



A Software Tool for the Exponential Power Distribution: The `normalp` package

Angelo M. Mineo
Università di Palermo

Mariantonietta Ruggieri
Università di Palermo

Abstract

In this paper we present the `normalp` package, a package for the statistical environment R that has a set of tools for dealing with the exponential power distribution. In this package there are functions to compute the density function, the distribution function and the quantiles from an exponential power distribution and to generate pseudo-random numbers from the same distribution. Moreover, methods concerning the estimation of the distribution parameters are described and implemented. It is also possible to estimate linear regression models when we assume the random errors distributed according to an exponential power distribution. A set of functions is designed to perform simulation studies to see the suitability of the estimators used. Some examples of use of this package are provided.

Keywords: exponential power distribution, R, estimation, linear regression.

1. Introduction

The Exponential Power Distribution (EPD) can be considered a general distribution for random errors. In statistical inference the usual hypothesis on random errors is that they are distributed as a normal distribution, but often this hypothesis is not suitable to the study of natural phenomena. Usually, in literature two alternative approaches are used. The first is related to the outliers theory and deals with robust methods, that are often disputable. The second approach looks for suitable distributional models more general than the gaussian one; in this paper we consider the latter approach.

The first formulation of the Exponential Power Distribution could be ascribed [Subbotin \(1923\)](#). Subbotin, starting from these two axioms:

1. The probability of a random error ε depends only on the absolute value of the error itself and can be expressed by a function $\varphi(\varepsilon)$ having continuous first derivative almost

everywhere;

2. The most likely value of a quantity, of which direct measurements are known, must not depend on the used unit of measure;

obtained a probability distribution with density function:

$$\varphi(\varepsilon) = \frac{mh}{2\Gamma(1/m)} \exp[-h^m |\varepsilon|^m]$$

with $-\infty < \varepsilon < +\infty$, $h > 0$ and $m \geq 1$; this distribution is also known as General Error Distribution (GED).

The second axiom considered by Subbotin is the same of the second axiom used by Gauss to derive the usual normal error distribution; the first axiom used by Subbotin is more general: indeed, Gauss considers that the arithmetic mean represents the best way to combine observations, instead of the condition on the first derivative of the function $\varphi(\varepsilon)$ used by Subbotin.

2. The exponential power distribution

Following the procedure introduced by Pearson (1895), Lunetta (1963) derived a different parametrization of the exponential power distribution by solving the differential equation:

$$\frac{d \log f}{dx} = p \cdot \frac{\log f - \log a}{x - c}$$

that gives the probability density function:

$$f(x) = \frac{1}{2\sigma_p p^{1/p} \Gamma(1 + 1/p)} \exp\left(-\frac{|x - \mu|^p}{p\sigma_p^p}\right) \quad (1)$$

with $-\infty < x < +\infty$, $-\infty < \mu < +\infty$, $\sigma_p > 0$ and $p > 0$.

This distribution, called also by Vianelli (1963) normal distribution of order p (this explains the name of the package described in the following paragraphs), is characterized by three parameters:

$$\mu = E[X] = \int_{-\infty}^{+\infty} x f(x) dx$$

the location parameter,

$$\sigma_p = \{E[|X - \mu|^p]\}^{1/p} = \left\{ \int_{-\infty}^{+\infty} |x - \mu|^p f(x) dx \right\}^{1/p}$$

the scale parameter and p the shape parameter. The last parameter determines the shape of the curve; in this way, it is linked to the thickness of the tails, and thus to the kurtosis, of the distribution. In fact, by changing the shape parameter p , the exponential power distribution describes both leptokurtic ($0 < p < 2$) and platikurtic ($p > 2$) distributions; in particular, we obtain the Laplace distribution for $p = 1$, the normal distribution for $p = 2$ and the Uniform distribution for $p \rightarrow \infty$. A recent review about this distribution family has been produced by Chiodi (2000). In previous papers, Chiodi (1986, 1995) presented some procedures for the

generation of pseudo-random numbers from a standardized exponential power distribution, i.e. an EPD with $\mu = 0$ and $\sigma_p = 1$. These procedures, based on a generalization of the well-known Box-Muller method (1958) for generating normal pseudo-random numbers, or on acceptance-rejection rules, are, of course, very useful in simulation studies, concerning the behavior of estimators under departures from normality. Other procedures for generating pseudo-random numbers from this distribution have been presented by Johnson (1979) and Tadikamalla (1980), among others.

