



**HAL**  
open science

## Standard statistical inference

Christophe Dutang

► **To cite this version:**

Christophe Dutang. Standard statistical inference. Arthur Charpentier. Computational Actuarial Science with R, Chapter 2, CRC press, <https://www.routledge.com/Computational-Actuarial-Science-with-R/Charpentier/p/book/9781466592599>, 2015, Chapman–Hall CRC-The-R-Series. hal-04723337

**HAL Id: hal-04723337**

**<https://hal.science/hal-04723337v1>**

Submitted on 7 Oct 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Chapter 2: Standard statistical inference

Christophe Dutang, *Université de Strasbourg*

March 2013

## 1 Probability distributions in actuarial science

Let  $X$  be our quantity of interest. Actuarial models rely on particular assumptions on the probability distribution of  $X$ . When  $X$  represents the claim amount or the life of an individual, one expects  $X$  to have a distribution on  $\mathbb{R}_+$ , whereas  $X$  represents the claim number, we deal with distribution on  $\mathbb{N}$ . But, characterizing the support of the random variable  $X$  is a necessary but not sufficient step to characterize our quantity of interest.

In the discrete case, probability distributions are generally characterized by the mass probability function  $p_X$  or the “elementary” probabilities:  $p_X(x) = \mathbb{P}(X = x)$  for  $x \in \mathbb{N}$ . In the continuous case, we define the probability distribution by its density  $f_X(x)$ , being the infinitesimal version of  $p_X$  such that  $f_X(x)dx = \mathbb{P}(X \in [x, x + dx])$ . A third case is when the random variable has both continuous and discrete parts, for which there is no proper density. In such a case, we define the distribution with the cumulative distribution function  $F_X(x) = \mathbb{P}(X \leq x)$ . We recall that for a discrete distribution on  $\mathbb{N}$ ,  $F_X(x) = \sum_{n=0}^{\lfloor x \rfloor} p_X(n)$ , while for a continuous distribution  $F_X(x) = \int_{-\infty}^x f_X(y)dy$ .

The purpose of this section is to present the most common distributions used in actuarial sciences, being continuous, discrete or mixed-type. As always in this book, a special emphasis is put on how this topic is implemented and can be extended in R.

### 1.1 Continuous distributions

There are a lot of ways to classify and to distinguish distributions. We present here the Pearson system and the exponential family, the latter being e.g. used in generalized linear models (GLM). Pearson (1895) consider the family of continuous distributions such that the density function  $f_X$  verifies the following ordinary differential equation

$$\frac{1}{f_X(x)} \frac{df_X(x)}{dx} = -\frac{a+x}{c_0+c_1x+c_2x^2},$$

where  $a, c_0, c_1, c_2$  are constants. Let  $p(x) = c_0 + c_1x + c_2x^2$ . The solution is defined up to a constant  $K$  which is derived by the constraint  $\int_{\mathbb{R}} f_X(x)dx = 1$ . Type 0 is obtained when  $c_1 = c_2 = 0$ : we get  $f_X(x) = Ke^{-(2a+x)x/(2c_0)}$ , which is the normal distribution. Type 1 is the case where the polynomial  $c_0 + c_1x + c_2x^2$  has two distinct real roots  $a_1$  and  $a_2$  such that  $a_1 < 0 < a_2$ : we get  $f_X(x) = K(x - a_1)^{m_1}(a_2 - x)^{m_2}$ . We recognize the beta distribution. Type 2 corresponds to the case where  $m_1 = m_2 = m$ .

Type 3 is obtained when  $c_2 = 0$  leading a first-order polynomial  $c_0 + c_1x$ . In this case, we get the gamma distribution with  $f_X(x) = K(c_0 + c_1x)^m e^{x+c_1}$ . Type 4 corresponds to the case where the polynomial  $p(x) = c_0 + c_1x + c_2x^2$  has no real roots, in which case  $p(x) = C_0 + c_2(x + C_1)^2$ . We get  $f_X(x) = K(C_0 + c_2(x + C_1)^2)e^{k \tan^{-1}((x+c_1)/\sqrt{c_0/c_2})}$ , which is closely linked to the generalized inverse Gaussian distribution of Barndorff-Nielsen.

We get type 5 when  $p$  is a perfect square, i.e.  $p(x) = (x + C_1)^2$ . The associated density is  $f_X(x) = K(x + C_1)^{-1/c_2} e^{k/(x+C_1)}$ . Two special cases are obtained when  $k = 0, c_2 > 0$  for type 8 and  $c_2 < 0$  for type 9.

Type 6 is obtained when  $p$  has two real roots  $a_1, a_2$  of the same sign for which we get  $f_X(x) = K(x - a_1)^{m_1}(x - a_2)^{m_2}$ , a generalized Beta distribution. Finally type 7 is obtained when  $a = c_1 = 0$ , leading to  $f_X(x) = K(c_0 + c_2x^2)^{-1/(2c_2)}$ .

Those distributions are implemented in the package **PearsonDS**. In Figure 1, we plot the densities for the first seven types in order to compare the different possible shapes.

```
> library(PearsonDS)
> x <- seq(-1, 6, by=1e-3)
> y0 <- dpearson0(x, 2, 1/2)
> y1 <- dpearsonI(x, 1.5, 2, 0, 2)
> y2 <- dpearsonII(x, 2, 0, 1)
> y3 <- dpearsonIII(x, 3, 0, 1/2)
> y4 <- dpearsonIV(x, 2.5, 1/3, 1, 2/3)
> y5 <- dpearsonV(x, 2.5, -1, 1)
> y6 <- dpearsonVI(x, 1/2, 2/3, 2, 1)
> y7 <- dpearsonVII(x, 3, 4, 1/2)
> plot(x, y0, type="l", ylim=range(y0, y1, y2, y3, y4, y5, y7),
+      ylab="f(x)", main="The Pearson distribution system")
> lines(x[y1 != 0], y1[y1 != 0], lty=2)
> lines(x[y2 != 0], y2[y2 != 0], lty=3)
> lines(x[y3 != 0], y3[y3 != 0], lty=4)
> lines(x, y4, col="grey")
> lines(x, y5, col="grey", lty=2)
> lines(x[y6 != 0], y6[y6 != 0], col="grey", lty=3)
> lines(x[y7 != 0], y7[y7 != 0], col="grey", lty=4)
> legend("topright", leg=paste("Pearson", 0:7), lty=1:4,
+      col=c(rep("black", 4), rep("grey", 4)))
```

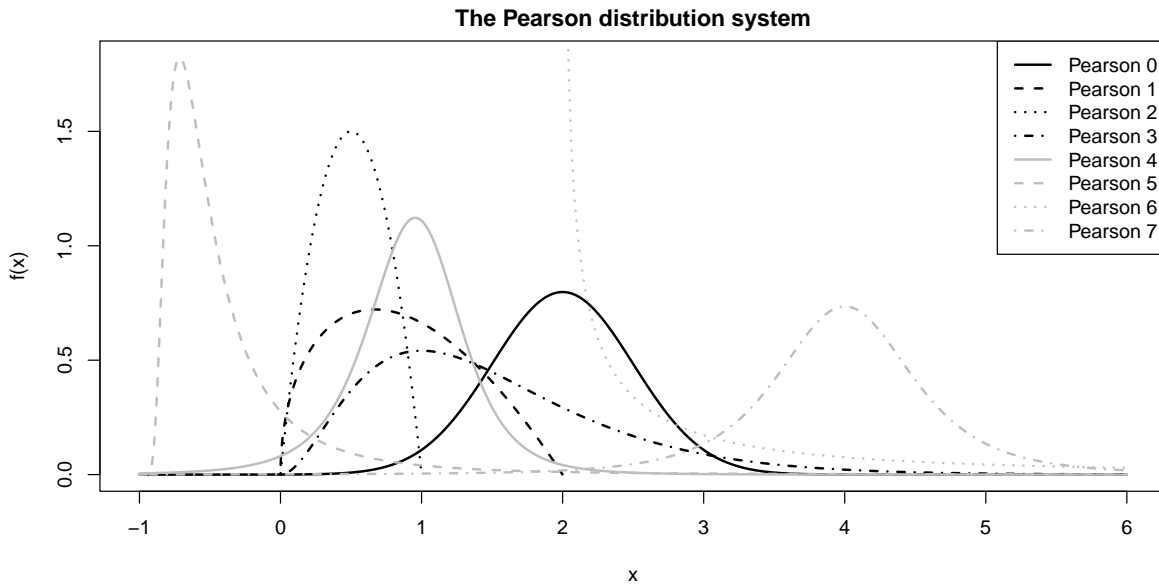


Figure 1: Pearson's distribution system

Another important class of distribution is the exponential family, that traces back to three independent works of Pitman, Darrois and Koopman in the mid 30's. This family contains distributions where the density function can be written as

$$f_X(x) = \exp \left( \sum_{j=1}^d a_j(x) \alpha_j(\theta) + b(x) + \beta(\theta) \right),$$

where  $\theta \in \mathbb{R}^d$  is the  $d$ -dimensional parameter vector,  $a_j, \alpha_j, b$  and  $\beta_j$  are known functions (see Bickel &

Doksum (2001) for a recent survey). The exponential family include many familiar distributions. We recover the exponential distribution, i.e.  $f_X(x) = \lambda e^{-\lambda x}$  with  $d = 1$ ,  $a(x) = x$ ,  $\alpha(x) = \lambda$ ,  $b(x) = 0$  and  $\beta(\lambda) = \log(\lambda)$ , the normal distribution, i.e.  $f_X(x) = e^{-(x-\mu)^2/(2\sigma^2)}/\sqrt{2\pi\sigma^2}$ , with  $d = 2$ ,  $a_1(x) = x^2$ ,  $\alpha_1(m, \sigma^2) = -1/(2\sigma^2)$ ,  $a_2(x) = x$ ,  $\alpha_2(m, \sigma^2) = m/\sigma^2$ ,  $b(x) =$  and  $\beta(m, \sigma^2) = -m/(2\sigma^2) - \log \sqrt{2\pi\sigma^2}$ . In the exponential family, the gamma and the inverse Gaussian distributions are also examples of particular interest in actuarial science.

In R, each probability distribution is implemented by a set of four functions and a particular root name `foo`: `dfoo` computes the density function  $f_X(x)$  or the mass probability function  $p_X(x)$ , `pfoo` the cumulative distribution function  $F_X(x)$ , `qfoo` the quantile function  $F_X^{-1}(x)$  and `rfoo` the random number generator. For instance, the gamma distribution with density  $f_X(x) = \lambda^\alpha x^{\alpha-1} e^{-\lambda x} / \Gamma(\alpha)$  is implemented in `dgamma`, `pgamma`, `qgamma` and `rgamma`, see example below.

```
> dgamma(1:2, shape=2, rate=3/2) #f_X(1), f_X(2)
```

```
[1] 0.5020429 0.2240418
```

```
> pgamma(1:2, shape=2, rate=3/2) #F_X(1), F_X(2)
```

```
[1] 0.4421746 0.8008517
```

```
> qgamma(1/2, shape=2, rate=3/2) #median
```

```
[1] 1.118898
```

```
> rgamma(5, shape=2, rate=3/2) #five random numbers
```

```
[1] 1.0377253 1.0755764 0.2381343 1.7131438 0.5217060
```

In Table 1.1, the continuous distributions implemented in R are listed. This set of distributions is rather limited, and in practice, other distributions such as Pareto are particularly relevant in actuarial science. Most of distributions are generally implemented in a dedicated package. The full list of non R-base distributions are listed on the corresponding task view: <http://cran.r-project.org/web/views/Distributions.html>. Among the numerous packages, two packages focus on distributions relevant to actuarial science : **actuar** and **ActuDistns**. Note that **actuar** provides the raw moment  $E(X^k)$ , the limited expected values  $E(\min(X, l)^k)$  and the moment generating functions  $E(e^{tX})$  for many distributions in three dedicated functions `mfoo`, `levfoo` and `mgffoo`.

Table 1: Continuous distributions in R.

Probability distribution	Root	Probability distribution	Root
beta	<b>beta</b>	logistic	<b>logis</b>
Cauchy	<b>cauchy</b>	lognormal	<b>lnorm</b>
chi-2	<b>chisq</b>	normal	<b>norm</b>
exponential	<b>exp</b>	Student t	<b>t</b>
Fisher F	<b>f</b>	uniform	<b>unif</b>
gamma	<b>gamma</b>	Weibull	<b>weibull</b>

When on a particular problem all classical distributions have been exhausted, it is sometimes appropriate to create new probability distributions. Typical transformations of a random variable  $X$  are listed : (i) translation  $X + c$  (e.g. the shifted lognormal distribution), (ii) scaling  $\lambda X$ , (iii) power  $X^\alpha$  (e.g. the generalized beta type 1 distribution), (iv) inverse  $1/X$  (e.g. the inverse gamma distribution), (v) the logarithm  $\log(X)$  (e.g. the loglogistic distribution), (vi) exponential  $\exp(X)$  and (vii) the odds ratio  $X/(1 - X)$  (e.g. the beta type 2 distribution). With the small code below, we test all transformations (except the last one) on gamma-distributed variables.

```

> set.seed(123)
> x <- rgamma(100, 2)
> x1 <- x+1
> x2 <- 2*x
> x3 <- sqrt(x)
> x4 <- 1/x
> x5 <- log(x)
> x6 <- exp(x)
> plot(density(x), ylim=c(0, 1), xlim=c(0, 10), main="Empirical densities",
+       lwd=2, xlab="x", ylab="f_X(x)")
> lines(density(x1), lty=2, lwd=2)
> lines(density(x2), lty=3, lwd=2)
> lines(density(x3), lty=4, lwd=2)
> lines(density(x4), lty=1, col="grey", lwd=2)
> lines(density(x5), lty=2, col="grey", lwd=2)
> lines(density(x6), lty=3, col="grey", lwd=2)
> legend("topright", lty=1:4, col=c(rep("black", 4), rep("grey", 3)),
+       leg=c("X", "X+1", "2X", "sqrt(X)", "1/X", "log(X)", "exp(X)"))

```

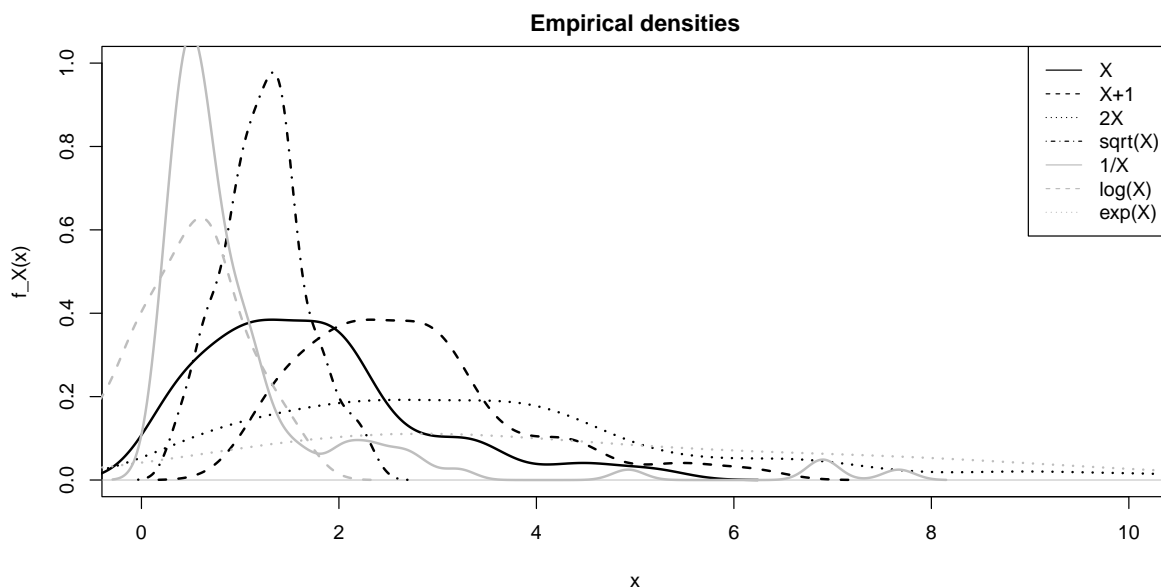


Figure 2: Distribution transformations

In Figure 2, we plot the empirical densities (as estimated by the `density` function). Note that the exponential transformation has a heavy-tailed distribution and only the right-tail is shown on the graphic. With these transformations in mind, we can now list the set of distributions generally used in actuarial science.

The most valuable (yet the simplest) distribution with finite-support is the uniform distribution with a density  $f_X(x) = \mathbb{1}_{[0,1]}(x)$ . The uniform distribution is always used for non-uniform random generation as the random variable  $F_Y(U)^{-1}$  with  $U$  a uniform variable has the same distribution as  $F_Y$ . Another particular used distribution is the beta distribution defined as

$$f_X(x) = \frac{x^{a-1}(1-x)^{b-1}}{\beta(a,b)} \mathbb{1}_{[0,1]}(x) \text{ and } F_X(x) = \frac{\beta(a,b,x)}{\beta(a,b)}.$$

where  $\beta(\cdot, \cdot)$  is the beta function and  $\beta(\cdot, \cdot, \cdot)$  is the incomplete lower beta function, see Olver et al. (2010). When  $a = b = 1$ , we get back to the uniform distribution, i.e.  $f_X$  is constant. When  $a, b < 1$ , the density  $f_X$  is U-shaped, whereas for  $a, b > 1$  the density is unimodal. A monotone density is obtained when  $a$  and  $b$  have opposite signs. Both of these distributions are implemented in R, see `?dunif` and `?dbeta`. By appropriate scaling and shifting, i.e.  $c + (d - c)X$ , a distribution on any interval  $[c, d]$  can be obtained. Finally, another

important distribution  $Tr(a, b, c)$  is the triangular distribution given by

$$f_X(x) = \frac{2(x-a)}{(b-a)(c-a)} \mathbb{1}_{[a,c]}(x) + \frac{2(x-a)}{(b-a)(c-a)} \mathbb{1}_{]c,b]}(x),$$

which as his name suggest has a triangular-shaped density. When  $b = (a+c)/2$ , the triangular is the sum of two uniform variates on interval  $[a, b]$ . The triangular distribution is available the **triangle** package.

