Statistical Methods in Physics (14P058)

Prof. Federico Sánchez (federico.sancheznieto@unige.ch)
Dr. Hepeng Yao (hepeng.yao@unige.ch)
Dr. Knut Zoch (knut.zoch@unige.ch)

Exercise I – Basics of statistics and python

22 & 29 September 2022, 9:15, room: SCI–202

Task 1: *Drawing random numbers from a PDF*

a) Use the `numpy.random` module to initialise a default random number generator (hint: check the numpy documentation).

b) Draw a batch of 100 random values from a uniform distribution in the interval $[0, 1)$.

c) Use the `matplotlib.pyplot` module to fill a histogram with the drawn values. As an example, the histogram should have 20 equidistant bins in the range $[0, 1]$.

d) Draw a batch of 1000 random values from a Gaussian distribution with mean $\mu = 15$ and standard deviation $\sigma = 1.0$. Fill these values into another histogram and plot it.

Task 2: *Defining and plotting analytical functions*

a) Write a function in python for a Gaussian distribution including the norm factor $\frac{1}{\sigma\sqrt{2\pi}}$. Hint: the python function should take three arguments, i.e., the value x at which it is evaluated, the mean μ and the standard deviation σ. It should return its evaluated value at x.

b) Create an array of 100 equidistantly spaced values in the range $[10, 20]$ (you can use the `numpy.linspace` method for that).

c) Create an array that contains the corresponding values of the evaluated Gaussian distribution for $\mu = 15$, $\sigma = 1.0$, such that you have 100 pairs of x and y values.

d) Plot the analytical function using `matplotlib.pyplot.plot` into the same figure as the random values drawn from a Gaussian distribution in the previous task.

Task 3: *Fitting functions to data*

As a last step, let's use the previously defined Gaussian function and perform a parameter estimate based on the drawn data points:

a) Histogramize the 1000 drawn values into a histogram with 50 bins in the range $[10, 20]$.

b) Extract the bin centres and bin values.

c) Perform a maximum likelihood estimate of the two function parameters, μ and σ, using the `scipy.optimize.curve_fit` method and the previously defined python function for Gaussian distributions.

Task 4: *Bonus question: the Monty Hall problem*

The Monty Hall problem is a probability puzzle[1], loosely based on the American television game show "Let's Make a Deal" and named after its original host, Monty Hall. More information can be found, e.g., on wikipedia. One popular formulation of the problem is the following:

> *Suppose you're on a game show, and you're given the choice of three doors. Behind one door is a car, behind the others, goats. You pick a door, say #1, and the host, who knows what's behind the doors, opens another door, say #3, which has a goat. He says to you, "Do you want to pick door #2?" Is it to your advantage to switch your choice of doors?*

Contrary to popular belief, switching doors is actually beneficial and increases chances of winning the car to ⅔, compared to an unchanged probability of ⅓ with the originally chosen door.

Write a small function that mimics the above problem: you pick a random door, then the "host" opens once of the other two doors, always one with a goat. You are left with the choice of keeping your originally chosen door or switching to the other unopened door.

a) Simulate a large number of toy experiments where you pick a door at random, but stick with your original door choice. What is the probability of winning?

b) Simulate a large number of toy experiments where you pick a door at random, but then switch your choice after one goat is revealed. What is the probability of winning?

[1] Original formulation: Selvin, Steve. "A problem in probability (letter to the editor)". The American Statistician. 29 (1): 67–71. doi:10.1080/00031305.1975.10479121