main

May 23, 2022

1 Ejercicio Obligatorio 2 | Grupo 1

- Florencia Chao
- Ariadna Fernandez Truglia
- Faustino Maggioni Duffy

[1]:
```
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

raw_data = pd.read_csv('europe.csv')
features = ['Area', 'GDP', 'Inflation', 'Life.expect', 'Military', 'Pop.
 ↪growth', 'Unemployment']

x = raw_data.loc[:, features].values
y = raw_data.loc[:,['Country']].values
x = StandardScaler().fit_transform(x)

pca = PCA(n_components=0.95, svd_solver='full')

components = pca.fit_transform(x)
df = pd.DataFrame(data = components)
final_df = pd.concat([df, raw_data[['Country']]], axis = 1)
first_component = pd.DataFrame(data=components[:, 0], index=y).T

print(final_df)
print(f'Primer Componente: \n {first_component}')
print(f'Varianza: \n {pca.explained_variance_ratio_}' + "\n")
print(f'Varianza acumulada: \n {pca.explained_variance_ratio_.cumsum()}')
print(f'Autovectores: \n {pca.components_}')
print('First componente: ', first_component.values)
print('First componente: ', first_component.columns)
```

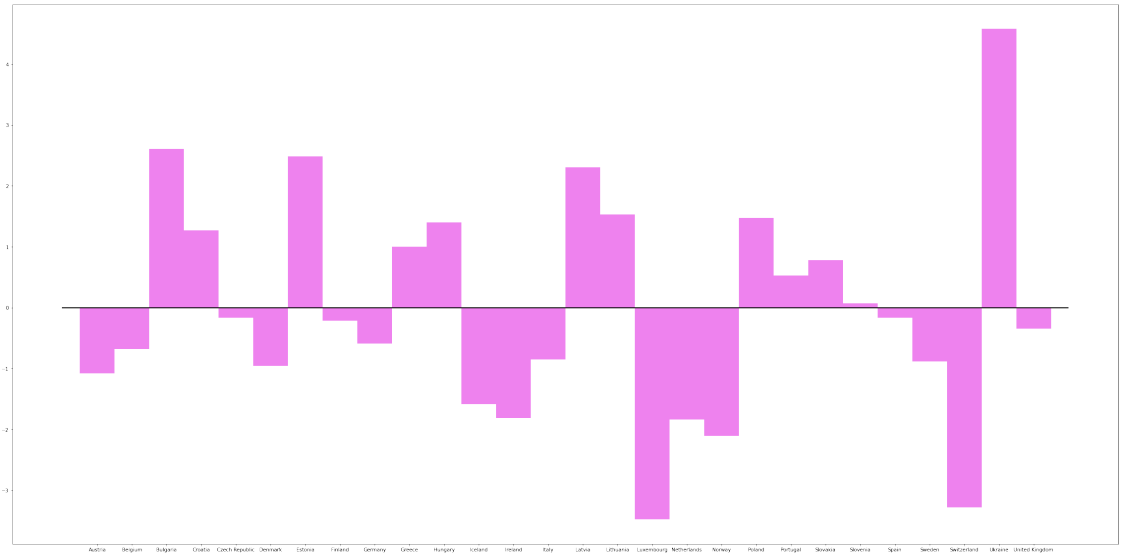
```
          0         1         2         3         4         Country
0  -1.081748 -1.270051 -0.514803 -0.413907  0.001251         Austria
1  -0.681094 -0.416041 -0.687292 -0.283364  0.273518         Belgium
2   2.609879  0.269638 -0.623031 -1.331941  0.001946        Bulgaria
3   1.270149  1.901427 -0.831925  0.371907 -0.377894         Croatia
```

```
4  -0.167209 -0.131943 -1.001979 -0.263449 -1.047409  Czech Republic
5  -0.955191 -0.409628 -0.732314 -0.391339  0.027888         Denmark
6   2.487735 -0.085779 -0.987538 -0.303984  0.555597         Estonia
7  -0.210563 -0.033717  1.174438 -0.614491 -0.145780         Finland
8  -0.592394 -0.471832  0.992255 -0.843702 -0.981331         Germany
9   1.000472  3.406849  0.738084 -0.793175  1.068531          Greece
10  1.396898 -0.034231 -0.695239 -0.097312  0.050388         Hungary
11 -1.583720 -1.477264 -0.306748  1.251670  0.167241         Iceland
12 -1.808918  0.521808 -0.301615  1.829776  0.372668         Ireland
13 -0.853224  0.327799  1.114486 -0.137694 -0.377246           Italy
14  2.306059 -0.675348 -1.301515  0.373985 -0.247019          Latvia
15  1.530100 -0.194997 -1.076834  1.142023 -0.283135       Lithuania
16 -3.478435 -1.076287 -0.361319  0.361099  2.021508      Luxembourg
17 -1.840053 -0.057229 -0.453754 -0.770093  0.132617     Netherlands
18 -2.106511 -0.143963  1.035463 -1.362686 -0.547047          Norway
19  1.471774  0.072985  0.741919  0.207883 -0.116252          Poland
20  0.526493  1.034774 -0.127654  0.056053  0.451807        Portugal
21  0.782966 -0.172260 -0.951409  0.930525  0.095982        Slovakia
22  0.067543  0.797672 -1.171579 -0.237358 -0.715185        Slovenia
