00200083e-02 -3.87437924e-02 7.19690499e-02 7.77150829e-07]  
[-3.44381592e-01 -2.80619560e-01 -1.76043004e-01 -4.84036700e-01  
4.70509187e-01 -2.59650351e-01 4.13726789e-01 -2.00002181e-01  
2.57609845e-02 3.70408885e-02 -8.66275750e-02 5.51278053e-02  
-7.36997603e-02 7.08486975e-02 -1.06857826e-01 -2.75093591e-02  
2.77512360e-02 -2.23080859e-02 7.00878535e-03 3.04085955e-03  
-3.86227388e-03 2.25618041e-03 2.91790197e-02 4.39957721e-04]  
[-2.37054453e-01 1.47560733e-01 -1.66136851e-02 3.47745431e-01  
1.25325712e-02 -3.28976129e-01 2.19182528e-01 -1.12192252e-01  
-1.02847423e-01 -4.55426124e-01 5.49562918e-01 -2.36947173e-02  
2.76974505e-01 -1.75385695e-02 -1.02714935e-01 5.44503151e-02  
4.10871764e-02 1.01580192e-01 -1.33785383e-02 -2.06300478e-02  
-1.74399168e-04 -8.32647366e-04 8.99805765e-02 3.36073365e-04]  
[-2.35920814e-01 1.53077758e-01 2.28829805e-02 -2.00808118e-01  
-3.96801942e-01 7.15428293e-02 2.56205792e-01 3.45472037e-01  
4.16303440e-01 -1.23252726e-01 -1.14892910e-01 4.73853666e-01  
2.53217369e-01 8.69181506e-02 -2.34942399e-02 -1.76631248e-01  
-1.03956958e-02 4.11315098e-03 -2.83051434e-02 -3.74517650e-02  
-1.51823570e-02 7.95880821e-03 1.46286241e-02 -2.31792351e-04]  
[-2.64801855e-01 7.98633406e-02 -1.29928182e-02 -1.44550919e-01  
8.86218002e-02 8.49789528e-02 -3.41959126e-01 1.58701933e-01  
-4.81875964e-02 1.73862209e-01 2.82487298e-02 -2.14385747e-01  
4.65656018e-01 -5.72362529e-02 -4.84726302e-01 1.13549571e-01  
7.28736023e-02 -4.23575046e-01 -1.11845326e-01 1.64805944e-02  
-1.11143898e-02 1.53065995e-03 1.82550730e-02 -3.46764777e-05]  
[-1.59425478e-02 2.61339752e-02 6.06215269e-02 1.93423849e-03  
-6.18254268e-02 1.09487232e-01 2.55798689e-01 1.35071750e-01  
-1.93941762e-01 2.17035576e-01 1.43343852e-01 -2.91194119e-02  
-6.69756810e-02 -6.54294974e-03 -3.24857579e-03 -5.07021697e-03  
1.09071135e-01 -3.34263649e-02 -1.49769761e-01 -2.51828816e-01  
3.26635668e-01 -7.60799391e-01 -2.74421161e-02 -9.83278800e-03]  
[-4.53173726e-02 4.59069189e-02 9.01607802e-02 1.00120966e-02]