As presented in Klugman et al. (2009), the two main families of positive continuous distributions are the gamma-transformed family and the beta-transformed family. Let  $X$  follow a gamma distribution  $Gamma(\alpha, 1)$ . The gamma-transformed family is the distribution of  $Y = X^{1/\tau}/\lambda$  for  $\tau > 0$ , which has the following density and distribution functions

$$f_Y(y) = \frac{\lambda^{\tau\alpha}}{\Gamma(\alpha)} \tau y^{\alpha\tau-1} e^{-(\lambda y)^\tau} \text{ and } F_Y(y) = \Gamma(\alpha, (\lambda y)^\tau) / \Gamma(\alpha),$$

where  $\Gamma(\cdot, \cdot)$  denotes the incomplete lower gamma function, see e.g. Olver et al. (2010). When  $\tau < 0$ , we get the inverse gamma-transformed family. Let  $\tau^* = -\tau$ . The density and the distribution function of  $Y = 1/(\lambda X^{1/\tau^*})$  are given by

$$f_Y(y) = \frac{\tau^* e^{-(\lambda y)^{-\tau^*}}}{\lambda^{\tau^*\alpha} y^{\alpha\tau^*+1} \Gamma(\alpha)} \text{ and } F_Y(y) = 1 - \Gamma(\alpha, (\lambda y)^{-\tau^*}) / \Gamma(\alpha).$$

On Figure 3, we list the different special cases of the transformed gamma distribution and their relationships.

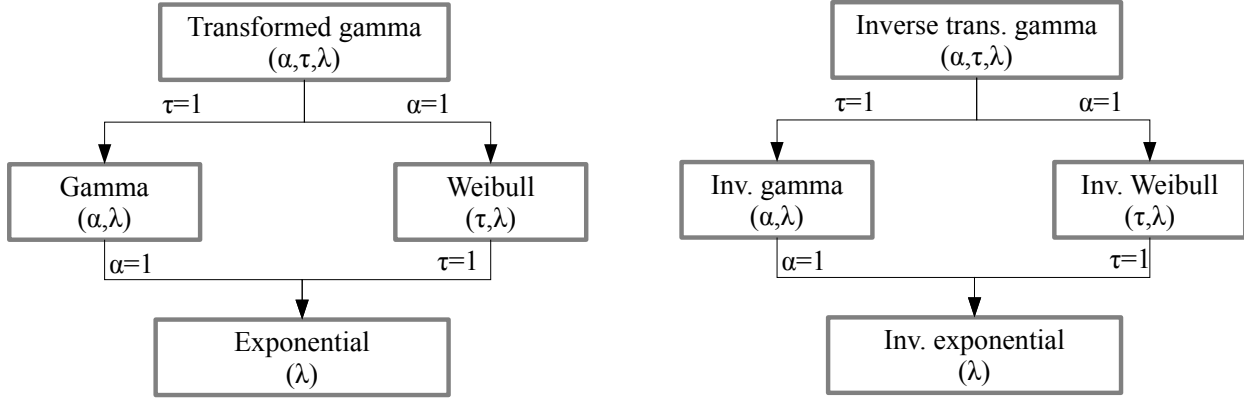


Figure 3: Transformed gamma family

The beta-transformed family is based on the beta distribution of the second kind (or type II), i.e. the distribution of  $X/(1-X)$  when  $X$  follows a beta distribution of type I, see the previous subsection. The beta distribution of type II has a density  $f_X(x) = \frac{x^{a-1}}{\beta(a,b)(1+x)^{a+b}}$ . Renaming  $a = \alpha$  and  $b = \tau$ , the transformed beta is the distribution of  $Y = \theta X^{1/\gamma}$  and has the following density and distribution function

$$f_Y(y) = \frac{1}{\beta(\alpha, \tau)} \frac{\gamma(y/\theta)^{\gamma\tau}}{y(1+(y/\theta)^\gamma)^{\alpha+\tau}} \text{ and } F_Y(y) = \frac{\beta(\alpha, \tau, \frac{x}{1+x})}{\beta(\alpha, \tau)},$$

where  $x = (y/\theta)^\gamma$  and  $\beta(\cdot, \cdot, \cdot)$  denotes the incomplete lower beta function. These two families are available in the **actuar** package.

## 1.2 Discrete distributions

The Sundt  $(a, b, 0)$  family of distributions is the set of distributions verifying

$$\frac{\mathbb{P}(X = k+1)}{\mathbb{P}(X = k)} = a + \frac{b}{k},$$

for  $k \in \mathbb{N}$  and  $a, b \geq 0$  positive parameters. This recurrence equation can be seen a simplified discrete equation of the Pearson system (see Johnson et al. (2005)). We get back to the binomial distribution  $\mathcal{B}(n, p)$

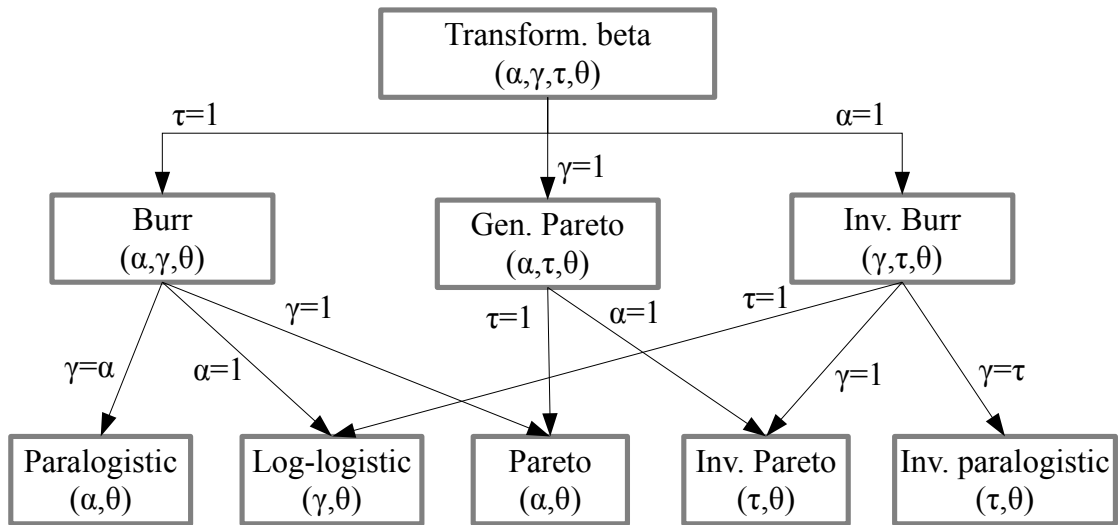


Figure 4: Transformed beta family

with  $a = -p/(1-p)$  and  $b = p(n+1)/(1-p)$ , the Poisson distribution  $\mathcal{P}(\lambda)$  with  $a = 0$  and  $b = \lambda$ , the negative binomial distribution  $\mathcal{BN}(m, p)$  with  $a = 1-p$  and  $b = (1-p)(m-1)$ . A generalization of the  $(a, b, 0)$  family is obtained by truncating the values smaller than  $n$ . Thus, the  $(a, b, n)$  family verifies

$$p_X(k) = p_X(k-1) \left( a + \frac{b}{k} \right) \mathbb{1}_{(k \geq n)}.$$

Furthermore, the exponential family also models discrete distributions by considering the mass probability function  $p_X$  verifies

$$p_X(k) = \exp \left( \sum_{j=1}^d a_j(k) \alpha_j(\theta) + b(k) + \beta(\theta) \right).$$

It includes many familiar distributions: the Bernoulli distribution with  $d = 1$ ,  $a(x) = x$ ,  $\alpha(p) = \log(p/(1-p))$ ,  $b(x) = 0$  and  $\beta(p) = \log(1-p)$ , the Poisson distribution with  $d = 1$ ,  $a(x) = x$ ,  $\alpha(\lambda) = \lambda$ ,  $b(x) = -\log(x!)$ ,  $\beta(\lambda) = -\lambda$ . Notably, the negative binomial distribution does not belong to the exponential family.

As for continuous distributions, discrete distributions are implemented in four functions: `dfoo` computes the mass probability function  $p_X$ , `pf00` the cumulative distribution function  $F_X$ , `qfoo` the quantile function  $F_X^{-1}$  and `rfoo` the random number generator. For instance, the poisson distribution is implemented in `dpois`, etc... Here is a standard call.

```
> dpois(0:2, lambda=3) #P(X=0, 1, 2)
```

```
[1] 0.04978707 0.14936121 0.22404181
```

```
> ppois(1:2, lambda=3) #F_X(1), F_X(2)
```

```
[1] 0.1991483 0.4231901
```

```
> qpois(1/2, lambda=3) #median
```

```
[1] 3
```

```
> rpois(5, lambda=3) #five random numbers
```

Common transformations of an integer-valued random variable  $X$  are listed: (i) translation  $X + m$  for a non-null integer  $m$  (e.g. the shifted Poisson distribution), (ii) scaling  $mX$ , (iii) zero-inflation  $(1 - B)X$  where  $B$  follows a Bernoulli distribution  $\mathcal{B}(q)$  and (iv) zero-modification  $(1 - B)(X + 1)$  where  $B$  follows a Bernoulli distribution. The resulting mass probability function for the transformed variable  $Y$  (i.e. one transformation at a time) is (i)  $\mathbb{P}(Y = k) = \mathbb{P}(X = k - m)$  for  $k \geq m$ , (ii)  $\mathbb{P}(Y = k) = \mathbb{P}(X = k/m)$  for  $k = 0, m, 2m, 3m, \dots$ , (iii)  $\mathbb{P}(Y = 0) = q + (1 - q)\mathbb{P}(X = 0)$  and  $\mathbb{P}(Y = k) = (1 - q)\mathbb{P}(X = k)$  for  $k \geq 1$ , (iv)  $\mathbb{P}(Y = 0) = q$  and  $\mathbb{P}(Y = k) = (1 - q)\mathbb{P}(Y = k - 1)$  for  $k \geq 1$ . The zero-modification and the zero-inflation are useful to add a parameter to standard discrete distributions, e.g. the Poisson distribution. A particular case of the zero-modification is the zero-truncation when the variable  $B$  equals 0 almost surely.

Such transformations are implemented in special packages, see the task view, but can be easily implemented. The zero-modification is presented below.

```
> dpoisZM <- function(x, prob, lambda)
+   prob*(x == 0) + (1-prob)*(x > 0)*dpois(x-1, lambda)
> ppoisZM <- function(q, prob, lambda)
+   prob*(q >= 0) + (1-prob)*(q > 0)*ppois(q-1, lambda)
> qpoisZM <- function(p, prob, lambda)
+   ifelse(p <= prob, 0, 1+qpois((p-prob)/(1-prob), lambda))
> rpoisZM <- function(n, prob, lambda)
+   (1-rbinom(n, 1, prob))*(rpois(n, lambda)+1)
> x <- rpoisZM(100, 1/2, 3)
> plot(ecdf(x), main="Zero-modified Poisson(prob=1/2, lam=3)")
> lines(z <- sort(c(0:12, 0:12-1e-6)),
+       ppoisZM(z, 1/2, 3), col="grey", lty=4, lwd=2)
> legend("bottomright", lty=c(1,4), lwd=1:2,
+       col=c("black", "grey"), leg=c("empir.", "theo."))
```

On Figure 5, we plot the empirical cumulative distribution function (sample size of 100) of a zero-modified Poisson distribution and compare it with the true distribution function.

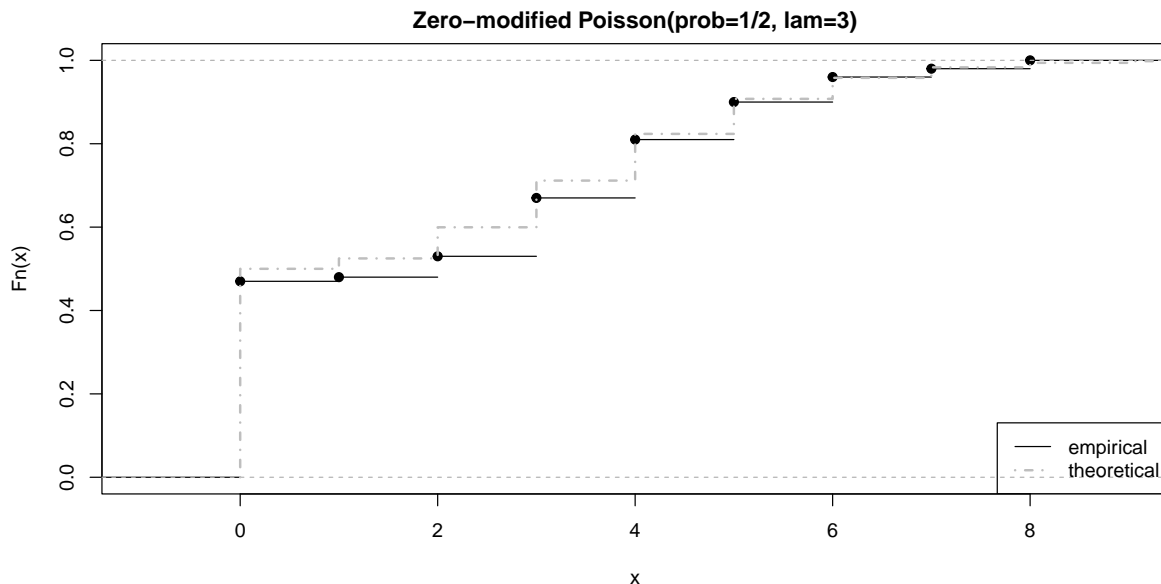


Figure 5: Distribution transformations

The main discrete distributions are the binomial  $\mathcal{B}(n, p)$ , the Poisson  $\mathcal{P}(\lambda)$  and the negative binomial  $\mathcal{Bn}(m, p)$  distributions, for which we recall the mass probability function  $p_X(x)$

$$\mathbb{P}(X = k) = \binom{n}{k} p^k (1 - p)^{n-k},$$



for  $p \in [0, 1]$  (the Bernoulli distribution is obtained with  $n = 1$ ).

$$\mathbb{P}(X = k) = \frac{\lambda^k}{k!} e^{-\lambda},$$

for  $\lambda > 0$ , and

$$\mathbb{P}(X = k) = \binom{m+k-1}{k} p^m (1-p)^k,$$

for  $p \in [0, 1]$ . The discrete analog of the Pareto distribution is the Zipf distribution whose mass probability function is given by

$$\mathbb{P}(X = k) = \frac{k^\eta}{\zeta(\eta)},$$

where  $\zeta(\cdot)$  is the zeta's Riemann function, see Olver et al. (2010).

### 1.3 Mixed-type distributions

Mixed-type distributions are distributions of random variables that are neither continuous nor discrete, i.e.  $0 < \sum_{x \in D_X} (F_X(x) - F_X(x_-)) < 1$  for  $D_X$  the set of discontinuities where the lower bound corresponds to continuous distributions and the upper bound discrete distributions. Thus, the distribution function has discontinuities and continuous parts. A first example of mixed-type distribution is the zero-modified gamma distribution which has the distribution function

$$F_X(x) = p \mathbb{1}_{x \geq 0} + (1-p) \frac{\Gamma(\alpha, \lambda x)}{\Gamma(\alpha)},$$

where  $\Gamma(\cdot, \cdot)$  denotes the incomplete gamma function.  $X$  has an improper density function  $f_X(x) = (1-p)\lambda^\alpha x^{\alpha-1} e^{-\lambda x} / \Gamma(\alpha)$ . In a similar way, zero-modified Pareto or zero-modified lognormal distributions can be defined.

An application of mixed-type distributions to destruction rate models is now presented. Destruction rate models focus on the distribution  $X = L/d$  where  $L$  is the loss amount and  $d$  the maximum possible loss (as defined in the insurance terms). By definition,  $X$  is bounded to the interval  $[0, 1]$ , and may have a mass at 1 when the object insured is entirely destroyed, as well as at 0 when there is no claim. In the application that will follow, we will consider the one-modified beta and the MBBEFD distributions. The one-modified beta is the distribution of  $X = BY$  where  $Y$  follows a beta distribution  $\mathcal{B}(a, b)$  and  $B$  follows a Bernoulli distribution  $\mathcal{B}(q)$ . The distribution function is given by

$$F_X(x) = \frac{\beta(a, b, x)}{\beta} (1-q) + q \mathbb{1}_{x \geq 1},$$

for which the improper density is  $f_X(x) = (1-q)x^{a-1}(1-x)^{b-1} / \beta(a, b)$ . In R, we define it as

```
> dbetaOM <- function(x, prob, a, b)
+   dbeta(x, a, b)*(1-prob)*(x != 1) + prob*(x == 1)
> pbetaOM <- function(q, prob, a, b)
+   pbeta(q, a, b)*(1-prob) + prob*(q >= 1)
```

The Maxwell-Boltzmann Bore-Einstein Fermi-Dirac (MBBEFD) distribution was introduced and popularized by Bernegger (1997) in the context of reinsurance treaties for which the pricing may be done on exposition curve (rather than on frequency/severity models). The distribution function is given by

$$F_X(x) = a \left( \frac{a+1}{a+b^x} - 1 \right) \mathbb{1}_{[0,1[}(x) + \mathbb{1}_{[1,+\infty[}(x),$$

where  $(a, b) \in ]-1, 0[ \times ]1, +\infty[$  or  $(a, b) \in (\mathbb{R} \setminus [-1, 0]) \times ]0, 1[$ . Note that there is a probability mass at 1, since  $\mathbb{P}(X = 1) = (a+1)b/(a+b) = q$ . The improper density function is  $f_X(x) = \frac{-a(a+1)b^x \ln(b)}{(a+b^x)^2} \mathbb{1}_{]0,1[}(x)$ . At the time this book is written, there is no package implementing the MBBEFD distribution, but this can be remedied by the following lines

```

> dMBBEFD <- function(x, a, b)
+   -a*(a+1)*b^x*log(b)/(a + b^x)^2 + (a+1)*b/(a+b)*(x == 1)
> pMBBEFD <- function(x, a, b)
+   a * ( (a+1) / (a + b^x) - 1) * (x < 1) + 1 * (x >= 1)

```

Those two distributions will be used in the subsequent section on destruction rate data.