23 -0.163767  1.152066  2.129796  2.613753 -0.725067           Spain
24 -0.885105 -0.402881  1.796021 -0.241101 -0.612589          Sweden
25 -3.281586 -0.108227 -0.753067 -0.476104 -0.813816     Switzerland
26  4.580268 -2.829042  1.893035 -0.011625  0.467664         Ukraine
27 -0.340819  0.505702  1.264115 -0.565349  1.301161  United Kingdom
Primer Componente:
    (Austria,)  (Belgium,)  (Bulgaria,)  (Croatia,)  (Czech Republic,)  \
0   -1.081748   -0.681094     2.609879    1.270149          -0.167209

   (Denmark,)  (Estonia,)  (Finland,)  (Germany,)  (Greece,)  …  (Norway,)  \
0   -0.955191    2.487735   -0.210563   -0.592394   1.000472  …  -2.106511

   (Poland,)  (Portugal,)  (Slovakia,)  (Slovenia,)  (Spain,)  (Sweden,)  \
0   1.471774     0.526493     0.782966     0.067543 -0.163767  -0.885105

   (Switzerland,)  (Ukraine,)  (United Kingdom,)
0       -3.281586    4.580268          -0.340819

[1 rows x 28 columns]
Varianza:
 [0.46102367 0.16958906 0.15188436 0.11005085 0.06540695]

Varianza acumulada:
 [0.46102367 0.63061273 0.78249709 0.89254794 0.95795489]
Autovectores:
 [[ 1.24873902e-01 -5.00505858e-01  4.06518155e-01 -4.82873325e-01
   1.88111616e-01 -4.75703554e-01  2.71655820e-01]
 [-1.72872202e-01 -1.30139553e-01 -3.69657243e-01  2.65247797e-01
   6.58266888e-01  8.26219831e-02  5.53203705e-01]
```

```
 [ 8.98296740e-01  8.39557607e-02  1.98194675e-01  2.46082460e-01
   2.43679433e-01  1.63697207e-01  5.00135736e-04]
 [ 4.48503976e-02 -8.42554739e-02  1.64685649e-01  2.67714373e-02
  -5.62374796e-01  3.92462767e-01  7.01967912e-01]
 [-3.24016926e-01  3.90632444e-01  6.89500539e-01 -1.01786561e-01
   3.68147581e-01  3.47867772e-01  1.01587422e-02]]
First componente:  [[-1.08174766 -0.68109407  2.60987882  1.27014885 -0.16720949
-0.9551908
   2.48773522 -0.21056316 -0.59239365  1.00047196  1.39689831 -1.5837197
  -1.80891761 -0.85322396  2.30605941  1.53009991 -3.47843496 -1.84005341
  -2.10651083  1.47177383  0.52649333  0.78296597  0.06754338 -0.16376696
  -0.88510531 -3.28158613  4.58026807 -0.34081935]]
First componente:  Index([       ('Austria',),        ('Belgium',),
('Bulgaria',),
              ('Croatia',), ('Czech Republic',),        ('Denmark',),
              ('Estonia',),        ('Finland',),        ('Germany',),
               ('Greece',),        ('Hungary',),        ('Iceland',),
              ('Ireland',),          ('Italy',),         ('Latvia',),
            ('Lithuania',),     ('Luxembourg',),    ('Netherlands',),
               ('Norway',),         ('Poland',),       ('Portugal',),
             ('Slovakia',),       ('Slovenia',),          ('Spain',),
               ('Sweden',),    ('Switzerland',),        ('Ukraine',),
       ('United Kingdom',)],
      dtype='object')
```

