General constraints for code submissions Please adhere to these rules to make our and your life easier! We will deduct points if your solution does not fulfill the following:

- If not stated otherwise, we will use exclusively Python 3.5.
- If not stated otherwise, we expect a Python script, which we will invoke exactly as stated on the exercise sheet.
- Your solution exactly returns the required output (neither less nor more) – you can implement a `--verbose` option to increase the verbosity level for developing.
- Add comments and docstrings, so we can understand your solution.
- (If applicable) The `README` describes how to install requirements or provides addition information.
- (If applicable) Add required additional packages to `requirements.txt`. Explain in your `README` what this package does, why you use that package and provide a link to it's documentation or GitHub page.
- (If applicable) All prepared unittests have to pass.
- (If applicable) You can (and sometimes have to) reuse code from previous exercises.

Having learned about different ways to empirically evaluate the performances of algorithms and automl systems in this exercise you will now implement some of these techniques. Add your code creating plots and outputting statistics to `main.py` (callable as `python main.py`). Furthermore, combine all plots and answers to the questions into one PDF. We expect all plots to have axes labels and a legend!

1. **Visualization and Evaluation** [3 points]
   `data.csv` contains *error* values of two algorithms A and B on $n = 419$ datasets. The method `load_data()` loads this dataset as an $n \times 2$ *numpy array*.
   (a) Create a scatterplot comparing the performance of A and B. Use different markers for the following three categories of datasets: [1pt.]
      - algorithm A achieved an *error* value that is 0.1 lower than that of B
      - algorithm B achieved an *error* value that is 0.1 lower than that of A
      - other

      How many datasets are in each category? Report overall mean and standard deviation as well as median and quartile values for A and B.
   (b) Plot the empirical cumulative distribution function (eCDF) of A and B as shown in the lecture. What are the probabilities of A and B to achieve an *error* value of 0.4 and lower (it is okay to estimate the value from the plot). [0.5pt.]
   (c) Create a boxplot[1] and a violin plot[2] for A and B. Write one sentence to describe the difference between these two kinds of plots. [0.5pt.]
   (d) Implement a paired permutation test to determine whether the mean of A is better than the mean of B. In your solution state what is H_0, H_1 and which α-value you used? [1pt.]

2. **Feedback** [Bonus: 1 points]
   For each question in this assignment, state:
   - How long you worked on it.
   - What you learned.
   - Anything you would improve in this question if you were teaching the course.

This assignment is due on 17.05.19 (10:00). Submit your solution for the tasks by uploading a PDF to your groups BitBucket repository. The PDF has to include the name of the submitter(s). Teams of at most 2 students are allowed.

---

[1] using e.g. `https://matplotlib.org/api/_as_gen/matplotlib.axes.Axes.violinplot.html`
[2] using e.g. `https://matplotlib.org/api/_as_gen/matplotlib.pyplot.boxplot.html`