Homework 1

All assignments need to be submitted via github classroom, assignment
https://classroom.github.com/a/7DmFzjkn
You also need to submit your assignment on gradescope as a single pdf containing all code for the assignments and all outputs.

Each assignment needs to be a separate subfolder in the repository called `task1` and `task2`. Please add a Readme.md to your repository stating your UNI. Please also add links to the output of the travis run for task1.

Task 1: Unit tests

Ensure that all tests that are written as part of this task are run by continuous integration and pass.

1.1 Set up travisci for your repository, running for Python3.6 and Python3.7.
You need to enable travis for your repository in the repository settings, then you can check the status at https://travis-ci.com/aml-spring-19/<your_repo_name> after pushing a commit.

You need to add at least a single test to the repository for the tests to run successfully, so don't worry if the tests fail at this point.

1.2 Write a function "fib" that computes the Fibonacci sequence. Write a unit test that ensures that the fib(2) is 1, that fib(5) is 5 and fib(12) is 144.

1.3 Write a test that reads `input.txt` as provided in your repository using pandas. It's countries population from 1980 to 2010. Test that the number of rows in the data is 225 and the number of columns is 31. Write a separate test that tests that the world population according to this table in 2010 is ca. 7065 million.

Task 2: Data Visualization and Analysis

The following plots need to be generated by python files within the task2 folder, called task21.py, task22.py and task23.py. Each file should both show the plot and store it as an image file (called task21.png, …). Also include the generated images in your submission. When asking to reproduce the graphs, you are only required to reproduce the essence of the graph. Tick locations and legend locations and specific colors and fonts are not required, but everything needs to be clearly labeled.

2.1 Pick a graph from http://www.tylervigen.com/spurious-correlations and recreate it using matplotlib (you can also use numpy and pandas). The code should include reading the data from the web if possible, otherwise from a text file for csv file (to be included in the repository). Ensure the axes are labeled properly.

2.2 Create a pair-plot of the iris dataset similar to Figure 1-3 in IMLP using only numpy and matplotlib (you can use scikit-learn to load the data with sklearn.datasets.load_iris, you are not allowed to use pandas). Ensure all axes are labeled. The diagonals need to contain histograms, the different species need to be distinguished by color or glyph, and there needs to be a legend for the species.

2.3 Reproduce the graphs on overlapping data from “Fundamentals of Data Visualization” https://serialmentor.com/dataviz/overlapping-points.html that is figures 18.1 to 18.4, using matplotlib as subplots in a single figure. You can find the dataset on the UCI repository here: http://archive.ics.uci.edu/ml/datasets/Auto+MPG
The code should include reading the data from the web if possible, otherwise from a text file for csv file (to be included in the repository).