

## Obrada informacija

# 4. Laboratorijska vježba: Multivarijlatni financijski vremenski nizovi

Prosinac 2023.

## 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 14.1.2024. 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 Fund	Equity: U.S. Industrials
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
import csv
from google.colab import drive

drive.mount('/content/drive')
prices = pd.read_csv('drive/My Drive/Colab Notebooks/prices.csv')
prices.set_index('Time', inplace=True)
prices.index = pd.to_datetime(prices.index)
csv_reader = csv.reader(prices)
prices.head()
```

```
print(prices)
```

	SPY	IEFA	VWO	EWJ	XLFX	
Time						
2019-01-02	231.492233	48.000053	33.417080	47.497765	21.776472	
2019-01-03	225.968170	47.684727	32.893154	47.227894	21.286907	
2019-01-04	233.537125	49.129986	33.932262	48.828522	21.994057	
2019-01-07	235.378525	49.366474	34.002121	48.949490	22.021255	
2019-01-08	237.589920	49.716839	34.194221	49.182148	22.039383	
...	...	...	...	...	...	
2023-12-06	454.760010	67.720001	40.250000	62.669998	35.770000	
2023-12-07	458.230011	68.050003	40.410000	63.150002	35.939999	
2023-12-08	460.200012	68.250000	40.209999	62.779999	36.130001	
2023-12-11	461.989990	68.349998	40.389999	63.139999	36.349998	
2023-12-12	462.309998	68.279900	40.229900	62.830002	36.485001	

  

	XLK	XLV	XLX	XLPL	XLUL	...	
Time							
2019-01-02	59.029892	78.483353	95.259102	44.591763	44.784126	...	
2019-01-03	56.050797	76.889748	93.196533	44.335598	44.775520	...	
2019-01-04	58.534962	79.183434	96.280815	45.280781	45.438545	...	
2019-01-07	59.058445	79.487411	98.457954	45.218948	45.128563	...	
2019-01-08	59.553371	80.104607	99.546509	45.634129	45.688255	...	
...	...	...	...	...	...	...	
2023-12-06	182.860001	132.009995	170.990005	70.300003	63.669998	...	
2023-12-07	185.080002	131.850006	172.399994	70.599998	63.529999	...	
2023-12-08	186.660004	132.070007	173.190002	70.110001	63.410000	...	
2023-12-11	188.289993	132.940002	173.649994	70.779999	63.820000	...	
2023-12-12	188.419998	133.009995	173.250000	70.915001	63.820000	...	

  

	XLB	BND	LQD	BNDX	SHV	
Time						
2019-01-02	46.068901	69.980576	97.039024	48.517841	101.234894	
2019-01-03	44.762505	70.253906	97.081886	48.517841	101.271584	
2019-01-04	46.522518	70.042305	97.021873	48.473194	101.299133	
2019-01-07	46.685822	69.936508	97.030441	48.357071	101.299133	
2019-01-08	47.175724	69.848312	97.244827	48.321350	101.289948	
...	...	...	...	...	...	
2023-12-06	81.080002	72.190002	107.930000	49.889999	110.190002	
2023-12-07	81.629997	72.169998	107.940002	49.830002	110.239998	
2023-12-08	81.930000	71.820000	107.360001	49.619999	110.239998	
2023-12-11	82.510002	71.830002	107.330002	49.599998	110.239998	
2023-12-12	82.730003	71.940002	107.489998	49.709999	110.264999	

  

	HYG	GLD	SLV	PDBC	USO
Time					
2019-01-02	63.253788	121.330002	14.560000	8.545244	78.800003
2019-01-03	63.285011	122.430000	14.750000	8.607411	79.599998
2019-01-04	64.347198	121.440002	14.730000	8.731748	81.440002
2019-01-07	64.971985	121.860001	14.670000	8.799566	82.320000
2019-01-08	65.393723	121.529999	14.690000	8.873038	84.000000
...	...	...	...	...	...
2023-12-06	75.879997	187.839996	21.930000	13.700000	64.760002
2023-12-07	76.010000	188.110000	21.700000	13.720000	65.110000

```

2023-12-07    76.810002    188.145554    21.750001    13.720000    65.110001
2023-12-08    75.860001    185.639999    21.100000    13.800000    66.529999
2023-12-11    75.809998    183.589996    20.910000    13.750000    66.800003
2023-12-12    75.754997    183.800003    20.910000    13.575000    64.720001

```

```
[1246 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)?

```

import numpy as np
import matplotlib.pyplot as plt

rets = prices.pct_change()
mean_rets = rets.mean()
volatility = rets.std()
annualMeanReturns = mean_rets * 252
annualVolatility = volatility * (252 ** 0.5)

data = pd.DataFrame({
    'Volatility': annualVolatility,
    'Mean Return': annualMeanReturns
})

plt.figure(figsize=(12, 8))
plt.scatter(data['Volatility'], data['Mean Return'], alpha=0.8)
for i, txt in enumerate(data.index):
    plt.annotate(txt, (data['Volatility'][i], data['Mean Return'][i]))
plt.title('Dijagram za volatilnost i srednje povrate')
plt.xlabel('Volatilnost')
plt.ylabel('Srednji povrat')

