DATA ANALYSIS AND KNOWLEDGE DISCOVERY

EXERCISE I:
SETTING UP THE PYTHON WORKING ENVIRONMENT
AND FUNDAMENTAL EXERCISES (PASS/FAIL -GRADING)

Return your exercise as a .ipynb-file (file format used by Jupyter) at the course Moodle page

TASK I: SETTING UP THE WORKING ENVIRONMENT

Install Anaconda (with Python3), Jupyter Notebook, Numpy, Scipy and Matplotlib-libraries (Anaconda should contain all the last four) on your computer.

See the following links for installation tutorials:

Windows: https://www.youtube.com/watch?v=Q0jGAZAdZqM

Ubuntu: https://www.youtube.com/watch?v=DY0DB_NwEu0

Quick tutorials to Jupyter Notebook:

- https://www.youtube.com/watch?v=jZ952vChhuI
- https://www.youtube.com/watch?v=HW29067qVWk

Python tutorial: https://www.tutorialspoint.com/python3/

Anaconda is a free and open source distribution of the Python and R programming languages for data science and machine learning related applications that aims to simplify package management and deployment.

Jupyter Notebook is an open-source integrated development environment (IDE) web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.

NumPy is the fundamental package for scientific computing with Python.

SciPy is a Python-based ecosystem of open-source software for mathematics, science, and engineering.

Matplotlib is a Python 2D plotting library for visualization tools.

TASK II: BASIC DATA HANDLING WITH NUMPY IN JUPYTER NOTEBOOK

1. **Load the comma-separated data matrix (task_II_data.txt) and delete any rows containing nan-values in it using NumPy.**
2. **Calculate the mean and standard deviation of each column in the edited data matrix (that is, the matrix without rows containing nan-values).**
3. **Select the 2nd row of the edited data matrix and print the mean value of this row.**
4. **Set all values of the 3rd row of the edited data matrix into 1. Print the 3rd row before and after the edit.**
5. **Find all row indices of the edited data matrix, where the 2nd column has a value greater or equal to 0.5 and print the corresponding rows of the edited data matrix.**

Link to NumPy reference manual: https://docs.scipy.org/doc/numpy/reference/

TASK III: INTERPOLATING DATA WITH SCIPY AND BASIC PLOTTING WITH MATPLOTLIB

1. **Load the comma-separated data matrix (task_III_data.txt) and calculate linear and cubic interpolation functions using the loaded data (first column is the x-values, second y-values). Use SciPy's interp1d-function.**

   Documentation:
   https://docs.scipy.org/doc/scipy-0.19.1/reference/tutorial/interpolate.html#d-interpolation-interp1d

   Take advantage of the example code in the documentation.

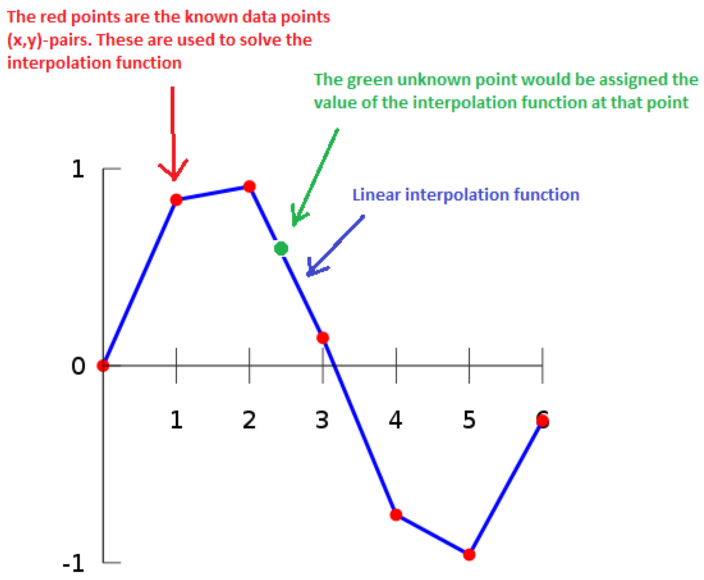
2. **Plot the loaded data and both interpolated functions. For figure title, x- axis, y-axis and legend, set the following labels (help can be found from the documentation of interp1d-function):**

   title text: "Input data and interpolated functions"
   x-axis text: "Input value"
   y-axis text: "Function value"
   legend: "data", "linear", "cubic"

   Link to SciPy reference manual: https://docs.scipy.org/doc/scipy/reference/

   What is interpolation? interpolation is a method of constructing new data points within the range of a discrete set of known data points.

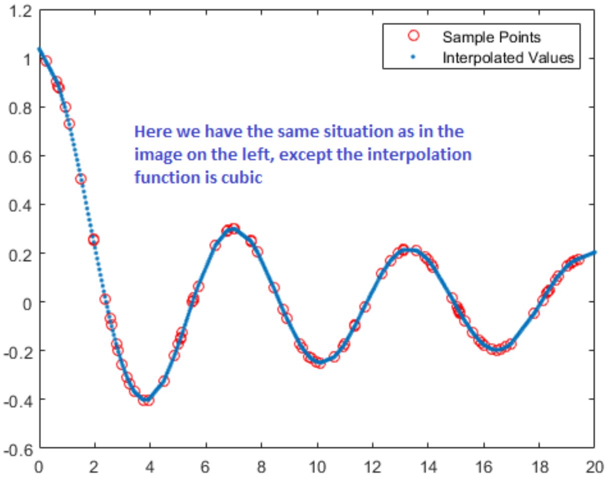
LINEAR INTERPOLATION


The red points are the known data points (x,y)-pairs. These are used to solve the interpolation function

The green unknown point would be assigned the value of the interpolation function at that point

Linear interpolation function

CUBIC INTERPOLATION


Here we have the same situation as in the image on the left, except the interpolation function is cubic