-8.34615849e-02	1.58036848e-01	4.04338246e-01	2.54333601e-01
-2.87427970e-01	3.66341123e-01	2.21004355e-01	-6.49556163e-02
-7.88159979e-02	-6.72929138e-02	7.82478004e-02	1.14120897e-01
2.04967929e-01	-3.38489029e-02	-4.96868893e-02	-1.45509463e-01
-5.09245446e-01	3.18105229e-01	1.92134782e-02	5.77795771e-03]
[-1.01901719e-02	2.04956067e-02	4.40032692e-02	-3.03471772e-03
-4.92857962e-02	5.73362138e-02	1.51704497e-01	7.69309823e-02
-1.01605651e-01	1.18972373e-01	6.24017049e-02	-1.64948834e-02
-5.37488547e-02	-2.30981514e-02	-4.97365177e-02	4.99259047e-02
4.14446424e-02	2.70903509e-02	-1.38786701e-01	-7.44641144e-02
7.73753192e-01	5.47545191e-01	-3.82229116e-03	-8.71297027e-03]
[2.75215077e-04	8.93226785e-04	1.63333424e-03	-3.10445270e-04
-9.17449890e-04	1.80938440e-03	2.89036288e-03	1.25260350e-04
-3.32046839e-03	1.45457026e-03	3.62662939e-04	5.18518293e-04
9.74692123e-04	-9.93487560e-05	-1.29335983e-03	-1.17281517e-03
-7.87191504e-04	8.78981407e-04	-2.58144122e-03	2.61554005e-03
1.30029838e-02	-4.91643068e-03	-5.84942543e-03	9.99861973e-01]
[-1.01200344e-01	4.79862657e-02	1.54067179e-02	3.33716320e-03
-4.49773014e-02	7.10167074e-02	1.54077763e-01	1.29920614e-01
-6.16940781e-02	1.18199736e-01	1.03595199e-01	-7.49687840e-03
-3.75032851e-02	-5.92661650e-02	8.90143531e-02	1.29202479e-01
2.25241426e-02	-5.90401683e-02	2.28388469e-01	8.92932117e-01
1.15975696e-01	-1.19704408e-01	1.17705744e-02	-4.48827535e-03]
[-4.17570375e-02	-1.85212059e-02	3.93841134e-01	-5.47198932e-02
-2.38786881e-02	3.91439319e-02	1.25772677e-01	-7.14101149e-02
-2.39700438e-01	1.09442745e-01	3.21230285e-02	-1.21325872e-01
1.58642924e-01	6.82444277e-03	-7.55136403e-02	-4.54157747e-01
-6.96352470e-01	3.92295319e-02	7.74125278e-02	2.57079099e-02
-2.72198845e-02	2.40448273e-02	2.97147180e-04	-2.74418828e-03]
[-1.44339939e-01	-6.79012133e-02	8.59560718e-01	-8.17324333e-02
1.28046162e-01	-1.89508381e-01	-1.76435844e-01	4.92410082e-02
1.17359976e-01	-6.83736327e-02	2.59717748e-02	1.08766978e-01
-1.75081039e-01	3.90710322e-04	2.56164099e-02	1.51351565e-01
2.35797631e-01	5.65394254e-03	-4.62033340e-02	5.51187237e-03
-2.28306077e-03	-1.38809007e-02	1.10232115e-03	4.54785971e-04]
[-1.33109982e-01	-3.05228786e-01	7.49147590e-02	4.81155549e-02
-2.22847973e-02	4.27040954e-02	3.10061604e-02	-2.21596244e-02
-1.53033753e-01	-3.92096229e-02	-2.73437510e-01	-1.90060344e-01
5.83984622e-01	9.67425569e-02	5.95531216e-01	7.05192676e-02
1.55554848e-01	4.61324637e-02	4.68820396e-03	-3.14563841e-02
6.41405513e-02	-1.85860461e-03	9.33014934e-03	-5.55131971e-04]
[-2.79892155e-01	-7.89156955e-01	-7.38919011e-02	2.60157484e-01
-3.22930035e-01	7.32829490e-02	-1.04078684e-01	7.87336881e-02
2.00689039e-02	-9.38281105e-03	1.04323827e-01	5.13349556e-02
-1.98439489e-01	-4.54895563e-02	-1.93289023e-01	-7.65877922e-03
-7.02846642e-02	-2.58693861e-02	-1.25809208e-02	1.60056044e-03
-2.02938101e-02	2.22687549e-03	-1.67174273e-04	9.95968355e-04]]

2.2. Izračunajte koliki udio ukupne varijance objašnjavaju prve tri komponente (kumulativno)?

```
total_variance = np.sum(sorted_eigenvalues)
explained_variance_ratio = sorted_eigenvalues[:3] / total_variance
cumulative_variance_ratio = np.sum(explained_variance_ratio)
```

```
print(f"Udio ukupne varijance objašnjen prvom komponentom:
{explained_variance_ratio[0]:.2%}")
print(f"Udio ukupne varijance objašnjen drugom komponentom:
{explained_variance_ratio[1]:.2%}")
print(f"Udio ukupne varijance objašnjen trećom komponentom:
{explained_variance_ratio[2]:.2%}")
print(f"Kumulativni udio varijance za prve tri komponente:
{cumulative_variance_ratio:.2%}")
```

```
Udio ukupne varijance objašnjen prvom komponentom: 52.67%
Udio ukupne varijance objašnjen drugom komponentom: 17.03%
Udio ukupne varijance objašnjen trećom komponentom: 8.95%
Kumulativni udio varijance za prve tri komponente: 78.65%
```

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.

**Prikažite grafički (npr. stupčastim dijagramom za svaku komponentu posebno) koeficijente prve 3 glavne komponente (elemente prva tri svojstvena vektora).**

