Pravděpodobnostní metody ve strojírenství

E.2 -Metody vícerozměrné statistické analýzy dat

FAKULTA STROJNÍ ČVUT V PRAZE

prof. RNDr. Gejza Dohnal, CSc.

ak. rok 2025/2026

E.2 - Metody vícerozměrné statistické analýzy dat

Základní informace:

E) Metody vícerozměrné statistické analýzy dat

Základní pojmy: Vícerozměrné rozdělení, marginální rozdělení, korelační koeficient, korelační matice, multikolinearita, Mahalanobisova vzdálenost.

Klíčové vztahy: Lineární model, hodnocení kvality regresního modelu;
Kontingenční tabulky, testy nezávislosti;
Redukce dimenze: metoda hlavních komponent (PCA), faktorová analýza;
Shluková analýza.

Literatura: [13]; [14]; [8];

Proč vícerozměrná analýza?

Reálné systémy = mnoho veličin současně

Příklady:

- Diagnostika strojů
- Kvalita výroby
- Senzorová data

Charakter dat

- Více veličin
- Korelace
- Šum
- Redundance

Klíčový problém:

Proměnné nejsou nezávislé

Proč vícerozměrná analýza?

Reálné systémy = mnoho veličin současně

Stavy systému jsou popsány náhodnými vektory:

$$\vec{X} = \big(X_1(\omega), X_2(\omega), \ldots, X_n(\omega)\big) = (X_1, X_2, \ldots, X_n)$$

se sdruženou distribuční funkcí $F(x_1, x_2, \ldots, x_n) = P(X_1 \leq x_1, X_2 \leq x_2, \ldots, X_n \leq x_n)$

vektorem středních hodnot $E\big(\vec{X}\big) = E(X_1, X_2, \ldots, X_n) = \big(E(X_1), E(X_2), \ldots, E(X_n)\big)$

a kovariančí maticí $\mathbf{D}(\vec{X}) = E(\vec{X} - E(\vec{X}))^T E(\vec{X} - E(\vec{X}))$

$$\mathbf{D}(\vec{X}) = \begin{pmatrix} Var(X_1) & Cov(X_1, X_2) & \cdots & Cov(X_1, X_n) \\ Cov(X_2, X_1) & Var(X_2) & \cdots & Cov(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_n, X_1) & Cov(X_n, X_2) & \cdots & Var(X_n) \end{pmatrix}$$

Vícerozměrná data

Reálné systémy = mnoho veličin současně

Měření (pozorování) systému v časech 1, 2, …, m lze zapsat ve tvaru datové matice:

$$\mathbf{X} = \begin{pmatrix} x_1^1 & x_2^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^m & x_2^m & \dots & x_n^m \end{pmatrix}$$

<- pozorování v čase 1

<- pozorování v čase 2

<- pozorování v čase m

Z dat odhadneme

$$\mu_i = \frac{1}{m}\sum_{k=1}^{m} x_i^k \qquad s_i^2 = \frac{1}{m-1}\sum_{k=1}^{m} (x_i^k - \mu_i)^2 \qquad c_{ij} = \frac{1}{m}\sum_{k=1}^{m} (x_i^k - \mu_i)(x_j^k - \mu_j)$$

a dostaneme výběrovou kovarianční matici

$$\Sigma = \begin{pmatrix} s_1^2 & c_{12} & \dots & c_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \dots & s_n^2 \end{pmatrix}$$

Vícerozměrná data

Reálné systémy = mnoho veličin současně

Měření (pozorování) systému v časech 1, 2, …, m lze zapsat ve tvaru datové matice:

$$\mathbf{X} = \begin{pmatrix} x_1^1 & x_2^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^m & x_2^m & \dots & x_n^m \end{pmatrix}$$

<- pozorování v čase 1

<- pozorování v čase 2

<- pozorování v čase m

Multikolinearita:

- Proměnné obsahují stejnou informaci, protože jsou silně závislé.
- Multikolinearita znamená, že máme víc proměnných než informací.
- Je to situace, kdy je jedna proměnná lineární kombinací ostatních.
- Pozná se podle vysokých hodnot korelačních koeficientů (>0,8)

Důsledky:

- nestabilní modely
- velké chyby odhadu
- „divné“ koeficienty
- máme více proměnných, ale ve skutečnosti měříme totéž

Vícerozměrná data

Reálné systémy = mnoho veličin současně

Měření (pozorování) systému v časech 1, 2, …, m lze zapsat ve tvaru datové matice:

$$\mathbf{X} = \begin{pmatrix} x_1^1 & x_2^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^m & x_2^m & \dots & x_n^m \end{pmatrix}$$

<- pozorování v čase 1

<- pozorování v čase 2

<- pozorování v čase m

Často se provádí standardizace (normalizace) dat

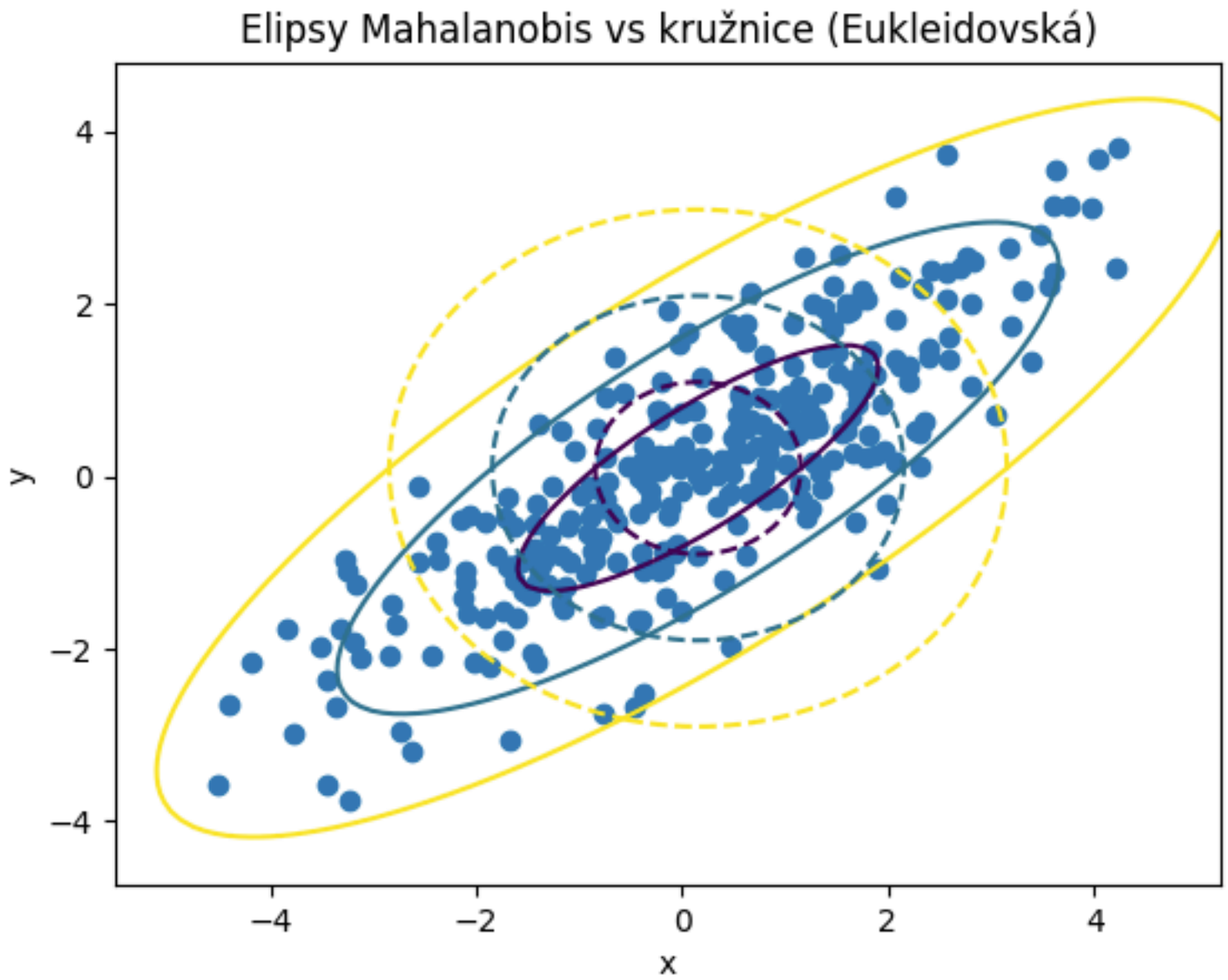
$$z_i^k = \frac{x_i^k - \mu_i}{s_i}, \quad i = 1,2,\dots,m, \;\; k = 1,2,\dots,n$$

Můžeme definovat **Mahalanobisovu vzdálenost**, která měří vzdálenost bodu od středu dat s ohledem na jejich rozptyl a korelace (normalizovaná vzdálenost vzhledem k variabilitě dat).

$$D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})}$$

Mahalanobisova vzdálenost

- Mahalanobisova vzdálenost říká, jak neobvyklý je stav systému vzhledem k tomu, co považujeme za normální.



- Mahalanobis mění kružnice na elipsy, protože realita není izotropní
- Normální není být blízko středu, ale normální je být uvnitř elipsy

Charakter dat

- Více veličin
- Korelace
- Šum
- Redundance

Klíčový problém:

Proměnné nejsou nezávislé

Kovariance ≈ míra společné změny veličin

- Kladná → rostou spolu
- Záporná → opačný trend
- ≈ fyzikální vazba mezi veličinami

$$\Sigma = \begin{pmatrix} s_1^2 & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & s_n^2 \end{pmatrix}$$

Data tvoří elipsoid, každý směr má jinou „šířku“

Vlastní čísla kovarianční matice říkají, kolik variability je v jednotlivých směrech prostoru

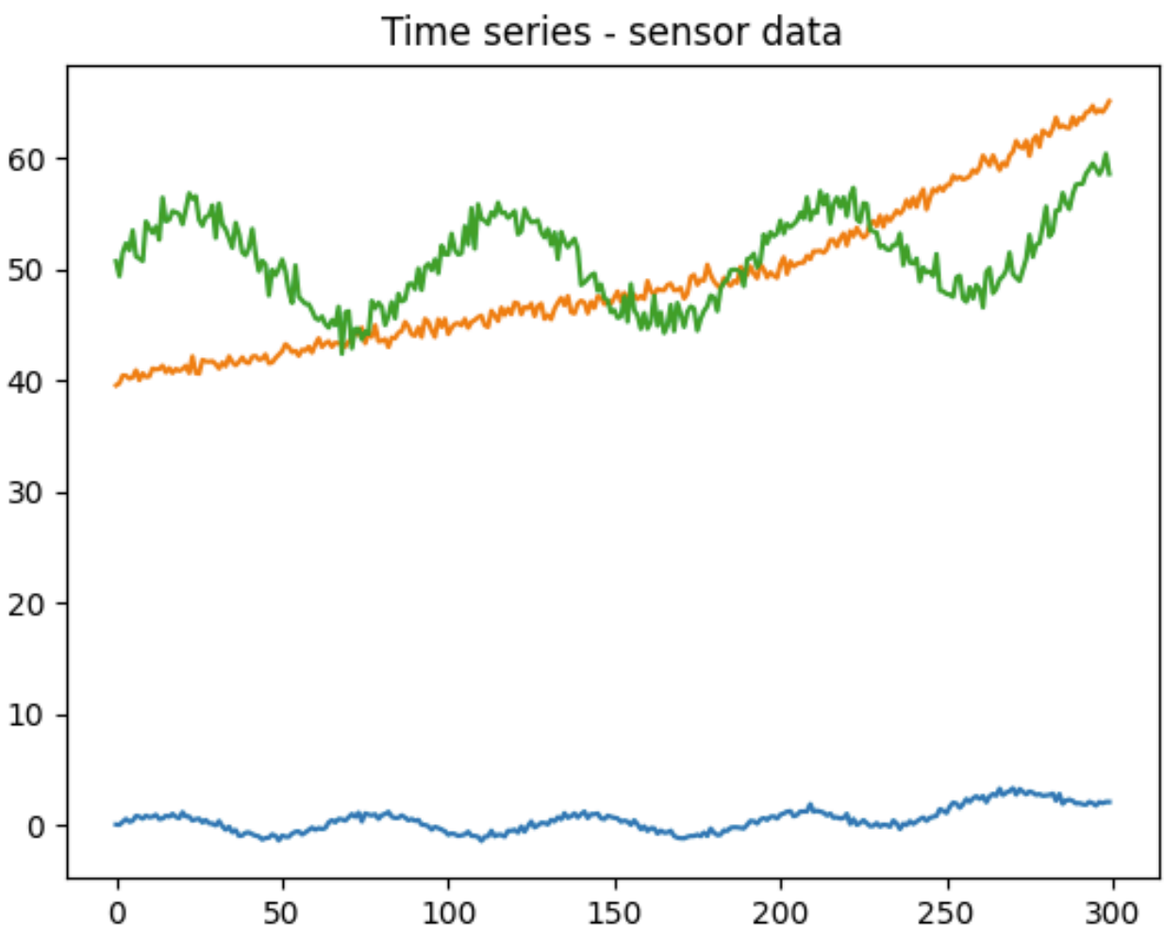
vlastní vektor → směr
vlastní číslo → **rozptyl v tomto směru**

