

Algoritmo PCA : Covariância, Autovalores e Autovetores

Alunos: Anabelle Elizabeth Araujo de Souza e Jules Severo Barcos

Prof. Me. Alexandre Garcia

Santos, 06 de Dezembro de 2022

1 Introdução

Neste trabalho iremos abordar sobre a Análise de Componentes Principais (em inglês PCA), técnica usada nos princípios de álgebra linear para transformar variáveis, possivelmente correlacionadas, em um número menor de variáveis chamadas de Componentes Principais. Para isso será selecionado um DataSet, e será realizado o tratamento e normalização de seus dados por meio da linguagem Python e por fim o Plot das informações em formato gráfico.

2 Desenvolvimento

- Definir Dataset escolhido pela plataforma Kaggle.
(Link: <https://www.kaggle.com/datasets/mukhazarahmad/22-years-of-us-treasury-bonds-data>);

```

1  Date,High,Low,Open,Close,Adj Close
2  1/2/2001,5.71999979,5.650000095,5.710000038,5.679999828,5.67999998
3  1/3/2001,5.760000229,5.53000021,5.679999828,5.53000021,5.53000021
4  1/4/2001,5.369999886,5.210000038,5.369999886,5.239999771,5.239999
5  1/5/2001,5.190000057,4.949999809,5.050000191,4.96999979,4.9699997
6  1/8/2001,5.059999943,4.989999771,5.019999981,5.050000191,5.050000
7  1/9/2001,5.130000114,5.059999943,5.059999943,5.090000153,5.090000
8  1/10/2001,5.159999847,5.090000153,5.090000153,5.139999866,5.13999
9  1/11/2001,5.159999847,5.159999847,5.159999847,5.159999847,5.15999
10 1/12/2001,5.130000114,5.130000114,5.130000114,5.130000114,5.13000
11 1/16/2001,5.21999979,5.159999847,5.170000076,5.210000038,5.210000
12 1/17/2001,5.260000229,5.199999809,5.230000019,5.21999979,5.219999

```

Figura 1: Prévia dos Dados Brutos.

- Utilizando-se do Jupyter Notebook por meio do Visual Studio Code foram feitos os Imports das bibliotecas usadas;

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

[18] ✓ 2.7s

```

Figura 2: Imports das Bibliotecas.

- Após isso, será chamado o arquivo em .CSV:

```
Data= pd.read_csv('economia/13-Week-Treasury-Bill.csv')
```

Figura 3: Chamar Arquivo .CSV.

- Os dados brutos estão com a coluna de "Data", que será removida com o comando abaixo ficando dessa forma:

```
16] Data= pd.read_csv('economia/13-Week-Treasury-Bill.csv') ✓ 0.1s
```

```
17] Data = Data.drop(columns=['Date']) ✓ 0.8s
```

```
18] Data.head() ✓ 0.1s
```

| | High | Low | Open | Close | Adj Close |
|---|------|------|------|-------|-----------|
| 0 | 5.72 | 5.65 | 5.71 | 5.68 | 5.68 |
| 1 | 5.76 | 5.53 | 5.68 | 5.53 | 5.53 |
| 2 | 5.37 | 5.21 | 5.37 | 5.24 | 5.24 |
| 3 | 5.19 | 4.95 | 5.05 | 4.97 | 4.97 |
| 4 | 5.06 | 4.99 | 5.02 | 5.05 | 5.05 |

Figura 4: Drop Coluna Data.

- Nesse passo será feita a covariância:

```
economia=np.cov(Data)
```

✓ 0.7s

Figura 5: Covariância.