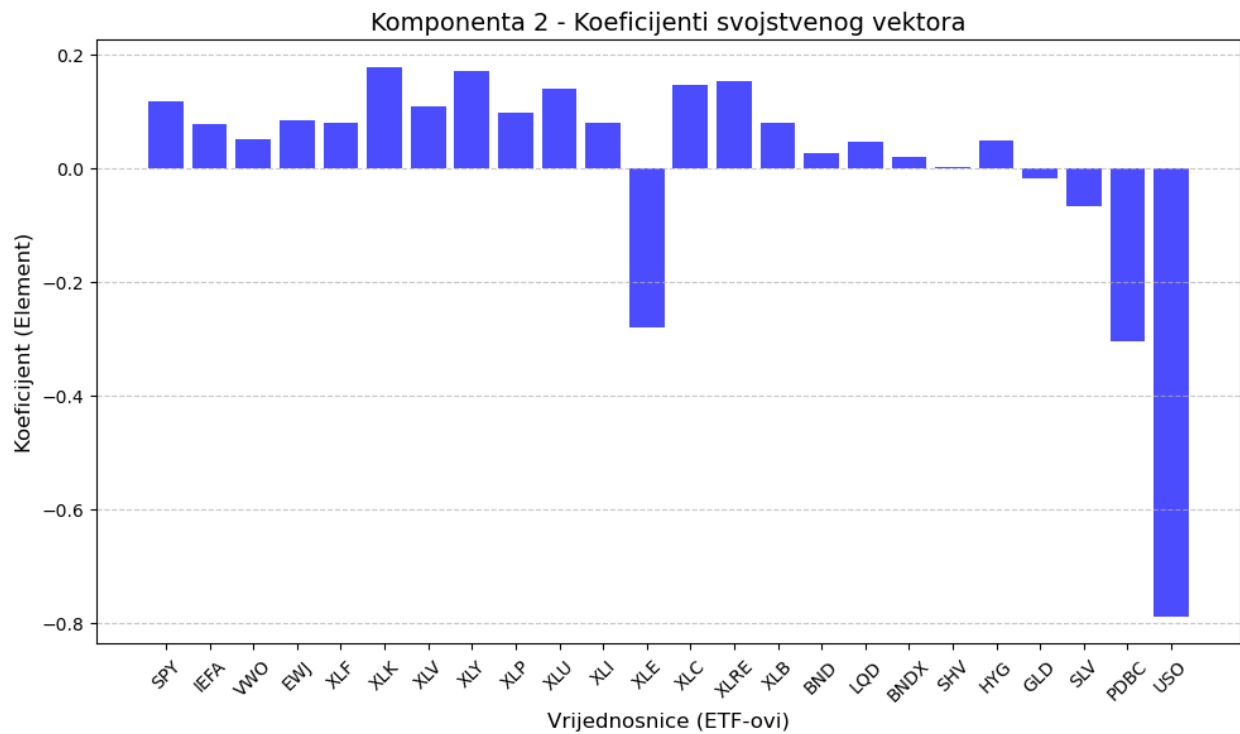
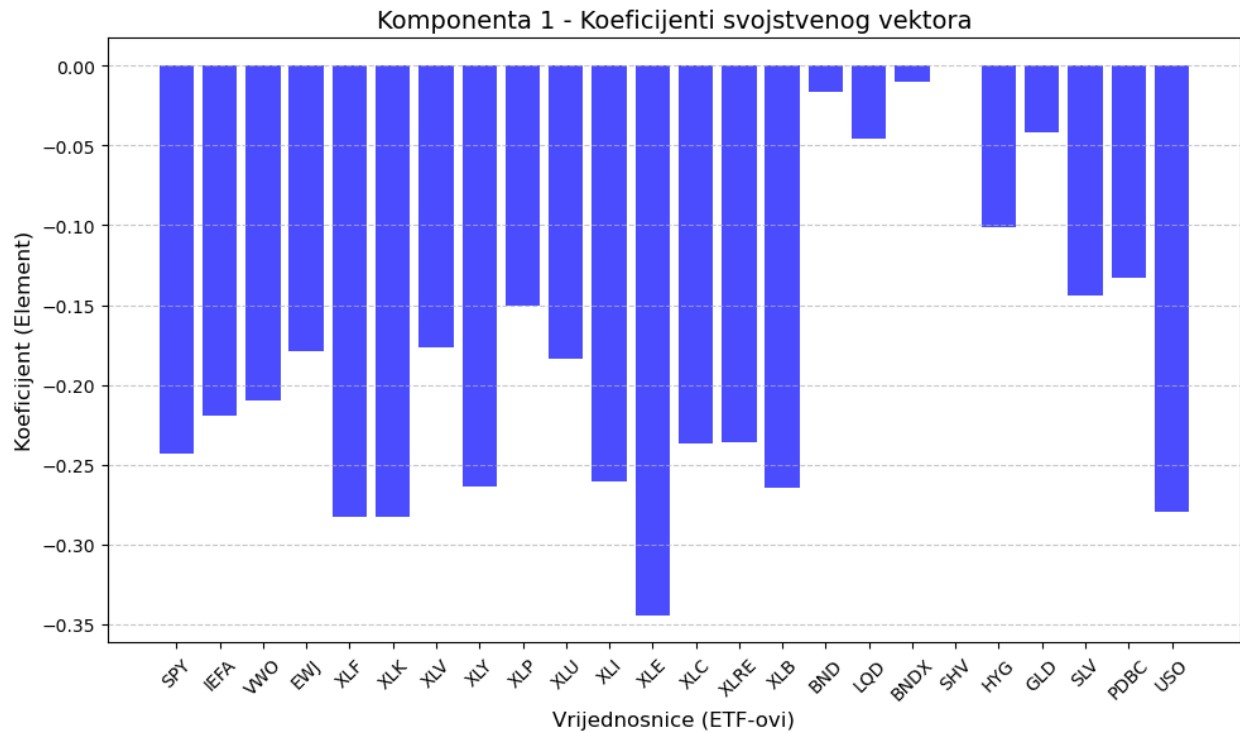
S obzirom na to koje vrijednosnice opisuju prve tri komponente, razmislite možete li zaključiti kakve zajedničke faktore u tržištu opisuju razmatrane komponente?