Charakter dat

Příklad: Sledování ložiska v čase

Máme senzorická data ze stroje:

time – čas (index měření)
vibration – vibrace (např. RMS / amplituda)
temperature – teplota ložiska
load – zatížení



dataset_lozisko n = 300

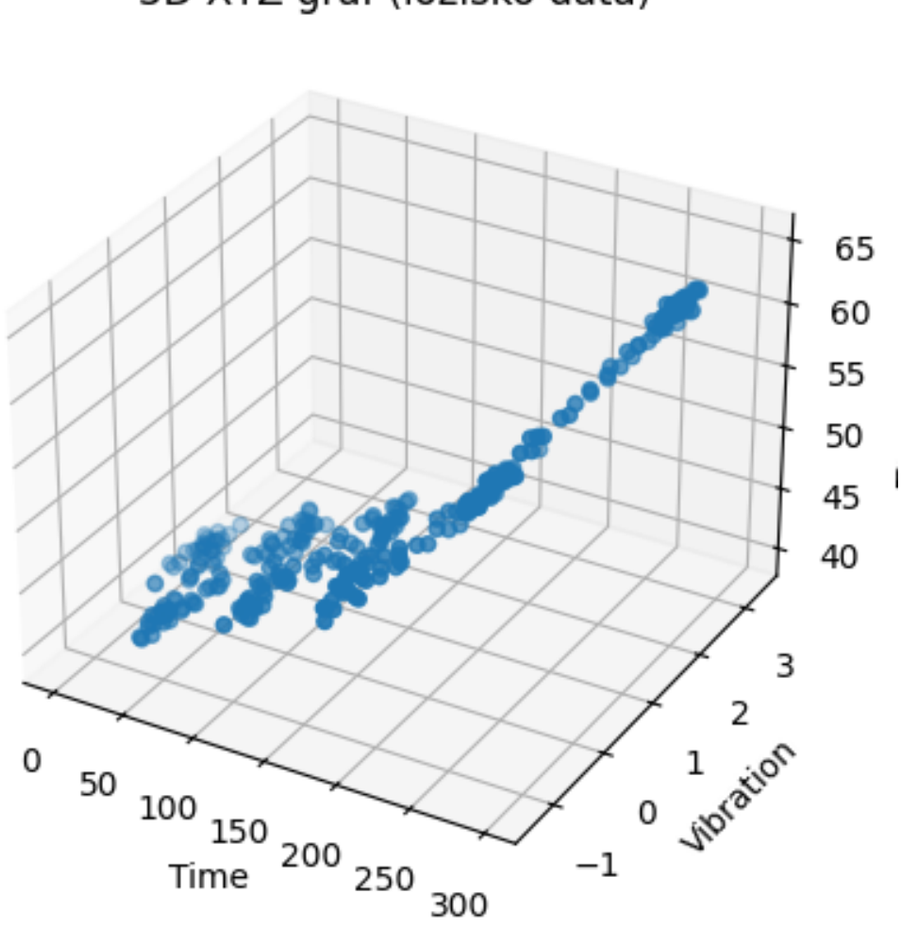
time	vibration	temperature	load
0	0,0993	39,5855	50,7570
1	0,0722	39,7699	49,4109
2	0,3282	40,4736	51,5343
3	0,6001	40,4552	52,3490
4	0,3426	40,1895	51,7310
5	0,4326	40,3087	53,5128
6	0,8805	40,9388	51,1733
7	0,7977	40,0542	51,0049
8	0,6235	40,6735	50,7633
9	0,8918	40,3489	54,3193
10	0,7488	40,3912	53,7462
11	0,7981	41,0994	53,2912
12	0,9804	41,0127	53,8667
13	0,5809	41,0568	52,6854
14	0,6405	41,3527	56,4638
15	0,8850	40,7605	54,3366
16	0,7970	41,1410	54,4874
17	1,0545	40,6949	55,2549
18	0,7922	41,0621	55,1412
19	0,6638	40,8849	54,9944
20	1,2024	41,0485	54,0692
21	0,8181	41,3476	55,3987
22	0,8220	40,6909	56,8549
23	0,4608	42,1962	56,3419
24	0,5666	40,6970	56,5911
25	0,6207	40,6429	54,4658

Charakter dat

Příklad: Sledování ložiska v čase

Máme senzorická data ze stroje:

time – čas (index měření)
vibration – vibrace (např. RMS / amplituda)
temperature – teplota ložiska
load – zatížení



dataset_lozisko n = 300

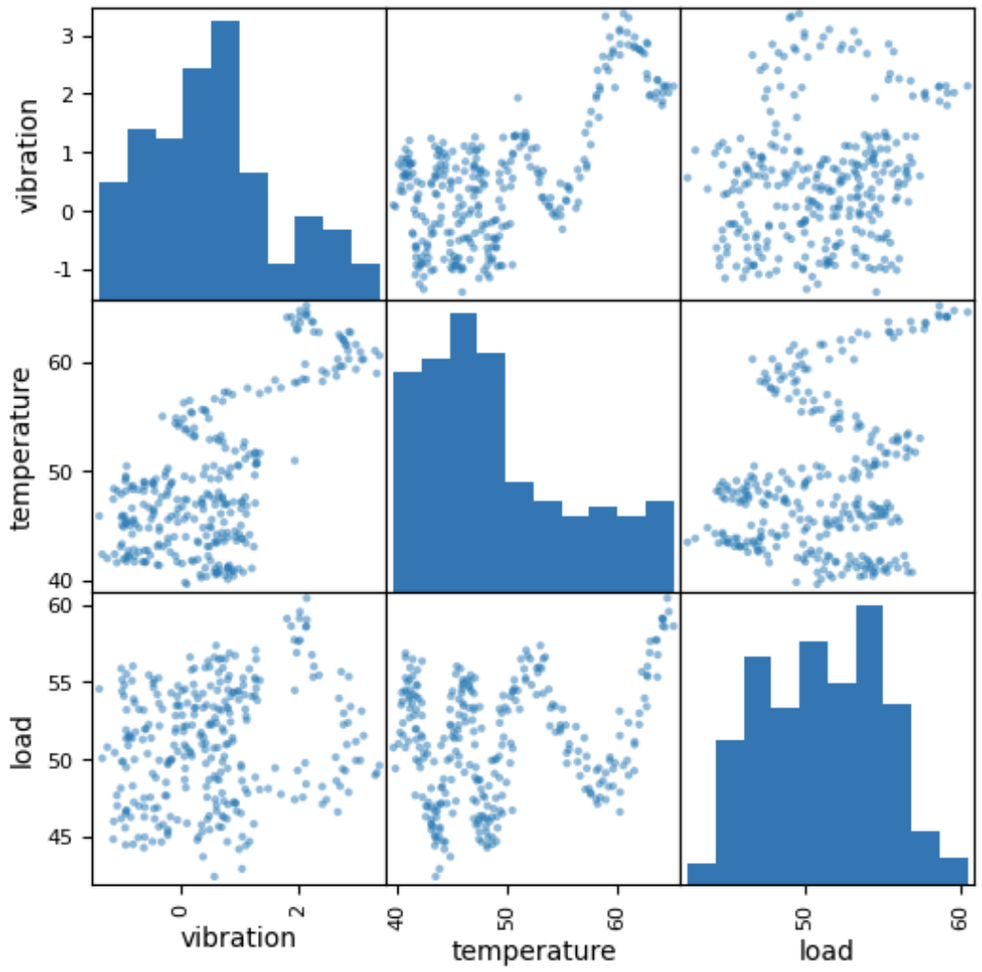
time	vibration	temperature	load
0	0,0993	39,5855	50,7570
1	0,0722	39,7699	49,4109
2	0,3282	40,4736	51,5343
3	0,6001	40,4552	52,3490
4	0,3426	40,1895	51,7310
5	0,4326	40,3087	53,5128
6	0,8805	40,9388	51,1733
7	0,7977	40,0542	51,0049
8	0,6235	40,6735	50,7633
9	0,8918	40,3489	54,3193
10	0,7488	40,3912	53,7462
11	0,7981	41,0994	53,2912
12	0,9804	41,0127	53,8667
13	0,5809	41,0568	52,6854
14	0,6405	41,3527	56,4638
15	0,8850	40,7605	54,3366
16	0,7970	41,1410	54,4874
17	1,0545	40,6949	55,2549
18	0,7922	41,0621	55,1412
19	0,6638	40,8849	54,9944
20	1,2024	41,0485	54,0692
21	0,8181	41,3476	55,3987
22	0,8220	40,6909	56,8549
23	0,4608	42,1962	56,3419
24	0,5666	40,6970	56,5911
25	0,6207	40,6429	54,4658

Charakter dat

Příklad: Sledování ložiska v čase

Máme senzorická data ze stroje:

time – čas (index měření)
vibration – vibrace (např. RMS / amplituda)
temperature – teplota ložiska
load – zatížení



Korelogram - závislosti mezi proměnnými

dataset_lozisko n = 300

time	vibration	temperature	load
0	0,0993	39,5855	50,7570
1	0,0722	39,7699	49,4109
2	0,3282	40,4736	51,5343
3	0,6001	40,4552	52,3490
4	0,3426	40,1895	51,7310
5	0,4326	40,3087	53,5128
6	0,8805	40,9388	51,1733
7	0,7977	40,0542	51,0049
8	0,6235	40,6735	50,7633
9	0,8918	40,3489	54,3193
10	0,7488	40,3912	53,7462
11	0,7981	41,0994	53,2912
12	0,9804	41,0127	53,8667
13	0,5809	41,0568	52,6854
14	0,6405	41,3527	56,4638
15	0,8850	40,7605	54,3366
16	0,7970	41,1410	54,4874
17	1,0545	40,6949	55,2549
18	0,7922	41,0621	55,1412
19	0,6638	40,8849	54,9944
20	1,2024	41,0485	54,0692
21	0,8181	41,3476	55,3987
22	0,8220	40,6909	56,8549
23	0,4608	42,1962	56,3419
24	0,5666	40,6970	56,5911
25	0,6207	40,6429	54,4658

Charakter dat

Příklad: Sledování ložiska v čase

Máme senzorická data ze stroje:

time – čas (index měření)
vibration – vibrace (např. RMS / amplituda)
temperature – teplota ložiska
load – zatížení

Kovarianční matice

	vibration	temperature	load
vibration	1.20	5.07	0.78
temperature	5.07	46.59	6.09
load	0.78	6.09	14.54

Korelační matice

	vibration	temperature	load
vibration	1.00	0.68	0.19
temperature	0.68	1.00	0.23
load	0.19	0.23	1.00

dataset_lozisko n = 300

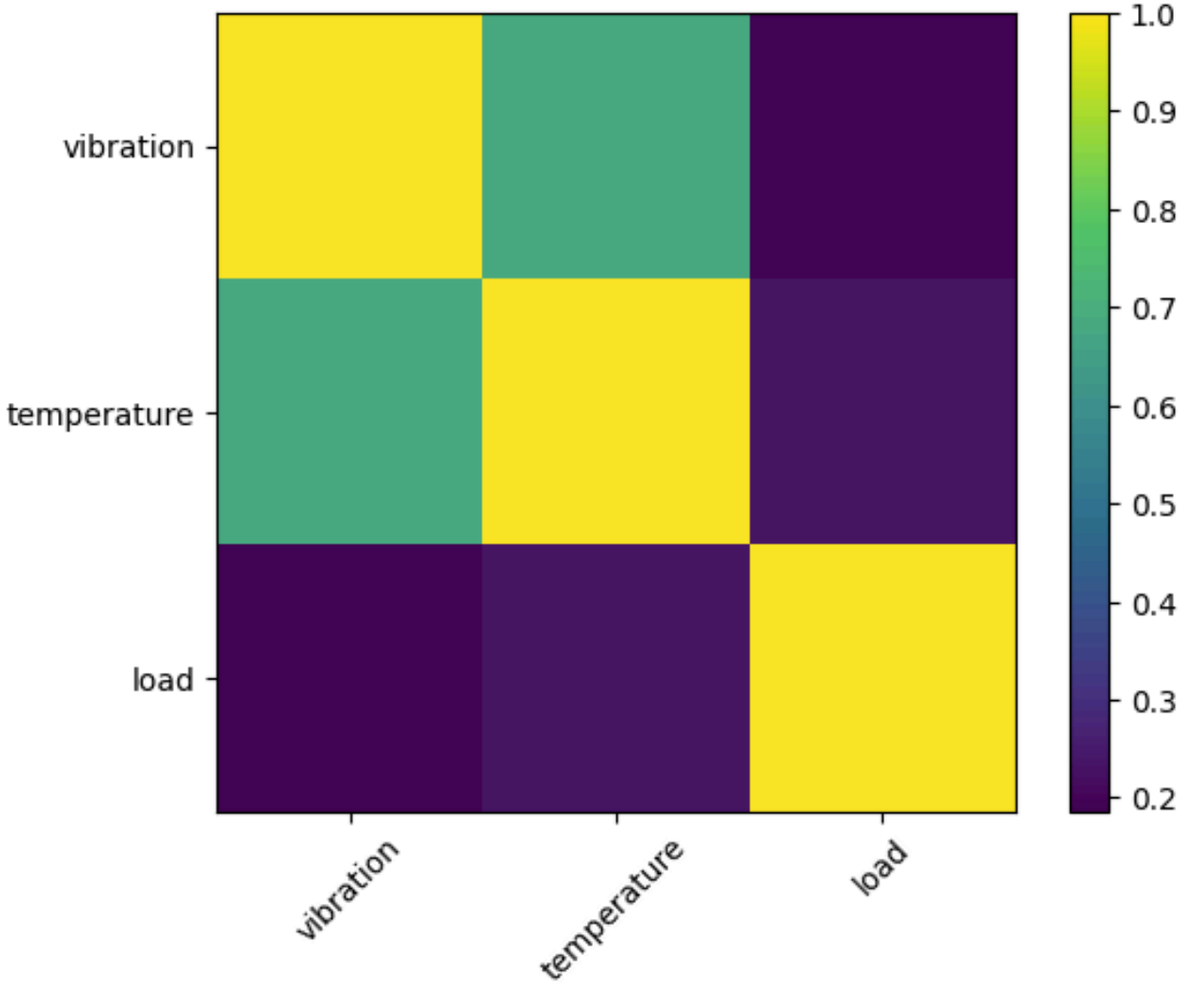
time	vibration	temperature	load
0	0,0993	39,5855	50,7570
1	0,0722	39,7699	49,4109
2	0,3282	40,4736	51,5343
3	0,6001	40,4552	52,3490
4	0,3426	40,1895	51,7310
5	0,4326	40,3087	53,5128
6	0,8805	40,9388	51,1733
7	0,7977	40,0542	51,0049
8	0,6235	40,6735	50,7633
9	0,8918	40,3489	54,3193
10	0,7488	40,3912	53,7462
11	0,7981	41,0994	53,2912
12	0,9804	41,0127	53,8667
13	0,5809	41,0568	52,6854
14	0,6405	41,3527	56,4638
15	0,8850	40,7605	54,3366
16	0,7970	41,1410	54,4874
17	1,0545	40,6949	55,2549
18	0,7922	41,0621	55,1412
19	0,6638	40,8849	54,9944
20	1,2024	41,0485	54,0692
21	0,8181	41,3476	55,3987
22	0,8220	40,6909	56,8549
23	0,4608	42,1962	56,3419
24	0,5666	40,6970	56,5911
25	0,6207	40,6429	54,4658

