

# Algorithms for Uncertainty Quantification

## Tutorial 2: Probability and statistics overview

In this worksheet, we focus on aspects related to probability theory and statistics.

### Biased vs. unbiased estimators

The formal definition of an estimator states that “an **estimator** is a procedure to construct estimates for a quantity  $q$  based on random samples  $X_1, \dots, X_n$ .” If  $x_1, \dots, x_n$  are realizations of  $X_1, \dots, X_n$ , an **estimate** is a realization of the estimator based on  $x_1, \dots, x_n$ . Example estimators include mean, variance, interval estimators.

#### Assignment 1

Assume that  $G = \{1.3, 1.7, 1.0, 2.0, 1.3, 1.7, 2.0, 2.3, 2.0, 1.7, 1.3, 1.0, 2.0, 1.7, 1.7, 1.3, 2.0\}$  represents a set of grades. Compute the mean and the variance of  $G$  using `numpy`'s functions `mean` and `var`. *Hint: if  $\mathbf{a}$  is a list with  $n$  elements  $X_i$ ,  $i = 1, \dots, n$ , `numpy.mean(a)` =  $\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$  and `numpy.var(a)` =  $S^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n}$ .*

An estimator is called *biased* if its mean value is **not** equal to the value of the parameter to be estimated. Otherwise, it is called unbiased.

#### Assignment 2

Check whether  $\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$  and  $S^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n}$  are biased or unbiased estimators. In case they are biased, how would you transform them into unbiased estimators? How would you modify the previous code to account for your modification?

## Univariate concepts

In the lecture, you saw some examples of discrete and continuous random variables. In this tutorial, we focus on continuous random variables.

Reminder: Let  $X$  be a random variable. Every random variable  $x$  has an associated *cumulative distribution function* (CDF)

$$F_X(x) = \mathcal{P}\{X \leq x\}.$$

Furthermore, a continuous random variable  $X$  has an associated probability density function (PDF)

$$f_X : \mathbb{R} \rightarrow [0, \infty[, \quad f_X(x) = \frac{dF_X(x)}{dx}.$$

Two of the most prominent examples of continuous random variables are the *uniform* and the *normal* or *Gaussian*:

- A random variable  $U$  is *uniformly distributed in the interval*  $[a, b]$ , denoted as  $U \sim \mathcal{U}(a, b)$ , if the associated PDF is  $f_U : [a, b] \rightarrow \{0, \frac{1}{b-a}\}$ ,

$$f_U(x) = \frac{1}{b-a} \mathcal{I}_{[a,b]}(x),$$

$$\text{where } \mathcal{I}_{[a,b]}(x) = \begin{cases} 1, & x \in [a, b] \\ 0, & \text{otherwise.} \end{cases}$$

- A random variable  $N$  is *normally distributed with mean*  $\mu$  *and variance*  $\sigma$ , denoted as  $N \sim \mathcal{N}(\mu, \sigma^2)$ , if its PDF is

$$f_N(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-(x - \mu)^2/2\sigma^2).$$

Sometimes, certain UQ formulations require *standard, reduced* random variables, i.e. random variables defined on standard domains, such as  $[0, 1]$ ,  $[0, \infty)$ , or  $(-\infty, \infty)$ . Two prominent examples of reduced random variables are  $U \sim \mathcal{U}(0, 1)$  or  $N \sim \mathcal{N}(0, 1)$ . However, the underlying uncertainty might be modeled in terms of random variables that are not reduced or not from a classical family.

### Assignment 3

Consider  $U \sim \mathcal{U}(0, 1)$  and  $U_g \sim \mathcal{U}(m, n)$ ,  $m < n \in \mathbb{N}$ . Write  $U_g$  in terms of  $U$ .

### Assignment 4

Consider  $N_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$  and  $N_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ . Show that

- $N_1 + c \sim \mathcal{N}(\mu_1 + c, \sigma_1^2)$ ,  $c \in \mathbb{R}$
- $cN_1 \sim \mathcal{N}(c\mu_1, c^2\sigma_1^2)$ ,  $c \in \mathbb{R}$
- $N_1 + N_2 \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$

Finally, show that if  $N \sim \mathcal{N}(0, 1)$  and  $N_g \sim \mathcal{N}(\mu, \sigma^2)$ ,  $N_g = \mu + \sigma N$ .