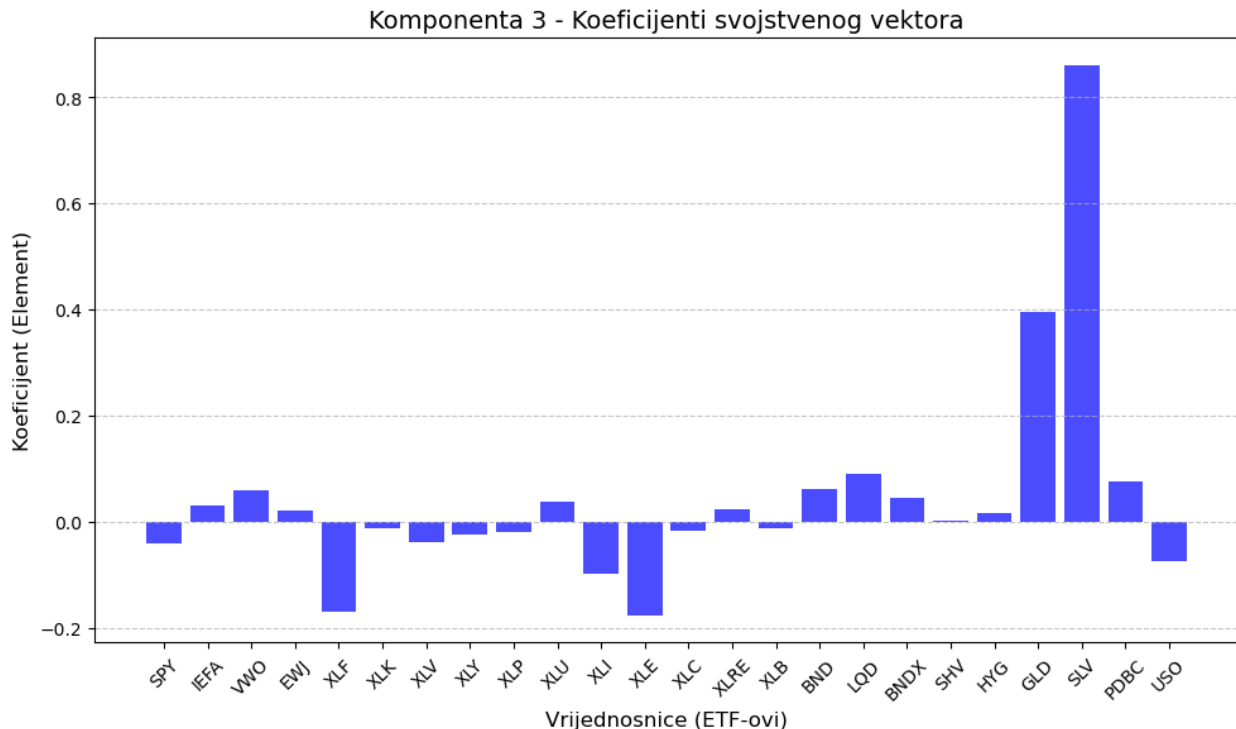
NAPOMENA: pripazite na to što vraća funkcija koju koristite i u kojoj se dimenziji (stupac ili red) nalaze svojstveni vektori.

```
components = sorted_eigenvectors[:, :3] # Prve tri komponente

etf_names = covariance_matrix.columns # Pretpostavka: ovo su nazivi
vrijednosnica

for i in range(3):
    plt.figure(figsize=(10, 6))
    plt.bar(etf_names, components[:, i], color='blue', alpha=0.7)
    plt.title(f"Komponenta {i+1} - Koeficijenti svojstvenog vektora",
    fontsize=14)
    plt.xlabel("Vrijednosnice (ETF-ovi)", fontsize=12)
    plt.ylabel("Koeficijent (Element)", fontsize=12)
    plt.xticks(rotation=45, fontsize=10)
    plt.grid(axis='y', linestyle='--', alpha=0.7)
    plt.tight_layout()
    plt.show()
```





### # Analiza glavnih komponenti (Komponenta 1, 2, 3)

#### # Komponenta 1:

# - Graf pokazuje da većina vrijednosnica ima slične negativne koeficijente, što sugerira da ova komponenta

# objašnjava zajednički tržišni faktor koji utječe na sve vrijednosnice.

# - SPY, IEFA, VWO, i XLF (financijski sektor) imaju značajan doprinos, što sugerira da je ova komponenta

# povezana s globalnim tržišnim trendovima i makroekonomskim uvjetima.

# - S druge strane, USO (nafta) i PDBC (sirovine) također imaju značajne negativne koeficijente, što implicira

# povezanost s energetske i sirovinske tržištima.

#### # Komponenta 2:

# - Ovdje su XLE (energetika) i USO (nafta) najistaknutiji s velikim negativnim koeficijentima, što upućuje na

# to da ova komponenta snažno opisuje dinamiku energetske sektora.

# - Pozitivni koeficijenti kod XLP (potrošački proizvodi) i XLU (uslužne djelatnosti) sugeriraju da ova komponenta

# odražava diverzifikaciju između defenzivnih sektora (poput XLU) i volatilnijih sektora (poput energetike).

#### # Komponenta 3:

# - Graf ukazuje na to da GLD (zlat) i SLV (srebro) imaju dominantan doprinos ovoj komponenti.



```

# - Ovo snažno upućuje na to da ova komponenta opisuje faktore
#   povezane s tržištima plemenitih metala.
# - Negativan doprinos USO (nafta) također ukazuje na razliku između
#   plemenitih metala i energetske tržišta,
#   što sugerira divergentne tržišne uvjete.

# Zaključak:
# - Komponenta 1: Objašnjava široke tržišne trendove, uključujući
#   globalnu tržišnu dinamiku i makroekonomske faktore.
# - Komponenta 2: Fokusirana na sektor energetike i njegovu razliku u
#   odnosu na defenzivne sektore.
# - Komponenta 3: Odražava utjecaje povezane s plemenitim metalima i
#   diverzifikacijom prema sirovinama.