```

```
plt.grid(True)
plt.show()
```

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.

```

cov_mat = rets.cov()
corr_mat= rets.corr()

print("Matrica kovarijance:")
print(cov_mat)
print("=====")
print("Matrica korelacije:")
print(cov_mat)

```

Matrica kovarijance:

	SPY	IEFA	VWO	EWJ	XLF \
SPY	1.761574e-04	1.457528e-04	1.376812e-04	1.170823e-04	1.918401e-04
IEFA	1.457528e-04	1.563030e-04	1.409613e-04	1.259424e-04	1.728926e-04
VWO	1.376812e-04	1.409613e-04	1.815550e-04	1.112717e-04	1.534720e-04
EWJ	1.170823e-04	1.259424e-04	1.112717e-04	1.299253e-04	1.357587e-04
XLF	1.918401e-04	1.728926e-04	1.534720e-04	1.357587e-04	2.808551e-04
XLK	2.158161e-04	1.701803e-04	1.706754e-04	1.387625e-04	2.036297e-04
XLV	1.318726e-04	1.061152e-04	9.604616e-05	8.357183e-05	1.406268e-04
XLY	1.934884e-04	1.614538e-04	1.574461e-04	1.310982e-04	1.994529e-04
XLP	1.107683e-04	8.944329e-05	7.605146e-05	7.151245e-05	1.225315e-04
XLU	1.274866e-04	1.049761e-04	8.575088e-05	8.290778e-05	1.461856e-04
XLI	1.753387e-04	1.551730e-04	1.373171e-04	1.224394e-04	2.245905e-04
XLE	1.907531e-04	1.835906e-04	1.738178e-04	1.417821e-04	2.810847e-04
XLC	1.796809e-04	1.441639e-04	1.424863e-04	1.176312e-04	1.748918e-04
XLRE	1.628937e-04	1.385278e-04	1.222450e-04	1.077890e-04	1.868533e-04
XLB	1.757895e-04	1.626840e-04	1.479333e-04	1.284975e-04	2.209945e-04
BND	1.032848e-05	1.106674e-05	9.632471e-06	9.707243e-06	5.550302e-06
LQD	3.073254e-05	3.035281e-05	2.706990e-05	2.332045e-05	2.590348e-05
BNDX	6.836683e-06	6.458734e-06	5.747600e-06	5.987623e-06	2.102074e-06
SHV	-1.782225e-07	-1.215006e-07	-1.699283e-07	2.821937e-08	-4.059010e-07
HYG	7.076129e-05	6.390899e-05	5.976081e-05	5.006845e-05	8.011112e-05
GLD	1.307499e-05	2.218935e-05	2.325321e-05	1.778782e-05	-1.998805e-06
SLV	6.410859e-05	7.962121e-05	8.086376e-05	5.718553e-05	5.123144e-05
PDBC	5.730408e-05	5.856160e-05	6.037230e-05	4.461771e-05	7.280871e-05
USO	1.171177e-04	1.116622e-04	1.196180e-04	7.435808e-05	1.564442e-04

	XLK	XLV	XLY	XLP	XLU \
SPY	2.158161e-04	1.318726e-04	1.934884e-04	1.107683e-04	1.274866e-04
IEFA	1.701803e-04	1.061152e-04	1.614538e-04	8.944329e-05	1.049761e-04
VWO	1.706754e-04	9.604616e-05	1.574461e-04	7.605146e-05	8.575088e-05
EWJ	1.387625e-04	8.357183e-05	1.310982e-04	7.151245e-05	8.290778e-05
XLF	2.036297e-04	1.406268e-04	1.994529e-04	1.225315e-04	1.461856e-04
XLK	2.996448e-04	1.510928e-04	2.403027e-04	1.222616e-04	1.316509e-04
XLV	1.510928e-04	1.385254e-04	1.271010e-04	9.811664e-05	1.172870e-04
XLY	2.403027e-04	1.271010e-04	2.591421e-04	1.078307e-04	1.199366e-04
XLP	1.222616e-04	9.811664e-05	1.078307e-04	1.154083e-04	1.193987e-04
XLU	1.316509e-04	1.172870e-04	1.199366e-04	1.193987e-04	2.069732e-04
XLI	1.910086e-04	1.299066e-04	1.859044e-04	1.148908e-04	1.390192e-04
XLE	1.834210e-04	1.329837e-04	1.816440e-04	1.091994e-04	1.356845e-04
XLC	2.298729e-04	1.227429e-04	2.057247e-04	1.007367e-04	1.094027e-04
XLRE	1.805967e-04	1.317512e-04	1.722832e-04	1.208262e-04	1.756801e-04
XLB	1.937529e-04	1.310538e-04	1.861967e-04	1.143999e-04	1.386152e-04
BND	1.330583e-05	7.160052e-06	1.404303e-05	8.633379e-06	1.322661e-05