Finally, we have to note that, although with a different parametrization, the exponential power distribution is dealt by Bayesian inference too, when there is the problem to specify a suitable a priori distribution (see, among others, Box and Tiao 1992, Choy and Smith 1997 and Achcar De Araujo Pereira 1999). A bivariate exponential power distribution has been introduced by De Simone (1968) and Taguchi (1978), while a multivariate formulation of this distribution can be found in Fang, Kotz, and Ng (1989) and Krzanowski and Marriott (1994).

3. Estimation of the exponential power distribution parameters

By assuming that the shape parameter p is known, the location and scale parameters could be easily obtained by using the maximum likelihood estimation method. The estimate of the shape parameter p is an open problem, so far. Anyway several procedures have been proposed, some of which are very interesting. In literature, the main proposals are based on the maximum likelihood estimation method or on methods based on the computation of kurtosis indices. The derivation of the maximum likelihood estimators does not bring, formally, many problems; on the other hand, the maximum likelihood estimators have suitable properties, at least asymptotically, but in this case, i.e. in the case of the estimation of the shape parameter p , we could meet with computational difficulties by using numerical methods to solve the obtained equation. A new approach, to estimate the location and scale parameter of this distribution based on a genetic algorithm, has been recently proposed by Vitrano and Baragona (2004), but it does not present substantial improvements in comparison with others numerical methods based on the direct evaluation of the likelihood function.

The problem of estimating the linear regression parameters when we have random errors distributed according to an exponential power distribution has been treated by Zeckhauser and Thompson (1970) and A.M. Mineo (1995).

3.1. Estimation of the location and scale parameters

Let's assume to have a sample of n i.i.d. observations drawn from (1), then the likelihood function is:

$$L(x; \mu, \sigma_p, p) = [2p^{1/p}\sigma_p\Gamma(1 + 1/p)]^{-n} \exp\left(-\frac{\sum_{i=1}^n |x_i - \mu|^p}{p\sigma_p^p}\right) \quad (2)$$

and the log-likelihood function is:

$$l(x; \mu, \sigma_p, p) = \log L(x; \mu, \sigma_p, p) = -n \log[2p^{1/p}\sigma_p\Gamma(1 + 1/p)] - \frac{\sum_{i=1}^n |x_i - \mu|^p}{p\sigma_p^p} \quad (3)$$

By deriving the log-likelihood function with respect to μ and σ_p and by equalizing the obtained

expressions to zero, we have the following equations:

$$\frac{dl(x)}{d\mu} = -\frac{1}{\sigma_p^p} \sum_{i=1}^n |x_i - \mu|^{p-1} \text{sign}(x_i - \mu) = 0 \quad (4)$$

$$\frac{dl(x)}{d\sigma_p} = -\frac{n}{\sigma_p} + \frac{1}{\sigma_p^{p+1}} \sum_{i=1}^n |x_i - \mu|^p = 0 \quad (5)$$

The equation (4) does not have, in general, an explicit solution and is solved by means of numerical methods, while from (5) we get the maximum likelihood estimator of σ_p :

$$\hat{\sigma}_p = \left(\frac{\sum_{i=1}^n |x_i - \mu|^p}{n} \right)^{1/p}. \quad (6)$$

Vianelli (1963) called $\hat{\sigma}_p$ *power deviation of order p*; it can be seen as a variability index that generalizes the standard deviation.