## 1.4 Mixture of distributions

Mixing distributions consists in randomly drawing a distribution among a finite set of distributions. Consider a set of distribution functions  $F_1, \dots, F_p$  and a set of weights  $\omega_1, \dots, \omega_p \in [0, 1]$ . The choice  $C$  of a distribution is such that  $\mathbb{P}(C = i) = \omega_i$  for  $i = 1, \dots, p$ . The random generation process given by (i) draw  $C$  according to  $\omega_i$ 's and (ii) draw according to  $F_c$  knowing that  $C = c$  is the mixture distribution among  $(F_1, \dots, F_p)$  according to  $\omega_1, \dots, \omega_p$ . This is characterized by the following distribution function

$$F_X(x) = \sum_{i=1}^p \omega_i F_i(x)$$

for all  $x \in \mathbb{R}$ .  $F_i$  are differentiable, then the density function of the mixture variable  $X$  is simply  $f_X(x) = \sum_{i=1}^p \omega_i f_i(x)$ .

A first simple example is the mixture of two normal distributions  $\mathcal{N}(m_1, s_1^2)$ ,  $\mathcal{N}(m_2, s_2^2)$  with the following density

$$f_X(x) = p e^{-(x-m_1)^2/(2s_1^2)} / \sqrt{2\pi s_1^2} + (1-p) e^{-(x-m_2)^2/(2s_2^2)} / \sqrt{2\pi s_2^2}$$

with a proportion  $p \in [0, 1]$  and  $x \in \mathbb{R}$ . This distribution is implemented in the package **mixtools** and **normlmix**. A second example of more interest in actuarial science is the mixture of a light-tailed and heavy-tailed claim distribution. Say for example the mixture of a gamma distribution  $\mathcal{G}(\nu, \lambda)$  and a Pareto distribution  $\mathcal{P}(\alpha, \theta)$ . The density is given by

$$f_X(x) = p \frac{\lambda^\nu x^{\nu-1} e^{-\lambda x}}{\Gamma(\nu)} + (1-p) \alpha / \theta \left( \frac{\theta}{\theta + x} \right)^{\alpha+1},$$

with a proportion  $p \in [0, 1]$  and  $x \in \mathbb{R}_+$ . In R, we implement it as

```

> library(actuar)
> dmixgampar <- function(x, prob, nu, lambda, alpha, theta)
+   prob*dgamma(x, nu, lambda) + (1-prob)*dpareto(x, alpha, theta)
> pmixgampar <- function(q, prob, nu, lambda, alpha, theta)
+   prob*pgamma(q, nu, lambda) + (1-prob)*ppareto(q, alpha, theta)

```

where **dpareto** is implemented in the **actuar** package.

## 2 Parametric inference

Parametric inference deals with the estimation of an unknown parameter of a chosen distribution. Experimenter assumes that  $(x_1, \dots, x_n)$  are realizations of a random sample  $(X_1, \dots, X_n)$  such that  $X_i$  are independent and identically distributed random variables according to a generic random variable  $X$  (this is the blanket assumption). The random variable  $X$  has a distribution function  $F(\cdot; \theta)$  for  $\theta \in \Theta \subset \mathbb{R}^d$ . For example,  $F(x; \theta) = (1 - e^{-\theta x}) \mathbb{1}_{\mathbb{R}_+}(x)$  when considering an exponential distribution  $\theta \in \mathbb{R}_+$ . In the following subsequent sections, classical estimation methods are presented and provide criterions to establish an estimator (i.e. a random variable)  $\hat{\theta}$  of  $\theta$ . Once a model is fitted, the experimenter can derive its quantities of interest (mean, variance, quantiles, survival probabilities, ...) from the fitted distribution  $F(x; \hat{\theta})$ . In most applications,  $X$  has either a continuous or a discrete distribution. Therefore, we work with either the density function  $f_X(\cdot; \theta)$  or the mass probability function  $p_X(\cdot; \theta)$ . For a general introduction to statistical inference, we refer to Casella & Berger (2002) and the references therein.

## 2.1 Maximum likelihood estimation

As its name suggests, maximum likelihood estimation consists in maximizing the likelihood with respect to  $\theta$ , which is defined as

$$\mathcal{L}(\theta, x_1, \dots, x_n) = \prod_{i=1}^n f_X(x_i; \theta) \text{ or } \prod_{i=1}^n p_X(x_i; \theta),$$

depending on the type of the random variable  $X$ . When the support of the distribution is independent of the parameter, it is more convenient to maximize the log-likelihood  $\log \mathcal{L}$  with respect to  $\theta$ . A school example is to consider the exponential distribution  $\mathcal{E}(\theta)$ , i.e.  $\log \mathcal{L}(\theta) = n \log(\theta) - \sum_{i=1}^n \theta x_i$ . The maximizer is  $1/\bar{x}_n$  leading to the estimator  $\hat{\theta} = 1/\bar{X}_n$ . In practice, closed-form formulas of the maximizers may not exist, thus we use a numeric optimization. The **fitdistrplus** package provides routines to compute the maximum likelihood estimator.

We consider a claim dataset `itamtplcost`, which contains large losses (in excess of 500 Keuro) of an Italian Motor-TPL company since 1997. For pedagogical purposes (despite the distribution is not really appropriate), we choose to fit a gamma distribution  $\mathcal{G}(\alpha, \lambda)$  defined as  $f_X(x) = \lambda^\alpha x^{\alpha-1} e^{-\lambda x} / \Gamma(\alpha)$ , i.e.  $\theta = (\alpha, \lambda) \in \mathbb{R}_+^2$ .

```
> library(CASdatasets)
> data(itamtplcost)
> library(fitdistrplus)
> x <- itamtplcost$UltimateCost / 10^6
> summary(x)
```

```
      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.002161 0.627719 0.844011 1.015352 1.224316 6.639500
```

```
> fgamMLE <- fitdist(x, "gamma", method="mle")
> fgamMLE
```

Fitting of the distribution ' gamma ' by maximum likelihood

Parameters:

```
      estimate Std. Error
shape 2.398306  0.1489467
rate  2.362096  0.1631285
```

```
> summary(fgamMLE)
```

Fitting of the distribution ' gamma ' by maximum likelihood

Parameters :

```
      estimate Std. Error
shape 2.398306  0.1489467
rate  2.362096  0.1631285
Loglikelihood: -385.1474  AIC:  774.2947  BIC:  782.5441
Correlation matrix:
      shape    rate
shape 1.000000 0.8992776
rate  0.8992776 1.0000000
```

Without a scaling of cost from euros to million of euros, the call to `fitdist` raises an error, thus we divide the ultimate cost by  $10^6$ . The previous optimization can be checked manually.

```

> loglik <- function(theta) -sum(log(dgamma(x, shape=theta[1], rate=theta[2])))
> op <- optim(par=c(2,2), fn=loglik, hessian=TRUE)
> op$par

```

```
[1] 2.398417 2.362019
```

```

> sqrt(diag(solve(op$hessian)))

```

```
[1] 0.1489539 0.1631228
```

In this example,  $\hat{\theta}$  is estimated as (2.398655, 2.362486). Note that the `fitdist` function returns an S3-object of class "fitdist", for which a `print`, `summary` and `plot` methods have been defined. In addition to the estimation of standard errors of  $\hat{\theta}$ , the `summary` method gives an estimation of the asymptotic correlation matrix (i.e. the inverse of the Hessian matrix  $\nabla_{\theta}^2 \log \mathcal{L}(x, \hat{\theta})$ ) as well as the (optimal) log-likelihood. This is based on the asymptotic normality of the maximum likelihood estimators (under the hypotheses of the Cramer-Rao model, see Casella & Berger (2002)). For the `method`, see the next section.

## 2.2 Moment matching estimation

The moment matching estimation is also commonly used to fit parametric distributions. This consists in finding the value of the parameter  $\theta$  that match the first theoretical raw moments of the parametric distribution to the corresponding empirical raw moments as

$$E(X^k|\theta) = \frac{1}{n} \sum_{i=1}^n x_i^k,$$

for  $k = 1, \dots, d$ , with  $d$  the number of parameters to estimate and  $x_i$  the  $n$  observations of variable  $X$ . For moments of order greater than or equal to 2, it may be relevant to match centered moments defined as

$$E(X|\theta) = \bar{x}_n, \quad E((X - E(X))^k|\theta) = m_k, \quad \text{for } k = 2, \dots, d,$$

where  $m_k = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}_n)^k$  denotes the empirical centered moments. For instance, consider the gamma distribution  $\mathcal{G}(\alpha, \lambda)$ . The moment matching estimation solves

$$\begin{cases} \alpha/\lambda = \bar{x}_n \\ \alpha/\lambda^2 = m_2 \end{cases} \Leftrightarrow \begin{cases} \alpha = (\bar{x}_n)^2/m_2 \\ \lambda = \bar{x}_n/m_2 \end{cases}$$

In general, there is no closed-form formulas for this estimator and use a numerical method. Still considering the gamma distribution fit on the MTPL dataset, we use the `fitdistrplus` package.

```

> fgamMME <- fitdist(x, "gamma", method="mme")
> cbind(MLE=fgamMLE$estimate, MME=fgamMME$estimate)

```

```

           MLE      MME
shape 2.398306 2.229563
rate  2.362096 2.195851

```

## 2.3 Quantile matching estimation

Fitting of a parametric distribution may also be done by matching theoretical quantiles of the parametric distribution (for some specified probabilities) against the empirical quantiles (Tse (2009)). The equation below is very similar to the previous equations for matching moments

$$F^{-1}(p_k; \theta) = Q_{n,p_k},$$

for  $k = 1, \dots, d$  and  $Q_{n,p_k}$  the empirical quantiles for specified probabilities  $p_k$ . Empirical quantiles  $Q_{n,p_k}$  are computed on observations  $x_1, \dots, x_n$  using the `quantile` function of the `stats` package. When  $p_k \times n$  is an integer, the empirical quantile is uniquely defined as  $Q_{n,p_k} = x_{p_k n}^*$  where  $(x_1^*, \dots, x_n^*)$  is the sorted sample. Otherwise, the empirical quantile is the convex combination of  $x_{\lfloor p_k n \rfloor}^*$  and  $x_{\lceil p_k n \rceil}^*$ , see `?quantile` and Hyndman & Fan (1996). The theoretical quantile  $F^{-1}(\cdot; \theta)$  generally has a closed-form formula. For example, when considering the exponential distribution  $\mathcal{E}(\lambda)$ , the quantile function is  $F^{-1}(p; \theta) = -\frac{\log(1-p)}{\lambda}$ . Solving the  $d$  equations  $F^{-1}(p_k; \theta) = Q_{n,p_k}$  is achieved by a numeric optimization in the `fitdist` function.

Continuing the MTPL example, we fit a gamma distribution against the probabilities  $p_1 = 1/3$  and  $p_2 = 2/3$ .

```
> fgamQME <- fitdist(x, "gamma", method="qme", probs=c(1/3, 2/3))
> cbind(MLE=fgamMLE$estimate, MME=fgamMME$estimate,
+       QME=fgamQME$estimate)
```

	MLE	MME	QME
shape	2.398306	2.229563	4.642362
rate	2.362096	2.195851	4.951094

Note that compared to the method of moments and the maximum likelihood estimation, the estimate parameter  $\hat{\theta}$  differs significantly when using the quantile matching estimation, despite considering probabilities in the heart of the distribution.

## 2.4 Maximum goodness-of-fit estimation

A last method of estimation called maximum goodness-of-fit estimation or (minimum distance estimation) is presented, see e.g. D'Agostino & Stephens (1986), Dutang et al. (2008). In this section, we focus on the Cramer - von Mises distance and refer to Delignette-Muller & Dutang (2013) for other distances (i.e. Kolmogorov-Smirnov and Anderson-Darling, the latter being of particular interest for skewed distributions). The Cramer - von Mises looks at the squared difference between the candidate distribution  $F(x; \theta)$  and the empirical distribution function  $F_n$ , the latter being defined as the percentage of observations below  $x$ :  $F_n(x) = \sum_{i=1}^n \mathbb{1}_{x_i \leq x}$ . The Cramer - von Mises distance is defined as

$$D(\theta) = \int_{-\infty}^{\infty} (F_n(x) - F(x; \theta))^2 dx,$$

and is estimated by in practice

$$\hat{D}(\theta) = \frac{1}{12n} + \sum_{i=1}^n \left( F(x_i; \theta) - \frac{2i-1}{2n} \right)^2.$$

The maximum goodness-of-fit estimation consists in finding the value of  $\theta$  minimizing  $\hat{D}(\theta)$ . The names comes from the fact that the Cramer - von Mises distance measures the goodness-of-fit of  $F(\cdot; \theta)$  against  $F_n$ . There is no closed-form formula for  $\arg \min_{\theta} \hat{D}(\theta)$  and a numerical optimization is used in the `fitdist` function.

Finally, we fit a gamma distribution by maximum goodness-of-fit estimation

```
> fgamMGE <- fitdist(x, "gamma", method="mge", gof="CvM")
> cbind(MLE=fgamMLE$estimate, MME=fgamMME$estimate,
+       QME=fgamQME$estimate, MGE=fgamMGE$estimate)
```

	MLE	MME	QME	MGE
shape	2.398306	2.229563	4.642362	3.720486
rate	2.362096	2.195851	4.951094	3.875886

As for quantile matching estimation, the value of  $\hat{\theta}$  differs widely. A practitioner approach could be take the average by component irrespectively of the methods tested. This leads to the question how to choose between fitted parameters and between fitted distribution.

### 3 Measures of adequacy

This section focuses on measures of adequacy either graphical methods or numerical methods.

#### 3.1 Histogram and empirical densities

A typical plot to assess the adequacy of a distribution is the histogram. We recall that for plotting an histogram, observed data are divided into  $k$  classes  $]a_{j-1}, a_j]$  for  $j = 1, \dots, k$  (generally  $k$  is proportional to  $\log(n)$ ); the number of observation in each class is computed, i.e. frequencies  $f_j$ ; finally rectangles are drawn such that the basis is a class  $]a_{j-1}, a_j]$  and the height is the absolute  $f_j$  or the relative  $f_j/n$  frequencies. Thus, the histogram is an estimator of the empirical density, as the area of a rectangle is proportional to  $\mathbb{P}(X \in ]a_{j-1}, a_j])$ . This graph is generically provided in the `plot` function of a "fitdist" object, but does not allow multiple fitted distributions. So in the example of a gamma fit to the MTPL dataset, we use the `denscomp` function.

```
> txt <- c("MLE", "MME", "QME(1/3, 2/3)", "MGE-CvM")
> denscomp(list(fgamMLE, fgamMME, fgamQME, fgamMGE), legendtext=txt,
+           fitcol="black", main="Histogram and fitted gamma densities")
```

Alternatively, we can estimate directly the density function by the popular kernel density estimation. This is implemented in the `density` function as shown below

```
> hist(x, prob=TRUE, ylim=c(0, 1))
> lines(density(x), lty=5)
```

In order to better assess the fitted gamma densities, the two above graphs are plotted on separate graphics. We observe that the MLE and the MME fits best approximates the density between  $x \in [0.5, 1.5]$  while the QME and the MGE fit best assess the density between  $x \in [1.5, 4]$ . However, it is clear that the gamma distribution cannot appropriately fit the whole distribution, mainly due to its light-tailedness.

