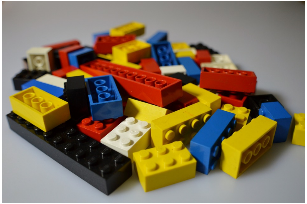
Reminder on the main probability distributions

M. L. Delignette-Muller
VetAgro Sup - LBBE

September 21, 2021

Why this reminder ?

- Bayesian inference enables to easily work with any probability distribution
  $\Rightarrow$ handled models use various distributions.
- In Bayesian inference the modeller has to explicitely write deterministic and **stochastic** links of his model.
  $\Rightarrow$ A good knowledge of classical distributions is required.



Stochastic modelling = playing lego with probability distributions

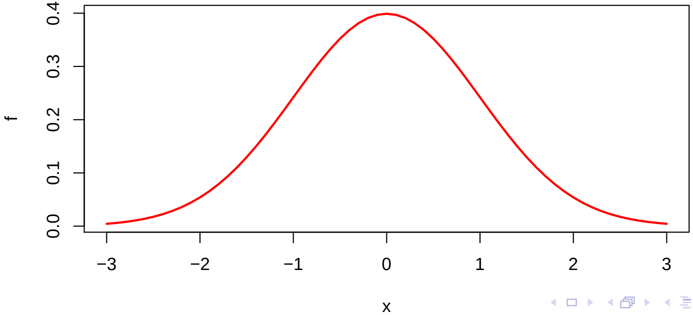
Learning objectives

- To know the main probability distributions to be able to build models.
- To know how to manipulate distributions using the **R** language (dpqr functions).

R handling of a probability distribution
The **density function** d...

Ex. with a Gaussian law: `dnorm(x, mean, sd)`

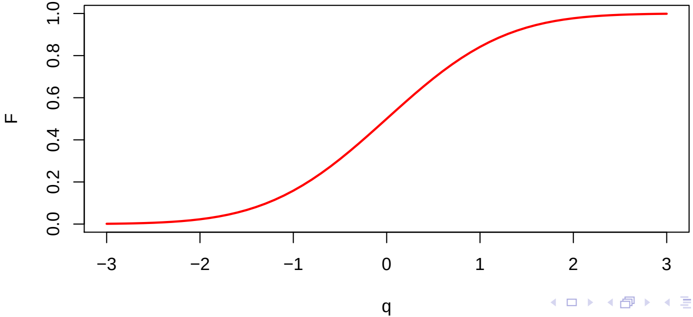
```
x <- seq(-3, 3, 0.1); f <- dnorm(x, mean = 0, sd = 1)
plot(x, f, type = "l", col = "red", lwd = 2)
```



R handling of a probability distribution
The **probability distribution** function p...

Ex. with a Gaussian law: `pnorm(q, mean, sd)`

```
q <- seq(-3, 3, 0.1); F <- pnorm(q, mean = 0, sd = 1)
plot(q, F, type = "l", col = "red", lwd = 2)
```

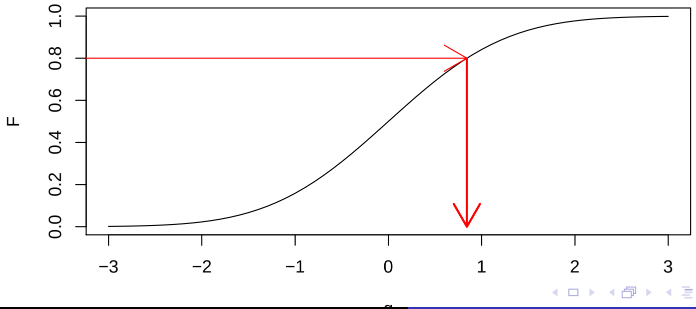


R handling of a probability distribution The **quantile** function q...

Ex. with a Gaussian law: `qnorm(p, mean, sd)`

```
qnorm(0.8, mean = 0, sd = 1)
```

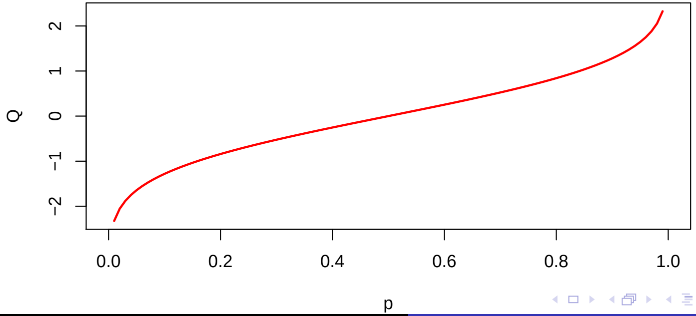
```
[1] 0.842
```



R handling of a probability distribution The **quantile** function q...