Charakter dat

Příklad: Sledování ložiska v čase

Máme senzorická data ze stroje:

time – čas (index měření)
vibration – vibrace (např. RMS / amplituda)
temperature – teplota ložiska
load – zatížení



Korelační matice - (multikolinearita)

dataset_lozisko n = 300

time	vibration	temperature	load
0	0,0993	39,5855	50,7570
1	0,0722	39,7699	49,4109
2	0,3282	40,4736	51,5343
3	0,6001	40,4552	52,3490
4	0,3426	40,1895	51,7310
5	0,4326	40,3087	53,5128
6	0,8805	40,9388	51,1733
7	0,7977	40,0542	51,0049
8	0,6235	40,6735	50,7633
9	0,8918	40,3489	54,3193
10	0,7488	40,3912	53,7462
11	0,7981	41,0994	53,2912
12	0,9804	41,0127	53,8667
13	0,5809	41,0568	52,6854
14	0,6405	41,3527	56,4638
15	0,8850	40,7605	54,3366
16	0,7970	41,1410	54,4874
17	1,0545	40,6949	55,2549
18	0,7922	41,0621	55,1412
19	0,6638	40,8849	54,9944
20	1,2024	41,0485	54,0692
21	0,8181	41,3476	55,3987
22	0,8220	40,6909	56,8549
23	0,4608	42,1962	56,3419
24	0,5666	40,6970	56,5911
25	0,6207	40,6429	54,4658

Principal Component Analysis, PCA

Příklad: **Sledování ložiska v čase**

Korelační matice

	vibration	**temperature**	**load**
vibration	1.00	0.68	0.19
temperature	0.68	1.00	0.23
load	0.19	0.23	1.00

Vlastní čísla

λ_1	≈	**48.27**
λ_2	≈	13.42
λ_3	≈	0.64

Vlastní vektory

PC1 ≈ [0.11	**0.98**	**0.18]**
PC2 ≈ [-0.01	-0.18	0.98]
PC3 ≈ [0.99	-0.11	-0.01]

Interpretace:

PC 1 (hlavní komponenta):

- téměř celá váha je na **teplotě**
- trochu přispívá zatížení
- vibrace skoro ne

PC1 ≈ „teplotní režim systému“

Principal Component Analysis, PCA

Příklad: Sledování ložiska v čase

Korelační matice

	vibration	**temperature**	**load**
vibration	1.00	0.68	0.19
temperature	0.68	1.00	0.23
load	0.19	0.23	1.00

Vlastní čísla

λ_1	≈	**48.27**
λ_2	≈	13.42
λ_3	≈	0.64

Vlastní vektory

PC1 ≈ [0.11	**0.98**	**0.18]**
PC2 ≈ [-0.01	-0.18	0.98]
PC3 ≈ [0.99	-0.11	-0.01]

Interpretace:

PC 1 (hlavní komponenta):

- téměř celá váha je na **teplotě**
- trochu přispívá zatížení
- vibrace skoro ne

PC1 ≈ „teplotní režim systému“

PC 2:

- dominuje **zatížení**
- slabý vliv teploty

PC2 ≈ „zatěžovací režim“

Principal Component Analysis, PCA

Příklad: Sledování ložiska v čase

Korelační matice

	vibration	**temperature**	**load**
vibration	1.00	0.68	0.19
temperature	0.68	1.00	0.23
load	0.19	0.23	1.00

Vlastní čísla

λ_1	≈	**48.27**
λ_2	≈	13.42
λ_3	≈	0.64

Vlastní vektory

PC1 ≈ [0.11	**0.98**	**0.18]**
PC2 ≈ [-0.01	-0.18	0.98]
PC3 ≈ [0.99	-0.11	-0.01]

Matematika nám říká, že většina chování systému je dána teplotou a zatížením a vibrace jsou až třetí v pořadí.

Interpretace:

PC 1 (hlavní komponenta):

- téměř celá váha je na **teplotě**
- trochu přispívá zatížení
- vibrace skoro ne

PC1 ≈ „teplotní režim systému“

PC 2:

- dominuje **zatížení**
- slabý vliv teploty

PC2 ≈ „zatěžovací režim“

PC 3:

- hlavně **vibrace,** málo důležité

PC3 ≈ „zbytková variabilita (šum)“

Principal Component Analysis, PCA

Příklad: Sledování ložiska v čase

Korelační matice

	vibration	**temperature**	**load**
vibration	1.00	0.68	0.19
temperature	0.68	1.00	0.23
load	0.19	0.23	1.00

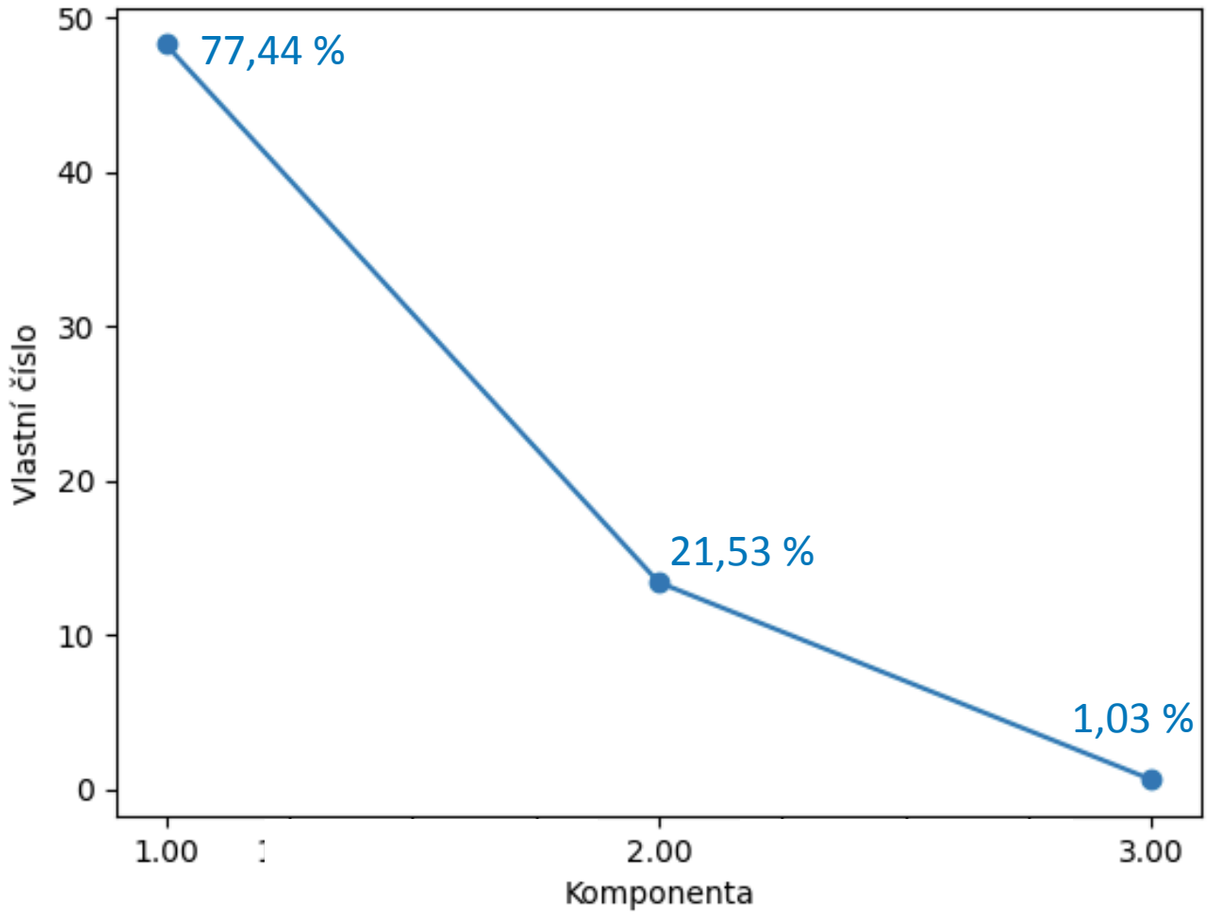
Vlastní čísla

λ_1	≈	**48.27**
λ_2	≈	13.42
λ_3	≈	0.64

Vlastní vektory

PC1 ≈ [0.11	**0.98**	**0.18]**
PC2 ≈ [-0.01	-0.18	0.98]
PC3 ≈ [0.99	-0.11	-0.01]

Scree plot říká, kolik dimenzí skutečně potřebujeme.



PC1 + PC2 ≈ ~99 % informace
PC3 ≈ zanedbatelná

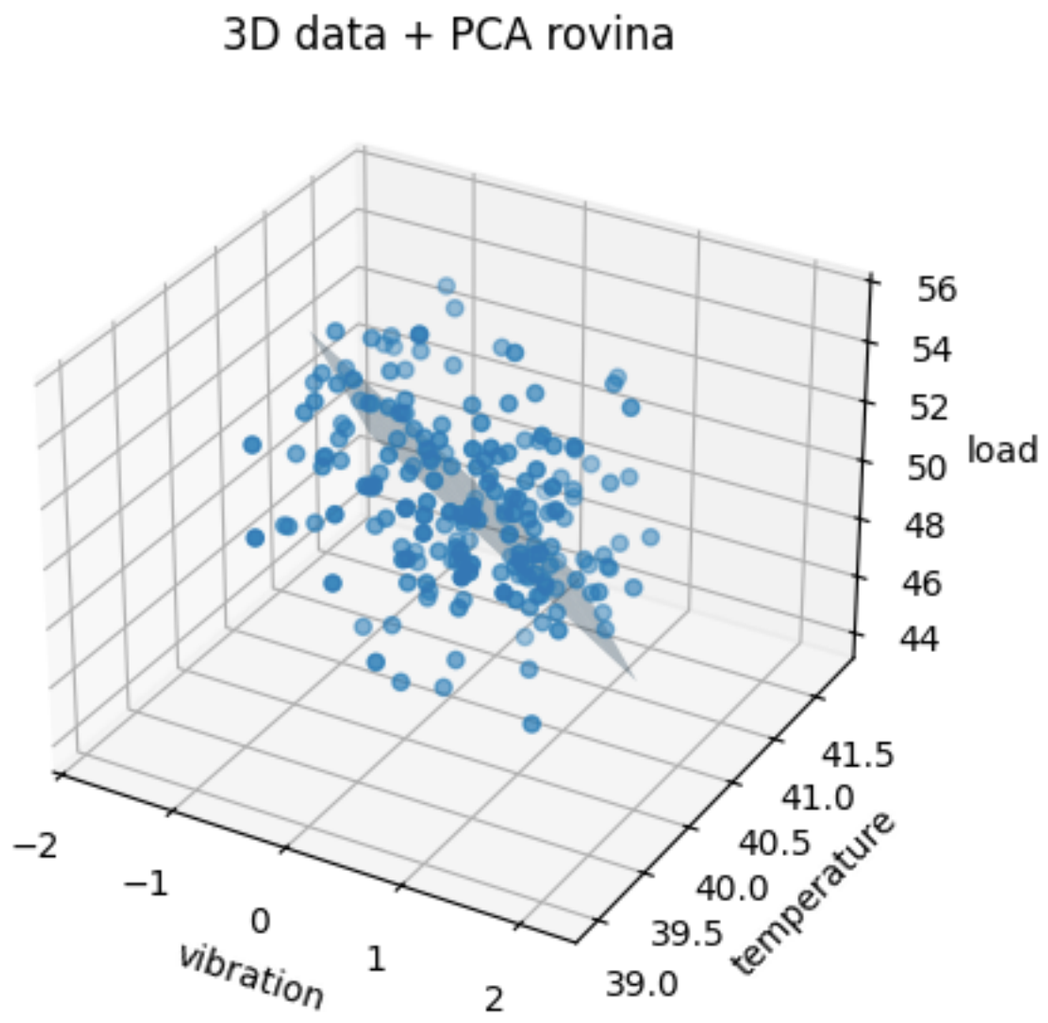
explained variance: $\frac{\lambda_i}{\sum \lambda}$

Systém je prakticky dvourozměrný

Principal Component Analysis, PCA

- PCA redukuje dimenzi při zachování podstatné části variability
- V našem případě redukuje 3D data do 2D roviny. Tato rovina co nejlépe vystihuje data - hlavní směry variability

- PCA není deformace dat – je to jen otočení pohledu.
  - PCA = změna souřadnic
  - nic se „nepočítá navíc“, jen se přeuspořádá informace
- umožní:
    - lepší vizualizaci
    - jednodušší modely



Principal Component Analysis, PCA

- PCA redukuje dimenzi při zachování podstatné části variability
- V našem případě redukuje 3D data do 2D roviny. Tato rovina co nejlépe vystihuje data - hlavní směry variability