#### 3.2 Distribution function plot

Another typical graph is to plot the fitted distribution  $F(\cdot; \hat{\theta})$  and the empirical cumulative distribution function  $F_n$ . As already given, the computation of  $F_n$  is simpler than for the empirical density  $F_n(x) = \sum_{i=1}^n \mathbb{1}_{x_i \leq x}$ . A new claim dataset is considered to illustrate this type of plot: we use the celebrated "Danish" dataset published in McNeil (1997). The dataset is stored in `danishuni` for the univariate version and contains fire loss amounts collected at Copenhagen Reinsurance between 1980 and 1990. We consider three distributions: a gamma distribution  $\mathcal{G}(\alpha, \lambda)$ , a Pareto distribution  $\mathcal{P}(\alpha, \theta)$ , a Pareto-gamma mixture  $\mathcal{PG}(p, \alpha_1, \theta_1, \alpha_2, \lambda_2)$  defined in Section 1.4 and a Burr distribution  $\mathcal{Bu}(\gamma, \tau, \theta)$ .

```
> data(danishuni)
> x <- danishuni$Loss
> fgam <- fitdist(x, "gamma", lower=0)
> fpar <- fitdist(x, "pareto", start=list(shape=2, scale=2), lower=0)
> fmixgampar <- fitdist(x, "mixgampar", start=
+                       list(prob=1/2, nu=1, lambda=1, alpha=2, theta=2), lower=0)
```

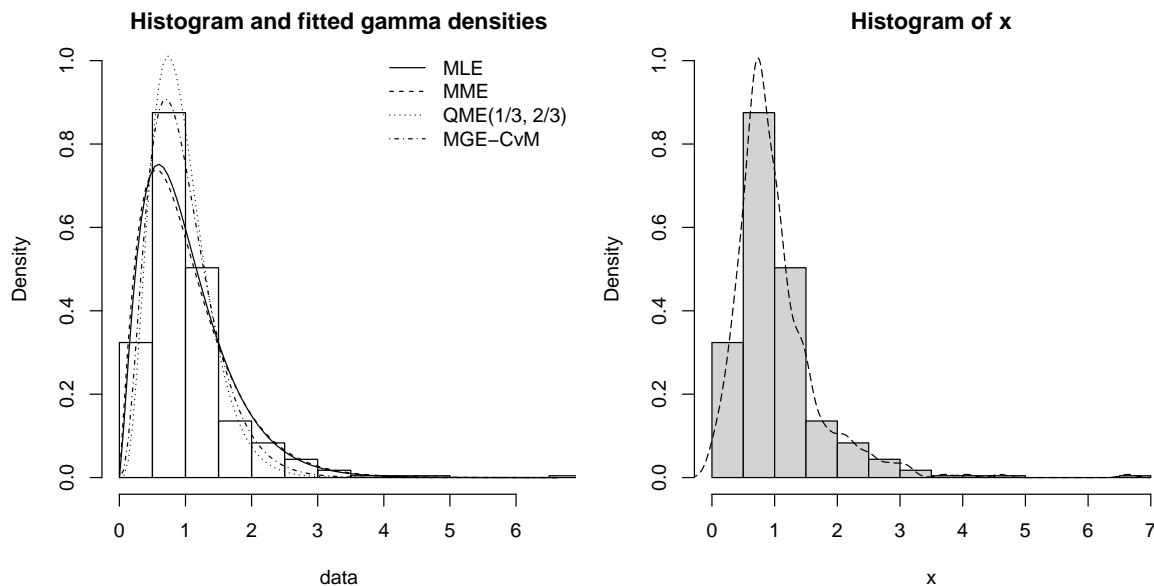


Figure 6: Comparison of fits on a MTPL dataset

```
> cbind(SINGLE= c(NA, fgam$estimate, fpar$estimate),
+        MIXTURE=fmixgampar$estimate)
```

	SINGLE	MIXTURE
	NA	0.6849901
shape	1.2975373	10.8671174
rate	0.3832863	6.5413112
shape	5.3689492	5.4070157
scale	13.8424418	29.9966023

When fitted alone, the parameters of the gamma distribution are estimated as  $(\hat{\alpha}, \hat{\lambda}) = (1.2976, 0.3833)$  and the parameters of the Pareto distribution are estimated as  $(\hat{\alpha}, \hat{\theta}) = (5.3689, 13.8413)$ . When used in the mixture, we get  $(\hat{p}, \hat{\alpha}_1, \hat{\theta}_1, \hat{\alpha}_2, \hat{\lambda}_2) = (0.6849, 10.8706, 6.5436, 5.4182, 30.0700)$ . As only the shape parameter  $\hat{\alpha}_2$  is of similar amplitude, only heavy-tailed distributions (like the Pareto) are appropriate for this dataset. Finally, we fit a Burr distribution.

```
> fburr <- fitdist(x, "burr", start=list(shape1=2, shape2=2,
+                                       scale=2), lower=c(0.1, 1/2, 0))
> fburr$estimate
```

shape1	shape2	scale
0.100000	14.441676	1.085243

Comparing the fitted densities is then carried out using the `cdfcomp` function.

```
> cdfcomp(list(fgam, fpar, fmixgampar, fburr), xlogscale=TRUE,
+          datapch=".", datacol="grey", fitcol="black", fitlty=2:5,
+          legendtext=c("gamma", "Pareto", "Par-gam", "Burr"),
+          main="Fitted CDFs on danish")
```

We also plot the tail of the distribution function on Figure 7. When using the maximum likelihood estimation, the best fit is provided by the Burr distribution, yet the first shape parameter hits the lower bound of 0.1.

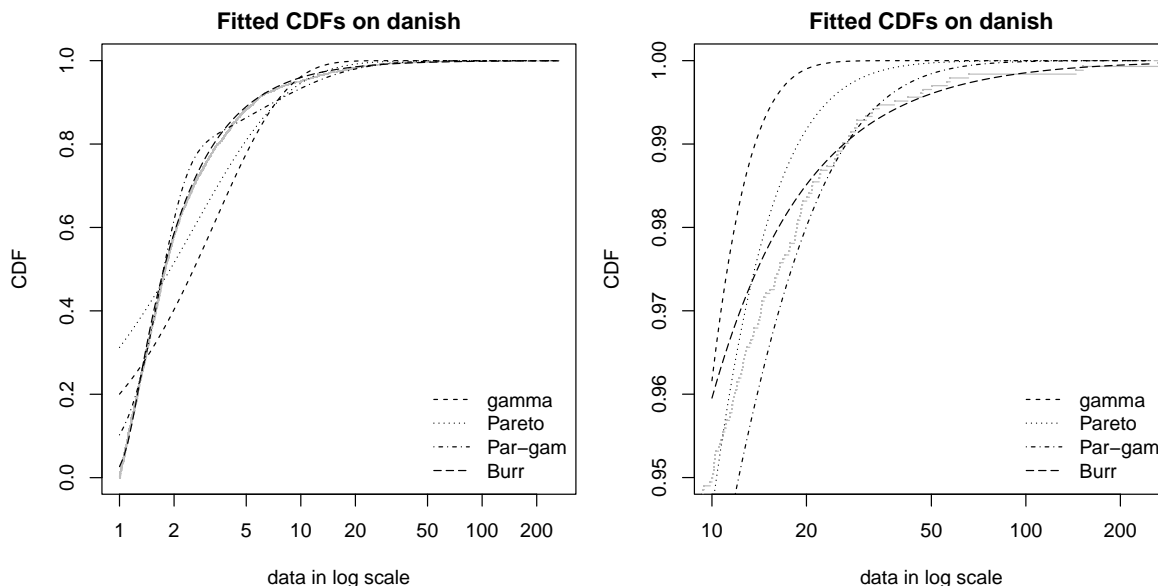


Figure 7: Comparison of fits on the Danish fire dataset

### 3.3 Qqplot, ppplot

On the two previous graphs, we consider the plot of the empirical density (resp. the empirical distribution function) and the fitted density (resp. the fitted distribution function). The pp-plot (resp. the qq-plot) consists in plotting (directly) the empirical distribution function  $F_n$  against the fitted distribution function  $F(\cdot; \hat{\theta})$  (resp. the empirical quantile function  $Q_n$ , against the fitted quantile function  $F^{-1}(\cdot; \hat{\theta})$ ). Those quantities are computed at the observations, which leads to further simplifications  $F_n(x_i) = \text{rank}(x_i)/n$  and  $Q_{n,i/n} = x_i^*$ . This is illustrated on the Danish fire dataset `danishuni` and the four distributions considered by using the `ppcomp` and `qqcomp` functions.

```
> qmixgampar <- function(p, prob, nu, lambda, alpha, theta)
+ {
+   L2 <- function(q, p)
+     (p - pmixgampar(q, prob, nu, lambda, alpha, theta))^2
+   sapply(p, function(p) optimize(L2, c(0, 10^3), p=p)$minimum)
+ }
> ppcomp(list(fgam, fpar, fmixgampar, fburr), xlogscale=TRUE,
+         ylogscale=TRUE, fitcol="black", main="PP-plot on danish",
+         legendtext=c("gamma", "Pareto", "Par-gam", "Burr"), fitpch=1:4)
> qqcomp(list(fgam, fpar, fmixgampar, fburr), xlogscale=TRUE,
+         ylogscale=TRUE, fitcol="black", main="QQ-plot on danish",
+         legendtext=c("gamma", "Pareto", "Par-gam", "Burr"), fitpch=1:4)
```

As there is no closed-form formula for the quantile function of the mixture distribution (i.e. inverse of  $x \mapsto p\gamma(\alpha, \nu x)/\Gamma(\nu) + (1-p)(1 - \theta^\alpha/(\theta + x)^\alpha)$ ), a numerical optimization is carried out using the Golden line-search (implemented in `optimize`). On Figure 8, quantiles and probabilities are plotted as a point, while the straight line corresponds to the identity function. The more points that are close to the line, the better fit the distribution. The pp-plot reveals that only the Burr distribution sufficiently fit the data, whereas the qq-plot shows that both the Pareto-gamma mixture and the Burr distributions best approximates the data.

The `plot` method of a `"fitdist"` object provides the four above graphs (for the fitted distribution) in the following order: an histogram with the fitted density, an emp. cumulative distribution function (ecdf-plot) with the fitted distribution function, a (theo.) quantile - (emp.) quantile plot and a (theo.) probability - (emp.) probability plot.



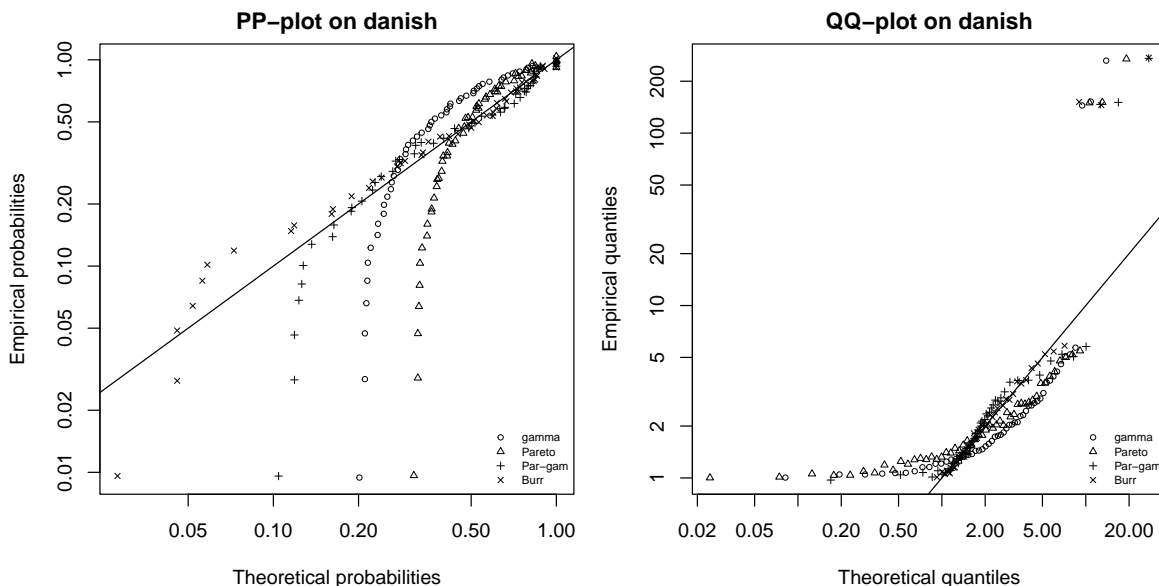


Figure 8: Comparison of fits on the Danish fire dataset

### 3.4 Goodness-of-fit statistics and tests

We turn our attention to goodness-of-fit statistics to complement the four previous goodness-of-fit graphs. For continuous distributions, the three statistics presented in Section 2.4 can be computed, that is Cramer - von Mises, Kolmogorov-Smirnov and Anderson-Darling statistics. For discrete distribution, the more common statistic is the chi-square statistic defined as

$$\Delta^2 = \sum_{i=0}^m \frac{(n_i - np_i)^2}{np_i}$$

where  $n_i$  the empirical frequency count for the level  $i$ ,  $n$  the total number of observations, the theoretical probability  $p_i = \mathbb{P}(X = i; \theta)$  (i.e.  $np_i$  the theoretical frequency count),  $m$  the number of cells. In practice, the number of cells is either fixed by the experimenter or chosen so that empirical frequencies are greater than five and  $p_i$  is replaced by  $\hat{p}_i = \mathbb{P}(X = i; \hat{\theta})$ . The chi-square statistic is linked to the Pearson's hypothesis test of goodness-of-fit, for which under the null hypothesis  $\Delta^2$  converge in law to a chi-square distribution  $\chi^2(m - d - 1)$  (where  $d$  is the number of parameters). For all distributions, we consider also the information criterions (AIC and BIC) proportional to the opposite of the log-likelihood. All of this is provided in the `gofstat` function. A numerical illustration is proposed on a TPL claim number dataset, for which a Poisson, a negative binomial and a zero-modified Poisson distributions are fitted by the maximum likelihood estimation.

```
> library(CASdatasets)
> data("freMTPLfreq")
> x <- freMTPLfreq$ClaimNb
> fpois <- fitdist(x, "pois")
> fnbinom <- fitdist(x, "nbinom")
> fpoisZM <- fitdist(x, "poisZM", start=list(
+   prob=sum(x == 0)/length(x), lambda=mean(x)),
+   lower=c(0,0), upper=c(1, Inf))
> gofstat(list(fpois, fnbinom, fpoisZM), chisqbreaks=c(0:4, 9),
+   discrete=TRUE, fitnames=c("Poisson", "NegBinomial", "ZM-Poisson"))
```

```
Chi-squared statistic: 1008.58 3.245916 26.39691
Degree of freedom of the Chi-squared distribution: 4 4 3
Chi-squared p-value: 4.934275e-217 0.5175469 7.875999e-06
```

the p-value may be wrong with some theoretical counts  $< 5$   
Chi-squared table:

	obscounts	theo Poisson	theo NegBinomial	theo ZM-Poisson
$\leq 0$	397779	3.973006e+05	3.977803e+05	3.977781e+05
$\leq 1$	14633	1.555969e+04	1.464111e+04	1.462057e+04
$\leq 2$	726	3.046861e+02	7.084345e+02	7.506765e+02
$\leq 3$	28	3.977527e+00	3.701326e+01	1.927131e+01
$\leq 4$	3	3.894349e-02	2.005249e+00	3.298210e-01
$> 4$	0	3.070350e-04	1.175589e-01	4.277418e-03

Goodness-of-fit criteria

	Poisson	NegBinomial	ZM-Poisson
Akaike's Information Criterion	138343.4	137792.5	137801.2
Bayesian Information Criterion	138354.3	137814.3	137823.0

From the chi-square statistic and the chi-square table  $(n_i, n\hat{p}_i)_i$ , the negative binomial distribution is clearly the best distribution. This is also confirmed by the AIC and the BIC criterions.

### 3.5 Skewness-kurtosis graph

When selecting a distribution, depending on the type of applications, the experimenter may give a particular attention to the tail, some quantiles or the body of the distribution for which a natural way of choosing the “best” distribution emerges. In actuarial science, a great care is put on the tail of distribution, and also on first moments. The code below provide values of quantiles (plotted before) as well as the first two raw moments.

```
> p <- c(.9, .95, .975, .99)
> rbind(
+   empirical= quantile(danishuni$Loss, prob=p),
+   gamma= quantile(fgam, prob=p)$quantiles,
+   Pareto= quantile(fpar, prob=p)$quantiles,
+   Pareto_gamma= quantile(fmixgampar, prob=p)$quantiles,
+   Burr= quantile(fburr, prob=p)$quantiles)

      p=0.9   p=0.95  p=0.975  p=0.99
empirical  5.541526  9.972647 16.26821 26.04253
gamma      7.309525  9.262005 11.19005 13.71334
Pareto     7.412939 10.342085 13.67489 18.79567
Pareto_gamma 7.091501 12.164170 17.93063 26.78132
Burr       5.345242  8.637989 13.95912 26.32738

> compmom <- function(order)
+   c(empirical= sum(danishuni$Loss^order)/length(x),
+     gamma=mgamma(order, fgam[[1]][1], fgam[[1]][2]),
+     Pareto=mpareto(order, fpar[[1]][1], fpar[[1]][2]),
+     Pareto_gamma= as.numeric(fmixgampar[[1]][1]*
+                               mgamma(order, fmixgampar[[1]][2], fmixgampar[[1]][3])+
+                               (1-fmixgampar[[1]][1])*
+                               mpareto(order, fmixgampar[[1]][4], fmixgampar$estimate[5])),
+     Burr=mburr(order, fburr[[1]][1], fburr[[1]][2], fburr[[1]][3]))
> rbind(Mean=compmom(1), Mom2nd= compmom(2))

      empirical   gamma   Pareto Pareto_gamma   Burr
Mean  0.0177542  3.385296  3.168369   3.282111  2.986242
Mom2nd 0.4395279 20.292516 26.036581  39.819947   Inf
```

For higher moments, it is typical to look at the skewness and the kurtosis coefficients defined as

$$sk(X) = \frac{E[(X - E(X))^3]}{Var(X)^{\frac{3}{2}}}, \quad kr(X) = \frac{E[(X - E(X))^4]}{Var(X)^2},$$

for a random variable  $X$ . For heavy-tailed distributions, such coefficients may not exist, yet empirically they always exist. The `descdist` function provides the so-called Cullen and Frey graph, which plots the empirical estimates of  $sk(X)$  and  $kr(X)$  as well as the possible values for some classic distributions (including the gamma family for continuous distributions and the Poisson distribution for discrete distributions) This is illustrated on the `danishuni` and the `tplclaimnumber` datasets on Figure 9. The fit analysis can also be completed by looking at the uncertainty of parameter estimate with a bootstrap analysis. This is possible with the `bootdist` function of `fitdistrplus`.

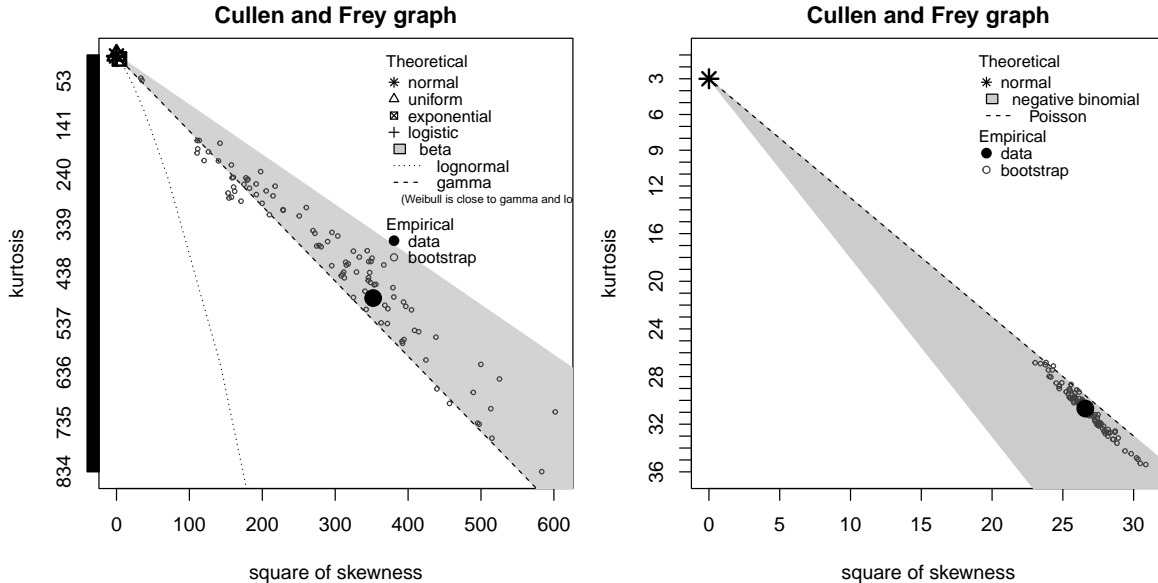


Figure 9: Cullen and Frey graph for danish and tplclaimnumber

## 4 Aggregate loss distribution

This section deals with the aggregate loss amount distribution, i.e. the distribution of the compound sum

$$S = \sum_{i=1}^N X_i,$$

where  $N$  is the claim number and  $(X_i)$ 's are the claim severities. Firstly, the computation of the distribution function  $F_S$  of  $S$  is studied. Then, an application to a TPL motor dataset is carried out. Finally, a continuous-time version of this problem is analyzed via the ruin theory framework.

