Covariance and Principal component analysis

In the first part of this assignment you will learn how principal component analysis (PCA) can be used for reducing dimensionality and visualizing global dataset structure. In the second part, which is theoretical, you will test your understanding of Bayesian Statistics through a few questions.

Exercise 1 (Performing PCA, 30 points)**.**

a) Implement PCA. You might want to use the supplied template function `pca.py`, especially for its comments. Your function should return i) unit vectors spanning the principal components, and ii) the variance captured by each of these components, where the principal components are sorted so that the variance is monotonically decreasing.

b) Perform PCA on the *murder* dataset as discussed in the appendix of this document. Remember that each principal component (PC) corresponds to an eigenvector of the covariance matrix, and that the corresponding eigenvalue corresponds to the variance of the data in the direction of the eigenvector. Produce a scatterplot of the dataset along with the mean and the principal eigenvectors pointing out of the mean, each eigenvector with a length scaled by the standard deviation of the data projected onto that eigenvector.

c) Perform PCA on the *pesticide* dataset (same as used in Assignment 2, see description in the Appendix below). Make a plot of variance versus PC index, where you should see the variance stabilizing (capturing primarily noise).

You can determine the cumulative normalized variance by normalizing the variance along all PCs such that the sum of all variances is 1, and then capture how large a proportion of the variance is described by the first, second, etc PC. Plot the cumulative variance versus the number of used PCs. How many PCs (dimensions) do you need to capture 90% of the variance in your dataset? 95%?

Deliverables. a) Uploaded code, b) the plot, and b) plot of variance versus PC; plot of cumulative variance versus PC; the numbers of dimensions needed to capture 90% and 95%.

Exercise 2 (Visualization in 2D, 30 points)**.** *Multidimensional scaling* is the process of visualizing a dataset in 2D or 3D while preserving pairwise distances between data points as well as possible. A classical way to do this is by projecting data points onto the first 2 or 3 principal components of the dataset.

Write a script to project the *Pesticide* dataset onto the first 2 principal components. You may want to use the supplied template `mds.py`, and produce a plot of the dataset. If you have not completed exercise 1, you may use a built-in PCA package from e.g. `scikit-learn`.

Deliverables. Uploaded code and plot.