In cases when the random variable  $X$  is not from a classical family, e.g.  $X \sim \text{lognormal}$ , we might want to write it in terms of classical random variables. One such approach stems from a basic technique for generating random numbers.

Let  $F_X(x)$  denote the cdf of the target random variable  $X$ . We assume that we have a (pseudo)random number generator capable of generating realizations  $U \sim \mathcal{U}(0, 1)$ . If we define the random variable  $Y = F_X^{-1}(U)$ , then  $Y$  and  $X$  have the same distribution, i.e. sampling  $X$  translates into sampling  $U \sim \mathcal{U}(0, 1)$  and then evaluating  $Y = F_X^{-1}(U)$ .

### Assignment 5 - optional

Based on the above setup, show that  $Y = F_X^{-1}(U)$  has the same distribution as  $X$ . *Hint: start from  $F_Y(y)$ .*

## Multivariate concepts

In the lecture, you saw the definition of the multivariate normal distribution.

The  $n$ -dimensional random vector  $\mathbf{X}$  is normally distributed with mean vector  $\boldsymbol{\mu} = [\mu_1, \mu_2, \dots, \mu_n]^T$  and covariance matrix  $V$ ,  $V_{ij} = \text{cov}(X_i, X_j)$ , written  $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, V)$ , if

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n |V|}} \exp \left[ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})V^{-1}(\mathbf{x} - \boldsymbol{\mu})^T \right]$$

where  $|V|$  is the *determinant* of  $V$ . The standard multivariate normal is defined for  $\boldsymbol{\mu} = [0, 0, \dots, 0]^T$ ,  $V = I_n$ .

All transformations from Assignment 4 can be extended to the multivariate case. However, we are most interested in writing a generic multivariate normal distribution in terms of the standard multivariate normal.

To this end, if  $\mathbf{N} \sim \mathcal{N}(\mathbf{0}, I_n)$  and  $\mathbf{N}_g \sim \mathcal{N}(\boldsymbol{\mu}, V)$ , it can be shown that

$$\mathbf{N}_g = \boldsymbol{\mu} + E\mathbf{N}, \tag{1}$$

where  $EE^T = V$  (for brevity, we will not prove this here).

## Assignment 6

$\mathbf{N}_1 \sim \mathcal{N}(\boldsymbol{\mu}, V)$ , where  $\boldsymbol{\mu} = [0.1, 0.5]^T$  and  $V = [[1.0, 0.2], [0.2, 1.0]]$ . Using Eq. (1), write a python program in which  $\mathbf{N}_1$  is defined in terms of  $\mathbf{N} \sim \mathcal{N}(\mathbf{0}, I_n)$ . Furthermore, for a comparison, plot the sampled points for both approaches. Additionally, you can use `matplotlib.mlab.bivariate_normal` to plot contour and the 3D representation of  $\mathbf{N}_1$ . *Hint: to obtain the matrix  $E$  from  $V = EE^T$ , you can use a Cholesky decomposition.*

## Assignment 7 - optional

We know that the entry  $ij$  in the covariance matrix  $V$  is defined as  $V_{ij} = cov(X_i, X_j)$ , where  $cov(X_i, X_j) = \mathbb{E}[X_i X_j] - \mathbb{E}[X_i]\mathbb{E}[X_j]$  measures the *covariance* between  $X_i$  and  $X_j$ . Hence,  $X_i$  and  $X_j$  are *independent* if  $cov(X_i, X_j) = 0$ .

Therefore, having the covariance matrix  $V$ , a quick way to check whether the underlying random variables are independent is to look at the off-diagonal entries of  $V$ ; if they are non-zero, the variables are dependent, otherwise, they are independent.

Given  $\mathbf{N}_1 \sim \mathcal{N}(\boldsymbol{\mu}, V)$  such that  $V$  has non-zeros off-diagonal entries, how could you use Eq. (1) to recast your problem in terms of independent random variables?