```

**2.4. Ponovite prethodnu analizu za matricu korelacije povrata  $C$  (prikažite svojstvene vrijednosti, udio varijance i koeficijente pojedinih komponenti za prve tri komponente).**

Usporedite rezultate - mijenjaju li se interpretacije komponenti?

```

# 1. Izračun svojstvenih vrijednosti i vektora za matricu korelacije
eigenvalues_corr, eigenvectors_corr =
np.linalg.eig(correlation_matrix)

# 2. Sortiranje svojstvenih vrijednosti i vektora
sorted_indices_corr = np.argsort(eigenvalues_corr)[::-1]
sorted_eigenvalues_corr = eigenvalues_corr[sorted_indices_corr]
sorted_eigenvectors_corr = eigenvectors_corr[:, sorted_indices_corr]

# 3. Izračun udjela varijance za prve tri komponente
total_variance_corr = np.sum(sorted_eigenvalues_corr)
explained_variance_ratio_corr = sorted_eigenvalues_corr[:3] /
total_variance_corr
cumulative_variance_ratio_corr = np.sum(explained_variance_ratio_corr)

# 4. Ispis rezultata
print("Svojstvene vrijednosti matrice korelacije (poredane
opadajuće):")
print(sorted_eigenvalues_corr)
print("\nUdio ukupne varijance objašnjen prve tri komponente
(kumulativno):")
print(f"Komponenta 1: {explained_variance_ratio_corr[0]:.2%}")
print(f"Komponenta 2: {explained_variance_ratio_corr[1]:.2%}")
print(f"Komponenta 3: {explained_variance_ratio_corr[2]:.2%}")
print(f"Kumulativni udio varijance:
{cumulative_variance_ratio_corr:.2%}")

# 5. Grafički prikaz koeficijenata svojstvenih vektora za prve tri
komponente
for i in range(3):
    plt.figure(figsize=(10, 6))

```

```

plt.bar(correlation_matrix.columns, sorted_eigenvectors_corr[:,
i], color='blue', alpha=0.7)
plt.title(f"Komponenta {i+1} - Koeficijenti svojstvenog vektora
(Matrica korelacije)", fontsize=14)
plt.xlabel("Vrijednosnice (ETF-ovi)", fontsize=12)
plt.ylabel("Koeficijent (Element)", fontsize=12)
plt.xticks(rotation=45, fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()

```

*# Interpretacije se mijenjaju između matrice kovarijance i korelacije:*  
*# - Kod kovarijance, komponente naglašavaju sektore s velikom*  
*apsolutnom volatilnošću (npr. XLE u prvoj komponenti).*  
*# - Kod korelacije, doprinos ETF-ova je uravnoteženiji, naglašavajući*  
*relativne odnose među ETF-ovima.*  
*# - Prve tri komponente kod kovarijance više objašnjavaju varijancu*  
*zbog apsolutne volatilnosti sektora.*  
*# - Kod korelacije, prve tri komponente ravnomjernije obuhvaćaju*  
*odnose među ETF-ovima bez dominacije volatilnih sektora.*

Svojstvene vrijednosti matrice korelacije (poredane opadajuće):

```

[1.19715098e+01 2.95530631e+00 2.06896137e+00 1.19524050e+00
 1.08076927e+00 8.65008701e-01 6.32826793e-01 4.24697344e-01
 3.42210938e-01 3.30235236e-01 3.01059505e-01 2.48249402e-01
 2.33104430e-01 1.99064293e-01 1.85967337e-01 1.70907389e-01
 1.61778121e-01 1.44099892e-01 1.33028132e-01 1.16396653e-01
 9.25834429e-02 7.65417069e-02 6.70455915e-02 3.40787259e-03]

```

Udio ukupne varijance objašnjen prve tri komponente (kumulativno):

