

**MUNI**  
FI

# **Correlation and matplotlib.animation**

Katarína Švecová

# Correlation

- Dependence
- Statistical relationship between two random variables
- Linear correlation (Pearsons correlation coefficient)

# Variance, Covariance

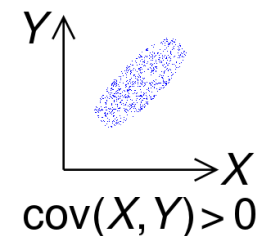
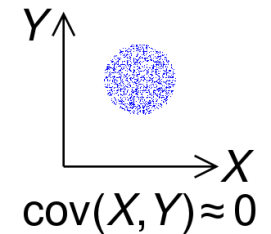
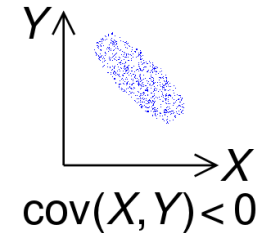
$$\text{Var}(X) = E[(X - \mu)^2].$$

$$\text{cov}(X, Y) = E[(X - E[X])(Y - E[Y])]$$

$$\text{cov}(X, Y) = \frac{1}{n} \sum_{i=1}^n (x_i - E(X))(y_i - E(Y)).$$

$$\text{Var}(X) = \text{Cov}(X, X).$$

- E is expected value (average)
- $\mu$  too
- $\text{corr}(X, Y) = \text{cov}(X, Y) / \sqrt{(\text{var}(X) * \text{var}(Y))}$