### 4.1 Computation of the aggregate loss distribution

A classical assumption on the aggregate amount  $S$  is to require that  $N$  is independent of claim amounts  $(X_i)_i$ . Another common assumption is that  $(X_i)_i \stackrel{i.i.d.}{\sim} X$ . Therefore, the distribution function simplifies to

$$F_S(s) = \sum_{n=0}^{+\infty} \mathbb{P}(N = n) \mathbb{P}(X_1 + \dots + X_n \leq s) = \sum_{n=0}^{+\infty} \mathbb{P}(N = n) F_X^{*n}(s),$$

where  $F_X^{*n}$  the  $n$ -order convolution product of  $F_X$ . In a small number of distribution of  $X$ , the distribution of the sum  $X_1 + \dots + X_n$  is easy. For instance, when  $X$  follows a gamma distribution  $\mathcal{G}(\alpha, \lambda)$ , then the sum  $X_1 + \dots + X_n$  follows a gamma distribution  $\mathcal{G}(n\alpha, \lambda)$ . This can be implemented by the following function.

```

> pgamsum <- function(x, dfreq, argfreq, shape, rate, Nmax=10)
+ {
+   tol <- 1e-10; maxit <- 10
+   nbclaim <- 0:Nmax
+   dnbclaim <- do.call(dfreq, c(list(x=nbclaim), argfreq))
+   psumfornbclaim <- sapply(nbclaim, function(n)
+     pgamma(x, shape=shape*n, rate=rate))
+   psumtot <- psumfornbclaim %>% dnbclaim
+   dnbclaimtot <- dnbclaim
+   iter <- 0
+   while( abs(sum(dnbclaimtot)-1) > tol && iter < maxit)
+   {
+     nbclaim <- nbclaim+Nmax
+     dnbclaim <- do.call(dfreq, c(list(x=nbclaim), argfreq))
+     psumfornbclaim <- sapply(nbclaim, function(n)
+       pgamma(x, shape=shape*n, rate=rate))
+     psumtot <- psumtot + psumfornbclaim %>% dnbclaim
+     dnbclaimtot <- c(dnbclaimtot, dnbclaim)
+     iter <- iter+1
+   }
+   as.numeric(psumtot)
+ }

```

In general, the distribution of the sum  $X_1 + \dots + X_n$  does not necessarily have the same distribution as  $X$ . So alternative computations are necessary. The Panjer recursion provides a recursive method to compute the mass probability function of  $S$  in the case  $X$  has a discrete distribution and  $N$  belongs to the  $(a, b, n)$  family, see Panjer (1981). The recursion formula for the mass probability function  $p_S$  is

$$p_S(s) = \frac{[p_X(1) - (a + b)p_X(0)]p_X(s) + \sum_{y=1}^{s \wedge m} (a + by/x)p_X(y)p_S(s - y)}{1 - ap_X(0)},$$

where  $s \in \mathbb{N}$ ,  $X$  has a discrete distribution on  $\{0, 1, \dots, m\}$  with a mass probability function  $p_X$ ,  $N$  belongs to  $(a, b, 0)$  family and starting at  $p_S(0) = G_N(p_X(0))$  with  $G_N$  the probability generating function. The recursion is stopped when the sum of elementary probabilities  $\mathbb{P}(S = 0, 1, \dots)$  is arbitrary closed to 1. In practice, the distribution of the claim amount is not discrete but can be discretized. The upper discretization is the forward difference  $\tilde{f}(x) = F_X(x + h) - F_X(x)$ , the lower discretization is the backward difference  $\tilde{f}(x) = F_X(x) - F_X(x - h)$ , the unbiased discretization is  $\tilde{f}(x) = (2E(X \wedge x) - E(X \wedge x - h) - E(X \wedge x + h))/h$  where  $h$  is the step of discretization, see Figure 10 and Dutang et al. (2008) for further details.

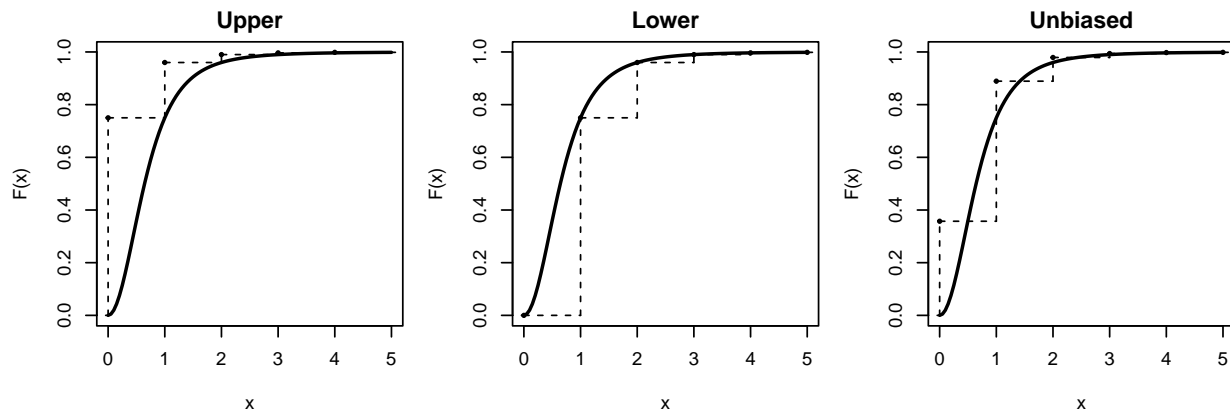


Figure 10: Comparison of three discretization methods (a paralogistic distribution)

Approximation based on the normal distribution are also available: (i) the normal approximation is given by

$$F_S(x) \approx \Phi\left(\frac{x - E(S)}{\sigma(S)}\right),$$

(ii) the normal-power approximation is given by

$$F_S(x) \approx \Phi \left( -\frac{3}{sk(S)} + \sqrt{\frac{9}{sk(S)^2} + 1 + \frac{6}{sk(S)} \frac{x - E(S)}{\sigma(S)}} \right),$$

where  $\sigma(S)$  is the standard deviation of  $S$  and  $sk(S)$  is the skewness coefficient of  $S$ . The skewness coefficient can be written as

$$sk(S) = \frac{sk(N)Var(N)^{3/2}E(X)^3 + 3Var(N)E(X)Var(X) + E(N)sk(X)Var(X)^{3/2}}{Var(S)^{3/2}}.$$

An approximation based on the gamma distribution is also possible, see Gendron & Crepeau (1989). These approximations are reasonably correct at the heart of the distribution but not at the tails of the distribution. The most common alternative to exact computation is the simulation procedure. It consists in simulating  $n$  claim numbers  $N_1, \dots, N_n$ ,  $N_i$ 's claim severities  $(X_{i,j})_j$  in order to get  $n$  realizations  $S_1, \dots, S_n$ . All these alternative methods are available in the `aggregateDist` function of the `actuar` package. We consider two examples: the gamma case ( $N$  follows a Poisson distribution  $\mathcal{P}(10)$  and  $X$  follows a gamma distribution  $\mathcal{G}(3, 2)$ ) and the Pareto case ( $N$  follows a Poisson distribution  $\mathcal{P}(10)$  and  $X$  follows a Pareto distribution  $\mathcal{P}(3.1, 4.2)$ ). The following code computes the gamma case.

```
> parsev <- c(3, 2); parfreq <- 10
> meansev <- mgamma(1, parsev[1], parsev[2])
> varsev <- mgamma(2, parsev[1], parsev[2]) - meansev^2
> skewsev <- (mgamma(3, parsev[1], parsev[2]) -
+           3*meansev*varsev - meansev^3)/varsev^(3/2)
> meanfreq <- varfreq <- parfreq[1]; skewfreq <- 1/sqrt(parfreq[1])
> meanagg <- meanfreq * meansev
> varagg <- varfreq * (varsev + meansev^2)
> skewagg <- (skewfreq*varfreq^(3/2)*meansev^3 + 3*varfreq*meansev*
+           varsev + meanfreq*skewsev*varsev^(3/2))/varagg^(3/2)
> Fs.s <- aggregateDist("simulation", model.freq = expression(y =
+           rpois(parfreq)), model.sev = expression(
+           rgamma(2, parsev[1], parsev[2])))
> Fs.n <- aggregateDist("normal", moments = c(meanagg, varagg))
> Fs.np <- aggregateDist("npower", moments = c(meanagg, varagg, skewagg))
> Fs.exact <- function(x) pgamsum(x, dpois, list(lambda=parfreq),
+           parsev[1], parsev[2], Nmax=100)
> x <- seq(25, 40, length=101)
> plot(x, Fs.exact(x), type="l",
+       main="Agg. Claim Amount Distribution", ylab="F_S(x)")
> lines(x, Fs.s(x), lty=2)
> lines(x, Fs.n(x), lty=3)
> lines(x, Fs.np(x), lty=4)
> legend("bottomright", leg=c("exact", "simulation",
+           "normal approx.", "NP approx."), col = "black",
+       lty = 1:4, text.col = "black")
```

Similarly, we have the Pareto case. We show here only the recursive computation calls.

```
> parsev <- c(3.1, 2*2.1); parfreq <- 10
> xmax <- qpareto(1-1e-9, parsev[1], parsev[2])
> fx2 <- discretize(ppareto(x, parsev[1], parsev[2]), from = 0,
+           to = xmax, step = 0.5, method = "unbiased",
+           lev = levpareto(x, parsev[1], parsev[2]))
> Fs2 <- aggregateDist("recursive", model.freq = "poisson",
+           model.sev = fx2, lambda = parfreq, x.scale = 0.5, maxit=2000)
> fx.u2 <- discretize(ppareto(x, parsev[1], parsev[2]), from = 0,
+           to = xmax, step = 0.5, method = "upper")
```

```

> Fs.u2 <- aggregateDist("recursive", model.freq = "poisson",
+                       model.sev = fx.u2, lambda = parfreq, x.scale = 0.5, maxit=2000)
> fx.l2 <- discretize(ppareto(x, parsev[1], parsev[2]), from = 0,
+                   to = xmax, step = 0.5, method = "lower")
> Fs.l2 <- aggregateDist("recursive", model.freq = "poisson",
+                       model.sev = fx.l2, lambda = parfreq, x.scale = 0.5, maxit=2000)

```

The two graphs are displayed on Figure 11. Despite that the expectation  $E(X)$  is identical in both cases, high-level quantiles of the aggregate claim distribution are significantly different. For the gamma case, the normal-power approximation suitably fits the exact distribution function, while for the gamma case, the normal-power approximation overestimates as the skewness  $sk(S)$  is very high and not representative of the shape of the distribution. As their name suggests, the upper and the lower recursive computations surrounds the true distribution function. The simulation number is voluntary chosen low (1000), but can be set to a much larger number. If convergence is achieved for a high number of simulation, parallelization, GPU computation and quasi - Monte Carlo sampling methods can be used to fasten the process, see <http://cran.r-project.org/web/views/HighPerformanceComputing.html> and <http://cran.r-project.org/web/views/Distributions.html>.

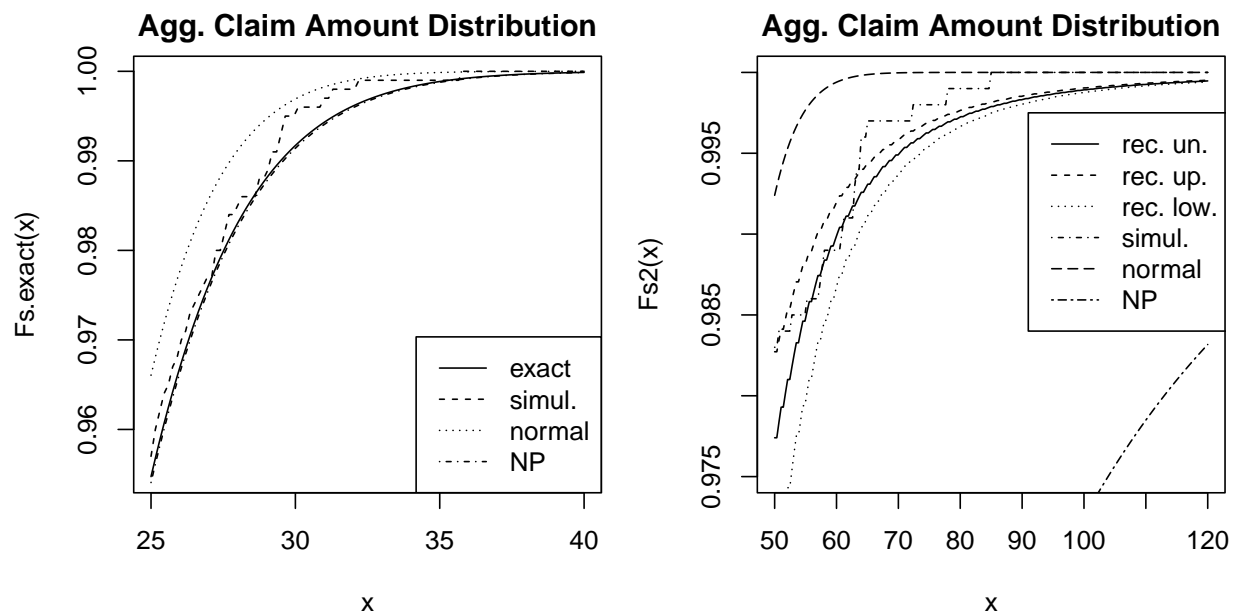


Figure 11: Aggr. claim distribution (gamma case / Pareto case)

## 4.2 Ruin models

Ruin theory deals with the study of stochastic processes linked to the wealth of an insurer, see Asmussen & Albrecher (2010) for a recent survey. A reserve risk process  $(U_t)_{t \geq 0}$  is considered. The initial model of Cramér-Lundberg assumes that the surplus  $(U_t)_{t \geq 0}$  of an insurance company at time  $t$  is represented by

$$U_t = u + ct - \sum_{i=1}^{N_t} X_i,$$

where  $u$  is the initial surplus,  $c$  is the premium rate,  $(X_i)_{i \geq 1}$  are i.i.d. successive claim amounts and  $(N_t)_{t \geq 0}$  is the claim arrival process assumed to be a Poisson process of intensity  $\lambda$ . Andersen (1957) generalized this model by proposing a renewal process for the claim arrival process  $(N_t)_{t \geq 0}$  (the claim waiting times are denoted by  $(T_i)_{i \geq 1}$ ). When claim severities and claim waiting times follow a phase-type distribution, closed-form formulas exist for the ruin probability

$$\psi(u) = \mathbb{P}(\exists t > 0 : U_t < 0 | U_0 = u),$$

see Asmussen & Rolski (1991). We provide below examples of that article.

```

> psi <- ruin(claims = "e", par.claims = list(rate = 1/0.6),
+           wait = "e", par.wait = list(rate = 1/0.6616858))
> ## Phase-type claims,
> ## exponential interarrival times
> p <- c(0.5614, 0.4386)
> r <- matrix(c(-8.64, 0.101, 1.997, -1.095), 2, 2)
> lambda <- 1/(1.1 * mphtype(1, p, r))
> psi2 <- ruin(claims = "p", par.claims = list(prob = p, rates = r),
+           wait = "e", par.wait = list(rate = lambda))
> ## Phase-type claims,
> ## mixture of two exponentials for interarrival times
> a <- (0.4/5 + 0.6) * lambda
> psi3 <- ruin(claims = "p", par.claims = list(prob = p, rates = r),
+           wait = "e", par.wait = list(rate = c(5 * a, a), weights =
+           c(0.4, 0.6)), maxit = 225)
> plot(psi, from = 0, to = 50)
> plot(psi2, add=TRUE, lty=2)
> plot(psi3, add=TRUE, lty=3)
> legend("topright", leg=c("Exp - Exp", "PH - Exp",
+ "PH - MixExp"), lty=1:3, col="black")

```

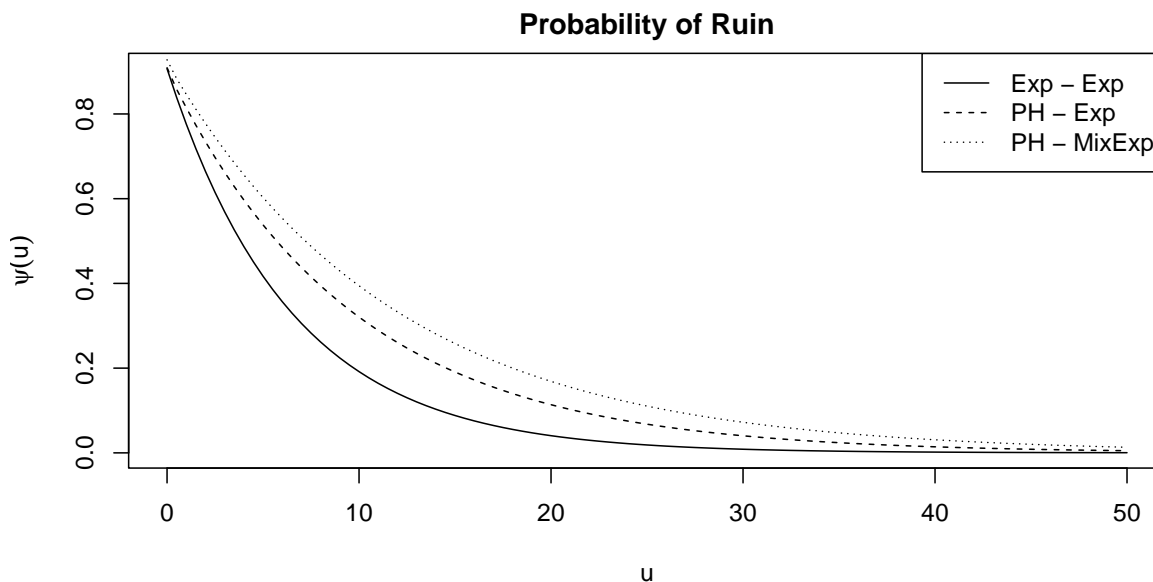


Figure 12: Ruin probability

## 5 Copulas and multivariate distributions

This section deals with distributions of multivariate random vectors  $\mathbf{X} = (X_1, \dots, X_d)$ . Due to the growing literature (e.g. Frees & Valdez (1998), Embrechts et al. (2001), Frees & Wang (2006)) on copulas (defined as multivariate distribution functions of random vector with uniform marginals) since 2000, we focus on copulas in this section.

