Tutors:

- Group 1: Ziyang Zheng
- Group 2: Nico Lorenz
- Group 3: Ziad Sakr
- Group 4: Adrian Schirra

Please, send (in pdf) or hand in the solution to this exercise sheet to your tutor, following their instructions, and carefully respecting the delivery date shown below. All exercise sheets will be graded. You can solve them individually or in pairs (with another student of the same tutorial group). In the latter case deliver please only one document with the solutions. Write the name of the file in the following format: X.Name1.Surname1 or X.Name1.Surname1-Name2.Surname2.pdf, with X being the number of the corresponding exercise sheet.

Exercise sheet 4

RECEIVED: May 16 - DELIVERY: May 23

Exercise 4.1: Moment Generating Function (5pt)

a) Show that for a Gaussian PDF $X \sim N(\mu, \sigma^2)$ with mean μ and variance σ the moment generating function is given by (1pt)

$$M_X(t) = \exp\left[\frac{1}{2}t^2\sigma^2 + \mu t\right]$$

b) Calculate the first six moments with the moment generating function using $\mu = 0$.(3pt)

c) can you see an iterative rule for the Gaussian moments ?(1pt)
(Hint: look at all the even and uneven moments separately)

Exercise 4.2: Python Exercise - Central Limit Theorem (6pt)

The central limit theorem makes sure that the sum of identically distributed independent random numbers with finite variance is approximately Gaussian distribution, and the approximation becomes better when summing over more random numbers. Try out these statements with a suitable python-script :

a) What's the distribution for the means of n Poisson-distributed random numbers? To calculate the distributions use the implemented random number generator of the scipy stats library `scipy.stats.poisson.rvs(mean,size = n)` and set the mean to 2 (2pt)

b) What about the means of n Cauchy-distributed random numbers? Generate the random numbers by `scipy.stats.cauchy.rvs(size = n)` (2pt)

c) Plot the distributions for the mean of $n = 100$ random numbers. Take the mean of $m = 50, 1000, 10000$ samples.
For the 10000 samples create the bins by: `bins = numpy.arange(xmin,xmax,stepsize)`. For the Cauchy distribution set $x_{min} = -7$ $x_{max} = 7$ and `stepsize` $= 0.1$. Use for the Poisson distribution `xmin` $= 1$ `xmax` $= 3$ and `stepsize` $= 0.01$. For the smaller sample sizes find yourself a suitable smaller step size. When arriving at the $m = 10000$ samples, draw a Gaussian PDF to the data. First fit the data by using `mean, sigma = scipy.stats.norm.fit(data)` and draw it onto the data by `plot(bins, scipy.stats.norm.pdf(bins,loc=mean,scale=sigma)`. Does the Gaussian distribution now fit the data / your expectation? If not you can change the fit function to the one you generated the data with (just replace norm with Cauchy or Poisson in the fit and the plot). What can you conclude? (2pt)

Exercise 4.3: Particles in a Box (5pt)

There are $N = 10^4$ particles randomly distributed in a volume V of 1 m^3; focus now on a small cubic sub-volume v of 10 cm by side.

a) What is the probability distribution of finding n particles in the volume v that applies here and why does it apply? (2pt)

b) What is the probability to find more than $n = 15$ particles inside the cubic volume v ? And less than $n = 5$ particles? Repeating this experiment infinitely many times, what is the variance of the distribution of particles? (2pt)

c) How big should N be in order to get a relative error in the number of particles in the cubic sub-volume of less than 1%? (1pt)