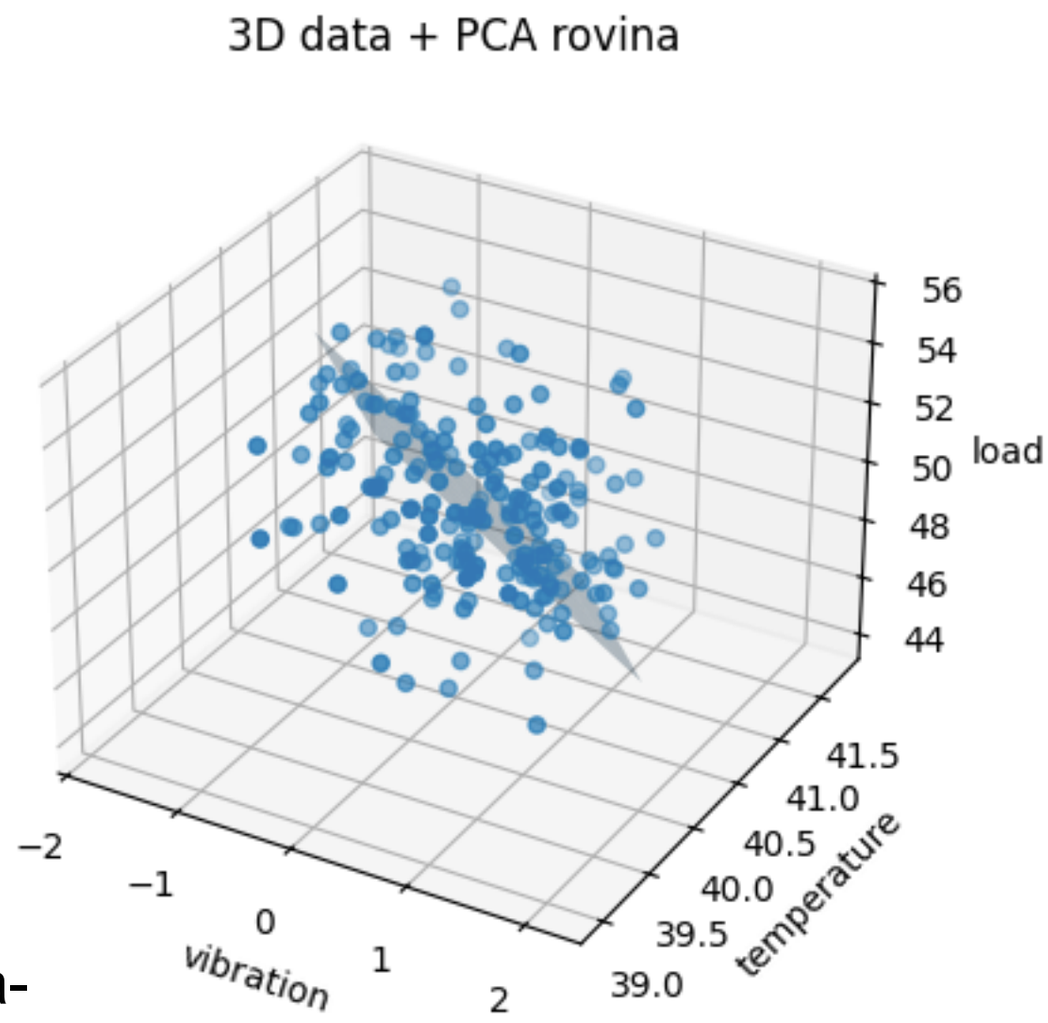
Jak to probíhá?

$\mathbf{X}$ = matice ($m \times n$) - m řádků dpovídajících pozorování n veličin

$\mathbf{W}$ = matice ($k \times n$) - řádky tvoří prvních k vlastních vektorů kovarianční matice

V našem příkladu je:

$$\mathbf{W} = \begin{pmatrix} 0{,}11 & 0{,}98 & 0{,}18 \\ -0{,}01 & -0{,}18 & 0{,}98 \end{pmatrix}$$



$\mathbf{X^{(k)}} = \mathbf{X} \,.\, \mathbf{W}^{\mathsf{T}}$ = redukovaná datová matice ($m \times k$) - má pouze k sloupců odpovídajících k hlavním komponentám

V našem příkladu dostaneme datovou matici o 300 řádcích (pozorování) a dvou sloupcích (PC1 a PC2).

Principal Component Analysis, PCA

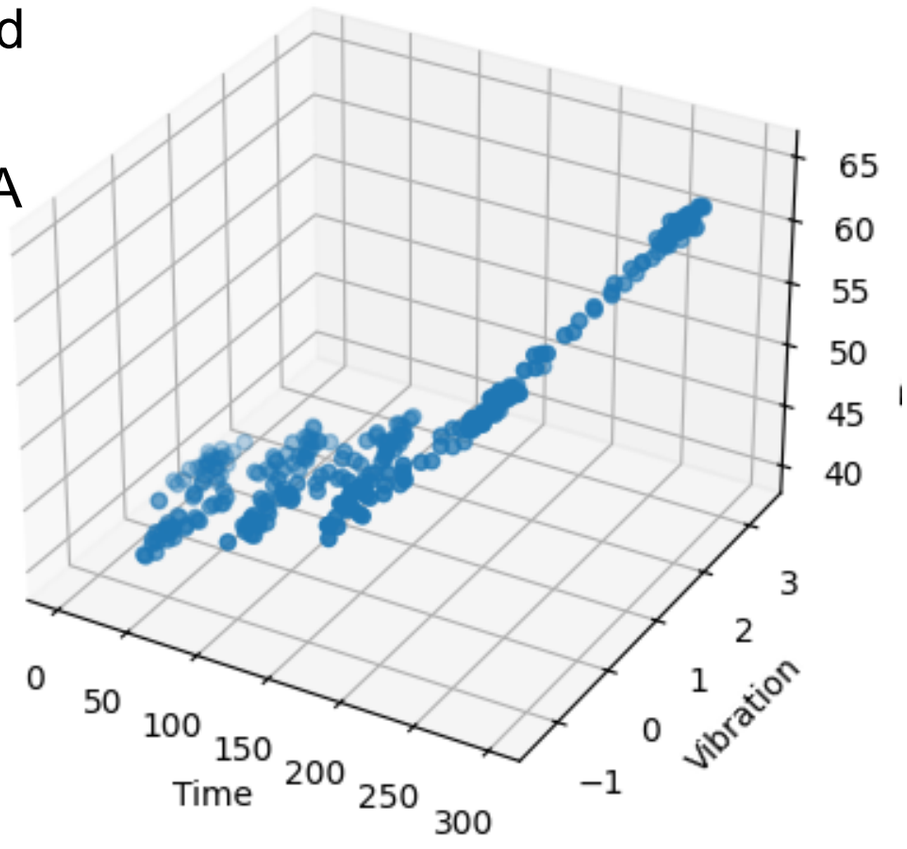
Příklad: Sledování ložiska v čase

Máme senzorická data ze stroje:

time – čas (index měření)
vibration – vibrace (např. RMS / amplituda)
temperature – teplota ložiska
load – zatížení

- Na první pohled nic nevidíme.
- Použijeme PCA
- Detekujeme vznik poruchy

3D XYZ graf (ložisko data)



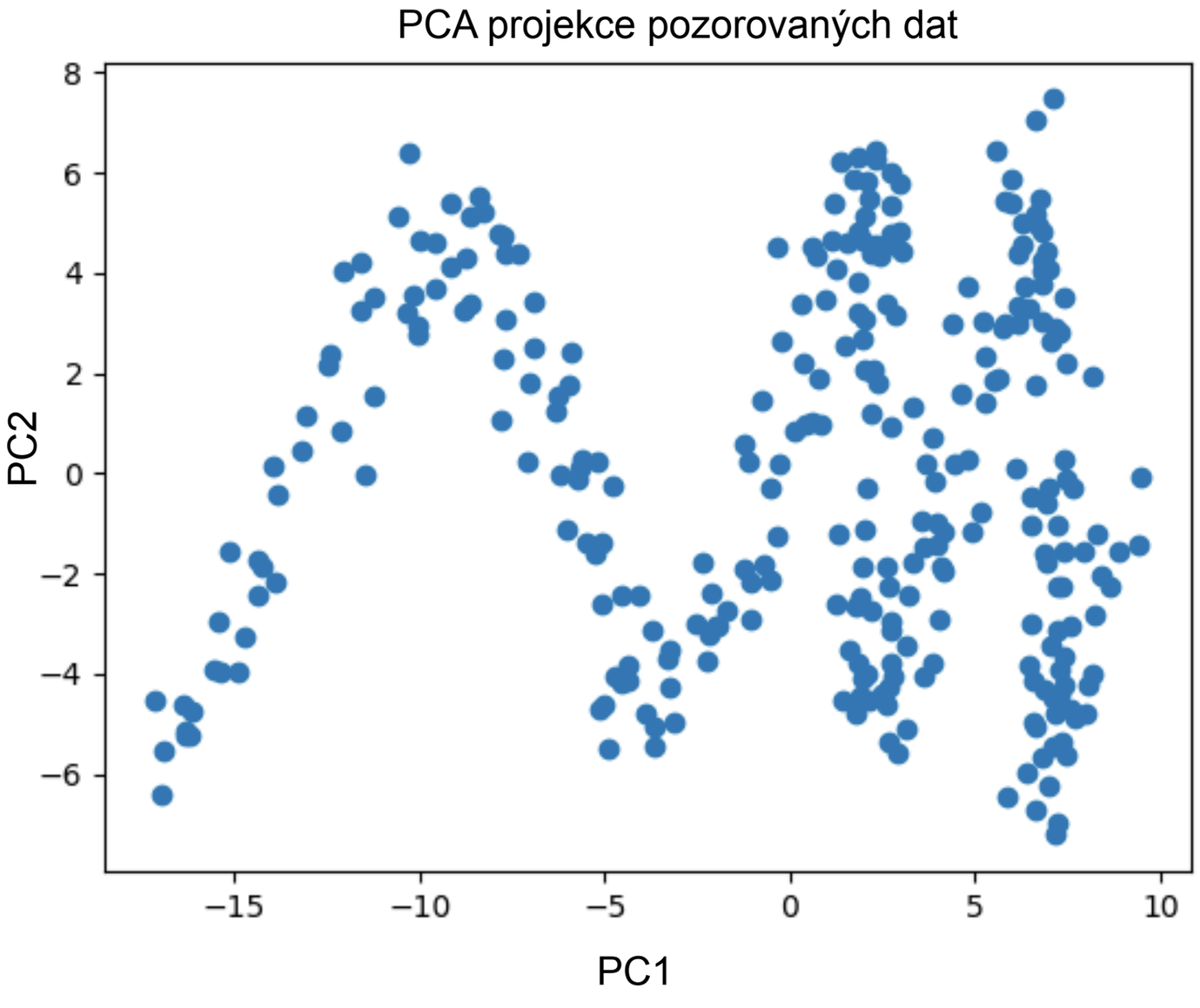
dataset_lozisko

time	vibration	temperature	load
0	0,0993	39,5855	50,7570
1	0,0722	39,7699	49,4109
2	0,3282	40,4736	51,5343
3	0,6001	40,4552	52,3490
4	0,3426	40,1895	51,7310
5	0,4326	40,3087	53,5128
6	0,8805	40,9388	51,1733
7	0,7977	40,0542	51,0049
8	0,6235	40,6735	50,7633
9	0,8918	40,3489	54,3193
10	0,7488	40,3912	53,7462
11	0,7981	41,0994	53,2912
12	0,9804	41,0127	53,8667
13	0,5809	41,0568	52,6854
14	0,6405	41,3527	56,4638
15	0,8850	40,7605	54,3366
16	0,7970	41,1410	54,4874
17	1,0545	40,6949	55,2549
18	0,7922	41,0621	55,1412
19	0,6638	40,8849	54,9944
20	1,2024	41,0485	54,0692
21	0,8181	41,3476	55,3987
22	0,8220	40,6909	56,8549
23	0,4608	42,1962	56,3419
24	0,5666	40,6970	56,5911
25	0,6207	40,6429	54,4658

Principal Component Analysis, PCA

Redukce dimenze pomocí PCA: každé měření → jeden bod

nové osy = kombinace původních veličin



kompaktní oblast → normální stav

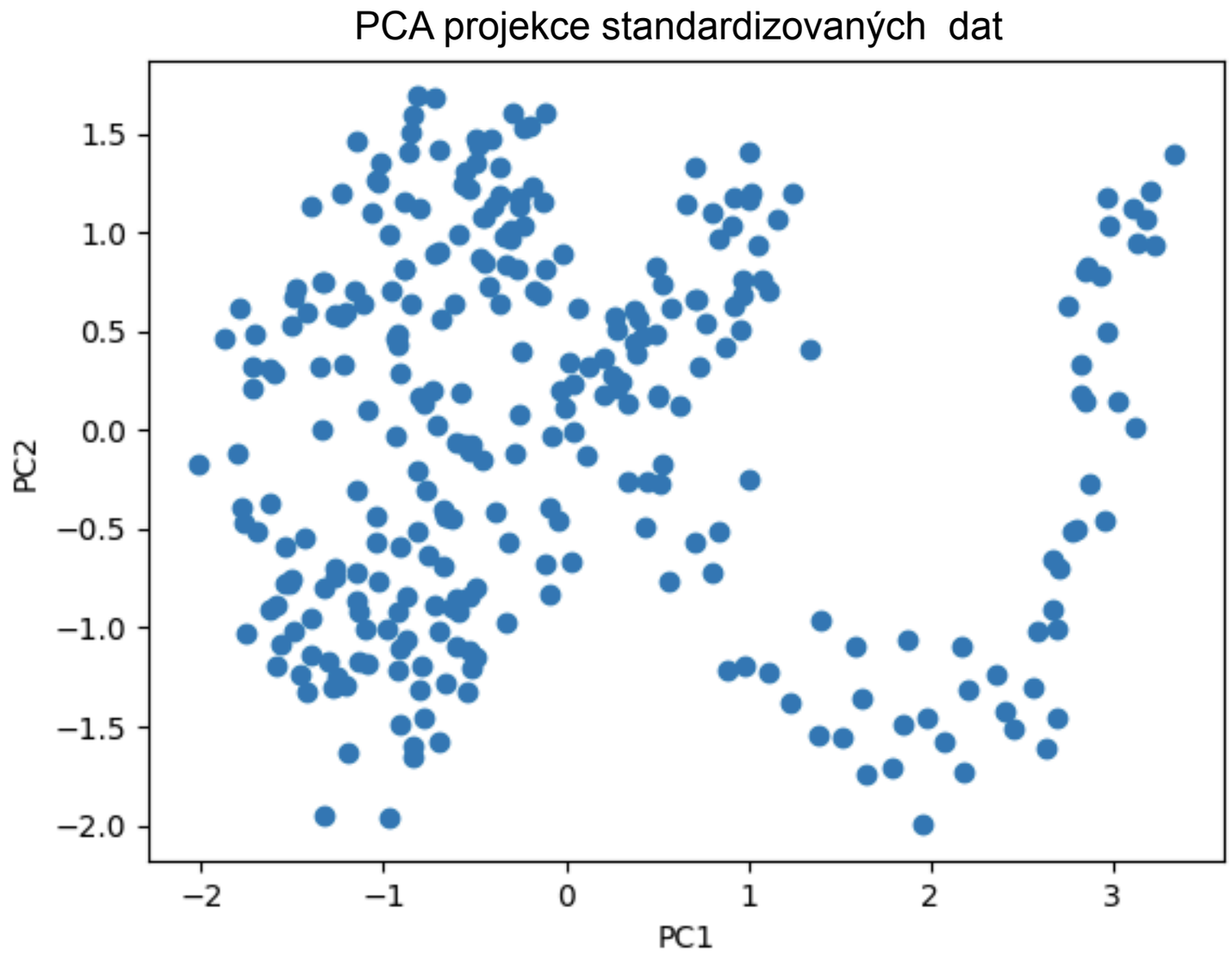
roztažení → změna chování systému

PCA nám odhalí strukturu, která v původních datech není vidět.