### 5.1 Definition of copulas

Let  $F_{\mathbf{X}}$  be the distribution function of  $\mathbf{X}$  with marginals  $F_{X_j}$  i.e.  $F_{\mathbf{X}}(\mathbf{x}) = \mathbb{P}(X_1 \leq x_1, \dots, X_d \leq x_d)$ . As  $F_{X_j}^{-1}(U)$  has the same distribution as  $X_j$  for  $U$  a uniform variate, it is easily checked that  $\mathbb{P}(X_1 \leq$

$F_{X_1}^{-1}(u_1), \dots, X_d \leq F_{X_d}^{-1}(u_d) = \mathbb{P}(F_{X_1}(X_1) \leq u_1, \dots, F_{X_d}(X_d) \leq u_d)$ . A copula function  $C$  is a multivariate distribution function such that  $C(u_1, \dots, u_d) = \mathbb{P}(F_{X_1}(X_1) \leq u_1, \dots, F_{X_d}(X_d) \leq u_d)$  for  $\mathbf{u} \in [0, 1]^d$ . The  $C$  function is bounded by the so-called Fréchet bound as  $(\sum_{i=1}^d u_i - (d-1))_+ \leq C(\mathbf{u}) \leq \min(u_1, \dots, u_d)$  generally denoted by  $W(\mathbf{u})$  and  $M(\mathbf{u})$ , see Nelsen (2006) for a recent introduction. By the Sklar theorem (Sklar (1959)) for any random vectors  $\mathbf{X}$  with marginals  $F_{X_j}$ , there exists a copula function  $C$  such that

$$\mathbb{P}(\mathbf{X} \leq \mathbf{x}) = \mathbb{P}(X_1 \leq x_1, \dots, X_d \leq x_d) = C(F_{X_1}(x_1), \dots, F_{X_d}(x_d)),$$

for  $\mathbf{x} \in \mathbb{R}^d$ . The copula  $C$  is unique on the support of  $\mathbf{X}$ , and not otherwise. Let us note that in the independent case where  $X_i \perp X_j$  for  $i \neq j$ , the copula function is simply  $C(\mathbf{u}) = u_1 \times \dots \times u_d$ . As described below, classical multivariate distributions such as the multivariate Gaussian distribution and the multivariate Pareto distribution can be represented using a copula function.

## 5.2 Archimedean copulas

A wide class of copulas is given by the family of Archimedean copulas. An Archimedean copula is characterized by a generator function  $\phi : [0, 1] \mapsto [0, \infty]$  such that

$$C(\mathbf{u}) = \phi^{-1} \left( \sum_{i=1}^d \phi(u_i) \right),$$

where  $\mathbf{u} \in [0, 1]^d$  and  $\phi$  is infinitely differentiable, completely monotone and invertible. We refer to Theorem 2.1 of Marshall & Olkin (1988) for the construction of Archimedean copulas. In this family, the three most classical copulas are the Gumbel copula  $\phi(t) = (-\log(t))^{-\alpha}$ , the Frank copula  $\phi(t) = \log(e^\alpha - 1) - \log(e^{\alpha t} - 1)$  and the Clayton copula  $\phi(t) = t^{-\alpha} - 1$  for a parameter  $\alpha$ . We get the following copula function

- Gumbel:  $C_{G_u}(\mathbf{u}) = \exp(-((-\log(u_1))^\alpha + \dots + (-\log(u_d))^\alpha)^{1/\alpha})$ , for  $\alpha \geq 1$ .
- Frank:  $C_F(\mathbf{u}) = -\log(1 + (e^{-\alpha u_1} - 1) \times (e^{-\alpha u_d} - 1)/(e^\alpha - 1))/\alpha$ , for  $\alpha \neq 0$ .
- Clayton:  $C_C(\mathbf{u}) = (1 - d + u_1^{-\alpha} + \dots + u_d^{-\alpha})^{-1/\alpha}$ , for  $\alpha \in [-1, +\infty[\setminus\{0\}$ .

In fact, the survival Clayton defined as  $\mathbb{P}(\mathbf{U} > \mathbf{u}) = C_C(1 - \mathbf{u})$  is linked to the multivariate Pareto distribution. According to Arnold (1983), the multivariate Pareto distribution is characterized by the following survival function

$$\mathbb{P}(\mathbf{X} > \mathbf{x}) = \left( 1 + \sum_{i=1}^d x_i^{\frac{1}{\gamma_i}} \right)^{-\alpha}.$$

The marginal distribution of  $X_i$  is also Pareto distributed, since  $\mathbb{P}(X_i > x_i) = (1 + x_i^{1/\gamma_i})^{-\alpha} = 1 - F_{X_i}(x_i)$ . It is easy to check that  $\mathbb{P}(X_1 > F_{X_1}^{-1}(u_1), \dots, X_d > F_{X_d}^{-1}(u_d)) = C_C(1 - \mathbf{u})$ . In other words, the multivariate Pareto distribution has a Clayton survival copula.

## 5.3 Elliptic copulas

Before introducing elliptic copulas, we define elliptical distributions. A random variable  $X$  has an elliptical distribution if its characteristic function  $\varphi_X$  verifies  $\varphi_X(t) = e^{it\mu}\psi(t^2\sigma)$  for some parameters  $\mu, \sigma$  and some function  $\psi$ . Generally a random vector  $\mathbf{X}$  follows an elliptical distribution if its characteristic function verifies

$$\varphi_{\mathbf{X}}(\mathbf{t}) = e^{it^T\mu}\psi(\mathbf{t}^T\Sigma\mathbf{t}),$$

for some column-vector  $\mu$ , some positive definite matrix  $\Sigma$ , some function  $\psi$  and  $\mathbf{t} \in \mathbb{R}^d$ . For such a distribution, the density function is given by

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{c_d}{\sqrt{\det(\Sigma)}} \phi((\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)/2),$$



for  $\mathbf{x} \in \mathbb{R}^d$ , some function  $\phi : \mathbb{R}_+ \mapsto \mathbb{R}$  such that  $\int_0^\infty x^{d/2-1} \phi(x) dx < \infty$  and some normalizing constant  $c_d$ . We get the multivariate normal distribution when  $\phi(t) = e^{-t}$  with mean vector  $\mu$  and covariance matrix  $\Sigma$ , the multivariate Student distribution with  $m$  degrees of freedom when  $\phi(t) = (1 + t/m)^{(d+m)/2}$ .

An elliptical copula is defined as

$$C(\mathbf{u}) = H(H_1^{-1}(u_1), \dots, H_d^{-1}(u_d)),$$

where  $H$  is a multivariate distribution with marginals  $H_i$  belonging to the elliptical family. In particular for a symmetric positive definite matrix  $\Sigma$ , the Gaussian and the student copulas are defined as

- Gaussian

$$C_{Ga}(\mathbf{u}) = \int_{-\infty}^{z_1} \dots \int_{-\infty}^{z_d} \tilde{c}_d e^{-\mathbf{x}^T \Sigma^{-1} \mathbf{x} / 2} dx_1 \dots dx_d,$$

where  $z_i = \Phi^{-1}(u_i)$  and  $\Phi^{-1}$  is the quantile function of the standard normal distribution.

- student

$$C_{St}(\mathbf{u}) = \int_{-\infty}^{z_1} \dots \int_{-\infty}^{z_d} \tilde{c}_d \left( 1 + \frac{\mathbf{x}^T \Sigma^{-1} \mathbf{x}}{2m} \right)^{\frac{d+m}{2}} dx_1 \dots dx_d,$$

where  $z_i = F_{St}^{-1}(u_i)$  is the quantile of a Student distribution with  $m > 0$  degrees of freedoms.

Note that in the bivariate case, the covariance matrix  $\Sigma$  simplifies to

$$\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix},$$

where  $\rho \in [0, 1]$ .

## 5.4 Properties and extreme copulas

The copulas presented in the previous subsections have a density function  $c : [0, 1]^d \mapsto [0, 1]$  since the copula function is differentiable with respect to all variables on the unit hypercube. The dependence induced by a particular copula can be quantified through the theory of concordance measures introduced by ?. The two main measures of concordance are the tau of Kendall or the rho of Spearman. The tau of Kendall of a bivariate vector  $(X, Y)$  is defined as

$$\tau(X, Y) = \mathbb{P}((X - \tilde{X})(Y - \tilde{Y}) > 0) - \mathbb{P}((X - \tilde{X})(Y - \tilde{Y}) < 0),$$

where  $(\tilde{X}, \tilde{Y})$  is an independent replicate of  $(X, Y)$ . Similarly, the rho of Spearman of  $(X, Y)$  is defined as

$$\tau(X, Y) = 3\mathbb{P}((X - \tilde{X})(Y - \bar{Y}) > 0) - 3\mathbb{P}((X - \tilde{X})(Y - \bar{Y}) < 0),$$

where  $(\tilde{X}, \tilde{Y})$  and  $(\bar{X}, \bar{Y})$  are independent replicates of  $(X, Y)$ . As these two measures satisfy the criterions of concordance measures,  $\tau(X, Y) \in [0, 1]$ ,  $\tau(X, Y) = 1$  means that the copula of  $(X, Y)$  is the upper Fréchet bound,  $\tau(X, Y) = 0$  means that the copula of  $(X, Y)$  is the independent copula (the same holds for  $\rho(X, Y)$ ). In the bivariate case, closed-form formulas are available for the copulas previously presented, see e.g. Nelsen (2006) and Joe (1997).

A desirable feature of copulas lies in the fact that they can model dependence between two or more variables with or without a tail dependency. This is characterized by the tail dependance coefficients. The upper tail coefficient of  $(X, Y)$  is defined as

$$\lambda_U(X, Y) = \lim_{t \rightarrow 1^-} \mathbb{P}(Y > F_Y^{-1}(t) | X > F_X^{-1}(t)),$$

while the lower tail coefficient  $\lambda_L(X, Y)$  is obtained when  $t \rightarrow 0^+$ . When  $X, Y$  have a continuous distribution with a dependency given by a copula  $C_{X,Y}$ , those coefficients can be rewritten as

$$\lambda_U(X, Y) = \lim_{t \rightarrow 1^-} \frac{1 - 2t - C_{X,Y}(t, t)}{1 - t} \text{ and } \lambda_L(X, Y) = \lim_{t \rightarrow 0^+} \frac{C_{X,Y}(t, t)}{t}$$

For the copulas presented here, we have  $\lambda_U = 0$  except for the Gumbel copula  $\lambda_U = 2 - 2^{1/\alpha}$  and the Student copula  $\lambda_U = 2F_{St}(\sqrt{m+1}\sqrt{1-\rho}/\rho + \rho)$ , whereas  $\lambda_L = 0$  except for the Clayton copula  $\lambda_L = 2^{-1/\alpha}$  and the Student copula  $\lambda_L = \lambda_U$ . In other words, copulas with  $\lambda_U = 0$  cannot model dependence at the right-hand tail.

Another desirable property of copulas is the max-stability. A copula function  $C$  is max-stable if

$$C(u_1, \dots, u_d) = \left( C(u_1^{1/k}, \dots, u_d^{1/k}) \right)^k,$$

for all  $k > 0$ . This property is linked to the extreme value theory since the right-hand side is the copula of component-wise maxima of a random vector sample  $(\mathbf{X}_1, \dots, \mathbf{X}_k)$ . Copulas verifying this property are called extreme copulas: the Gumbel and the Husler-Reiss copulas belong to this family. The Husler-Reiss copula is defined as follows in the bivariate case

$$C_{HR}(u_1, u_2) = \exp(\log(u_1)\Phi(d_+) + \log(u_2)\Phi(d_-)),$$

where  $d_{\pm} = 1/\alpha \pm \alpha/2 \log(\log(u_1)/\log(u_2))$  and  $\phi$  is the distribution function of the standard normal distribution.

## 5.5 Copula fitting methods

There are four main methods to calibrate copulas which differ on how the marginals are considered in the fitting process. Consider a sample of random vectors  $(\mathbf{X}_1, \dots, \mathbf{X}_n)$  and corresponding observations  $\mathbf{x}_1, \dots, \mathbf{x}_n$  where the  $i$ th marginal has a density  $f_i(\cdot; \theta_i)$  and a distribution function  $F_i(\cdot; \theta_i)$ . A (full) maximum likelihood estimation is the first option, which consists in maximizing the likelihood

$$\mathcal{L}(\alpha, \theta_1, \dots, \theta_d, \mathbf{x}_1, \dots, \mathbf{x}_n) = \prod_{i=1}^n c(F_{1,i}, \dots, F_{d,i}; \alpha) f_{1,i} \times \dots \times f_{d,i},$$

where  $F_{j,i} = F_j(x_{j,i}; \theta_j)$ ,  $f_{j,i} = f_j(x_{j,i}; \theta_j)$ ,  $\alpha$  is the parameter (vector) of the copula  $C$  and  $\theta_i$  its parameter. The optimization is carried out over the whole parameter space.

The second estimation method is the method of moments which consists as in the univariate in matching theoretical moments and empirical moments. Marginal parameters  $\theta_i$  are set by equalizing the empirical moments of the sample  $(X_{i,1}, \dots, X_{i,n})$ , while the copula parameters  $\alpha$  are determined by matching the tau of Kendall and/or the rho of Spearman.

The third estimation called inference for margins is a two-step procedure. First marginal distributions are fitted by maximum likelihood, then a pseudo sample is defined as  $\mathbf{u}_i = (u_{1,i}, \dots, u_{d,i}) = (F(x_{1,i}, \hat{\theta}_1), \dots, F(x_{d,i}, \hat{\theta}_d))$  for  $i = 1, \dots, n$ . Then the copula is fitted on  $\mathbf{u}_1, \dots, \mathbf{u}_n$  by maximizing the likelihood

$$\mathcal{L}(\alpha, \mathbf{u}_1, \dots, \mathbf{u}_n) = \prod_{i=1}^n c(u_{1,i}, \dots, u_{d,i}; \alpha).$$

The inference for margins method takes the advantage of the two steps to reduce the dimension of the likelihood from  $(\alpha, \theta_1, \dots, \theta_d)$  to  $\alpha$ . Finally, the canonical maximum likelihood method is similar to the inference for margins and consists in replacing the parametric estimate by the non-parametric estimates in the pseudo data. That is to say,  $\mathbf{u}_i = (F_n(x_{1,i}), \dots, F_n(x_{d,i}))$  which further simplifies to  $\mathbf{u}_i = (\text{rank}(x_{1,i})/n, \dots, \text{rank}(x_{d,i})/n)$ . In the following section, we only consider the inference for margins method, yet our code is generic and can be easily adapted for other estimation methods.

## 5.6 Application and copula selection

Numerical illustrations of copulas and their estimation is carried out on the loss-ALAE data of Frees & Valdez (1998). The dataset comprises 1,500 general liability claims (expressed in USD) where each claim is a two-component vector: an indemnity payment (loss) and an allocated loss adjustment expense (ALAE).

```

> data(lossalae)
> par(mfrow=c(1,2))
> plot(lossalae, log="xy", main="Scatterplot of loss-ALAE")
> plot(apply(lossalae, 2, rank)/NROW(lossalae),
+      main="rank transform of loss-ALAE")

```

On Figure 13, we plot the scatter plots of the data  $(x_i, y_i)$  and the empirical distributions evaluated at  $(x_i, y_i)$ , i.e.  $(F_{n,X}(x_i), F_{n,Y}(y_i)) = (\text{rank}(x_i)/n, \text{rank}(y_i)/n)$ .