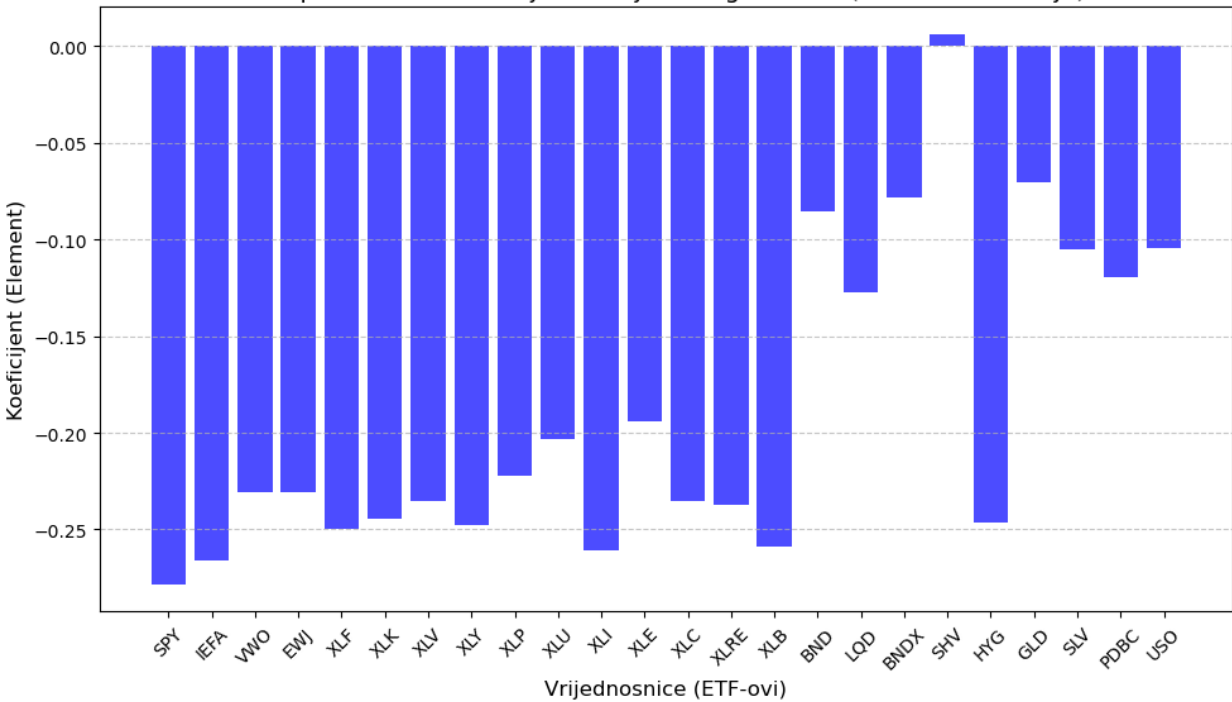
Komponenta 1: 49.88%

Komponenta 2: 12.31%

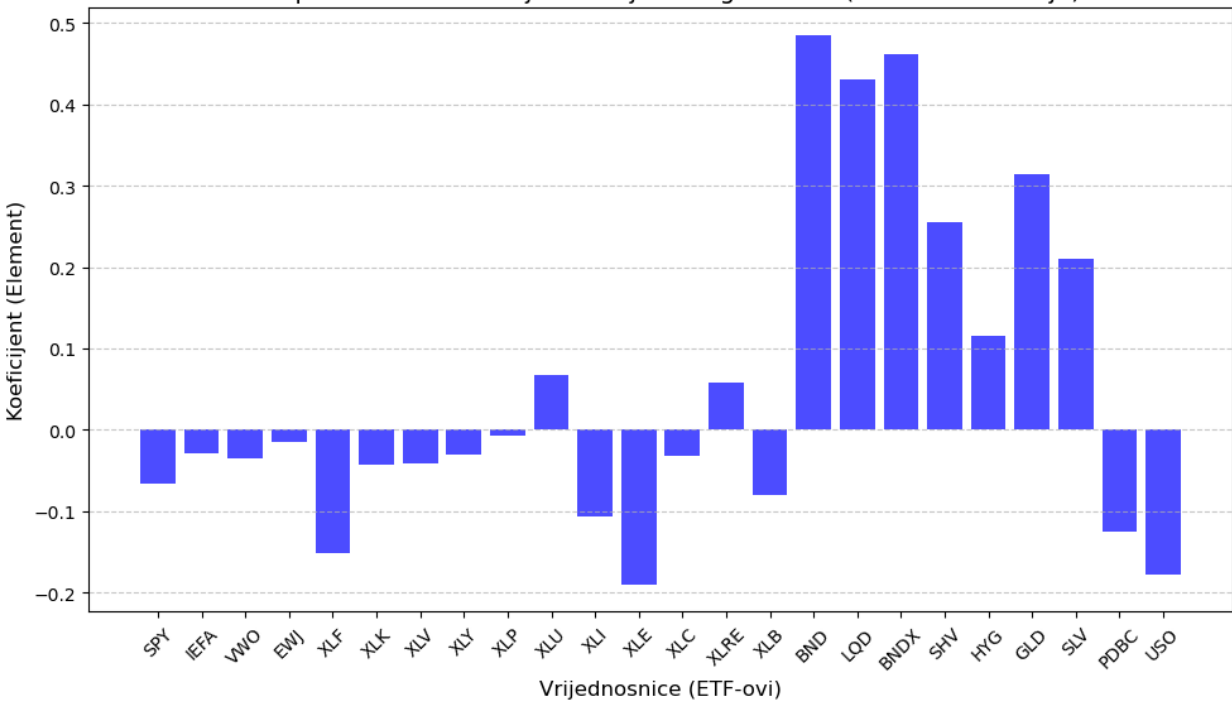
Komponenta 3: 8.62%

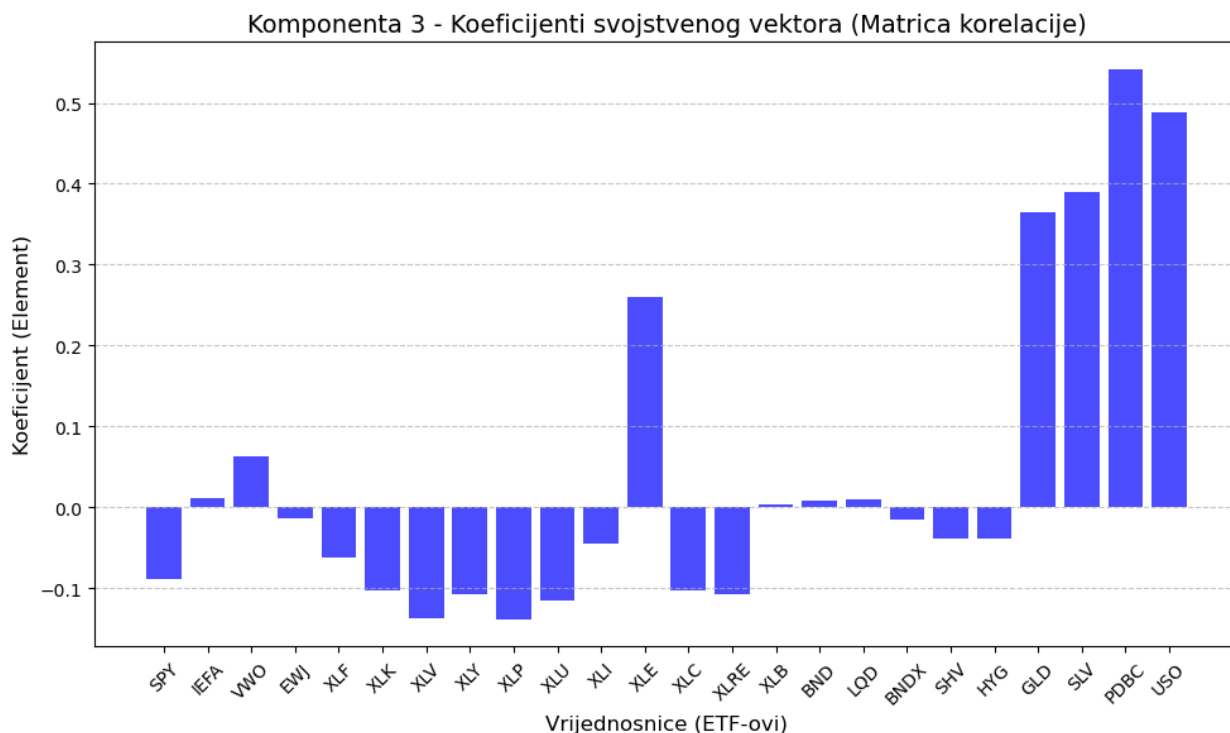
Kumulativni udio varijance: 70.82%

Komponenta 1 - Koeficijenti svojstvenog vektora (Matrica korelacije)



Komponenta 2 - Koeficijenti svojstvenog vektora (Matrica korelacije)





## 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. Same težine svojstvenih portfelja mogu biti rotirane i skalirane u odnosu na elemente svojstvenih vektora.

U ovoj analizi ćemo 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, potrebno je i skalirati svojstvene portfelje sa sumom njihovih apsolutnih vrijednosti:

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

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

Ukoliko znamo povrate  $R \in R^{T \times N}$  (gdje je  $R_i \in R^T$  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 adekvatno skalirane i rotirane svojstvene portfelje  $\tilde{w}$  koji proizlaze iz prve tri glavne komponente dobivene iz matrice kovarijance  $\Sigma$ . Za ta tri svojstvena portfelja izračunajte povijesne povrate kroz razmatrani period. Grafički prikazite vremensko kretanje njihovih vrijednosti (njihove povrate "vratite" natrag u cijene, s tim da početna cijena bude jednak za sve portfelje, npr. 100).

```
# Uzmi svojstvene vektore za prve tri glavne komponente
w1 = sorted_eigenvectors[:, 0]
w2 = sorted_eigenvectors[:, 1]
w3 = sorted_eigenvectors[:, 2]

# Funkcija za skaliranje i rotaciju težina
def scale_and_rotate_weights(w):
    w_rotated = w * np.sign(np.sum(w)) # Rotiraj težine tako da suma
    bude pozitivna
    w_scaled = w_rotated / np.sum(np.abs(w_rotated)) # Skaliraj
    težine
    return w_scaled

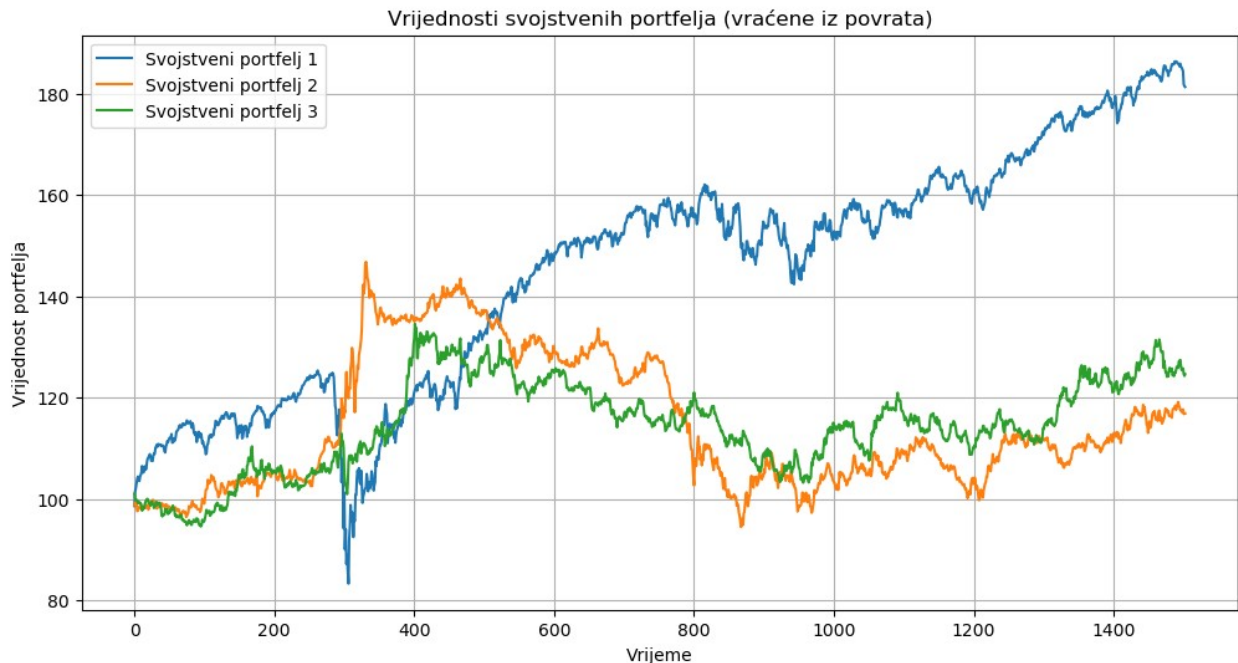
# Skalirane težine za prve tri glavne komponente
w1_scaled = scale_and_rotate_weights(w1)
w2_scaled = scale_and_rotate_weights(w2)
w3_scaled = scale_and_rotate_weights(w3)

# Dnevni povrati (pretpostavljamo da su izračunati)
daily_returns = prices.pct_change().dropna()

# Povrati svojstvenih portfelja - radimo matrično množenje
portfolio1_returns = daily_returns.values @ w1_scaled
portfolio2_returns = daily_returns.values @ w2_scaled
portfolio3_returns = daily_returns.values @ w3_scaled

# "Vraćanje" povrata u cijene (normaliziramo početnu cijenu na 100)
portfolio1_prices = 100 * (1 + np.cumsum(portfolio1_returns))
portfolio2_prices = 100 * (1 + np.cumsum(portfolio2_returns))
portfolio3_prices = 100 * (1 + np.cumsum(portfolio3_returns))

# Grafički prikaz svojstvenih portfelja
plt.figure(figsize=(12, 6))
plt.plot(portfolio1_prices, label="Svojstveni portfelj 1")
plt.plot(portfolio2_prices, label="Svojstveni portfelj 2")
plt.plot(portfolio3_prices, label="Svojstveni portfelj 3")
plt.title("Vrijednosti svojstvenih portfelja (vraćene iz povrata)")
plt.xlabel("Vrijeme")
plt.ylabel("Vrijednost portfelja")
plt.legend()
plt.grid()
plt.show()
```



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 3 umjesto originalnih 24) koje najbolje opisuju cijeli razmatrani skup podataka. Dobra procjena tih komponenti je ključna u razumijevanju zajedničkog kretanja većih skupova dionica i upravljanju financijskim rizikom.

Razmislite što to znači za tržište koje smo analizirali - koji su glavni izvori rizika prevladavali u razmatranom periodu?

**Izračunajte korelacijske koeficijente između povrata tri razmatrana svojstvena portfelja (realizacije prve tri glavne komponente) i svih pojedinih ETF-ova - za svaki od tri svojstvena portfelja pronađite ETF koji mu je najbliži (najviši iznos korelacije povrata).**

```
portfolio_returns_df = pd.DataFrame({
    'Portfolio 1': portfolio1_returns,
    'Portfolio 2': portfolio2_returns,
    'Portfolio 3': portfolio3_returns
})

# 1. Izračun korelacijske matrice između portfelja i ETF-ova
correlation_matrix = np.corrcoef(portfolio_returns_df.values.T,
daily_returns.values.T)[:3, 3:]

# 2. Pronalazak ETF-a s najvećom korelacijom za svaki portfelj
max_correlation_indices = np.argmax(correlation_matrix, axis=1) #
Indeksi ETF-ova s najvećom korelacijom
max_correlation_values = np.max(correlation_matrix, axis=1) #
Vrijednosti najvećih korelacija
```

```
# 3. Dobivanje naziva ETF-ova
```

```
etf_names = daily_returns.columns.tolist()
```

```
# 4. Ispis rezultata
```

```
for i in range(3): # Iteracija kroz tri portfelja
```

```
    etf_name = etf_names[max_correlation_indices[i]]
```

```
    correlation = max_correlation_values[i]
```

```
    print(f"Portfolio {i+1}: Najsličniji ETF je {etf_name} s  
korelacijom {correlation:.2f}")
```

```
Portfolio 1: Najsličniji ETF je SPY s korelacijom 0.94
```

```
Portfolio 2: Najsličniji ETF je XLY s korelacijom 0.30
```

```
Portfolio 3: Najsličniji ETF je SLV s korelacijom 0.90
```