Chiodi shows (1988b) that the maximum likelihood estimator of μ is unbiased, with asymptotic variance given by:

$$\text{var}(\hat{\mu}) = \frac{\Gamma(1/p)}{\Gamma(2-1/p)} \frac{p^{2/p-2}}{n} \sigma_p^2, \quad (7)$$

while Lunetta (1966), when μ is known, obtains the sampling distribution of $\hat{\sigma}_p^p$:

$$f(x) = \frac{\lambda^c}{\Gamma(c)} x^{c-1} e^{-\lambda x} \quad (8)$$

i.e. a Gamma distribution with $\lambda = n/(p\sigma_p^p)$, $c = n/p$, $E(\hat{\sigma}_p^p) = \sigma_p^p$, $\text{var}(\hat{\sigma}_p^p) = (p \cdot \sigma_p^{2p})/n$.

If μ is unknown, these results are only asymptotically unbiased, while $\hat{\sigma}_p^p$ is biased for small samples. An unbiased estimator, when p is known and $\hat{\mu}$ is the maximum likelihood estimator of μ , is given by Chiodi:

$$\bar{\sigma}_p^p = \frac{\sum_{i=1}^n |x_i - \hat{\mu}|^p}{n - p/2}, \quad (9)$$

that provides also its asymptotic sampling distribution, i.e. the Gamma distribution (8) but with $c = n/p - 1/2$.

A paper where are showed some procedures to test hypothesis on homoskedasticity when samples are drawn from an exponential power distribution has been presented by Chiodi (1988a), as well.

Finally, Chiodi (1994) uses saddle point approximations to obtain asymptotic sampling distributions of the maximum likelihood estimators for the exponential power distribution parameters. The sampling distributions, obtained in this way, work well even for small samples, for example $n = 5$.

Moreover, we can compute the Fisher information matrix on μ and σ_p (Agrò 1995):

$$\mathbf{I} = \begin{bmatrix} I_{\mu\mu} & I_{\mu\sigma_p} \\ I_{\sigma_p\mu} & I_{\sigma_p\sigma_p} \end{bmatrix}$$

where

$$I_{\mu\mu} = -E \left[\frac{d^2 \ln f(x)}{d\mu^2} \right] = \frac{(p-1)p^{\frac{p-2}{p}} \Gamma\left(\frac{p-1}{p}\right)}{\sigma_p^2 \Gamma\left(\frac{1}{p}\right)} \quad (10)$$

$$I_{\mu\sigma_p} = I_{\sigma_p\mu} = -E \left[\frac{d^2 \ln f(x)}{d\mu d\sigma_p} \right] = 0 \quad (11)$$

$$I_{\sigma_p\sigma_p} = -E \left[\frac{d^2 \ln f(x)}{d\sigma_p^2} \right] = \frac{p}{\sigma_p^2} \quad (12)$$

Then, the parameters μ and σ_p are orthogonal, according to the Fisher's information matrix: for the estimation of the parameters, this fact makes possible the use of the conditional profile log likelihood that has better properties than the ordinary profile log likelihood (Cox and Reid 1987).

By inverting the information matrix, we obtain the asymptotic matrix of variance of the maximum likelihood estimators ($\hat{\mu}, \hat{\sigma}_p$):

$$\mathbf{I}^{-1} = \begin{bmatrix} \frac{\sigma_p^2 \Gamma(\frac{1}{p}) p^{(2-p)/p}}{(p-1) \Gamma(\frac{p-1}{p})} & 0 \\ 0 & \frac{\sigma_p^2}{p} \end{bmatrix}$$

that for $p=1$ and $p=2$ becomes, respectively, the asymptotic matrix of variance for the maximum likelihood estimators of the Laplace and the Gauss distribution parameters.

3.2. Estimation of the shape parameter p

The methods presented in literature are based on the likelihood function and on indices of kurtosis. In particular, a comparison among different approaches, i.e. the whole log-likelihood, the profile log-likelihood (Barndorff-Nielsen 1988) and the conditional profile log-likelihood (Cox and Reid 1987), proposed by Agrò (1999), and an index of kurtosis, proposed by A.M. Mineo (1994), has been made by A.M. Mineo (2003b). In this paper, according to a simulation study, it seems that the best method is the one based on the index of kurtosis, while generally the whole log-likelihood approach works better than those based on the profile log-likelihood and the conditional profile log-likelihood. Indeed, the method based on the index of kurtosis VI (see the formula 15) gives estimates of p that seem less biased and more efficient than the estimates obtained by using the three methods based on the log likelihood function. In some way, these results are expected, given the link between the shape parameter p and the kurtosis (cfr. 3.2.2).

