Principal Components Analists
Algebra Linear

Augusto Rodrigo Camblor Santos
Gabriela Duarte Maciel

06/12/2022

- Dataset selecionado.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import random as rd
from sklearn import preprocessing
from sklearn.decomposition import PCA

data = pd.read_csv("./input/ProjetosRenunciaFiscal1.csv", encoding="utf-8")

dataTitulo = data["TITULO_PROJETO"]

del data["TITULO_PROJETO"]
del data["CNPJ_PROPONENTE"]
del data["Unnamed: 0"]
del data["Unnamed: 2"]
del data["UF_PROPONENTE"]
del data["SITUACAO_REGISTRO"]
del data["RAZAO_SOCIAL_PROPONENTE"]

data["LEI_8313"]= data["LEI_8313"].str.replace(",", ".")
data["ART1"] = data["ART1"].str.replace(",", ".")
data["ART1A"]= data["ART1A"].str.replace(",", ".")
data["ART3"] = data["ART3"].str.replace(",", ".")
data["ART3A"]= data["ART3A"].str.replace(",", ".")
data["ART39"]= data["ART39"].str.replace(",", ".")
data["FUNCINES"] = data["FUNCINES"].str.replace(",", ".")
data["TOTAL_CAPTADO"]= data["TOTAL_CAPTADO"].str.replace(",", ".")

print(data)
print(data.head())
print(data.shape)
```

- Cálculo da matriz de covariância, autovalores e autovetores.

```
scaled_data = preprocessing.scale(data)

dataIndex = object

for gene in data.index:
dataTitulo = data.loc[gene]

pca = PCA()
pca.fit(scaled_data)
pca_data = pca.transform(scaled_data)

per_var = np.round(pca.explained_variance_ratio_* 100, decimals = 1)
labels = ['PC' + str (x) for x in range (1, len(per_var)+1)]

plt.bar(x=range(0, len(per_var)), height=per_var, tick_label = labels)
plt.ylabel('Percentage of Explained Variance')
plt.xlabel('Principal Component')
plt.title('Scree Plot')
plt.show()
```

- Dois maiores autovalores.

```
pca_df = pd.DataFrame(pca_data, index = data, columns = labels)

plt.scatter(pca.PC1, pca_df.PC2)
plt.title('My PCA Graph')
plt.xlabel('PC1'.format(per_var))
plt.ylabel('PC2'.format(per_var))

plt.show()
```

- Plotagem das colunas correspondentes aos dois autovalores.

