Setup

For this exercise, you should install the packages `matplotlib`, `pandas`, `scikit-learn`, `scipy`, and `seaborn`. If you don't want to install them individually, you may also use the file `requirements.txt` from ILIAS. See `DS1_Python_Setup.pdf` for more information.

All these packages have detailed documentation as well as countless Stack Overflow questions, which both might help to solve the following tasks.

Task: Statistics and Plots

The aim of this exercise is to compute descriptive statistics and create different types of plots. We use the world-renowned `iris` dataset, which was not only featured in the lecture, but even has its own Wikipedia entry.

a) **[Loading]** Obtain the dataset with the function `load_iris()` from the package `sklearn.datasets`. Combine the values of the properties `data`, `feature_names`, `target`, and `target_names` from the loaded object into one `pandas.DataFrame`. Name the target column `species`. View the resulting `DataFrame`.

b) **[Descriptive Statistics]** Compute descriptive statistics like mean, standard deviation etc. for the numeric features of the dataset. You may call methods to compute individual statistics like `mean()` as well as a summary with `describe()`, both applicable to the whole `DataFrame` as well as single columns.
To summarize the target variable, count how often each `species` occurs.

c) **[Distribution Plots]** Choose at least one of the features and create a histogram as well as a boxplot with the package `matplotlib.pyplot`, `seaborn`, or using the `plot()` method of `DataFrame`. Try changing the number of buckets used in the histogram.

d) **[Scatter Plots]** Create a scatter plot with one feature on one axis and another feature on the other axis. Color the data points according to `species`.

e) **[Grouped Boxplots]** Use `seaborn` to create a boxplot of one numeric feature, having a separate box for each `species`. Repeat this procedure for each feature.

f) **[χ^2 Test]** Conduct χ^2 tests to examine the relationship between each of the numeric features and the target `species`. You may use `chi2_contingency()` from `scipy.stats` for the tests. As preparation, you may use `crosstab()` from `pandas` to create contingency tables and `cut()` from `pandas` to discretize features.
How do you interpret the results? What is the relationship between the results of the statistical tests and the plots from the previous sub-task?