DATA.ML.100 Introduction to Pattern Recognition and Machine Learning
TAU Computing Sciences
Exercise - Week 2: *LSQFit in Python*

Be prepared for the exercise sessions (watch the demo lecture). You may ask TAs to help if you cannot make your program to work, but don't expect them to show you how to start from the scratch.

1. **Linear model fit with $N > 2$ training points** (30 points)

   (a) During the lecture we started to derive solutions for the parameters a and b of the linear model $y = ax + b$ and for N training samples $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$.

   You should first finish the derivation. Do not google, but allow yourself to do the math since errors will be found in the next steps.

   (b) Implement a Python function *my_linfit(x,y)* that solves and returns a and b. Use your own derivations in the function - no matter how "ugly" they are - to convince yourself about your super powers.

   (c) Write a Python program that asks user to give N points with a mouse (left click: add point, right click: stop collecting) and then plots the points and a fitted linear model.

   ```
   (dataml100)$ python fit_line.py
   ```

   You may start with this code snippet or write your own:

```
# Linear solver
def my_linfit(x,y):
    a = 0
    b = 0
    return a,b

# Main
import matplotlib.pyplot as plt
import numpy as np

x = np.random.uniform(-2,5,10)
y = np.random.uniform(0,3,10)
a,b = my_linfit(x,y)
plt.plot(x,y,'kx')
xp = np.arange(-2,5,0.1)
plt.plot(xp,a*xp+b,'r-')
print(f"My␣fit:␣a={b}␣and␣b={b}")
plt.show()
```

   Return the following items:

   - Python code: <surname>_fit_line.py
   - A full desktop screenshot that includes the terminal window or IDE for code execution and the plot window after line fitting:
     <surname>_fit_line_screenshot.png