Principal Component Analysis, PCA

Redukce dimenze pomocí PCA: každé měření → jeden bod

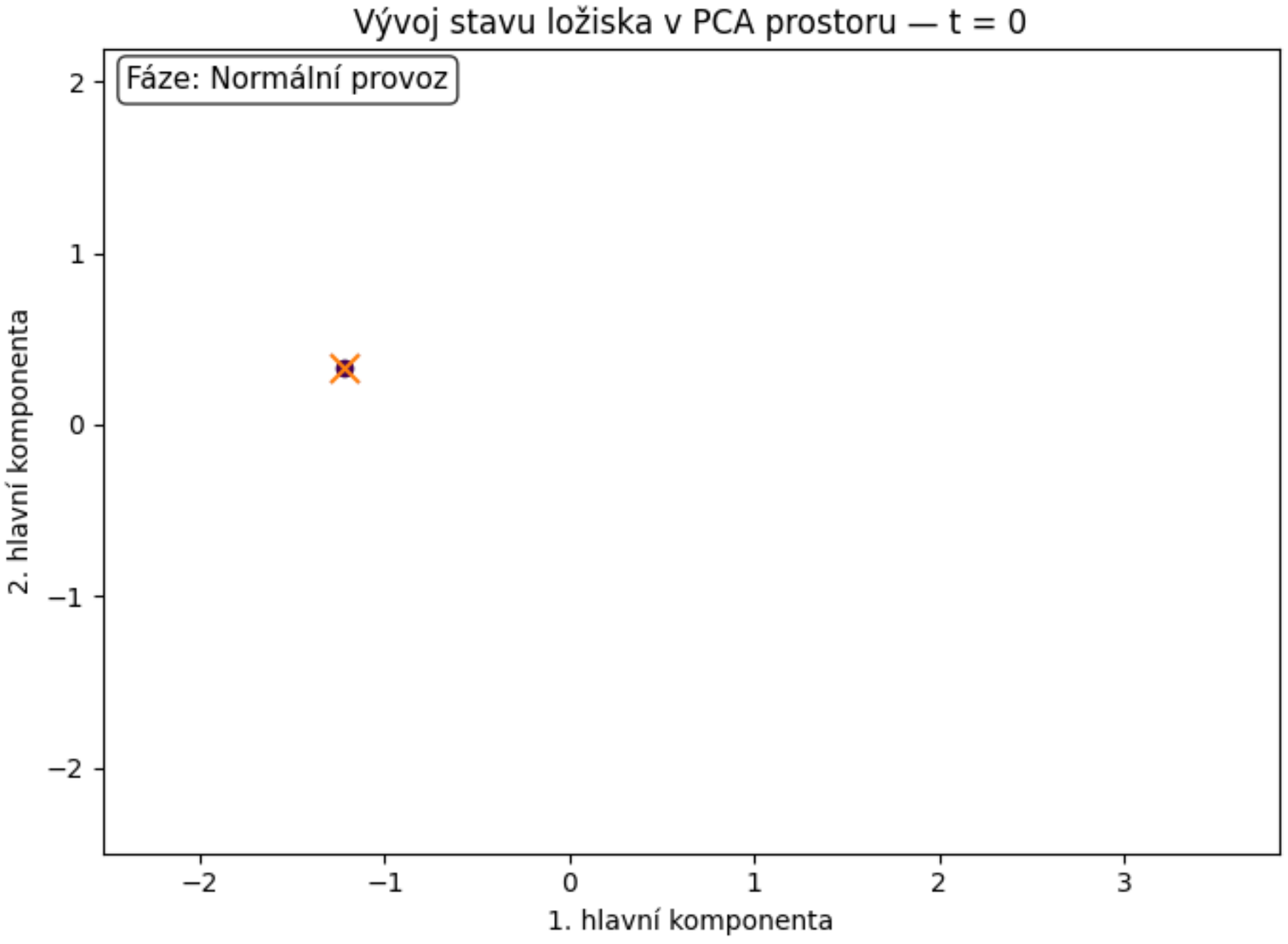
nové osy = kombinace původních veličin



PCA se zpravidla provádí na standardizovaných datech:

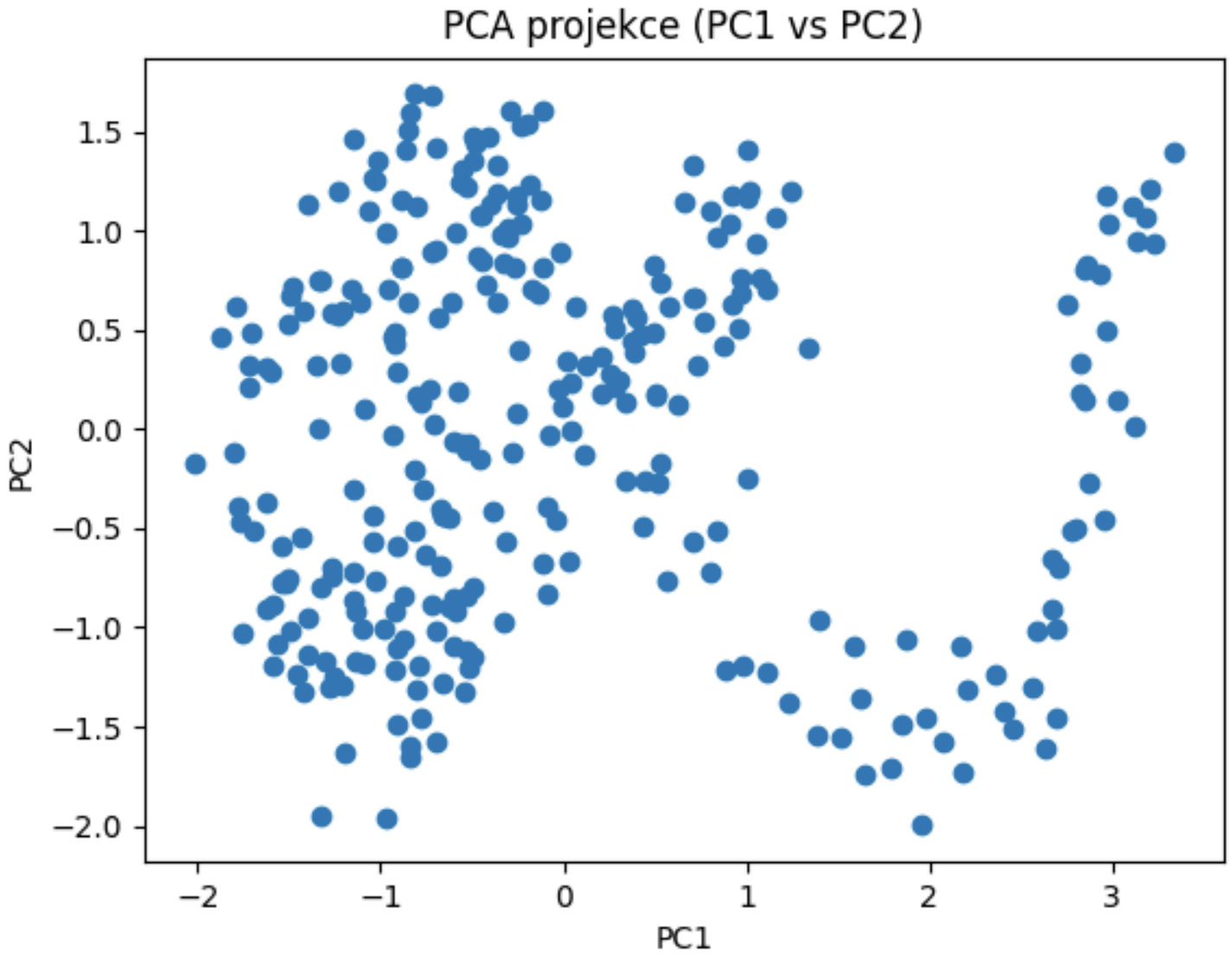
$$Z_t = \frac{X_t - \mu_X}{\sigma_X}$$

PCA nám odhalí strukturu, která v původních datech není vidět.


Vývoj stavu ložiska v PCA prostoru — t = 0
Fáze: Normální provoz
2
1
0
−1
−2
2. hlavní komponenta
−2
−1
0
1
2
3
1. hlavní komponenta

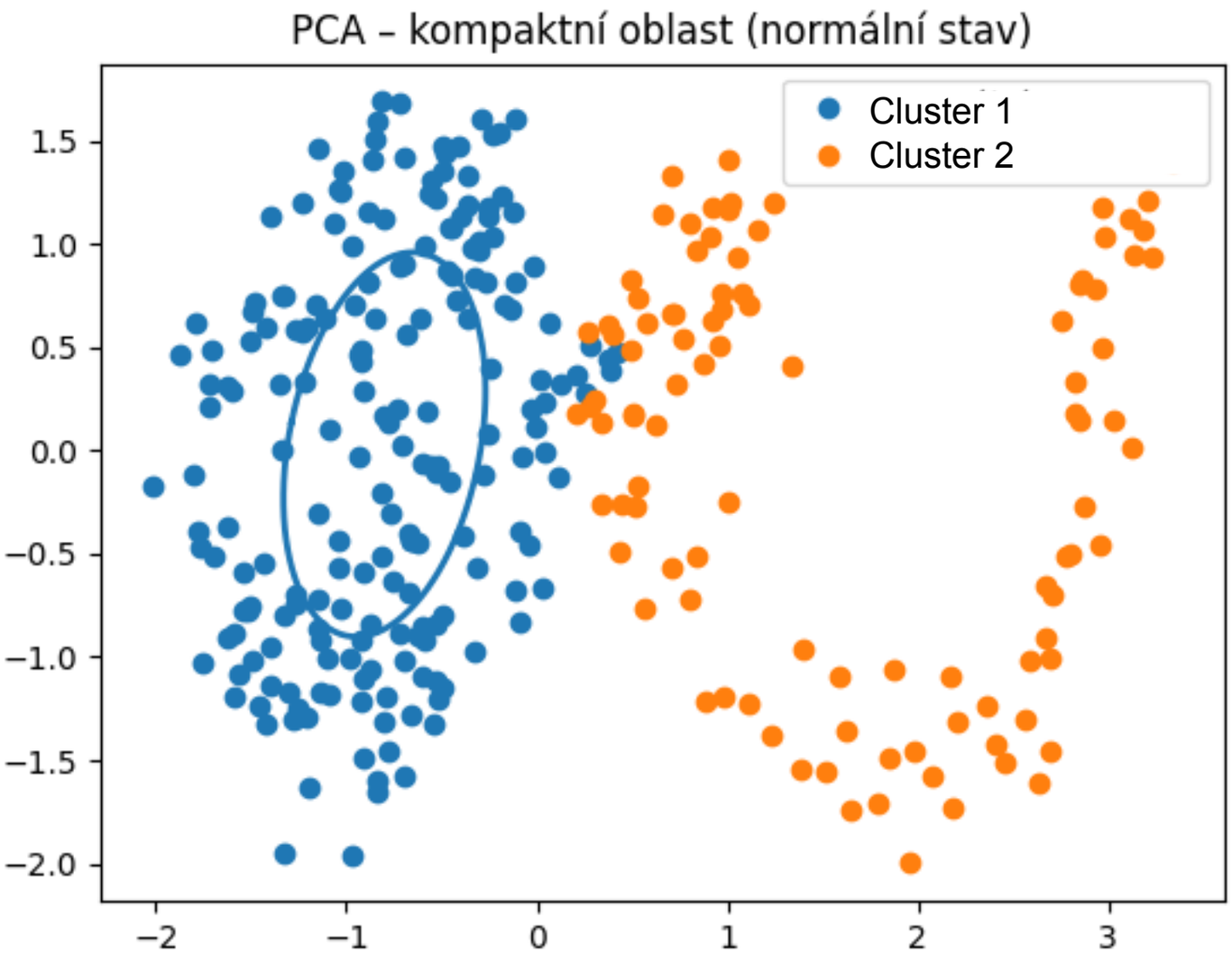
Clustering (shlukování)