Ex. with a Gaussian law: `qnorm(p, mean, sd)`

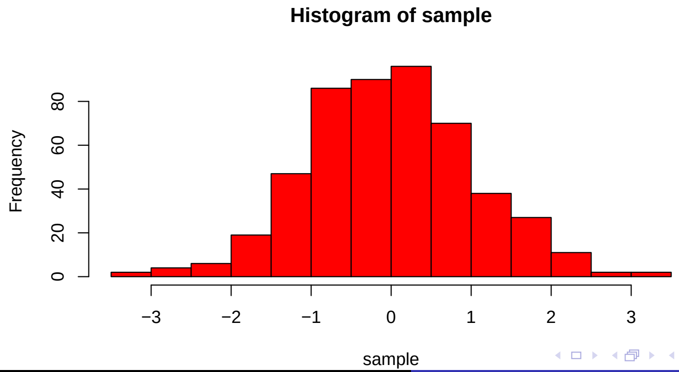
```
p <- seq(0, 1, 0.01); Q <- qnorm(p)
plot(p, Q, type = "l" , col = "red", lwd = 2)
```



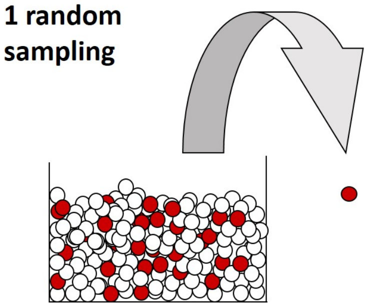
R handling of a probability distribution
The **random generator function** r...

Ex. with a Gaussian law: `rnorm(n, mean, sd)`

```
sample <- rnorm(500, mean = 0, sd = 1)
hist(sample, col = "red")
```



Bernoulli process and distribution

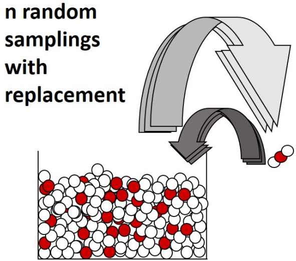


Discrete probability distribution of Z, the variable coding for a success
(ex.: 1 if the ball is red) :

$$Z \sim Bern(p)$$

with p the success probability (here the proportion of red balls).

Bernoulli process and binomial distribution



Discrete probability distribution of R, the number of success among n random draws:

$$R \sim Binom(p, n)$$

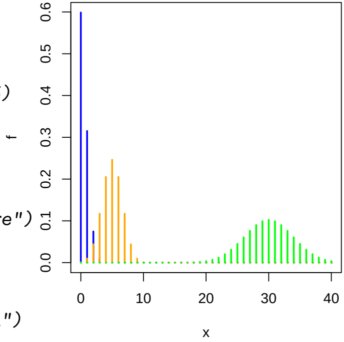
Asymptotic properties:

- $Binom(p, n) \to Poisson(np)$ for large n
- $Binom(p, n) \to N(np, \sqrt{np(1-p)})$ for large np and $n(1-p)$

Binomial distribution in R

dbinom(x, size = n, prob = p)

```
x <- 0:40
#
# n = 10, p = 0.05
f <- dbinom(x, size = 10, prob = 0.05)
plot(x, f, type = "h", col = "blue")
#
# n = 10, p = 0.5
f <- dbinom(x, size = 10, prob = 0.5)
points(x, f, type = "h", col = "orange")
#
# n = 60, p = 0.5
f <- dbinom(x, size = 60, prob = 0.5)
points(x, f, type = "h", col = "green")
```



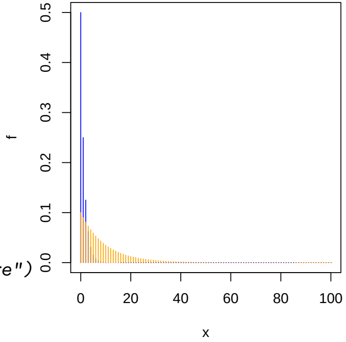
Geometric distribution

For the same Bernoulli process (success probability p),
with T **the number of random draws needed to have one succes**
($T-1$ is the number of failures before one succes) :

$$T - 1 \sim Geom(p)$$

Geometric distribution in R
dgeom(x, prob = p)

```
x <- 0:100
#
# p = 0.5
f <- dgeom(x, prob = 0.5)
plot(x, f, type = "h", col = "blue")
#
# p = 0.1
f <- dgeom(x, prob = 0.1)
points(x, f, type = "h", col = "orange")
```



