

▼ ANALIZA SKŁADOWYCH GŁÓWNYCH

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

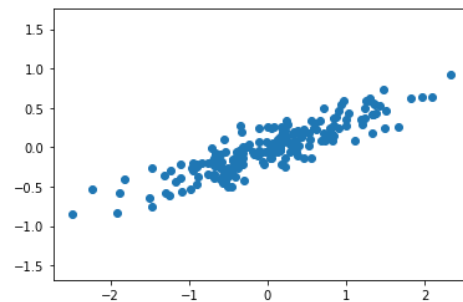
Ładowanie zbioru danych:

```
data=pd.read_csv('PCA_dataset.csv')
data
```

	x	y
0	-0.625302	-0.170064
1	0.960695	0.590901
2	-0.598543	-0.402593
3	-2.228059	-0.532577
4	-0.461430	-0.498867
...
195	0.838409	0.278380
196	0.312708	-0.010600
197	1.955876	0.638224
198	-1.108304	-0.395054
199	0.398647	0.023187

200 rows × 2 columns

```
plt.scatter(data.iloc[:,0], data.iloc[:,1])
plt.axis('equal');
```



▼ Dwie składowe główne

Wyliczamy dwie składowe główne (**n_components=2**) czyli wektory bazowe nowego układu współrzędnych:

+ Kod + Tekst

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
pca.fit(data)

PCA(n_components=2)
```

Wektory bazowe nowego układu współrzędnych:

```
print(pca.components_)

[[-0.944446029 -0.32862557]
 [-0.32862557  0.944446029]]
```

Wariancje dla nowych współrzędnych:

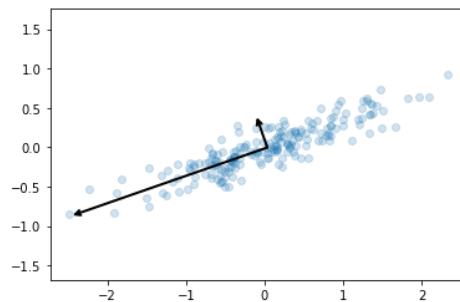
```
print(pca.explained_variance_)

[0.7625315  0.0184779]
```

Dodajemy do wykresu wektory wyznaczające nowy układ współrzędnych. Ich długość określona jest przez wariancje:

```
def draw_vector(v0, v1, ax=None):
    ax = ax or plt.gca()
    arrowprops=dict(arrowstyle='->',
                    linewidth=2,
                    shrinkA=0, shrinkB=0, color='black')
    ax.annotate('', v1, v0, arrowprops=arrowprops)

plt.scatter(data.iloc[:, 0], data.iloc[:, 1], alpha=0.2)
for length, vector in zip(pca.explained_variance_, pca.components_):
    v = vector * 3 * np.sqrt(length)
    draw_vector(pca.mean_, pca.mean_ + v)
plt.axis('equal');
```



Współrzędne punktów w **nowym układzie współrzędnych**:

```

data_pca = pca.transform(data)
data_pca

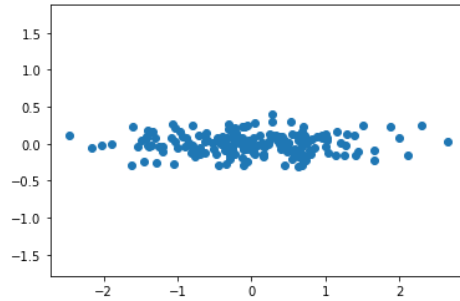
[ -9.24887775e-02, -6.30062621e-02],
[ -3.05962249e-02, -3.57199779e-03],
[  1.30795754e-01, -4.00179911e-03],
[ -7.74659629e-02, -7.83158849e-02],
[ -4.20826569e-01,  5.54588300e-03],
[  6.78334448e-01, -1.36015287e-01],
[ -6.35104074e-01,  1.06651554e-01],
[  2.72075594e-01,  3.93203001e-01],
[ -2.26801066e-01, -1.36909181e-01],
[ -1.45908094e+00, -2.45286954e-01],
[  4.03275391e-01, -7.98143251e-02],
[  4.88618199e-01,  2.35232561e-03],
[ -3.77797862e-02,  4.11805137e-02],
[  2.25514691e-01,  2.18823208e-02],
[  3.73320407e-01, -9.42631804e-03],
[  9.96559672e-01,  1.35680064e-01],
[  6.68655132e-01,  5.84565987e-02],
[ -3.09207055e-01,  1.97249840e-01],
[  1.44746288e+00, -1.14083649e-01],
[ -1.27674147e-01,  7.26979511e-02],
[  1.95898129e-02, -6.44892347e-02],
[ -4.68331172e-01,  7.43248455e-02],
[ -7.59794861e-01, -9.50500102e-02],
[  2.11566325e+00, -1.50228254e-01],
[ -1.28843614e+00, -2.56574123e-01],
[  5.24455206e-01,  2.90379472e-01],
[  2.68082969e-01,  1.04520533e-01],
[  4.06271559e-02, -1.31461762e-01],
[ -1.63087335e+00, -2.89691466e-01],
[  4.50273668e-01,  2.87481726e-02],
[ -1.41736985e+00,  9.84653747e-02],
[ -3.20579341e-01,  7.34129168e-02],
[ -2.16095416e+00, -6.30801894e-02],
[  7.55938440e-01,  1.58567217e-02],
[  1.13147728e+00, -1.63990641e-01],
[ -4.01022769e-01,  9.35581760e-02],
[ -1.33261395e-01,  2.30758543e-01],
[ -1.20765775e-01, -2.63426989e-02],
[  1.03185993e+00, -1.29167415e-01],
[ -1.29878689e-01, -1.84090463e-01],
[ -4.08011754e-01,  6.01636158e-02],
[  4.17084437e-01,  9.04137707e-02],
[ -1.00930809e-01, -6.78582909e-02],
[  7.22839507e-02, -2.66653403e-02],
[  6.47903117e-01, -6.91638080e-02],
[  4.74689466e-01, -1.14557938e-03],
[  6.85499472e-01, -2.13675563e-01],
[ -1.49366216e+00,  5.05889730e-02],
[ -3.49297457e-01, -2.69159245e-01],
[ -7.79713261e-01, -1.13958001e-01],
[  5.67446775e-01, -1.25273441e-01],
[  5.18831382e-02,  2.76513979e-01],
[  1.25350822e+00, -1.59067690e-01],
[ -8.53016941e-01,  2.26324489e-03],
[ -2.61547685e-01, -9.79084715e-02],
[ -2.02667441e+00, -2.51066183e-02],
[  1.20688282e+00,  5.97125713e-03],
[ -3.53816725e-01, -9.42400167e-02]])

```

```

plt.scatter(data_pca[:,0], data_pca[:,1])
plt.axis('equal');

```



▼ Jedna składowa główna

Wyliczamy jedną składową główną (**n_components=1**) - chcemy wyeliminować jeden wymiar danych.

```
pca = PCA(n_components=1)
pca.fit(data)

PCA(n_components=1)
```

Współrzędne punktów w **nowym układzie współrzędnych**:

```
data_pca = pca.transform(data)
```

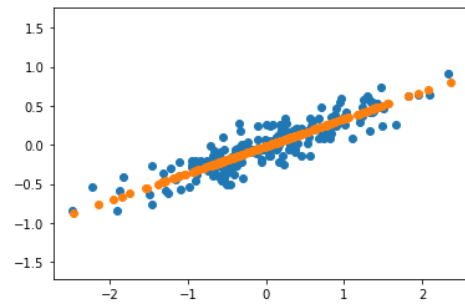
Porównanie kształtów danych początkowych i po redukcji jednego wymiaru:

```
print("Początkowy shape: ", data.shape)
print("Po transformacji shape:", data_pca.shape)

Początkowy shape: (200, 2)
Po transformacji shape: (200, 1)
```

Dane początkowe i po redukcji wymiaru:

```
data_new = pca.inverse_transform(data_pca)
plt.scatter(data.iloc[:, 0], data.iloc[:, 1]),
plt.scatter(data_new[:, 0], data_new[:, 1]),
plt.axis('equal');
```



✓ 0 s ukończono o 09:53