```
loading_scores = pd.Series(pca.components_[0], index = dataTitulo)
sorted_loading_scores = loading_scores.abs().sort_values(ascending=False)
top_10_projetos = sorted_loading_scores[0:10].index.values
print(loading_scores[top_10_projetos])
```

- Link do repositório do GitHub
(Link: Link do repositório no GitHub: https://github.com/AugustoRC/P2Algebra);
- Prints dos resultados gerados:

	LEI_8313	ART1	ART1A	ART3	ART3A	ART39	FUNCINES	TOTAL_CAPTADO
0	190000	440000	0	0	0	0	0	630000
1	0	2604045	970000	0	0	0	0	3574045
2	0	310251	1000000	0	0	0	0	1310251
3	162500	0	0	0	0	0	0	162500
4	0	0	0	0	0	704999.78	0	704999.78
...	...	...	...	...	...	...	...	...
2831	0	1330434	0	859632.64	0	0	0	2190066.64
2832	1737500	3000000	0	1800000	0	0	0	6537500
2833	260000	0	0	0	0	0	0	260000
2834	0	0	1340000	0	0	0	0	1340000
2835	0	0	250000	0	0	0	0	250000

Figure 1: Banco de dados selecionado.

Figure 2: Banco de dados após o tratamento.

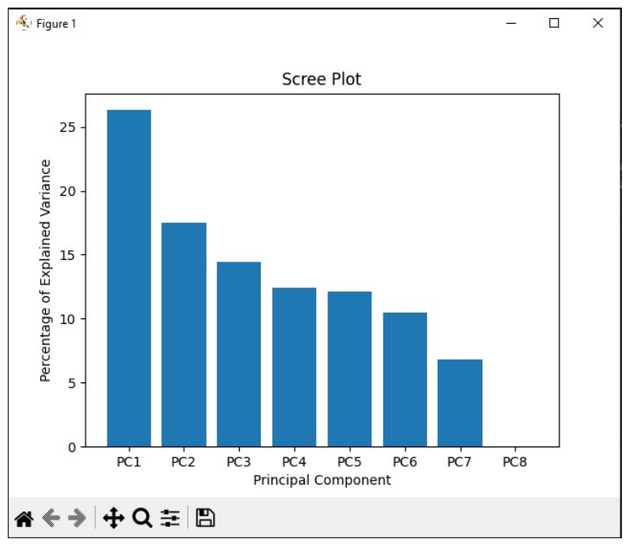

Figure 3: Cálculo de autovetores e autovalores.

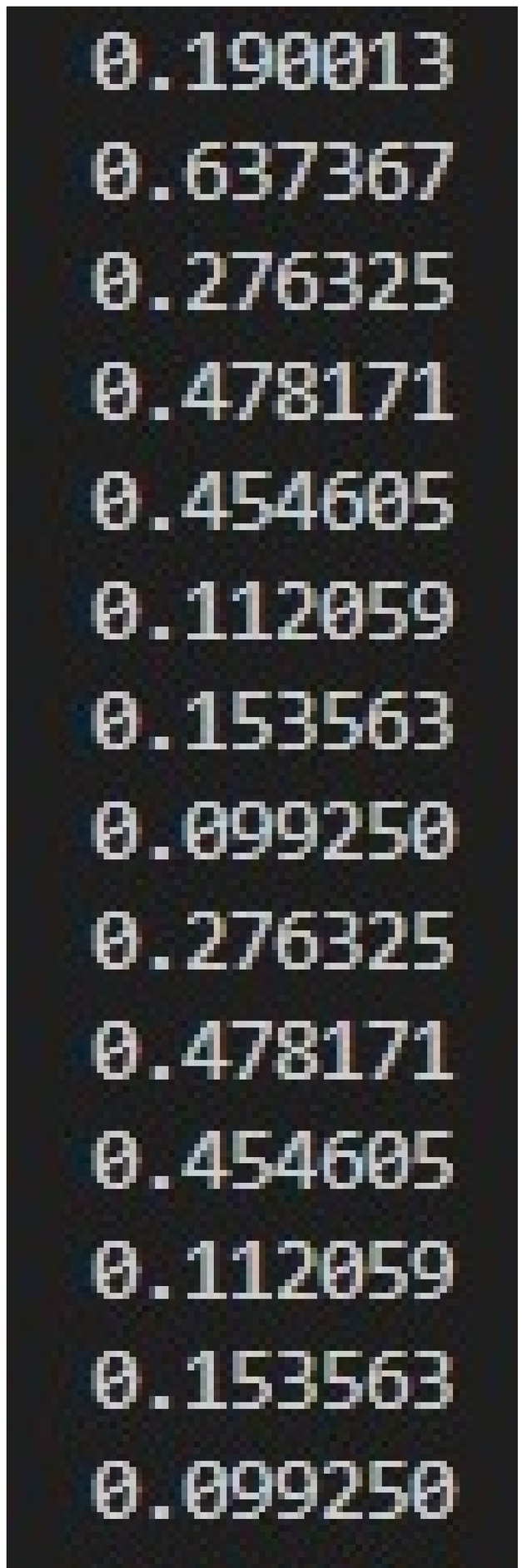

Figure 4: Maiores Autovalores.

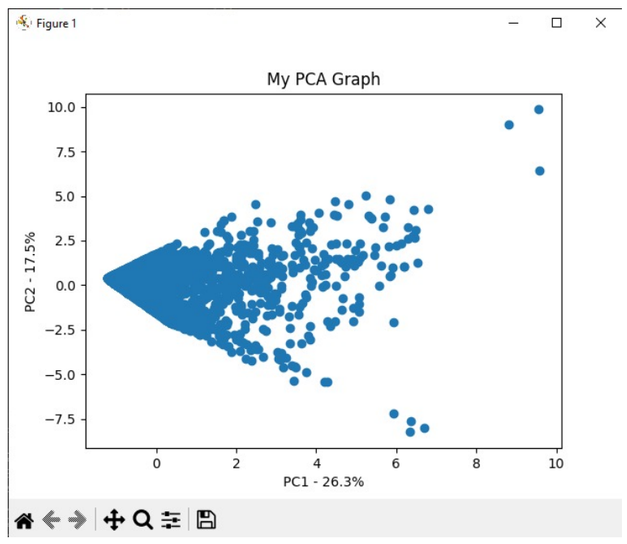

Figure 5: Plotagem dos maiores autovalores.