- 1. Joint, marginal and conditional distributions I  $\star$
- 2. Joint, marginal and conditional distributions II \*
- 3. Compound distribution
- Sampling vs. evaluating random variables ★
- Generating data ★
- 6. Analyzing and visualizing data ★
- Working with dataframes ★
- 8. Error propagation and functions

## Introductory exercises

Eawag Summer School in Environmental Systems Analysis

Note, for the first three exercises you need only pencil and paper.

# 1. Joint, marginal and conditional distributions I ★

The joint discrete probability table of  $P_{A,B}(a,b)$  is given below:

	B.1	B.2	B.3
A.1	0.2	0.1	0.3
A.2	0.1	0.1	0.2

Derive the following probabilities:

- $P_{A,B}(1,2)$
- $P_B(2)$
- $P_{A|B}(1|2)$
- Are A and B independent?

# 2. Joint, marginal and conditional distributions II ★

Assume the probability densities  $p(E \mid B)$ , p(B),  $p(A, D \mid E)$ , and  $p(C \mid B, E)$  are known.

 Draw the corresponding directed acyclic graph of the conditional probabilities to visualize the independence structure.

- Derive p(B, C, E)
- Derive the joint distribution of *A*, *B*, *C*, *D*, and *E*.
- Derive  $p(A, B \mid C, D, E)$
- Derive  $p(A \mid D)$
- Derive  $p(A \mid B, E)$

## 3. Compound distribution

Assume that:

$$\mu \sim f_{\mu}(\mathbf{n}) = \begin{cases} 0.1 \exp(-0.1\mathbf{n}) & \mathbf{n} \ge 0 \\ 0 & \text{else} \end{cases}$$

and

$$X \sim f_{X|\mu \# m}(x \mid m) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x - m)^2}{2}\right)$$

This means X is normal distributed with mean  $\mu$  and  $\mu$  itself is exponentially distributed.

Derive and interpret:

- $f_{X,\mu}(x,m)$
- $f_X(x)$ , a so called compound distribution.
- $P(\mu > 5)$
- $f_{X,\mu|\mu>5}(x,\mu x)$   $f_{X|\mu>5}(x)$

It is not the aim to find closed forms for the integrals.

## 4. Sampling vs. evaluating random variables ★

Assume-two random variables X and Y with following distributions:

The 
$$X \sim \text{Uniform}(0, 1)$$
  
 $Y \sim \text{Normal}(2, 10)$ 

- 1. Evaluate the probability density  $f_X(0.8)$  and  $f_Y(0.8)$ .
- 2. Generate 10000 samples from both random variables. Visualize the distributions as histograms.

Another random variable is defined as a function of X as follows:

$$Z = \sin(2\pi X)\sqrt{X}$$

While it is difficult to derive the probability density of Z, sampling from it is easy.  $\angle 3$  Derive the dusty

Generate 10000 samples from Z by first sampling from X and then transforming the samples.  $\neq_{\geq}$  ( $\geq$  ) Visualize as histogram.

10.03.23, 15:05

#### **Hints**

R

Julia

Most important univariate probability distributions are already implemented in R. Type ?Distributions to get an overview. For every distribution \_\_ four functions are defined with the following naming scheme:

```
d__(x, ...) # evaluate pdf at x
p__(x, ...) # evaluate cdf at x
q__(p, ...) # evaluate the p-th quantile
r__(n, ...) # sample n random numbers
```

For example, for the normal distribution the functions are called <code>dnorm()</code>, <code>pnorm()</code>, <code>qnorm()</code>, and <code>rnorm()</code>.

Histograms are generated with the function <code>hist</code>. You can adjust the number of bins with the argument <code>breaks</code>, e.g. <code>hist(rnorm(10000)</code>, <code>breaks=100)</code>.

### 5. Generating data ★

Generate two samples of fictional observations (each of class matrix ) denoted as  $Y_{\rm obs,indep}$  and  $Y_{\rm obs,dep}$ . The former should contain 1000 realisations of two *independent* random variables (as two columns of the matrix) and the latter of two *dependent* ones. Use means of  $\mu=(3,8)$  for both samples. Use the standard deviations  $\sigma_{\rm obs,indep}=(2,5)$  for the independent variables constituting  $Y_{\rm obs,indep}$ , and the covariance matrix.

$$\Sigma_{\text{obs,dep}} = \begin{pmatrix} 4 & 8 \\ 8 & 25 \end{pmatrix}$$

for the dependent variables constituting  $Y_{
m obs,dep}$  .

#### **Hints**

R

Julia

In R, objects of a certain class can often be constructed by a function that matches the class name, such as matrix(). You can use rnorm() and cbind() to  $construct Y_{obs,indep}$  and rmvnorm() from the package mvtnorm to  $construct Y_{obs,dep}$ .

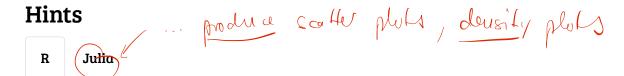
## 6. Analyzing and visualizing data ★

Perform some preliminary analysis of the data generated in Task 5:

- a. What are the interquartile and the 90%-interquantile ranges of your samples?
- b. Plot and compare the histograms and the densities of all the marginals
- c. Compare the scatterplots of  $Y_{\rm obs,indep}$  and  $Y_{\rm obs,dep}$

d. Compute the covariance and the correlation matrix of  $Y_{
m obs,indep}$  and  $Y_{
m obs,dep}$ 

Which of the above steps reveal a potential correlation structure in your data?

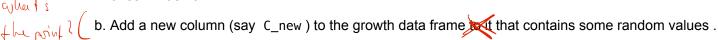


Try to arrange multiple plots in the same window by setting par(mfrow=c(<nrow>,<ncol>)). Use quantile() to calculate the interquantile range; use hist() and plot(density()) to visualize the data; use cov(), cor() for the covariance and the correlation, respectively.

## 7. Working with dataframes $\star$

Real data contain often columns of different data types (e.g. numbers and strings). Dataframes are designed to work with this kind of data conveniently.

a. Import the file ./data/model\_growth.csv as dataframe. Perform some analyses similar to the ones in Task 6.



#### **Hints**

R Julia

Read the data using read.table("</path/to/somefile.txt>", header=TRUE) to indicate that the first row are the column names (use file ../data/model\_growth.csv). To select the column C\_M from a dataframe, say data, you can use data $C_M$  or data[,"C\_M"].

To add columns, you can use <code>cbind</code> to column-bind the new data to the available matrix and convert everything to a <code>data.frame</code>. Then, you can rename the columns by assigning the desired names with function <code>colnames()</code> applied to the newly created <code>data.frame</code>.

### 8. Error propagation and functions

It is generally known that,  $\underline{\mathbf{f}}f()$  is non-linear:  $f(E[X]) \neq E[f(X)]$ 

where E is the expected value and X is a random variable. Define a non-linear function in R, e.g.  $f(x) = \sin \sqrt{x}$ . In order to avoid negative values, generate some realizations of a log-normally distributed random variable X. Calculate f(E[X]), E[f(X)], Var[X], Var[f(X)] and compare them.

#### **Hints**

R Julia

Use rlnorm to sample from a log-normal distribution, which accepts the mean and the standard deviation on the log-scale, not on the original scale. Additionally, keep in mind that in R a general function can be defined as:

```
function.name <- function(arg1,arg2){
  result <- arg1 + arg2 # or any other operation
  return(result)
}</pre>
```

Most basic functions are already available, and those include both sin and sqrt. Try ?sin in the R console to get access to the manual of the harmonic functions. These can be used anywhere in the code, including inside a custom function.