LQD	3.825942e-05	2.096342e-05	3.828408e-05	1.900692e-05	3.080993e-05
BNDX	8.851570e-06	5.695455e-06	9.282111e-06	5.575668e-06	1.193018e-05
SHV	-1.452970e-07	-9.666214e-08	-1.939843e-07	-4.617104e-08	4.848319e-08
HYG	8.294049e-05	5.245450e-05	7.940941e-05	4.330333e-05	5.912695e-05
GLD	1.719533e-05	1.085658e-05	1.310897e-05	1.405772e-05	2.878039e-05
SLV	7.808044e-05	4.390190e-05	7.235107e-05	4.033467e-05	6.391738e-05
PDBC	5.823096e-05	3.552949e-05	5.444496e-05	2.980149e-05	3.136934e-05
USO	1.189163e-04	7.579423e-05	1.034852e-04	6.289437e-05	5.387372e-05

	...	XLB	BND	LQD	BNDX \
SPY	...	1.757895e-04	1.032848e-05	3.073254e-05	6.836683e-06
IEFA	...	1.626840e-04	1.106674e-05	3.035281e-05	6.458734e-06
VWO	...	1.479333e-04	9.632471e-06	2.706990e-05	5.747600e-06
EWJ	...	1.284975e-04	9.707243e-06	2.332045e-05	5.987623e-06

## ✓ 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.**

```
#Vaš kod ide ovdje
eigenvalues_cov, eigenvectors_cov = np.linalg.eig(cov_mat)
sorted_indices = np.argsort(eigenvalues_cov)[::-1]
sorted_ev = eigenvalues_cov[sorted_indices]
sorted_evec = eigenvectors_cov[:, sorted_indices]

plt.figure(figsize=(12, 8))
plt.bar(range(len(sorted_ev)), sorted_ev)
plt.title('Graf')
plt.xlabel('Indeks komponente')
plt.ylabel('Svojstvena vrijednost')
plt.grid(True)
plt.show()
```

## 2.2. Izračunajte koliki udio varijance objašnjavaju prve tri komponente?

```
#Vaš kod ide ovdje
totalVar = sum(sorted_ev)
var3 = sum(sorted_ev[:3]) / totalVar

print("Udio prve tri komponente:",var3*100,'%')
    Udio prve tri komponente: 79.71142095322995 %
```

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.

```
#Vaš kod ide ovdje
fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(15, 6), sharey=True)
etfs = rets.columns

for i in range(3):
    axes[i].bar(etfs, sorted_evec[:,i])
    axes[i].set_title(f'Glavna komponenta {i+1}')
    axes[i].tick_params(axis='x', rotation=90)
    axes[i].set_ylim(-0.8,0.9)
plt.show()
```

## 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?

```
#Vaš kod ide ovdje
eigenvalues_corr, eigenvectors_corr = np.linalg.eig(corr_mat)
sorted_indices_corr = np.argsort(eigenvalues_corr)[::-1]
sorted_ev_corr = eigenvalues_corr[sorted_indices_corr]
sorted_evec_corr = eigenvectors_corr[:, sorted_indices_corr]

plt.figure(figsize=(10, 6))
plt.bar(range(len(sorted_ev_corr)), sorted_ev_corr)
plt.title('Svojstvene vrijednosti matrice korelacije povrata')
plt.xlabel('Indeks komponente')
plt.ylabel('Svojstvena vrijednost')
plt.grid(True)
plt.show()

total_var_corr = sum(sorted_ev_corr)
var3corr = sum(sorted_ev_corr[:3]) / total_var_corr

fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(14, 6), sharey=True)
for i in range(3):
    axes[i].bar(etfs, sorted_evec_corr[:, i])
    axes[i].set_title(f'Glavna komponenta {i+1} (Korelacija)')
    axes[i].tick_params(axis='x', rotation=90)
plt.show()

print("Udio prve tri komponente: ", var3corr*100, '%')
```

### ✓ 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, 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 volatilnostima 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 tri glavne komponente dobivene iz matrice kovarijance  $\Sigma$ . Za ta tri 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).**

```
#Vaš kod ide ovdje
pf = np.copy(sorted_evec[:, :3])
for i in range(3):
    if np.sum(pf[:, i]) < 0:
        pf[:, i] = -pf[:, i]
    pf[:, i] /= np.sum(np.abs(pf[:, i]))
pfRets = np.dot(rets.fillna(0), pf)
initial_price = 100
portfolio_prices = np.cumprod(1 + pfRets, axis=0) * initial_price

plt.figure(figsize=(14, 8))
for i in range(3):
    plt.plot(rets.index, portfolio_prices[:, i], label=f'Portfelj {i+1}')
plt.title('Promjena cijena portfelja')
plt.xlabel('Datum')
plt.ylabel('Cijena')
plt.legend()
plt.grid(True)
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?