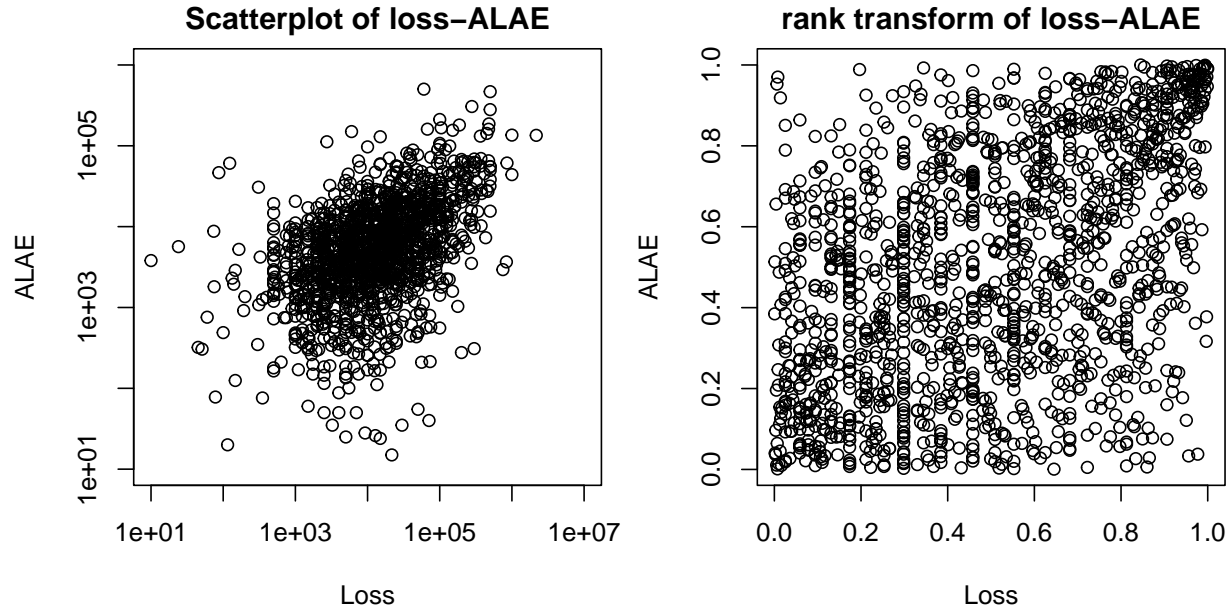


Figure 13: Loss-ALAE dataset

On this dataset, we choose to fit the following bivariate copulas: (i) the Gaussian copula  $C_{Ga}(\cdot, \cdot; \rho)$ , (ii) the Student copula  $C_{St}(\cdot, \cdot; \rho, m)$ , (iii) the Gumbel copula  $C_{Gu}(\cdot, \cdot; \alpha)$ , (iv) the Frank copula  $C_F(\cdot, \cdot; \alpha)$  and (v) the Husler-Reiss copula  $C_{HR}(\cdot, \cdot; \alpha)$ . We use the implementation done in the **fCopulae** package (part of the Rmetrics project, see <https://www.rmetrics.org/>). For ease of use, we define the following functions

```

> dnormcop <- function(U, param)
+   as.numeric(dellipticalCopula(U, rho=param[1], type="norm"))
> dtcop <- function(U, param)
+   as.numeric(dellipticalCopula(U, rho=param[1], type="t",
+                               param=param[2]))
> dgumcop <- function(U, param)
+   as.numeric(devCopula(U, type="gumbel", param=param[1]))
> dHRcop <- function(U, param)
+   as.numeric(devCopula(U, type="husler.reiss", param=param[1]))
> dfrankcop <- function(U, param)
+   as.numeric(darchmCopula(U, type="5", alpha=param[1]))

```

In addition to finding an appropriate copula, a choice of distribution must be done for marginals. A Pareto chart on both marginals shows that they follow heavy-tailed distributions.

```

> paretochart <- function(x, ...)
+   plot(-log((1:length(x))/(length(x)+1)), log(sort(x)), ...)
> paretochart(lossalae$Loss)
> paretochart(lossalae$ALAE)

```

Therefore, we choose a Pareto type II distribution and a lognormal distribution for candidate distributions

of marginals. As there is no package fitting copulas for any kind of copula, we implement the inference for margins method in the following function.

```

> fit.cop.IFM.2 <- function(obs, copula, marg, arg.margin=list(),
+                           method.margin="mle", arg.cop=list(), initpar, ...)
+ {
+   Obs1 <- obs[,1]
+   Obs2 <- obs[,2]
+   #marginal fit
+   if(marg %in% c("exp","gamma","lnorm","pareto","burr"))
+   {
+     Obs1 <- Obs1[Obs1 > 0]
+     Obs2 <- Obs2[Obs2 > 0]
+   }
+   marg1 <- do.call(fitdist, c(list(data= Obs1, distr=marg,
+                                   method=method.margin), arg.margin))
+   marg2 <- do.call(fitdist, c(list(data= Obs2, distr=marg,
+                                   method=method.margin), arg.margin))
+
+   #fitted distrib. func.
+   comput.cdf <- function(fit, obs)
+   {
+     para <- c(as.list(fit$estimate), as.list(fit$fix.arg))
+     distname <- fit$distname
+     pdistname <- paste("p", distname, sep = "")
+     do.call(pdistname, c(list(q = obs), as.list(para)))
+   }
+   #pseudo data
+   pseudomarg1 <- comput.cdf(marg1, Obs1)
+   pseudomarg2 <- comput.cdf(marg2, Obs2)
+   U <- cbind(pseudomarg1, pseudomarg2)
+   #log likelihood
+   copLogL <- function(x)
+   {
+     if(all(arg.cop$lower <= x) && all(arg.cop$upper >= x))
+       res <- -sum(remove.naninf(log(copula(U, param=x))))
+     else res <- Inf
+     return(res)
+   }
+   resopt <- optim(par=initpar, fn=copLogL, method="L-BFGS-B",
+                  lower=arg.cop$lower, upper=arg.cop$upper, ...)
+
+   list(marg1=marg1, marg2=marg2, copula=
+         list(name=arg.cop$name, alpha=resopt$par))
+ }
> remove.naninf <- function(x)
+   x[!is.nan(x) & is.finite(x)]

```

The copulas are now fitted using the function `fit.cop.IFM.2` by defining the corresponding `arg.cop` argument. Note that the marginal distributions are fitted using `fitdist`.

```

> library(fCopulae)
> argnorm <- list(length=1, lower=0, upper=1, name="Gaussian")
> argt <- list(length=2, lower=c(0,0), upper=c(1,1000),
+             name="Student")
> arggum <- list(length=1, lower=1, upper=100, name="Gumbel")
> argHR <- list(length=1, lower=0, upper=1000, name="Husler-Reiss")
> argfrank <- list(length=1, lower=-1000, upper=1000, name="Frank")

```

```

> #Gaussian copula
> fgausspareto <- fit.cop.IFM.2(lossalae, copula= dnormcop,
+   marg="pareto", arg.margin=list(start=list(shape=10, scale=100),
+   lower=c(1, 1/2)), arg.cop= argnorm, initpar=1/2)
> #Student copula
> ftpareto <- fit.cop.IFM.2(lossalae, copula= dtcop,
+   marg="pareto", arg.margin=list(start=list(shape=10, scale=100),
+   lower=c(1, 1/2)), arg.cop= argt, initpar=c(1/2, 4))
> ftpareto$copula$alpha[2] <- round(ftpareto$copula$alpha[2])
> #Gumbel copula
> fgumbelpareto <- fit.cop.IFM.2(lossalae, copula= dgumcop,
+   marg="pareto", arg.margin=list(start=list(shape=10, scale=100),
+   lower=c(1, 1/2)), arg.cop= arggum, initpar=10)
> #Husler-Reiss copula
> fHRpareto <- fit.cop.IFM.2(lossalae, copula= dHRcop,
+   marg="pareto", arg.margin=list(start=list(shape=10, scale=100),
+   lower=c(1, 1/2)), arg.cop= argHR, initpar=10)
> #Frank copula
> ffrankpareto <- fit.cop.IFM.2(lossalae, copula= dfrankcop,
+   marg="pareto", arg.margin=list(start=list(shape=10, scale=100),
+   lower=c(1, 1/2)), arg.cop= argfrank, initpar=10)
> recap <- function(x)
+ {
+   res <- c(alpha=x$copula$alpha, x$marg1$estimate, x$marg2$estimate)
+   if(length(res) < 6)
+     res <- c(res[1], NA, res[2:5])
+   res <- as.matrix(res)
+   colnames(res) <- x$copula$name
+   res
+ }
> round(cbind(recap(fgausspareto), recap(ftpareto),
+   recap(fHRpareto), recap(fgumbelpareto),
+   recap(ffrankpareto)      ), 4)

```

	Gaussian	Student	Husler-Reiss	Gumbel	Frank
alpha	0.4783	0.4816	1.1133	1.4444	3.1141
	NA	10.0000	NA	NA	NA
shape	1.2379	1.2379	1.2379	1.2379	1.2379
scale	16236.7759	16236.7759	16236.7759	16236.7759	16236.7759
shape	2.2233	2.2233	2.2233	2.2233	2.2233
scale	15142.4430	15142.4430	15142.4430	15142.4430	15142.4430

The level of dependency seems low since the value of the first parameter is either close to 1 (for non-elliptic copulas) or close to 0 for elliptic copulas. Unsurprisingly, the fitted parameters of the marginal distributions are identical. In order to assess the quality of the fit, we look at the tail coefficients  $\lambda_U(X, Y)$  and  $\lambda_L(X, Y)$ , which are computable given a copula. They can be estimated on data by using a non-parametric estimate of  $C(t, t)$ , i.e. the empirical bivariate distribution function  $C_n(t, t) = \sum_{i=1}^n \mathbb{1}_{x_i \leq t} \mathbb{1}_{y_i \leq t} / n$ . This is done by the following function.

```

> Lemp <- function(u, obs)
+   sapply(1:length(u), function(i)
+     1/NROW(obs)*sum(obs[,1] <= u[i] & obs[,2] <= u[i]) )/u
> Uemp <- function(u, obs)
+   (1-2*u+sapply(1:length(u), function(i)
+     1/NROW(obs)*sum(obs[,1] <= u[i] & obs[,2] <= u[i]) ))/(1-u)
> Lcop <- function(u, pcop, param=param)
+   pcop(cbind(u, u), param=param)/u

```

```
> Ucop <- function(u, pcop, param=param)
+ (1-2*u+pcop(cbind(u, u), param=param))/(1-u)
```

The plot of the two tail coefficients is now possible.

```
> u <- seq(0, 0.4, length=101)
> rklossalae <- apply(lossalae, 2, rank)/NROW(lossalae)
> plot(u, Lemp(u, rklossalae), type="l", main="Lower coefficient",
+      ylim=c(0, .6), xlab="u", ylab="L(u)")
> lines(u, Lcop(u, pgumcop, fgumbelpareto$copula$alpha), lty=2)
> lines(u, Lcop(u, pHRcop, fHRpareto$copula$alpha), lty=3)
> lines(u, Lcop(u, ptcop, ftpareto$copula$alpha), lty=4, col="grey25")
> lines(u, Lcop(u, pnormcop, fgausspareto$copula$alpha), lty=5, col="grey25")
> lines(u, Lcop(u, pfrankcop, ffrankpareto$copula$alpha), lty=6, col="grey25")
> legend("bottomright", lty=1:6, col=c(rep("black", 3), rep("grey25", 3)),
+      leg=c("emp.", "Gumbel", "Husler-Reiss", "Student", "Gaussian", "Frank"))
```

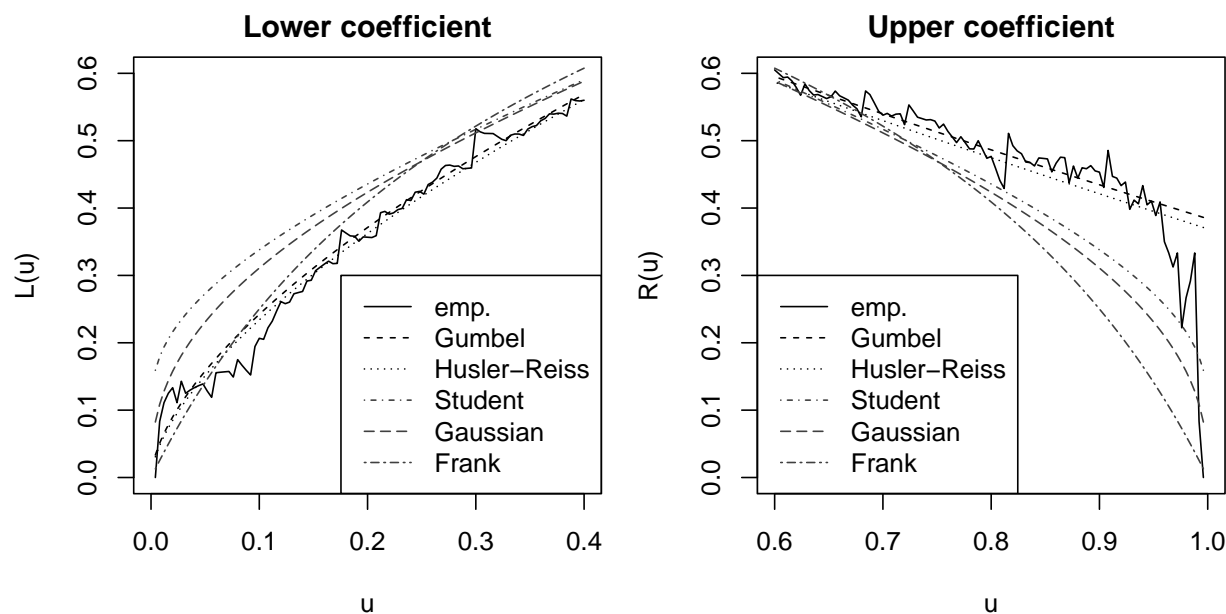


Figure 14: Tail coefficients

As shown on Figure 14, the tail coefficients are best approximated by the extreme copulas : Gumbel and Husler-Reiss. When considering the tau of Kendall and the rho of Spearman, these two copulas are also reasonably good. Therefore, we continue the analysis only with the Gumbel and the Husler-Reiss copula.

```
> cbind(emp=cor(lossalae, method="kendall")[1,2],
+      Frank=taufrankcop(ffrankpareto$copula$alpha),
+      Gumbel= taugumcop(fgumbelpareto$copula$alpha),
+      HR= tauHRcop(fHRpareto$copula$alpha),
+      Gauss = taunormcop(fgausspareto$copula$alpha),
+      Student = tautcop(ftpareto$copula$alpha))
> cbind(emp=cor(lossalae, method="spearman")[1,2],
+      Frank=rhofrankcop(ffrankpareto$copula$alpha),
+      Gumbel= rhogumcop(fgumbelpareto$copula$alpha),
+      HR= rhoHRcop(fHRpareto$copula$alpha),
+      Gauss = rhonormcop(fgausspareto$copula$alpha),
+      Student = rhotcop(ftpareto$copula$alpha))
```

	emp	Frank	Gumbel	HR	Gauss	Student
Tau	0.3154175	0.3171205	0.3076784	0.3008567	0.317476	0.3198715

	emp	Frank	Gumbel	HR	Gauss	Student
Rho	0.451872	0.4622857	0.4424592	0.4360085	0.45	0.45

Given a copula, quantities of interest can be estimated by a Monte-Carlo method. Focusing on the distribution of the total expense, i.e. the sum of loss and ALAE, we simulate this sum with the following template function

```
> simul.cop.2 <- function(n, rcopula, fit)
+ {
+   U <- rcopula(n, fit$copula$alpha)
+   qmarg1 <- paste("q", fit$marg1$distname, sep="")
+   qmarg2 <- paste("q", fit$marg2$distname, sep="")
+   cbind(
+     X1=do.call(qmarg1, c(list(p=U[,1]), fit$marg1$estimate)),
+     X2=do.call(qmarg2, c(list(p=U[,2]), fit$marg2$estimate))
+   )
+ }
```

where `rcopula` is the copula random generator defined as

```
> rgumcop <- function(n, param)
+   revCopula(n, type="gumbel", param=param[1])
> rHRcop <- function(n, param)
+   revCopula(n, type="husler.reiss", param=param[1])
> rindep <- function(n,param)
+   cbind(runif(n), runif(n))
```

Choosing a sample size of  $n = 1e4$ , we plot the empirical distribution functions of the sum for the Gumbel, the Husler-Reiss and the independent copulas.

```
> sumlossalae <- rowSums(lossalae)
> n <- 10^4
> sumgumpareto <- rowSums(simul.cop.2(n, rgumcop, fgumbelpareto))
> sumHRpareto <- rowSums(simul.cop.2(n, rHRcop, fHRpareto))
> sumindeppareto <- rowSums(simul.cop.2(n, rindep, fgumbelpareto))
> plot(ecdf(sumlossalae), log="x", main="", xlim=range(sumlossalae))
> z <- 10^seq(1, log(max(sumlossalae)), length=201)
> lines(z, ecdf(sumgumpareto)(z), lty=2)
> lines(z, ecdf(sumHRpareto)(z), lty=3)
> lines(z, ecdf(sumindeppareto)(z), lty=4)
> legend("bottomright", lty=1:4, col="black",
+       leg=c("emp.", "Gumbel", "Husler-Reiss", "indep."))
```

On Figure 15, we observe that both extreme copulas are particularly adapted for the `lossalae` dataset irrespective of the choice of the marginal distribution. Yet, we notice that the choice of the lognormal distribution better suits the tail of the empirical distribution. However, as extreme claims above 1 million of dollar are less observed (by definition) than large claims (between 100 thousand and 1 million), the experimenter must take a prudential view of the right-hand tail without overfitting data. The independent copula seems to be a reasonable approximation of the tail of the distribution (yet not of the rest of the distribution).