- Hledání režimů provozu
- Normální vs. porucha
- Segmentace dat



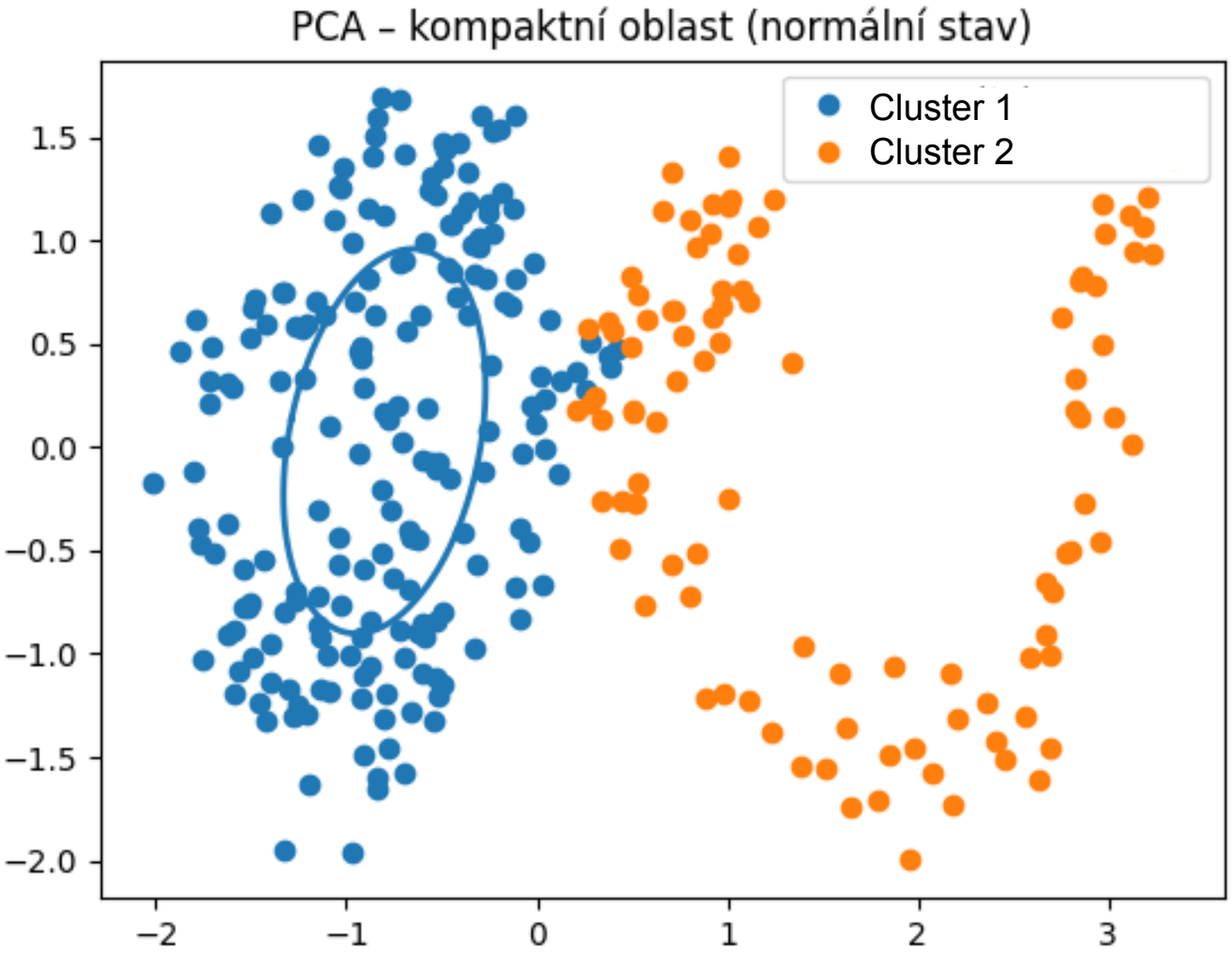
Clustering (shlukování)

- Hledání režimů provozu
- Normální vs. porucha
- Segmentace dat



Clustering (shlukování)

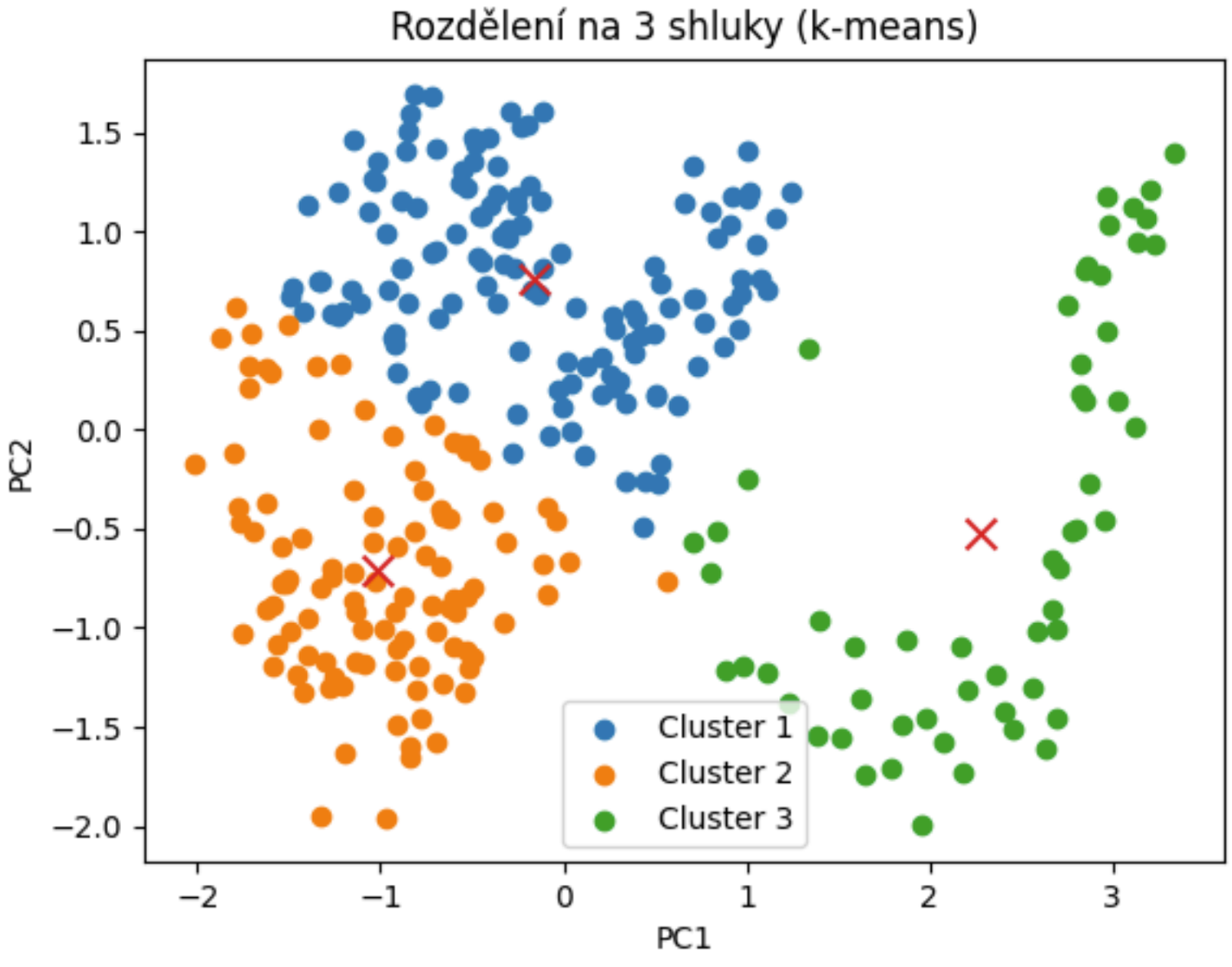
Metoda k-means: Minimalizace vzdálenosti ke centroidům $\sum_{i=1}^{n} ||x_i - \mu_{c_i}||^2$



1. zvolíme počet shluků (např. k = 2 nebo 3)
2. náhodné centroidy
3. přiřazení bodů
4. aktualizace

Clustering (shlukování)

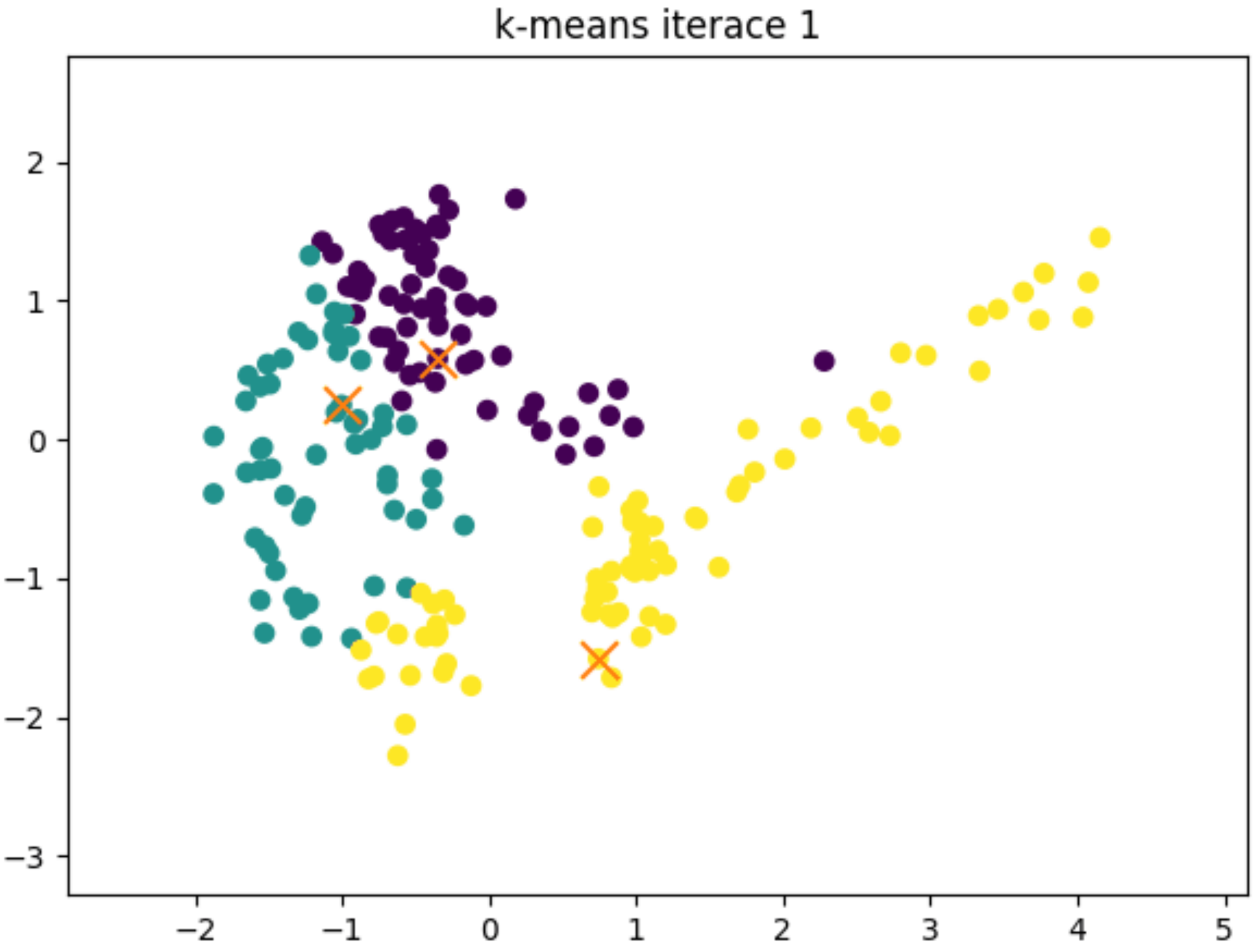
Metoda k-means: Minimalizace vzdálenosti ke centroidům $\sum_{i=1}^{n} ||x_i - \mu_{c_i}||^2$



1. zvolíme počet shluků (např. k = 2 nebo 3)
2. náhodné centroidy
3. přiřazení bodů
4. aktualizace

Shlukování nám rozdělí data na podobné skupiny. Neposkytuje nám však jejich interpretaci