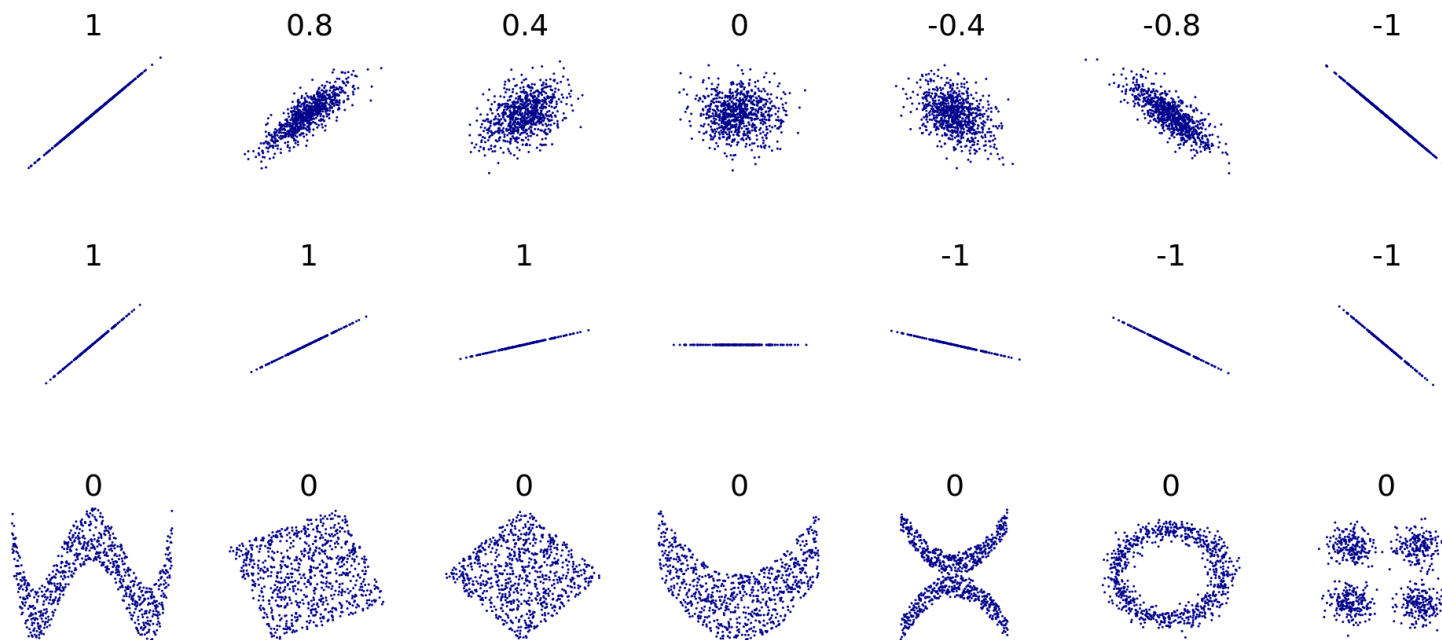


# Pearsons correlation coefficient

$$\text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{\text{E}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

- $\sigma = \sqrt{\text{var}}$  (rozptyl =  $\sqrt{\text{šstandardnej odchýlky}}$ )
- Linear correlation of numerical variables

# Pearsons correlation coefficient illustration



# Other correlation measures

- Spearman correlation coefficient
  - Pearson coefficient between rank variables
  - Assesses monotonic correlation

- Kendall rank correlation coefficient

$$\tau = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{\binom{n}{2}}$$

- Point biserial for one binary variable

# scipy.stats

- Big library with a lot of statistical functions
- For correlation:
  - `scipy.stats.pearsonr(x, y)`
  - `scipy.stats.spearmanr(x, y)`
  - `scipy.stats.kendalltau(x, y)`

# Matplotlib

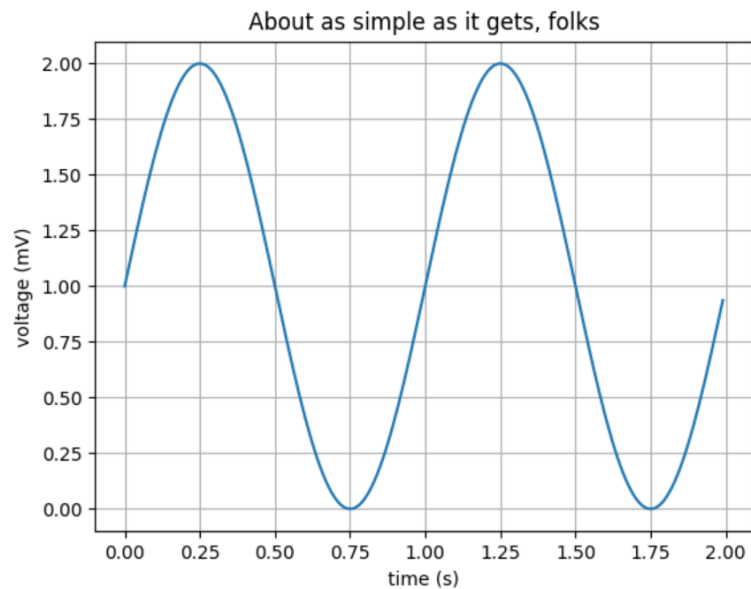
- A python library for plots
- Simple plot, subplots, images, heatmaps, histograms, 3d...
- Animations
- Examples:

[https://matplotlib.org/stable/tutorials/introductory/sample\\_plots.html](https://matplotlib.org/stable/tutorials/introductory/sample_plots.html)



# Matplotlib

## – Simple example



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

# Data for plotting
t = np.arange(0.0, 2.0, 0.01)
s = 1 + np.sin(2 * np.pi * t)

fig, ax = plt.subplots()
ax.plot(t, s)

ax.set(xlabel='time (s)', ylabel='voltage (mV)',
       title='About as simple as it gets, folks')
ax.grid()

fig.savefig("test.png")
plt.show()
```

# Matplotlib.animation

- Live animations
- Example from

[https://matplotlib.org/2.0.2/examples/animation/simple\\_anim.html](https://matplotlib.org/2.0.2/examples/animation/simple_anim.html)

# Bar Chart Race

- Built on matplotlib.animation
- Works on 3-dimensional data
- Example from <https://pypi.org/project/bar-chart-race/>