Negative binomial distribution

For the same Bernoulli process (success probability p),
with T **the number of random draws needed to have s succes**
($T - s$ is the number of failures before s succes)

- if the last draw is a success (we stop draws at s success):

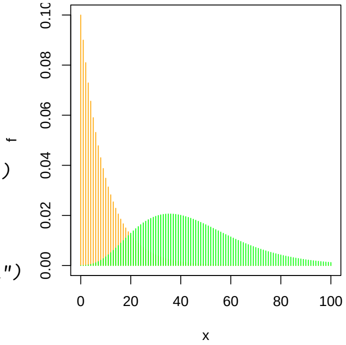
$$T - s \sim NegBin(s, p)$$

- if the last draw is a success or a failure (we count s success and we want to know the distribution of the number of draws) :

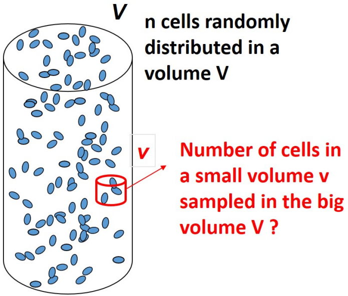
$$T_{bis} - s \sim NegBin(s + 1, p)$$

Negative binomial distribution in R
`dnbinom(x, size = s, prob = p)`

```
x <- 0:100
#
# s = 1, p = 0.1
f <- dnbinom(x, size = 1, prob = 0.1)
plot(x, f, type = "h", col = "orange")
#
# s = 5, p = 0.1
f <- dnbinom(x, size = 5, prob = 0.1)
points(x, f, type = "h", col = "green")
```



Poisson distribution on an example from microbiology



Distribution of R the number of cells in v :

By analogy with the Bernoulli process, with cells randomly placed in V, a succes if the cell is in v and n draws with a success probability $= \frac{v}{V}$,

$R \sim Binom(p = \frac{v}{V}, n)$

Asymptotic properties:

$R \rightarrow Poisson(\lambda = n \times \frac{v}{V})$ for large n

$Pois(\lambda) \rightarrow N(\lambda, \sqrt{\lambda})$ for large λ

Note that λ is both the mean and the variance of the distribution.

Poisson process - classical definition

Number of occurences of an event in a time interval,
with p **the probability of this event in a small time interval**

- **proportional** to the interval size,
- **independent** from the occurence of the same event in another time interval (as soon as there is no overlap of both intervals).

Distribution of N the number of occurences of the event in the interval δt,

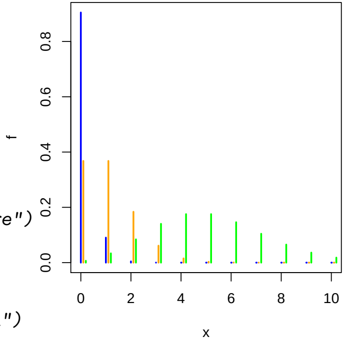
$$N \sim Pois(\lambda = \delta t \times I)$$

with I named the intensity of the process.

Poisson distribution in **R**

dpois(x, lambda = λ)

```
x <- 0:10
#
# lambda = 0.1
f <- dpois(x, lambda = 0.1)
plot(x, f, type = "h", col = "blue")
#
# lambda = 1
f <- dpois(x, lambda = 1)
points(x, f, type = "h", col = "orange")
#
# lambda = 5
f <- dpois(x, lambda = 5)
points(x, f, type = "h", col = "green")
```



Exponential distribution

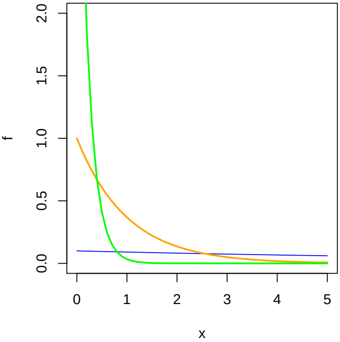
For a Poisson process of mean λ
($\lambda =$ mean number of events per time, surface or volume unit),
x **the time, surface or volume needed to observe one event:**

$$x \sim Exp(\lambda)$$

The mean of this distribution is $\frac{1}{\lambda}$ and its variance is $\frac{1}{\lambda^2}$.