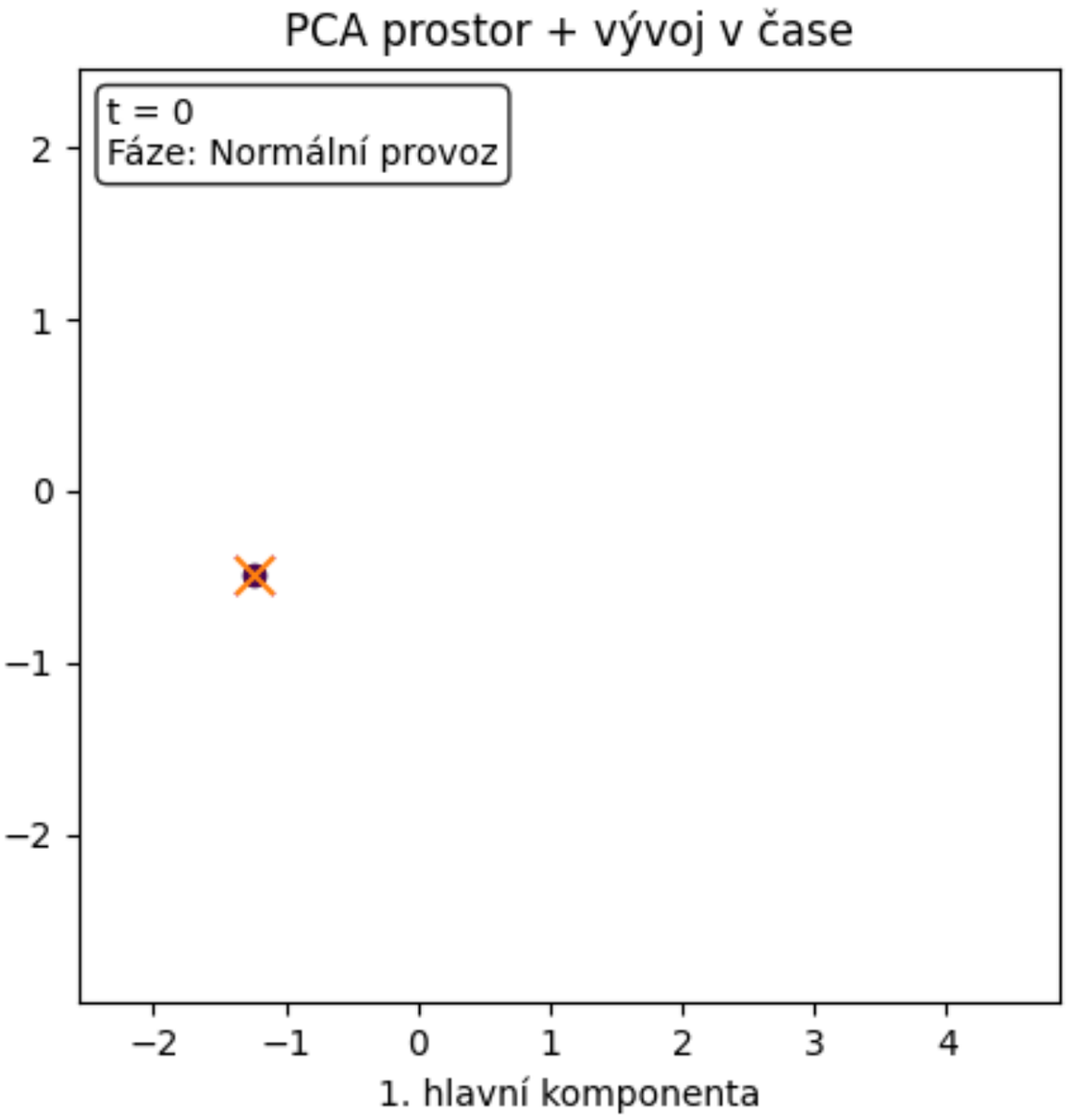
Clustering (shlukování)

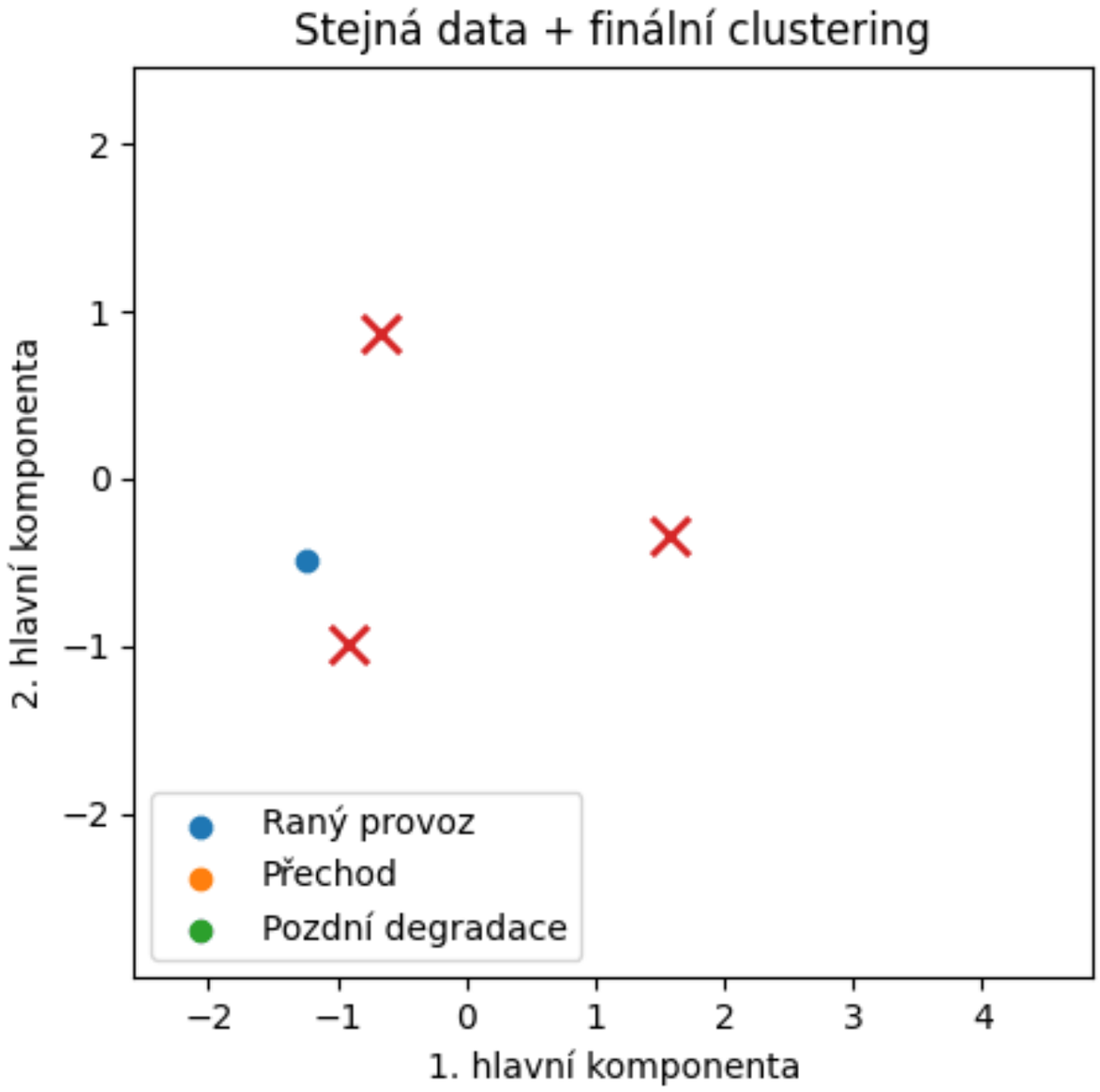


Clustering (shlukování)

Příklad: **Sledování ložiska v čase**

Shlukování nám rozdělí data na podobné skupiny.
Neposkytuje nám však jejich interpretaci

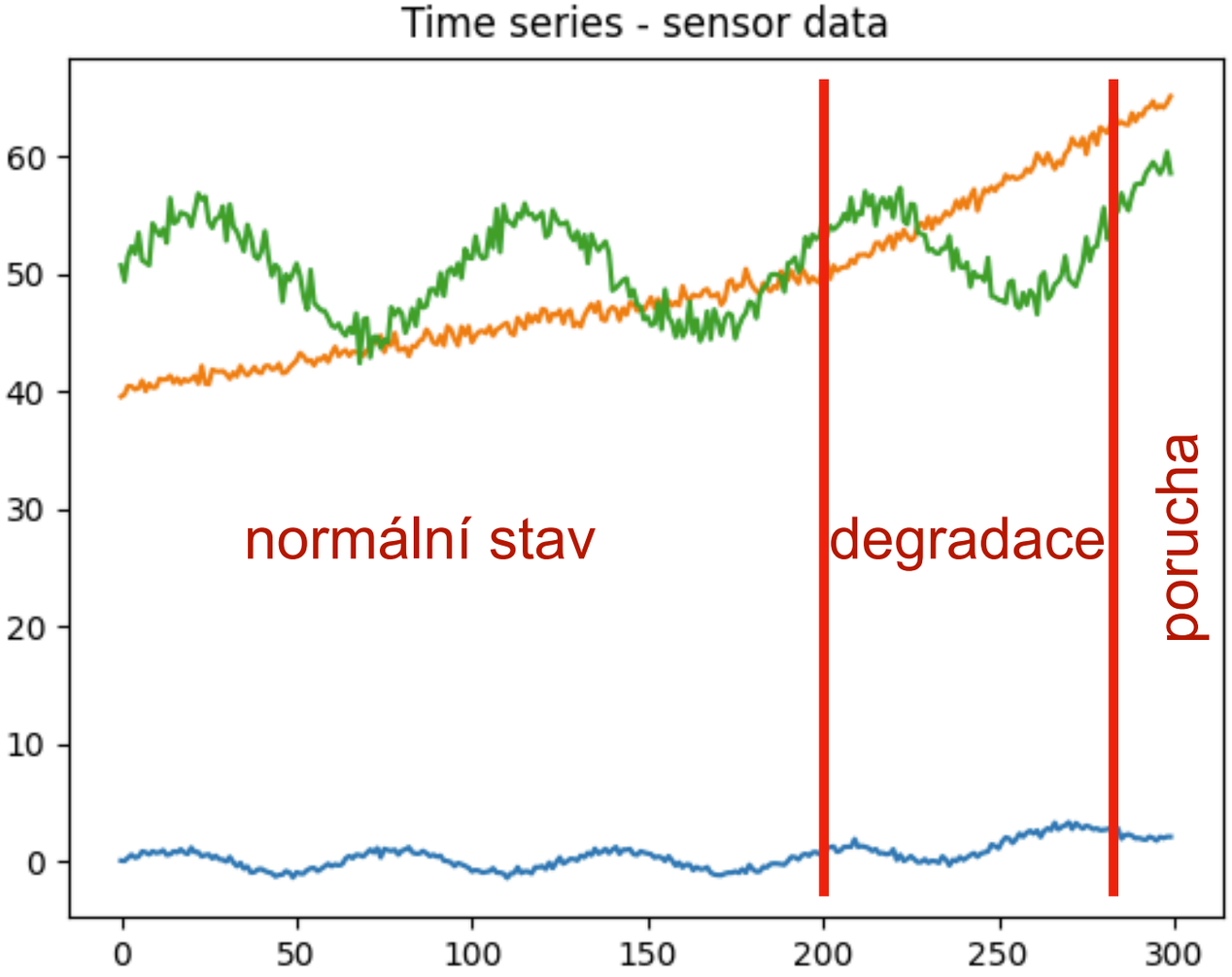




Clustering (shlukování)

Příklad: Sledování ložiska v čase

time – čas (index měření)
vibration – vibrace (např. RMS / amplituda)
temperature – teplota ložiska
load – zatížení



vidíme vibrace, teplotu a zatížení
na první pohled:
- data jsou hlučná
- není jasné, kdy vzniká problém

systém se postupně mění
vzniká degradace → porucha

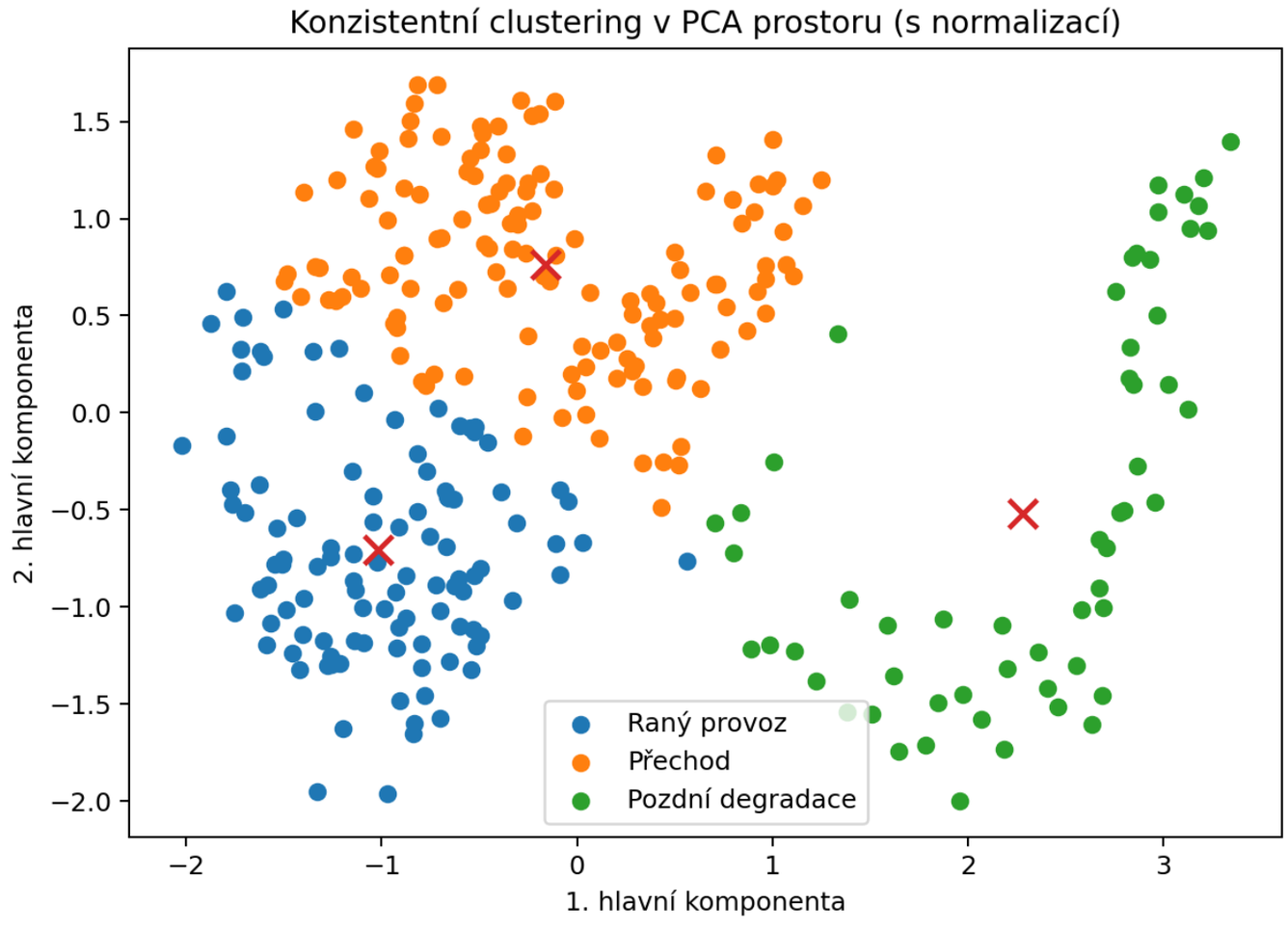
Z jedné veličiny poruchu spolehlivě nepoznáme.