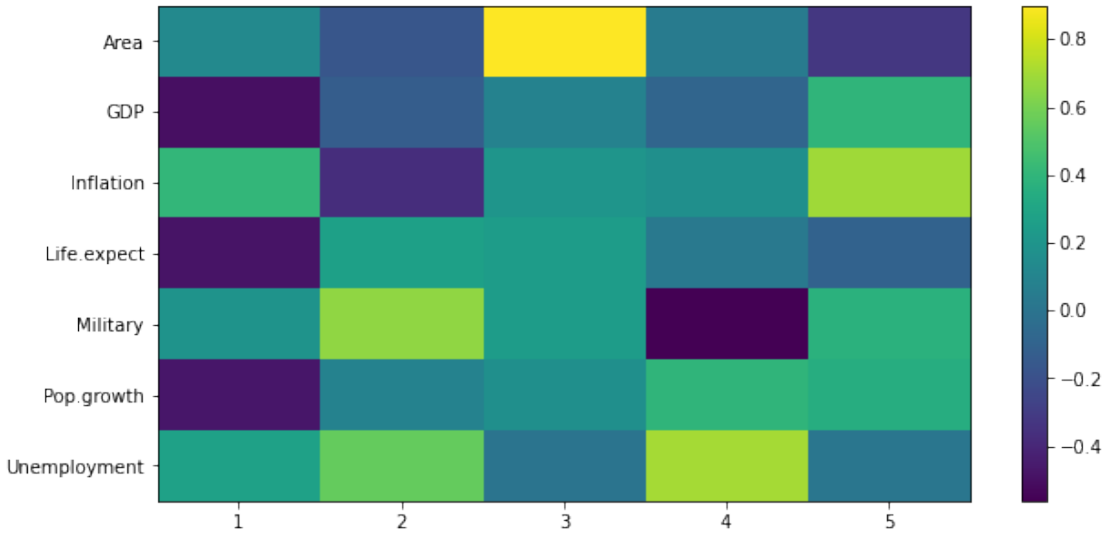
1.1 Valor del primer componente principal para cada pais

[2]:
```
import numpy as np
import matplotlib.pyplot as plt

bars = first_component.values[0]
labels = list(map(lambda t: t[0], first_component.columns))
plt.figure(figsize=(40, 20))
plt.bar(labels,bars, width=1, align='center', color='violet')
plt.plot(np.linspace(-1,28,num=28), np.zeros(len(bars)), linewidth=2,␣
 ↪color='black')
plt.show()
```



1.2 Influencia de las variables en cada componente

[3]:
```
fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(10, 5))
componentes = pca.components_
plt.imshow(componentes.T, cmap='viridis', aspect='auto')
plt.yticks(range(len(raw_data.columns[1:])), raw_data.columns[1:])
plt.xticks(range(len(componentes)), np.arange(pca.n_components_) + 1)
plt.grid(False)
plt.colorbar();
```



1.3 Porcentaje de variancia explicado por cada componente

[4]:
```
fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(6, 4))
ax.bar(
    x      = np.arange(pca.n_components_) + 1,
    height = pca.explained_variance_ratio_
)

for x, y in zip(np.arange(len(raw_data.columns[1:])) + 1, pca.
 ↪explained_variance_ratio_):
    label = round(y, 2)
    ax.annotate(
        label,
        (x,y),
        textcoords="offset points",
        xytext=(0,10),
        ha='center'
    )

ax.set_xticks(np.arange(pca.n_components_) + 1)
ax.set_ylim(0, 1.1)
ax.set_title('Porcentaje de varianza explicada por cada componente')
ax.set_xlabel('Componente principal')
ax.set_ylabel('Por. varianza explicada');
```