Exponential distribution in R
dexp(x, rate = λ)

```
x <- seq(0, 5, 0.1)
#
# lambda = 0.5
f <- dexp(x, rate =  0.1)
plot(x, f, type = "l", col = "blue",
     ylim = c(0,2))
#
# lambda = 1
f <- dexp(x, rate =  1)
lines(x, f, col = "orange")
#
# lambda = 5
f <- dexp(x, rate =  5)
lines(x, f, col = "green")
```



Gamma distribution

For the same Poisson process of mean λ
x **the time, surface or volume needed to observe** α **events:**

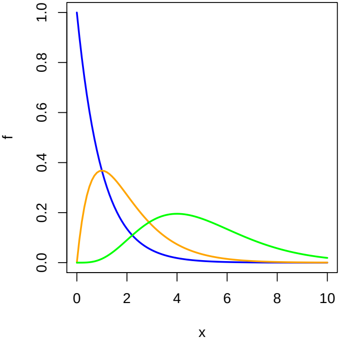
$$x \sim Gamma(\alpha, \lambda)$$

α is the shape parameter,
$\beta = \frac{1}{\lambda}$ is the scale parameter.
The mean of this distribution is $\frac{\alpha}{\lambda}$ and its variance is $\frac{\alpha}{\lambda^2}$.

Gamma distribution in R

dgamma(x, shape = α, rate = λ)

```
x <- seq(0, 10, 0.1)
#
# alpha = 1, lambda = 1
f <- dgamma(x, shape = 1, rate =  1)
plot(x, f, type = "l", col = "blue")
#
# alpha = 2, lambda = 1
f <- dgamma(x, shape = 2, rate =  1)
lines(x, f, col = "orange")
#
# alpha = 5, lambda = 1
f <- dgamma(x, shape = 5, rate =  1)
lines(x, f, col = "green")
```



Go back to the negative binomial distribution

The negative binomial distribution is classically used to model overdispersion in the Poisson model.

The negative binomial distribution is also called the Gamma-Poisson, as it corresponds to the mixture of a Poisson distribution and a Gamma distribution:

Poisson distribution of parameter λ, with λ following a Gamma distribution.

Overview of distributions based on stochastic processes

- **Bernoulli process**
  - success for one draw : Bernoulli distribution
  - number of success for n draws with replacement: binomial distribution
  - number of failures before 1 success: geometric distribution
  - number of failures before s success: negative binomial distribution
- **Poisson process**
  - number of cells in a small volume: Poisson distribution
  - volume to observe one cell: exponential distribution
  - volume to observe α cells: gamma distribution

Normal (Gaussian) distribution

Often used especially due to the central limit theorem:

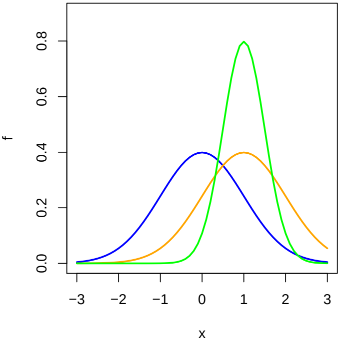
$$x \sim N(\mu, \sigma)$$

BE CAREFUL, do not forget it is defined on $]-\infty, +\infty[$ and thus can generate negative values even with a small standard deviation in comparison to the mean, a soon as the number of random draws is high (truncation in simulation may be necessary in some cases).

Normal distribution in R

dnorm(x, mean = μ, sd = σ)

```
x <- seq(-3, 3, 0.1)
#
# mu = 0, sigma = 1
f <- dnorm(x, mean = 0, sd = 1)
plot(x, f, type = "l", col = "blue",
     ylim = c(0, 0.91))
#
# mu = 1, sigma = 1
f <- dnorm(x, mean = 1, sd = 1)
lines(x, f, col = "orange")
#
# mu = 1, sigma = 0.5
f <- dnorm(x, mean = 1, sd = 0.5)
lines(x, f, col = "green")
```



Lognormal distribution

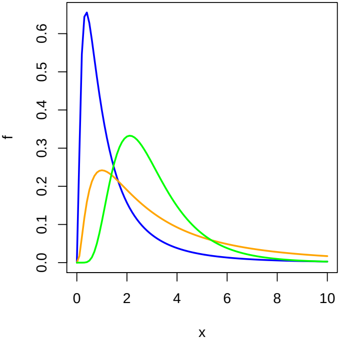
Asymetrical positive distribution often used when x is varying on different orders of magnitude (e.g. 10^2, 10^3, 10^9, ..., as a microbial concentration for example)

$$x \sim LN(\mu_l, \sigma_l) \Leftrightarrow \ln(x) \sim N(\mu_l, \sigma_l)$$

BE CAREFUL, parameters of the lognormal distribution, μ_l and σ_l, correspond to mean and standard deviation in the natural logarithm scale.