Porucha nevzniká skokem – je to proces, který můžeme pomocí PCA velmi brzy odhalit.

Clustering (shlukování)

Příklad: Sledování ložiska v čase

time – čas (index měření)
vibration – vibrace (např. RMS / amplituda)
temperature – teplota ložiska
load – zatížení



Shlukování nám rozdělí data na podobné skupiny. Neposkytuje nám však jejich interpretaci:

Cluster 1 – normální stav: kompaktní oblast, stabilní chování

Cluster 2 – přechod (degradace): body mezi shluky, systém se mění

Cluster 3 – porucha: oddělený region, výrazně jiné chování

Faktorová analýza

Předpokládáme, že data jsou generována malým počtem skrytých faktorů + šumem.

Základní model: $$\mathbf{x} = \Lambda \cdot \mathbf{f} + \epsilon$$

kde $\mathbf{x}$ jsou pozorovaná data
$\mathbf{f}$ jsou latentní (skryté) faktory
$\mathbf{\Lambda}$ jsou váhy (loadings)
ε je šum

Variabilita se rozkládá na:

- **Společnou variabilitu (common)** způsobená faktory
- **Specifickou variabilitu (unique)** způsobenou šumem / individuální vlivy

$$\Sigma = \Lambda\Lambda^{\mathsf{T}} + \Psi$$

kde Ψ je šum.

Faktorová analýza

Předpokládáme, že data jsou generována malým počtem skrytých faktorů + šumem.

Základní model: $\mathbf{x} = \Lambda \cdot \mathbf{f} + \epsilon$

kde $\mathbf{x}$ jsou pozorovaná data
$\mathbf{f}$ jsou latentní (skryté) faktory
$\mathbf{\Lambda}$ jsou váhy (loadings)
ε je šum

Výpočet:

1. Inicializace - počáteční odhad $\mathbf{\Psi}$ = diag(ψ_1, …, ψ_n) např. jako ψ_i = Var(x_i)
2. Odhad vah faktorů $\mathbf{\Lambda}$ - jaké $\mathbf{\Lambda}$ nejlépe vysvětlí $\mathbf{\Sigma} - \mathbf{\Psi}$
3. Aktualizace $\mathbf{\Psi}$ (spočte se $\mathbf{\Psi} = \mathbf{\Sigma} - \mathbf{\Lambda}\mathbf{\Lambda}^T$)
4. Iterace - opakuje se aktualizace $\mathbf{\Lambda}$ a aktualizace $\mathbf{\Psi}$.

Jiný přístup:

(ML přístup) hledá se $\max_{\Lambda,\Psi} \log p(\mathbf{X} \mid \Lambda, \Psi)$

Faktorová analýza

V našem příkladu:

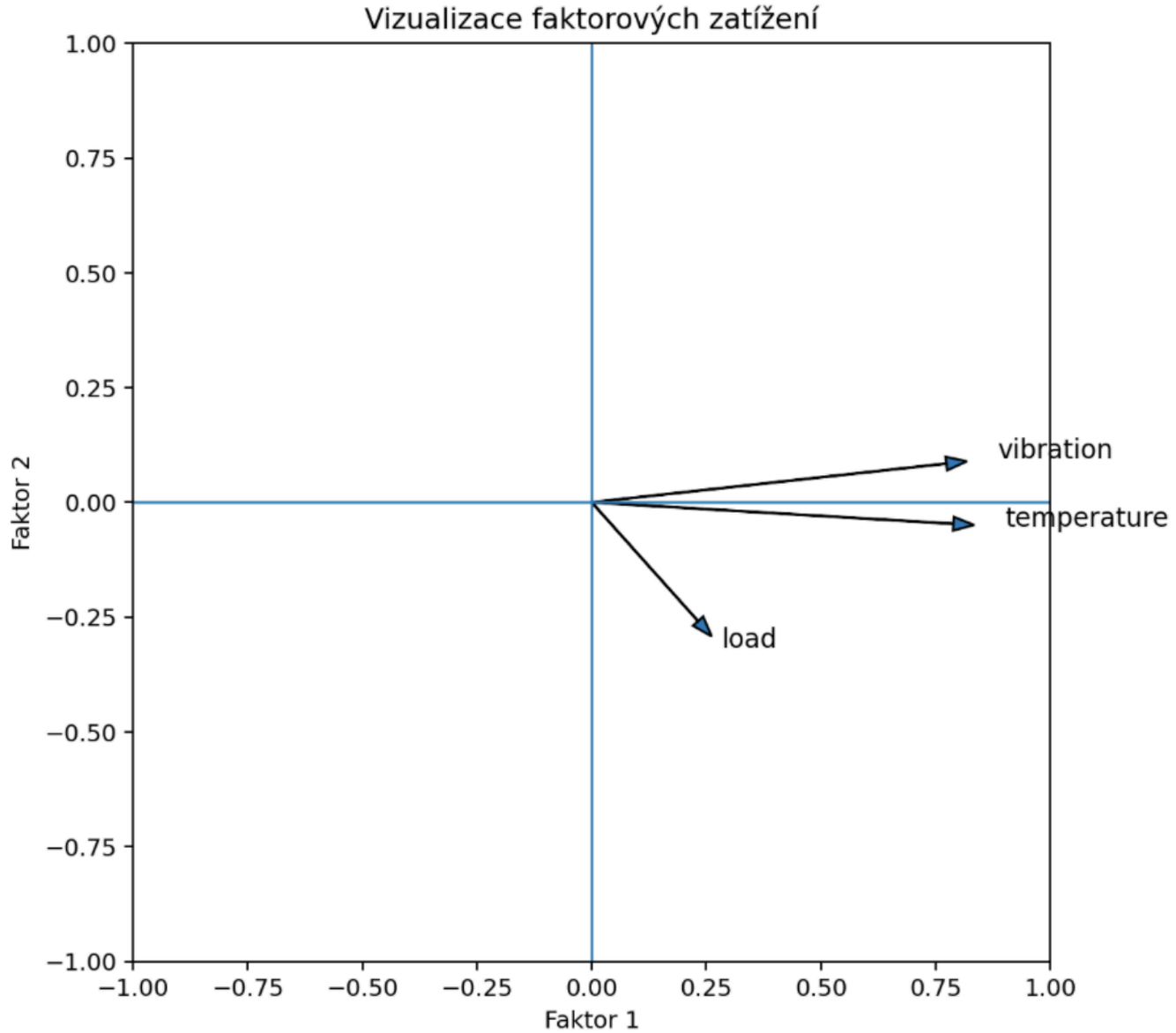
	Faktor	**1**	**Faktor**	**2**
vibration	0.82		0.09	
temperature	0.83		-0.05	
load	0.26		-0.29	

Faktor 1: „hlavní provozní režim“
(silný vliv teploty a vibrací, slabý vliv zatížení)

Faktor 2: „zatížení / sekundární efekt“
(prakticky nulový vliv teploty a vibrací)

Faktorová analýza se snaží vysvětlit, proč spolu veličiny souvisejí. Hledá skryté příčiny, které generují pozorovaná data.

Faktorová analýza nám říká, které veličiny patří k sobě a jsou řízené stejným skrytým vlivem.



PCA vs. FA

PCA popisuje data. Faktorová analýza data vysvětluje.

PCA (popis dat)

👉 „Jak data vypadají“

PC1 ≈ hlavní variabilita (teplota)
PC2 ≈ sekundární variabilita (zatížení)
redukce: **3D → 2D**

👉

PCA = změna souřadnic

Faktorová analýza (vysvětlení dat)

👉 „Proč data tak vypadají“

Faktor 1 → společný vliv (teplota + vibrace)
Faktor 2 → specifický vliv (zatížení)

👉

FA = skryté příčiny

PCA	Faktorová analýza
hledá směry variability	hledá skryté příčiny
čistě matematická	statistický model
žádný šum	explicitní šum
komponenty = kombinace dat	faktory = „skutečné vlivy“

Regrese

Jak vstupy ovlivňují výstup?

Příklad (jak využít PCA k odstranění multikolinearity)

y = měřená veličina (výstup, odezva), závisející na hodnotách měření x_1, x_2 ze dvou čidel
(například opotřebení ložiska, amplituda vibrací, účinnost systému, riziko poruchy apod.)

x_1, x_2 = vstupní veličiny ze dvou čidel, měřících teplotu. Jejich hodnoty budou silně korelované.

Model: $y = \alpha + \beta x_1 + \gamma x_2 + \varepsilon$

Odhady parametrů regrese: $\alpha = 0{,}36$; $\beta = 2{,}32$; $\gamma = -0{,}33$

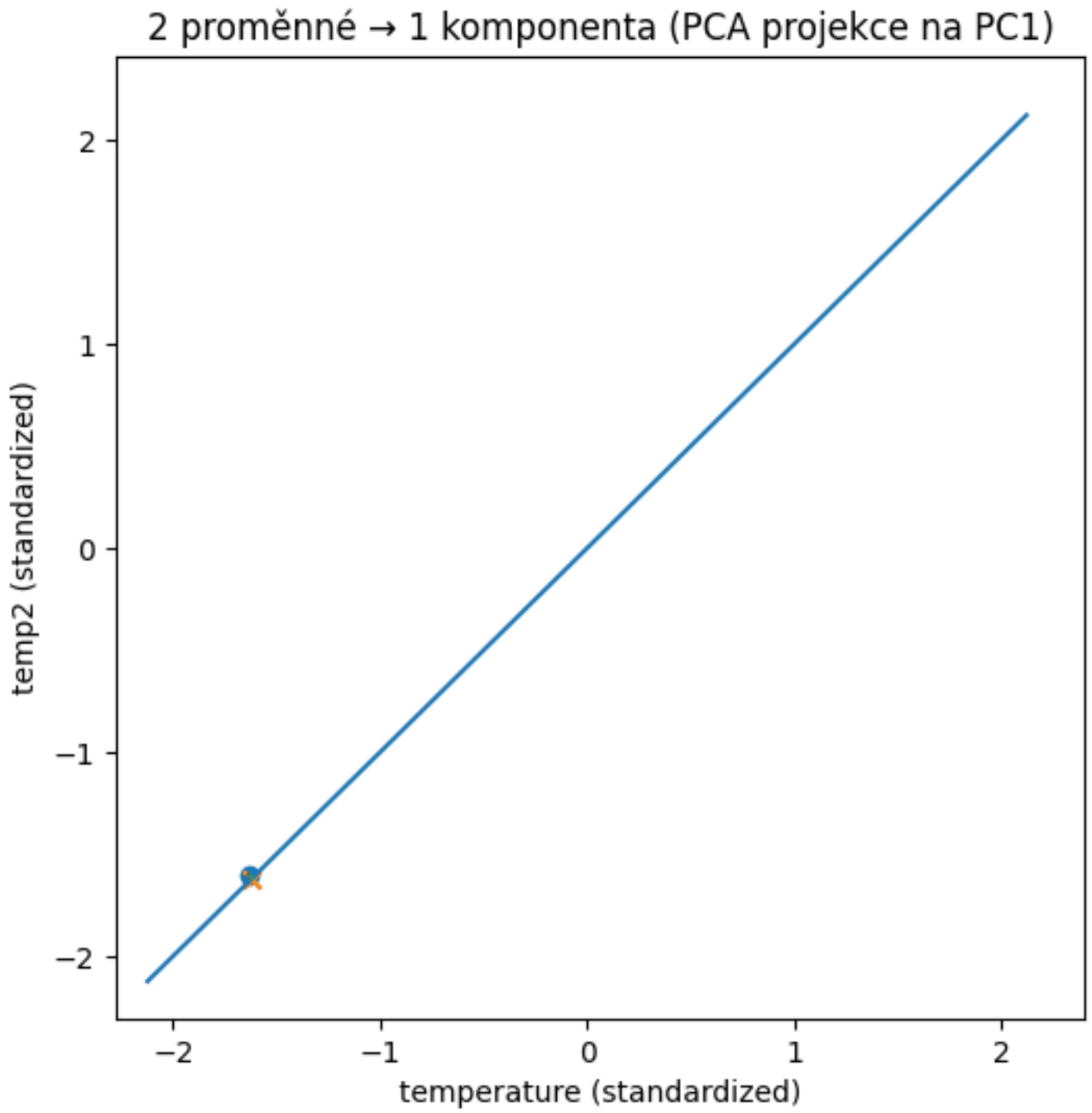
$y = 0{,}36 + 2{,}32x_1 - 0{,}33x_2$ <- to je nesmysl!

Regrese s jednou proměnnou: $y = 0{,}36 + 1{,}99\,x + \varepsilon$

Model ví, že teplota ovlivňuje výstup, ale protože máme dvě téměř stejné proměnné, neví, kterou použít – a začne ‚blbnout'. To je důsledek multikolinearity.

Regrese

Příklad:



Regrese

Jak vstupy ovlivňují výstup?

Příklad (jak využít PCA k odstranění multikolinearity)

y = měřená veličina (výstup, odezva), závisející na hodnotách měření x_1, x_2 ze dvou čidel (například opotřebení ložiska, amplituda vibrací, účinnost systému, riziko poruchy apod.)

x_1, x_2 = vstupní veličiny ze dvou čidel, měřících teplotu. Jejich hodnoty budou silně korelované (korelace blízká 1).

Použijeme PCA a x_1, x_2 nahradíme PC1: $y = \alpha + \beta PC1 + \varepsilon$

Odhady parametrů regrese: $\alpha = 89{,}82$; $\beta = 4{,}16$

$y = 89{,}82 + 4{,}16PC1$

PCA odstranila redundanci, vytvořila ortogonální proměnné a regrese pak funguje správně. PCA odstraní multikolinearitu tím, že sloučí redundantní informaci.