Workshop #1 Exercises

1. The file **`Weather.csv`** contains weather data in great detail. There are four *feature-vectors* in the table **T** (temperature in $^\circ\mathrm{C}$), **P** (air pressure in hPa), **H** (humidity in %) and **W** (wind speed in $\mathrm{km/h}$). The data is indexed by time, as given by the column **Local Time**.

- There are some missing data in the vectors. Before we start any calculations, we need to take care of this (and there are many ways in which you can handle this). Remove the rows that contain missing data in any of the columns.
- The vectors contain values which are of different magnitudes and variability. We "fix" this in a two step process:
  - Normalize the vectors using the Euclidean norm.
  - In the last step, we calculate the similarity matrix of the normalized feature-vectors. This matrix contains the cosine similarity coefficients for every pair of vectors:

	T	P	H	W
T	1	🟨	🟩	🟦
P	🟨	1	🟥	🟧
H	🟩	🟥	1	🟪
W	🟦	🟧	🟪	1

Are there any vectors that are not similar, based on their similarity measure?

---

2. The file **`Random_Matrix.csv`** contains a 4×4 symmetric matrix of random data. We use that matrix in the following calculations.

- Calculate SVD of the matrix
- Next, we will analyze the structure of the original data. Construct a rank 3 approximation of the original matrix (i.e. approximate the matrix using the three largest singular values)
- Let us measure how close the approximation matrix is to the original one. For this purpose, we use a **matrix norm** called **Frobenius norm.**[1] Calculate the difference of the two matrices (original minus approximate), and then calculate the Frobenius norm of the difference. Smaller numbers indicate that the matrices are "closer". Is this approximation a good one?
- Repeat the process, now with a rank 2 approximation (construct it, and measure how close to the original matrix this approximation is). Is the rank 2 approximation managing to "capture" the original data? In other words – would you use it in practical applications?

---

3. Use low-rank approximation of a matrix to construct approximations of the image **`cake.jpg`**. Follow the process from class. You should produce 10 images in the rank range from $k = 10$ to $k = 100$ (increment by 10, best to use a for-loop or similar). What is the (lowest) rank of the approximation you would use if you had to choose from the ten images you produced? Note: the last question is subjective 😊

[1] To obtain the Frobenius norm of a matrix $A = [a_{ij}]_{m\times n}$, denoted $\|A\|_F$, we use the formula:

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2}.$$

Basically, the Frobenius norm is a "glorified" Euclidean norm. To calculate it you must "flatten" the matrix, i.e. write it as a vector of its columns, and then calculate the Euclidean norm of this vector.