→ ANALIZA SKŁADOWYCH GŁÓWNYCH

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
Pobranie danych:
data=pd.read_csv('PCA_simpledata.csv')
data
         х у
     0 2.5 2.4
     1 0.5 0.7
     2 2.2 2.9
     3 1.9 2.2
     4 3.1 3.0
     5 2.3 2.7
     6 2.0 1.6
     7 1.0 1.1
     8 1.5 1.6
     9 1.1 0.9
Wykres rozrzutu:
plt.scatter(data.iloc[:,0],data.iloc[:,1])
```

```
<matplotlib.collections.PathCollection at 0x11612f5b0>
      3.0
Macierz korelacji:
      2.5 +
CorrMatrix = np.array(data.corr())
CorrMatrix
array([[1.
                      , 0.92592927],
            [0.92592927, 1.
Wartości i wektory własne macierzy korelacji:
#TODO
w, v = np.linalg.eig(CorrMatrix)
print(w)
print(v)
    [1.92592927 0.07407073]
    [[ 0.70710678 -0.70710678]
     [ 0.70710678  0.70710678]]
v[:,0]
    array([0.70710678, 0.70710678])
v[:,1]
    array([-0.70710678, 0.70710678])
```

Sprawdź poprawność wyliczeń wektorów i wartości własnych. Wykorzystaj równanie na wartości własne.

```
#TODO
u=v[:,1]
print(u)
      [-0.70710678  0.70710678]

lam = w[1]
print(lam)
      0.07407072730775455

print(np.dot(CorrMatrix,u))
      [-0.05237591  0.05237591]

print(lam*u)
      [-0.05237591  0.05237591]
```

Dane w nowym układzie współrzędnych wyznaczonym przez wektory własne macierzy korelacji (zobacz slajdy 30 i 31 z ED_teoria_4.pdf):

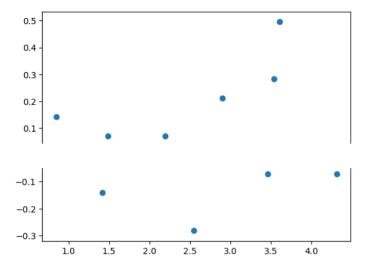
```
#TODO
#print(data.iloc[0,0]*v[0,0]+data.iloc[0,1]*v[1,0],data.iloc[0,0]*v[0,1]+data.iloc[0,1]*v[1,1])
T=np.array([])
for i in range (10):
 x=data.iloc[i,0]*v[0,0]+data.iloc[i,1]*v[1,0]
 y=data.iloc[i,0]*v[0,1]+data.iloc[i,1]*v[1,1]
 T=np.append(T,[x,y],axis=0)
 print(x,y)
 #print(data.iloc[i,0]*v[0,0]+data.iloc[i,1]*v[1,0],data.iloc[i,0]*v[0,1]+data.iloc[i,1]*v[1,1])
    3.4648232278140827 -0.07071067811865461
    0.8485281374238569 0.14142135623730956
    3.6062445840513924 0.4949747468305832
    2.899137802864845 0.21213203435596473
    4.31335136523794 -0.07071067811865417
    3.5355339059327378 0.28284271247461956
    2.5455844122715714 -0.2828427124746187
    1.48492424049175 0.07071067811865495
    2.1920310216782974 0.07071067811865506
    1.4142135623730951 -0.14142135623730945
T=np.reshape(T,(10,2))
Wykres rozrzutu danych w nowym układzie:
np.info(T)
    class: ndarray
    shape: (10, 2)
    strides: (16, 8)
    itemsize: 8
    aligned: True
```

plt.scatter(T[:,0],T[:,1])
plt.show()

contiguous: True fortran: False

byteorder: little
byteswap: False
type: float64

data pointer: 0x600002a2c3c0



• ×