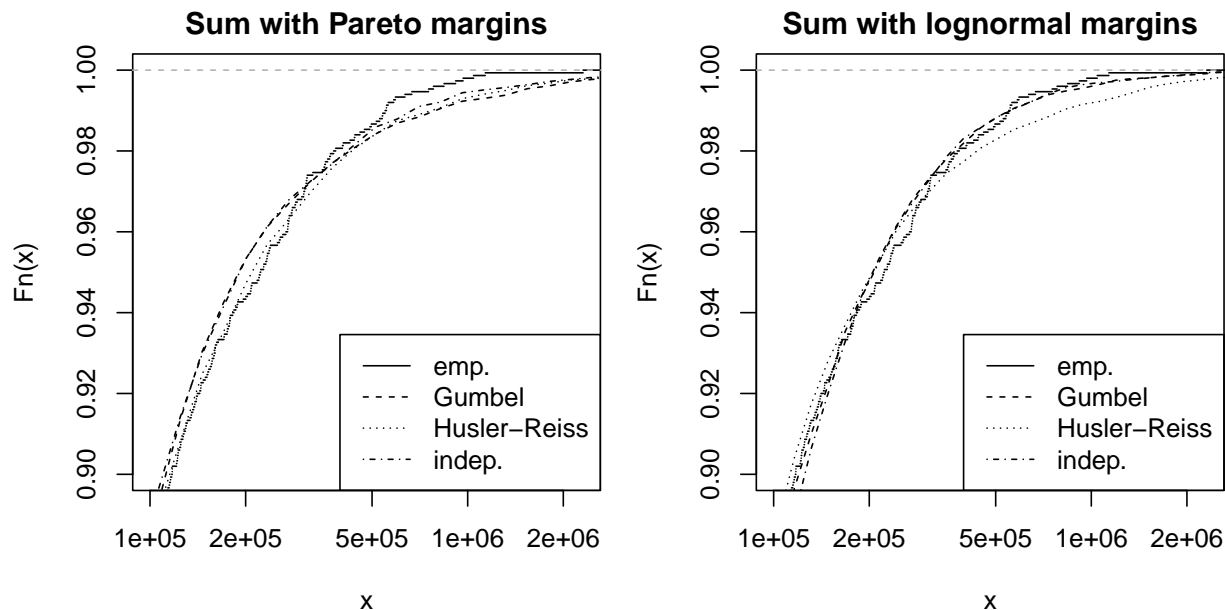


Figure 15: Tails of the distribution of the total claim

## 6 Parametric inference with covariates

Assume as in Section 2 that the  $X_i$ 's are independent and identically distributed random variables according to a generic random variable  $X$  with identical distribution function  $F(\cdot; \theta)$  for  $\theta \in \Theta \subset \mathbb{R}^d$  is a strong assumption. As discussed in Chapter 4, if  $X$  is a binary random variable indicating if a loaner had experienced non-payment, the probability  $\mathbb{P}(X = 1)$  can be function of some covariate, such as the length and the amount of the loan. Similarly, in chapter 8, if  $X$  is a binary random variable indicating if an insured is still alive, or not, at some specific date,  $\mathbb{P}(X = 1)$  can be function of the age of the insured today (or at the specific date). And finally, in Chapter 13, if  $X$  denotes the number of claims per claim in motor insurance,  $X$  is usually a function of the age of the driver. Hence, it would be realistic to have models that could incorporate covariates.

Using standard notations,  $y$  will now denote the variable of interest, and  $\mathbf{x} = (x_1, \dots, x_k)$  will denote a collection of  $k$  covariates. The extension of the previous framework will be to assume that not only we observe  $(y_1, \dots, y_n)$ , but also associated covariates,  $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ . And  $(y_1, \dots, y_n)$  are independent realizations of a random sample  $(Y_1, \dots, Y_n)$  such that  $Y_i$  has distribution  $F(\cdot; \theta(\mathbf{x}_i))$  where now the parameter is a function  $\theta: \mathbb{R}^k \rightarrow \Theta \subset \mathbb{R}^d$ .

### 6.1 Linear regression

Consider the case where only one covariate is considered here, and that it is assumed to be a continuous variable. In the (simple) linear regression model, we assume that  $Y|X = x$  has a normal distribution  $\mathcal{N}(\theta(x), \sigma^2)$ , for some  $\sigma^2$ . Since all conditional variances are assumed to be identical, such models will be called homoscedastic. In order to have a *linear* model, we will assume that  $\theta(\cdot)$  is a linear function of the parameter (not necessarily of the covariate  $x$ ).

Note that instead of writing

$$Y|X = x \sim \mathcal{N}(\beta_0 + \beta_1 x, \sigma^2),$$

linear models are usually written as

$$Y = \beta_0 + \beta_1 x + \varepsilon \quad \text{where } \varepsilon \sim \text{mathcal{N}}(0, \sigma^2).$$

The R function to estimate parameters of the model (namely  $(\hat{\theta}, \hat{\sigma}^2) = (\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}^2)$ ) is the `lm` function (standing for linear model). The generic code is



```
> linearmodel <- lm(y~x)
```

where the intercept here is implicit. In other words, the previous code is equivalent to

```
> linearmodel <- lm(y~x)
```

where  $\mathbf{x}$  and  $\mathbf{y}$  are two numeric vectors with the same length. If one observation is missing for one vector, the associated value will be removed on the other vector.

Under the Gaussian noise assumption, the estimator of  $\beta$  by maximum likelihood estimation or by the least-square method are equivalent. using maximum likelihood. The likelihood function is here

$$\mathcal{L}(\beta_0, \beta_1, \sigma^2, (y_1, x_1), \dots, (y_n, x_n)) = \prod_{i=1}^n \frac{e^{-(y_i - [\beta_0 + \beta_1 x_i])^2 / 2\sigma^2}}{\sqrt{2\pi\sigma^2}}$$

and the argument of the maximum of this function can be written explicitly, since

$$\log \mathcal{L}(\beta_0, \beta_1, \sigma^2, (y_1, x_1), \dots, (y_n, x_n)) = -\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - [\beta_0 + \beta_1 x_i])^2 - \log(\sqrt{2\pi\sigma^2}).$$

Equivalently,  $(\hat{\beta}_0, \hat{\beta}_1)$  is the unique solution of a least square program, since

$$(\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}^2) = \operatorname{argmin} (y_i - [\beta_0 + \beta_1 x_i])^2.$$

For the estimator of  $\sigma^2$ , maximum likelihood and least-square are the same up to a constant.

For a multiple regression model, with  $k$  explanatory variables,

$$Y|\mathbf{X} = \mathbf{x} \sim \mathcal{N}(\beta_0 + \beta_1 x_1 + \dots + \beta_k x_k = \boldsymbol{\beta}'\mathbf{x}, \sigma^2),$$

with the convention that  $\mathbf{x} = (1, x_1, \dots, x_k)$ . As previously, this model will usually be written

$$Y = \beta_0 + \beta_1 x + \dots + \beta_k x_k + \varepsilon = \boldsymbol{\beta}'\mathbf{x} + \varepsilon \quad \text{where } \varepsilon \sim \mathcal{N}(0, \sigma^2).$$

This is a large class of models that includes simple and multiple regression as well as ANOVA models (where  $x \in \{0, 1\}$  indicates groups). The generic code to estimate this model is

```
> linearmodel <- lm(y~x1+x2+x3)
```

$\mathbf{x}_1, \dots, \mathbf{x}_k$  and  $\mathbf{y}$  are numeric vectors with the same length. As discussed in the first chapter,  $\mathbf{y} \sim \mathbf{x}_1 + \mathbf{x}_2$  is a symbolic expression since the  $+$  is not the standard addition symbol, but denotes the operator to regress on  $\mathbf{x}_1$  and  $\mathbf{x}_2$ . In order to run a regression on  $X_1 + X_2$ , one should use the `I` operator, that will inhibit the interpretation of operators such as  $+$  (but also  $*$  or  $\wedge$ ) as a formula operator, so it will be used as the arithmetical operator.

```
> linearmodel <- lm(y~I(x1+x2)+x3)
```

Similarly, to run a regression on  $X_1^2$  (or any power transformation), one should use

```
> linearmodel <- lm(y~I(x1^2)+x2+x3)
```

The interest of using functions of initial variates, instead of creating new variates (like `x1square<- x1^2`) is that prediction can be obtained directly from initial variates  $\mathbf{x}_1, \dots, \mathbf{x}_k$ .

In the case where  $\mathbf{X}$  is the matrix containing as vectors all explanatory variables, and where the first column is the unit vector  $\mathbf{1}$ , and  $\mathbf{y} = (y_1, \dots, y_n)$ , then the maximum likelihood estimator of the linear model coefficient is then

$$\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_k)' = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}.$$

If we assume that noises are i.i.d. Gaussian noise (i.e.  $\varepsilon_i \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$ ), then from the Gauss-Markov theorem, in the class of estimators that are linear functions of the explanatory covariates,  $\hat{\boldsymbol{\beta}}$  is the minimum variance unbiased estimators of  $\boldsymbol{\beta}$  (see e.g. de Jong & Heller (2008) or Frees (2010) for more details, and exemplified on insurance data).

**Remark 1** *In the theory of linear regression, the normality assumption is not necessary to have the previous result (centered with identical variance will be a sufficient condition). The Gaussian assumption is only necessary for building confidence intervals and hypothesis testing.*

The estimator for the mean square error is defined as

$$\hat{\sigma}^2 = \frac{1}{n - (k + 1)} \sum_{i=1}^n \underbrace{[y_i - \mathbf{x}'_i \hat{\boldsymbol{\beta}}]}_{\hat{\varepsilon}_i}$$

The estimated variance matrix of estimator  $\hat{\boldsymbol{\beta}}$  is then  $\hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}$ . All those quantities can be obtained using `summary` of an "lm" object.

Consider for instance the age of a husband and his wife, when they purchase a joint life insurance contract,

```
> data(canlifins)
> ages <- subset(canlifins, EntryAgeM > 20)
> jointinsurance <- data.frame(Male=ages$EntryAgeM,
+                               Female=ages$EntryAgeF)
> linearmodel <- lm(Female~Male,data=jointinsurance)
> summary(linearmodel)
```

Call:

```
lm(formula = Female ~ Male, data = jointinsurance)
```

Residuals:

Min	1Q	Median	3Q	Max
-66.101	-2.251	0.408	2.762	40.367

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	6.490587	0.420821	15.42	<2e-16 ***
Male	0.862359	0.006155	140.11	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.692 on 14886 degrees of freedom

Multiple R-squared: 0.5687, Adjusted R-squared: 0.5687

F-statistic: 1.963e+04 on 1 and 14886 DF, p-value: < 2.2e-16

The column `Estimate` contains estimates  $\hat{\boldsymbol{\beta}}$ , and their standard deviation are in the column `Std. Error`. The ratio of the two is the  $t$ -statistic of a test  $H_0 : \beta_j = 0$  again  $H_1 : \beta_j \neq 0$ , and this statistic is given in the `t value` column. The  $p$ -value associated to this test is the `Pr(>|t|)` column. The `***` indicate that both coefficient are significant with probability lower than 0.1%. In the lines above, the `Residual standard error` is  $\hat{\sigma}$  (the square root of  $\hat{\sigma}^2$ ). The  $R^2$ , defined as

$$R^2 = \frac{\sum_{i=1}^n [\mathbf{x}'_i \hat{\boldsymbol{\beta}} - \bar{y}]^2}{\sum_{i=1}^n [y_i - \bar{y}]^2} = \frac{\sum_{i=1}^n [\hat{y}_i - \bar{y}]^2}{\sum_{i=1}^n [y_i - \bar{y}]^2} = 1 - \frac{\sum_{i=1}^n [y_i - \hat{y}_i]^2}{\sum_{i=1}^n [y_i - \bar{y}]^2}$$

is the **Multiple R-squared**, while the **Adjusted R-squared** is defined as

$$\bar{R}^2 = R^2 - (1 - R^2) \frac{k}{n - (k + 1)}.$$

Finally, the **F-statistic** is the Fisher statistic of a test  $H_0 : \beta_{(0)} = \mathbf{0}$ , where  $\beta_{(0)} = (\beta_1, \dots, \beta_k)$ .

## 6.2 Generalized linear models

The model described in the previous section can be extended to the class of distributions in the exponential family (as in McCullagh & Nelder (1989)). A Generalized Linear Model is characterized by three components:

1. a random component:  $Y_i$  follows a distribution of the exponential family  $\mathcal{F}_{exp}(\theta_i, \phi_i, a, b, c)$ ,
2. a systematic component: the covariate vector  $X_i$  produces a linear predictor  $\eta_i = X_i^T \beta$ ,
3. a link function  $g : \mathbb{R} \mapsto S$  which is monotone, differentiable and invertible, such that  $E(Y_i) = g^{-1}(\eta_i)$ ,

for  $i \in \{1, \dots, n\}$ , where  $\theta_i$  is the shape parameter,  $\phi_i$  the dispersion parameter and  $a, b, c$  three functions.  $\mathcal{F}_{exp}(\theta, \phi, a, b, c)$  denotes the natural form of the exponential family defined as

$$f_Y(y) = \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} - c(y, \phi)\right),$$

for  $y \in \mathbb{N}, \mathbb{R}_+$  or  $\mathbb{R}$ . The link function  $g$  is said to be canonical when  $\theta_i = \eta_i$ .

For instance, one can consider the case where  $Y$  is a binary variable, taking values 0 and 1. The logistic regression will be obtained when

$$Y|X = x \sim \mathcal{B}(\theta(x)), \text{ where } \theta(x) = \mathbb{P}(Y = 1|X = x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}},$$

while the probit model is obtained when

$$\theta(x) = \mathbb{P}(Y = 1|X = x) = \Phi(\beta_0 + \beta_1 x),$$

where  $\Phi$  is the cumulative distribution function of the  $\mathcal{N}(0, 1)$  distribution. In the case where  $Y$  is a count variable, such that conditionally on  $X$ ,  $Y$  has a Poisson distribution, then

$$Y|X = x \sim \mathbb{P}(\theta(x)), \text{ where } \theta(x) = \mathbb{E}(Y|X = x) = e^{\beta_0 + \beta_1 x}.$$

For those models, maximum likelihood estimation is used, as discussed in Chapters 4 and 13. In actuarial science, typical GLMs are summarized in the following table.

Law	Canonical link	Mean	Used for
$\mathcal{N}(\mu, \sigma^2)$	identity $\eta_i = \mu_i$	$\mu = X\beta$	std linear reg
$\mathcal{B}(\mu)$	logit $\eta_i = \log\left(\frac{\mu}{1-\mu}\right)$	$\mu = \frac{1}{1+e^{-X\beta}}$	rate modelling
$\mathcal{P}(\mu)$	log $\eta_i = \log(\mu_i)$	$\mu = e^{X\beta}$	claim frequency
$\mathcal{G}(\alpha, \beta)$	inverse $\eta_i = \frac{1}{\mu_i}$	$\mu = (X\beta)^{-1}$	claim severity
$\mathcal{IN}(\mu, \lambda)$	squared inverse $\eta_i = -\frac{1}{\mu_i^2}$	$\mu = (X\beta)^{-2}$	claim severity

Table 2: Family and link functions

## References

- Andersen, S. (1957), ‘On the collective theory of risk in case of contagion between claims’, *Bulletin of the Institute of Mathematics and its Applications* **12**, 2775–279.
- Arnold, B. C. (1983), *Pareto Distributions*, International Co-operative Publishing House.
- Asmussen, S. & Albrecher, H. (2010), *Ruin Probabilities*, 2nd edn, World Scientific New Jersey.
- Asmussen, S. & Rolski, T. (1991), ‘Computational methods in risk theory: A matrix algorithmic approach’, *Insurance: Mathematics and Economics* **10**(4), 259–274.
- Bernegger, S. (1997), ‘The Swiss Re exposure curves and the MBBEFD distribution class’, *Astin Bull.* **27**(1), 99–111.
- Bickel, P. & Doksum, K. (2001), *Mathematical Statistics: Basic Ideas and Selected Topics*, Vol. 1, Prentice Hall.
- Casella, G. & Berger, R. (2002), *Statistical Inference*, Duxbury Thomson Learning.
- D’Agostino, R. & Stephens, M. (1986), *Goodness-of-Fit Techniques*, first edn, Dekker.
- de Jong, P. & Heller, G. (2008), *Generalized Linear Models for Insurance Data*, Cambridge University Press.
- Delignette-Muller, M. & Dutang, C. (2013), fitdistrplus: an R Package for Fitting Distributions. working paper.
- Dutang, C., Goulet, V. & Pigeon, M. (2008), ‘actuar: an R package for Actuarial Science’, *Journal of Statistical Software* **25**(7).
- Embrechts, P., Lindskog, F. & McNeil, A. (2001), Modelling dependence with copulas and applications to risk management, Technical report, ETH Zurich.
- Frees, E. (2010), *Regression Modeling with Actuarial and Financial Applications*, Cambridge University Press.
- Frees, E. W. & Valdez, E. (1998), ‘Understanding Relationships Using Copulas’, *North American Actuarial Journal* **2**(1).
- Frees, E. W. & Wang, P. (2006), ‘Copula credibility for aggregate loss models’, *Insurance: Mathematics and Economics* **38**, 360–373.
- Gendron, M. & Crepeau, H. (1989), ‘On the computation of the aggregate claim distribution when individual claims are Inverse Gaussian’, *Insurance: Mathematics and Economics* **8**, 251–258.
- Hyndman, R. & Fan, Y. (1996), ‘Sample quantiles in statistical packages’, *American Statistician* **50**, 361–365.
- Joe, H. (1997), Multivariate dependence measure and data analysis, in ‘Monographs on Statistics and Applied Probability’, Vol. 73, Chapman & Hall.
- Johnson, N., Kemp, A. & Kotz, S. (2005), *Univariate Discrete Distributions*, 3rd edn, Wiley-Interscience.
- Klugman, S. A., Panjer, H. H. & Willmot, G. E. (2009), *Loss Models: From Data to Decisions*, Wiley Series in Probability and Statistics.
- Marshall, A. W. & Olkin, I. (1988), ‘Families of multivariate distributions’, *Journal of the American Statistical Association* **83**(403), 834–841.
- McCullagh, P. & Nelder, J. A. (1989), *Generalized linear models (Second edition)*, London: Chapman & Hall.
- McNeil, A. (1997), ‘Estimating the tails of loss severity distributions using extreme value theory’, *ASTIN Bulletin* **27**(1), 117–137.
- Nelsen, R. B. (2006), *An introduction to copulas*, Springer.

Olver, F. W. J., Lozier, D. W., Boisvert, R. F. & Clark, C. W., eds (2010), *NIST Handbook of Mathematical Functions*, Cambridge University Press.

**URL:** <http://dlmf.nist.gov/>

Panjer, H. H. (1981), 'Recursive evaluation of a family of compound distributions', *Astin Bull.* **12**(1), 22–26.

Pearson, K. (1895), 'Contributions to the mathematical theory of evolution, ii: Skew variation in homogeneous material', *Philosophical Transactions of the Royal Society of London* .

Sklar, A. (1959), 'Fonctions de répartition à n dimensions et leurs marges', *Publications de l'ISUP de Paris* **8** 8, 229–231.

Tse, Y. (2009), *Nonlife Actuarial Models: Theory, Methods and Evaluation*, International Series on Actuarial Science, Cambridge University Press.