

Daniszewska_lab4_exported

October 23, 2025

```
[5]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pathlib import Path

CSV_PATH = Path("1.csv")

#1 Wczytanie i przygotowanie danych
df = pd.read_csv(CSV_PATH)
numeric_df = df.select_dtypes(include=[np.number])

use_synthetic = False
reason = ""
if numeric_df.shape[1] < 2:
    use_synthetic = True
    reason = "CSV ma < 2 kolumny numeryczne."
elif numeric_df.shape[0] < 2:
    use_synthetic = True
    reason = f"CSV ma zbyt mało wierszy: {numeric_df.shape[0]} (wymagane 2)."

if not use_synthetic:
    X = numeric_df.iloc[:, :2].to_numpy(dtype=float)
    col_names = list(numeric_df.columns[:2])
    src_desc = "Dane z pliku CSV"
else:
    # syntetyczne dane 2D (odchylne eliptyczne + rotacja)
    rng = np.random.default_rng(42)
    n_syn = 200
    t = rng.normal(size=(n_syn, 2)) @ np.array([[2.0, 0.0], [0.0, 0.5]])
    theta = np.deg2rad(35)
    R = np.array([[np.cos(theta), -np.sin(theta)],
                  [np.sin(theta), np.cos(theta)]])
    X = t @ R.T + np.array([5.0, -3.0])
    col_names = ["x", "y"]
    src_desc = f"Dane syntetyczne (powód: {reason})"

pd.DataFrame(X, columns=col_names).head()
```

```

#2 PCA via SVD: centrowanie, macierz kowariancji, SVD, wariancje, kąty
n = X.shape[0]
xbar = X.mean(axis=0)                # wektor średnich  $[\bar{x}, \bar{y}]$ 
B = X - xbar                         # centrowanie
C = (B.T @ B) / (n - 1)              # macierz kowariancji (nieobciążona)

# Rozkład SVD macierzy B
U, S, Vt = np.linalg.svd(B, full_matrices=False)
V = Vt.T                             # kolumny V to kierunki składowych głównych
variances = (S**2) / (n - 1)          # wariancje wzdłuż PC (  $= \Sigma^2/(n-1)$ )

def angle_deg(vec: np.ndarray) -> float:
    """Kąt wektora (w stopniach) względem osi X."""
    return float(np.degrees(np.arctan2(vec[1], vec[0])))

angle_pc1 = angle_deg(V[:, 0]) if V.shape[1] >= 1 else np.nan
angle_pc2 = angle_deg(V[:, 1]) if V.shape[1] >= 2 else np.nan

print("Źródło danych:", src_desc)
print("Środek (średnia) =", xbar)
print("Macierz kowariancji C =\n", C)
print("Wektory główne (kolumny V) =\n", V)
print("Wartości osobliwe  $\Sigma$  =", S)
print("Wariancje  $= \Sigma^2/(n-1)$  =", variances)
print(f"Kąt PC1 względem osi X: {angle_pc1:.4f}°")
print(f"Kąt PC2 względem osi X: {angle_pc2:.4f}°")

#3 Wykres punktów i osi głównych (skala osi = 2 * std dla czytelności)
fig = plt.figure(figsize=(7, 7))
ax = fig.add_subplot(111)
ax.scatter(X[:, 0], X[:, 1], s=12)
ax.set_aspect('equal', 'box')
ax.set_xlabel(col_names[0])
ax.set_ylabel(col_names[1])
ax.set_title("Dane i osie główne (PCA via SVD)")

origin = xbar
scale = 2.0
if len(variances) >= 1 and V.shape[1] >= 1 and np.isfinite(variances[0]):
    pc1 = V[:, 0] * np.sqrt(variances[0]) * scale
    ax.plot([origin[0] - pc1[0], origin[0] + pc1[0]],
            [origin[1] - pc1[1], origin[1] + pc1[1]],
            linewidth=2, label="PC1")
if len(variances) >= 2 and V.shape[1] >= 2 and np.isfinite(variances[1]):
    pc2 = V[:, 1] * np.sqrt(variances[1]) * scale
    ax.plot([origin[0] - pc2[0], origin[0] + pc2[0]],
            [origin[1] - pc2[1], origin[1] + pc2[1]],

```

```

[origin[1] - pc2[1], origin[1] + pc2[1]],
linewidth=2, label="PC2")

```

```

ax.legend(loc="best")
plt.show()

```

Źródło danych: Dane syntetyczne (powód: CSV ma zbyt mało wierszy: 1 (wymagane 2).)

Środek (średnia) = [5.04541158 -2.98798945]

Macierz kowariancji C =

```
[[2.64139602 1.75737111]
```

```
[1.75737111 1.47874142]]
```

Wektory główne (kolumny V) =

```
[[ -0.81057292  0.58563772]
```

```
[ -0.58563772 -0.81057292]]
```

Wartości osobliwe Σ = [27.89816676 6.44977845]

Wariancje = $\Sigma^2/(n-1)$ = [3.91109401 0.20904343]

Kąt PC1 względem osi X: -144.1519°

Kąt PC2 względem osi X: -54.1519°

