

▼ ANALIZA SKŁADOWYCH GŁÓWNYCH

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

import math
from sympy import Matrix

from google.colab import drive
drive.mount('/content/drive',force_remount=True)

import os
os.chdir('/content/drive/My Drive/DM')

Mounted at /content/drive
```

Pobranie danych:

```
data=pd.read_csv('ED_lab_8/PCA_simplesdata.csv')
data
```

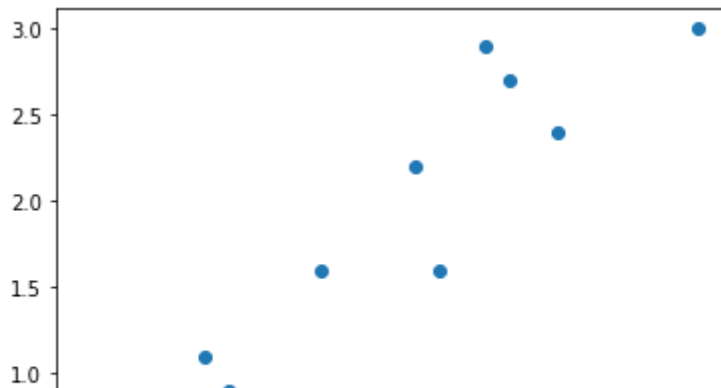
	X	Y
0	2.5	2.4
1	0.5	0.7
2	2.2	2.9
3	1.9	2.2
4	3.1	3.0
5	2.3	2.7
6	2.0	1.6
7	1.0	1.1
8	1.5	1.6
9	1.1	0.9

Wykres rozrzutu:

```
plt.scatter(data.iloc[:,0],data.iloc[:,1])
```



```
<matplotlib.collections.PathCollection at 0x7f757fce66d8>
```



Macierz korelacji:

```
CorrMatrix = np.array(data.corr())
CorrMatrix

array([[1.          , 0.92592927],
       [0.92592927, 1.          ]])
```

Wartości i wektory własne macierzy korelacji:

```
values, vectors = np.linalg.eig(CorrMatrix)
print("Wartości własne:\n ")
print(values)
print("Wektory własne:")
print(vectors)
print("")
print(data)
```

Wartości własne:

```
[1.92592927 0.07407073]
```

Wektory własne:

```
[[ 0.70710678 -0.70710678]
 [ 0.70710678  0.70710678]]
```

	X	Y
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

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

```
m = Matrix(CorrMatrix)
```

```

dane = np.array(data)

print(m)
1 - (m[1] * m[2])
x1 = 1.925929
x2 = 0.074070

print("\ndla x1: ",x1)
print(m[0] - x1, "x + ", m[1], "y = 0")
print(m[3] - x1, "x + ", m[2], "y = 0")
print("\nSpełnia warunek dla wektorow takich, że [-x = y]\n")

print("dla x2: ",x2)
print(m[0] - x2, "x + ", m[1], "y=0")
print(m[3] - x2, "x + ", m[2], "y=0")
#nieskonczenie wiele rozwiązań
print("\nSpełnia warunek dla wektorow takich, że [x = y]\n")

print(dane)
print(np.mean(data['X']))
print(np.mean(data['Y']))
meanX = np.mean(data['X'])
meanY = np.mean(data['Y'])

Matrix([[1.00000000000000, 0.925929272692246], [0.925929272692246, 1.00000000000000]])

dla x1: 1.925929
-0.925929000000000 x + 0.925929272692246 y = 0
-0.925929000000000 x + 0.925929272692246 y = 0

Spełnia warunek dla wektorow takich, że [-x = y]

dla x2: 0.07407
0.925930000000000 x + 0.925929272692246 y=0
0.925930000000000 x + 0.925929272692246 y=0

Spełnia warunek dla wektorow takich, że [x = y]

[[2.5 2.4]
 [0.5 0.7]
 [2.2 2.9]
 [1.9 2.2]
 [3.1 3. ]
 [2.3 2.7]
 [2.  1.6]
 [1.  1.1]
 [1.5 1.6]
 [1.1 0.9]]
1.81
1.9099999999999997

```

Dane w nowym układzie współrzędnych wyznaczonym przez wektory własne macierzy korelacji (zobacz slajdy 30 i 31 z ED_teorja_4.pdf):

```
wektorv = Matrix(vectors)
```

```
wektory = dane[2:\text{rows},
```

```
Wek_data = []
```

```
for i in range(0, len(dane), 2):
```

```
    print(dane[i : i + 2], "\n")
```

```
    mx = Matrix(dane[i : i + 2])
```

```
    Wek_data.append((mx * wektory))
```

```
Wek_data
```

```
[[2.5 2.4]
 [0.5 0.7]]
```

```
[[2.2 2.9]
 [1.9 2.2]]
```

```
[[3.1 3. ]
 [2.3 2.7]]
```

```
[[2.  1.6]
 [1.  1.1]]
```

```
[[1.5 1.6]
 [1.1 0.9]]
```

```
[Matrix([
 [ 3.46482322781408, -0.0707106781186542],
 [0.848528137423857,  0.14142135623731]]), Matrix([
 [3.60624458405139, 0.494974746830584],
 [2.89913780286485, 0.212132034355965]]), Matrix([
 [4.31335136523794, -0.0707106781186537],
 [3.53553390593274,  0.28284271247462]]), Matrix([
 [2.54558441227157, -0.282842712474618],
 [1.48492424049175, 0.0707106781186551]]), Matrix([
 [ 2.1920310216783, 0.0707106781186555],
 [1.41421356237309, -0.141421356237309]])]
```

```
tab = []
```

```
for i in Wek_data:
```

```
    tab.append([i[0], i[1]])
```

```
    tab.append([i[2], i[3]])
```

```
for i in tab:
```

```
    print(i[0], " ", i[1])
```

```
    plt.scatter(i[0], i[1], c = 'y')
```

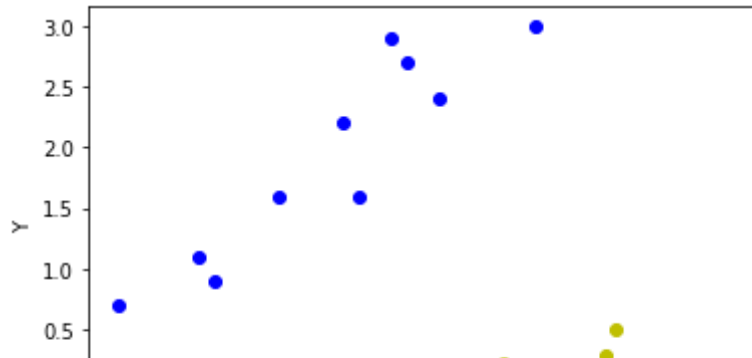
```
plt.scatter(data['X'], data['Y'], c = 'b')
```

```
plt.xlabel('X')
```

```
plt.ylabel('Y')
```

```
plt.show()
```

3.46482322781408	-0.0707106781186542
0.848528137423857	0.141421356237310
3.60624458405139	0.494974746830584
2.89913780286485	0.212132034355965
4.31335136523794	-0.0707106781186537
3.53553390593274	0.282842712474620
2.54558441227157	-0.282842712474618
1.48492424049175	0.0707106781186551
2.19203102167830	0.0707106781186555
1.41421356237309	-0.141421356237309



Wykres rozrzutu danych w nowym układzie:

0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5

```
for i in A:
    plt.scatter(i[0] + meanX, i[1] + meanY, c = 'y')
plt.scatter(data['X'], data['Y'], c = 'b')
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```