```
print(economia)
```

```
[[0.00077    0.002665  0.00204    ... 0.000507    0.00030499 0.000828   ]
 [0.002665   0.01162999 0.00797999 ... 0.00182401 0.00075998 0.00160351]
 [0.00204    0.00797999 0.00603    ... 0.00141901 0.00068499 0.001536   ]
 ...
 [0.000507   0.00182401 0.00141901 ... 0.0003447   0.0001905   0.0004773   ]
 [0.00030499 0.00075998 0.00068499 ... 0.0001905   0.0001575   0.0004895   ]
 [0.000828   0.00160351 0.001536   ... 0.0004773   0.0004895   0.0016647   ]]
```

Figura 6: Covariância Resultado.

- Impressão dos autovalores e autovetores:

```
def z_score(m):
    """Remove a média e normaliza os pelo desvio padrão"""
    return (m - m.mean()) / m.std()
```

Figura 7: Desvio padrão.

```
m=np.array([[0.00077, 0.002665, 0.00204, 0.000507, 0.00030499, 0.000828 ],
 [0.002665, 0.01162999, 0.00797999, 0.00182401, 0.00075998, 0.00160351],
 [0.00204, 0.00797999, 0.00603, 0.00141901, 0.00068499, 0.001536 ],
 [0.000507, 0.00182401, 0.00141901, 0.0003447, 0.0001905, 0.0004773 ],
 [0.00030499, 0.00075998, 0.00068499, 0.0001905, 0.0001575, 0.0004895 ],
 [0.000828, 0.00160351, 0.001536, 0.0004773, 0.0004895, 0.0016647 ]])
```

Figura 8: Matriz de covariância

```
autovalores, autovetores= np.linalg.eig(m)
w, v = np.linalg.eig(m)
print("Impressão dos autovalores:\n",
      w)
print("Impressão dos autovetores:\n",
      v)
```

Figura 9: Autovalores e Autovetores

```
... Impressão dos autovalores:
[ 1.86518629e-02  1.65970801e-03  2.85321699e-04  7.54065697e-09
 -5.69028033e-09 -4.49482258e-09]
```

Figura 10: Autovalores Resultado.

```
Impressão dos autovetores:
[[-0.19143155 -0.228249 -0.00730498  0.68819554  0.65222304 -0.11043096]
 [-0.78100667  0.32344177  0.52704956 -0.08499581 -0.01966005  0.0046861 ]
 [-0.55965785 -0.10586464 -0.77034918  0.01205246 -0.24714172 -0.14461542]
 [-0.13075701 -0.10503208 -0.16208657 -0.22650491  0.31313123  0.89232459]
 [-0.06100716 -0.2299014 -0.03531334 -0.68387313  0.55338196 -0.41019799]
 [-0.13909121 -0.87646683  0.31813276 -0.00551464 -0.32993454  0.04861962]]
```

Figura 11: Resultado dos Autovetores.

```
print("Printar matriz original:\n", m)
[10] ✓ 0.9s
... Printar matriz original:
[[0.00077  0.002665  0.00204  0.000507  0.00030499 0.000828 ]
 [0.002665  0.01162999 0.00797999 0.00182401 0.00075998 0.00160351]
 [0.00204  0.00797999 0.00603  0.00141901 0.00068499 0.001536 ]
 [0.000507  0.00182401 0.00141901 0.0003447  0.0001905  0.0004773 ]
 [0.00030499 0.00075998 0.00068499 0.0001905  0.0001575  0.0004895 ]
 [0.000828  0.00160351 0.001536  0.0004773  0.0004895  0.0016647 ]]
```

Figura 12: Matriz original.

- Dois Maiores autovalores:

1.86518629e-02 1.65970801e-03

Figura 13: Maiores autovalores.

- Colunas dos respectivos dois maiores autovalores para plotagem 2D:

```
x=[5.72, 5.76, 5.37, 5.19, 5.06]
y=[5.65, 5.53, 5.21, 4.95, 4.99]
plt.plot(x,y)
```

Figura 14: Colunas dos maiores autovalores.

- Plot das respectivas colunas em 2D:

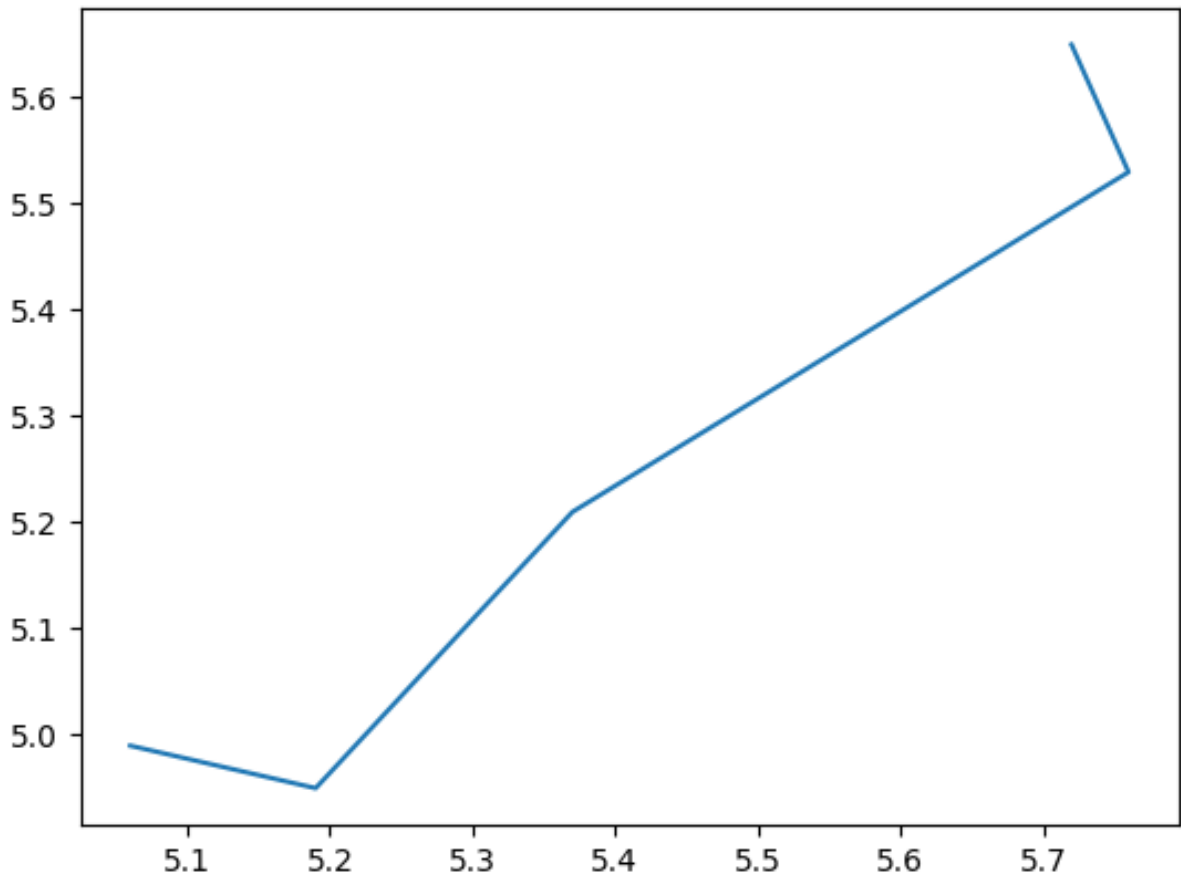


Figura 15: Plot Final 1.

- Colunas dos respectivos dois maiores autovalores para plot 3D:

```
ax = plt.axes(projection='3d')  
x=[5.72, 5.76, 5.37, 5.19, 5.06]  
y=[5.65, 5.53, 5.21, 4.95, 4.99]  
  
ax.plot3D(x, y, 'red')  
  
ax.scatter3D(x, y)
```



0.8s

Figura 16: Colunas dos maiores autovalores.

- Plot das respectivas colunas em 3D:

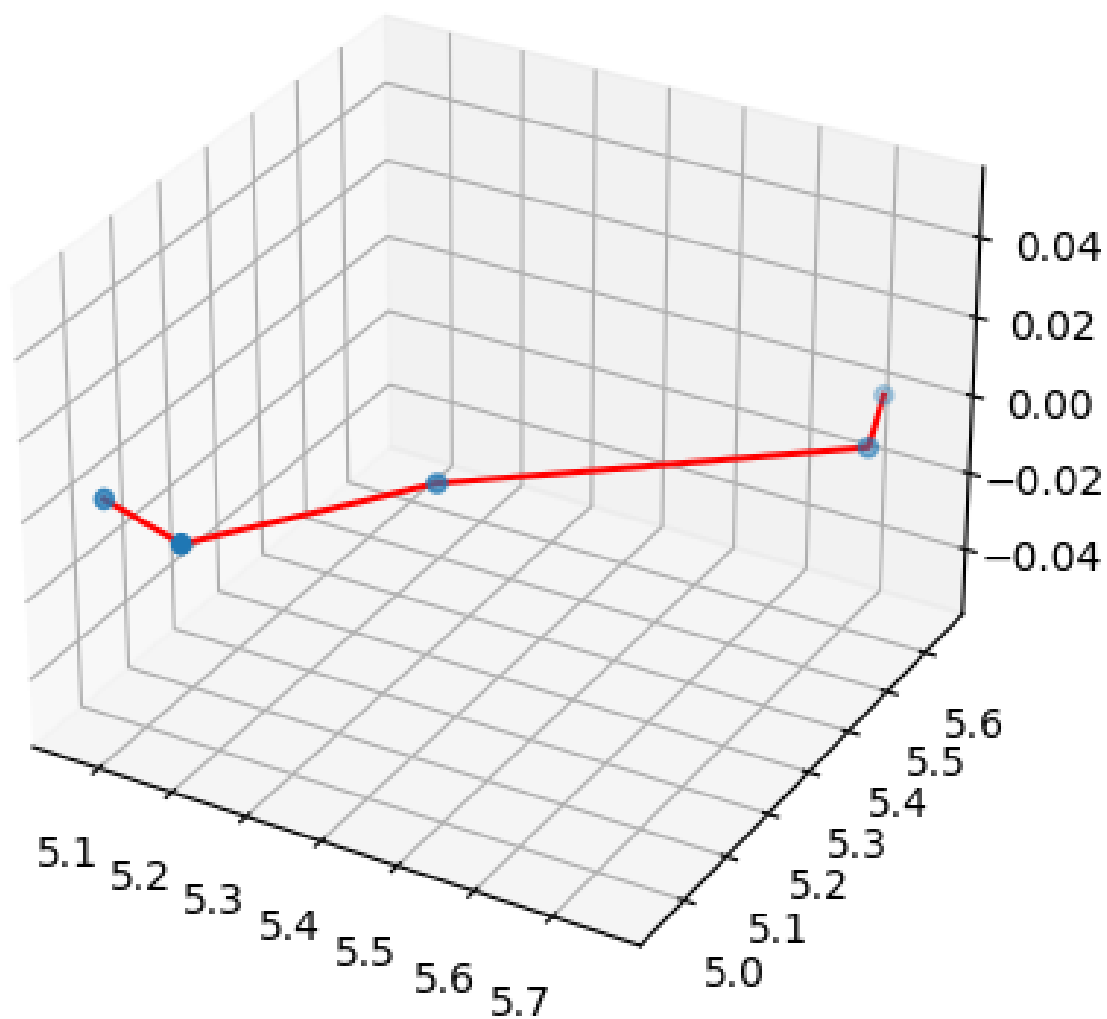


Figura 17: Plot Final 2.

- Link do repositório do Github.
(Link: <https://github.com/AnabelleSouza/DataSet-AlgoritmoPCA>);