Lognormal distribution in R
dlnorm(x, meanlog = μ_l, sdlog = σ_l)

```
x <- seq(0, 10, 0.1)
#
# mu = 0, sigma = 1
f <- dlnorm(x, mean = 0, sd = 1)
plot(x, f, type = "l", col = "blue")
#
# mu = 1, sigma = 1
f <- dlnorm(x, mean = 1, sd = 1)
lines(x, f, col = "orange")
#
# mu = 1, sigma = 0.5
f <- dlnorm(x, mean = 1, sd = 0.5)
lines(x, f, col = "green")
```



Student distributions

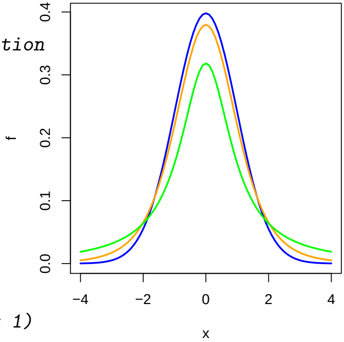
Symetrical distributions defined on $]-\infty, +\infty[$ with heavy tails, heavier than those of the normal distribution for low degrees of freedom ν:

$$x \sim T(\mu, \sigma, \nu)$$

The Cauchy distribution is the one with the heaviest tails ($\nu = 1$).

Student distributions in R with $\mu = 0$ et $\sigma = 1$ `dt(x, df =` ν`)`

```
x <- seq(-4, 4, 0.1)
#
# nu = 100 - close to normal distribution
f <- dt(x, df = 100)
plot(x, f, type = "l", col = "blue")
#
# nu = 5
f <- dt(x, df = 5)
lines(x, f, col = "orange")
#
# nu = 1 - Cauchy distribution
f <- dt(x, df = 1)
# equivalent alternative
f <- dcauchy(x, location = 0, scale = 1)
lines(x, f, col = "green")
```



Beta distribution

Flexible distribution defined on $]0, 1[$,
symetrical only if both parameters are identical.

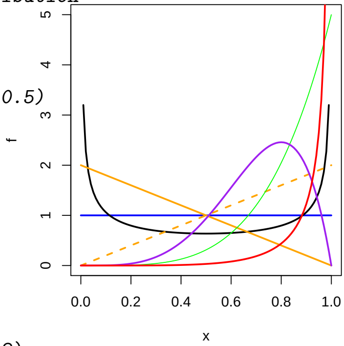
$$x \sim Beta(\alpha, \beta)$$

Its mean is $\frac{\alpha}{\alpha+\beta}$ and its variance $\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$.

The beta(1, 1) distribution corresponds to the uniform distribution Unif(0,1).

Beta distribution in R
dbeta(x, shape1 = α, shape2 = β)

```
x <- seq(0, 1, 0.01)
# alpha = 1, beta = 1 - uniform distribution
f <- dbeta(x, shape1 = 1, shape2 = 1)
plot(x, f, type = "l", col = "blue",
     ylim = c(0,5))
f <- dbeta(x, shape1 = 0.5, shape2 = 0.5)
lines(x, f, col = "black")
f <- dbeta(x, shape1 = 1, shape2 = 2)
lines(x, f, col = "orange")
f <- dbeta(x, shape1 = 2, shape2 = 1)
lines(x, f, col = "orange", lty = 2)
f <- dbeta(x, shape1 = 5, shape2 = 1)
lines(x, f, col = "green")
f <- dbeta(x, shape1 = 5, shape2 = 2)
lines(x, f, col = "purple")
f <- dbeta(x, shape1 = 5, shape2 = 0.2)
lines(x, f, col = "red")
```



Overview of previous distributions

- **Distributions based on the Gaussian one**
  - **normal / Gaussian**
  - **lognormal**
    defined on $]0; +\infty[$
- **Distributions based on the Student one**
  - **Student**
  - **Cauchy**
    Student distribution with the degree of freedom equal to one (distribution with heavy tails)
- **Beta distribution**
  defined on $]0; 1[$ and very flexible

It is good to know those classical distributions !

The knowledge of those few **classical distributions** and **stochastic processes** will help you **to build and implement models** in the context of **Bayesian inference**, but could also help you to **prevent misuses of classical models in frequentist inference**.