Scientific Computing: Basic Methods
J. Bürki, M. Block and R. Barniol Duran

Homework 7: Histograms and Function Fitting in Python

- Make sure that you scripts run without error in order to get credit. Do not hesitate to ask for help if needed!
- Check the output of the Autograder for any issue that should be fixed. In case of "Unexpected error", email me so I can take a look as it may be an issue with the Autograder rather than your code.
- Take ownership of your learning! Remember that you are responsible for the work you turn in. Simply copying somebody else's answers, copying from the Internet, using AI to generate your code, sharing your code (or part of your code) in any way, or copying it from someone else will be considered academic dishonesty. Please, contact me if you have any questions about collaborations.

Problem 1 Distribution of random numbers

For this problem, we will use a different function from `numpy.random` that generates numbers between 0. and 1. with an asymmetric distribution that depends on 2 parameters `a` and `b`. The function within the module that generates the numbers is `beta(a, b, N)`. Use it with `a=1.7` and `b=8.5` to generate 500 numbers, from which you will build a histogram. Following the procedure described in the notes, try and fit the histogram to the function

$$\beta(x, a, b) = d_0 x^a (1-x)^b, \tag{1}$$

where d_0, a, and b will be the fitting parameters.

Add the midpoints as blue circles and the fit as a dashed, red line to your plot. Play around with the number of bins you use. You should also add to your plot a legend, some text giving the values of the fit parameters, as well as vertical lines of different styles at the position of the mean and the median of the distribution (including labels giving their values.)

Repeat the whole thing for 50000 random numbers and have the two plots showing in two subplots of the same figure. What do you notice?

Problem 2 Linear fit

Import the data stored in the file `HW07-linearFit.dat` and fit it to several polynomials of order n

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \tag{2}$$

for various values of the polynomial order n. In other words, start fitting to a line, then a parabola, a cubic, ... until you get a reasonable looking fit.

Plot the data and fits for various values of $n = 1, 2, 3, \ldots$ (stop at the value of n when you find a reasonable fit) together on the same graph (make sure to include a legend that indicates the value of n used for each fit).

What minimum value of n do you need to have in the polynomial to obtain a reasonable fit?

Problem 3 Data import & Histogram

In this problem, you'll analyze some data that, while made up for this problem, was inspired by experimental results.

Use the data contained in the attached file `HW07-prob3-data.txt`. Let's call the data G, which is the symbol commonly used for conductance (the reciprocal of the resistance, which was measured in the experiment inspiring this problem).

Start by importing the data and plot it as a scatter plot of G vs. the point number. You should observe data that seems to follow a curve that overall increases while moving along the horizontal axis, but seems to flatten a couple of times. These flatter parts, or *plateaus*, mean that some values of G are more likely than others to be observed.

In order to get a better idea of the relative probabilities of observing any value of G, we need to "count" how many times each value of G occurs, or in other words, make a histogram of the G values.

(a) Make such a histogram of the G data. Try different number of bins until you get a "good-looking" histogram.

(b) Make sure to obtain the midpoints of the edges, as described in the notes. Add the midpoints to your figure at the top of each bar. You should observe a double-peaked distribution that corresponds to the two plateaus in the initial plot.

(c) Try to fit the histogram data to a sum of two Gaussian functions

$$f(x) = A \exp\left(-\frac{(x-\bar{x})^2}{2\sigma^2}\right), \tag{3}$$

each with its own set of fitting parameters $A_1, \bar{x}_1, \sigma_1$ and $A_2, \bar{x}_2, \sigma_2$.
Depending on how many bins you used (feel free to experiment with that number), `curve_fit` may give a pretty good fit without any further work. If not, try and get estimates of the positions, height, and half-width of the peaks and provide these as initial guesses to `curve_fit`.

(d) Once you have a decent fit, add it to the plot; also add each of the two Gaussian separately using dashed lines and vertical dotted lines showing the positions of the two peaks (label them).

What to submit: For each problem, you'll have to submit one `Python` script and one figure, using the usual naming convention `HW07_buerki-P1.py`.