Estimation of p by means of the maximum likelihood method

If we want to determine the maximum likelihood estimator of the shape parameter p , the equation that we obtain by deriving the log-likelihood function (3) is:

$$\begin{aligned} \frac{dl(x)}{dp} &= -\frac{n}{p^2} [\log p + \Psi(1 + 1/p) - 1] + \frac{1}{p^2 \sigma_p^p} \sum_{i=1}^n |x_i - \mu|^p + \\ &+ \frac{1}{p \sigma_p^p} \left[\log \sigma_p \sum_{i=1}^n |x_i - \mu|^p - \sum_{i=1}^n |x_i - \mu|^p \log |x_i - \mu| \right] = 0 \end{aligned} \quad (13)$$

where $\Psi(\cdot)$ is the digamma function, that is the first derivative of the logarithm of the gamma function.

The equation (13) can be solved by using numerical methods. Moreover, Agrò (1992) uses this method showing that it does not work well for small samples, even though it provides good results for samples of size greater than 50 – 100.

In a following paper, Agrò (1995) computes the inverse of the Fisher information matrix that defines the asymptotic matrix of variance of the maximum likelihood estimators $(\hat{\mu}, \hat{\sigma}_p, \hat{p})$:

$$\mathbf{I}^{-1} = \begin{bmatrix} \frac{\sigma_p^2 \Gamma(\frac{1}{p}) p^{(2-p)/p}}{(p-1)\Gamma(\frac{p-1}{p})} & 0 & 0 \\ 0 & \frac{\sigma_p^2}{p} \left\{ 1 + \frac{[\log p + \Psi(1+1/p)]^2}{(1+1/p)\Psi'(1+1/p)-1} \right\} & p\sigma_p \frac{[\log p + \Psi(1+1/p)]^2}{(1+1/p)\Psi'(1+1/p)-1} \\ 0 & p\sigma_p \frac{[\log p + \Psi(1+1/p)]^2}{(1+1/p)\Psi'(1+1/p)-1} & \frac{p^3}{(1+1/p)\Psi'(1+1/p)-1} \end{bmatrix}$$

where $\Psi'(\cdot)$ is the trigamma function, that is the second derivative of the logarithm of the gamma function. From this matrix we can see how the shape parameter p is orthogonal to the location parameter μ , but is not orthogonal to the scale parameter σ_p .

Capobianco (2000) observes how the asymptotic variance of the scale parameter estimator is larger than the asymptotic variance of the same estimator obtained for the Gaussian or the Laplace distribution. This is an expected result, since the Gaussian and the Laplace distribution are exponential power distributions when $p = 1$ and $p = 2$, respectively; if we substitute these values in the expression of the asymptotic variance, we obtain the same results. The apparent loss of efficiency is just due to the need of estimating p .

Estimation of p by means of indices of kurtosis

These estimation procedures take into account the relationship between the shape parameter p and the kurtosis. The usually used indices of kurtosis are:

$$\beta_2 = \frac{\mu_4}{\mu_2^2} = \frac{\Gamma(1/p)\Gamma(5/p)}{[\Gamma(3/p)]^2} \quad (14)$$

$$VI = \frac{\sqrt{\mu_2}}{\mu_1} = \frac{\sqrt{\Gamma(1/p)\Gamma(3/p)}}{\Gamma(2/p)} \quad (15)$$

$$I = \frac{1}{VI} \quad (16)$$

$$\beta_p = \frac{\mu_{2p}}{\mu_p^2} = p + 1 \quad (17)$$

where

$$\mu_r = \sigma_p^p p^{r/p} \frac{\Gamma[(r+1)/p]}{\Gamma(1/p)} \quad (18)$$

is the absolute moment of grade r . The index β_p , called *generalized index of kurtosis*, has been proposed by Lunetta (1963) and gives for $p = 2$ the Pearson's index of kurtosis β_2 .

The estimators of the indices of kurtosis above described are given by:

$$\hat{\beta}_2 = \frac{n \sum_{i=1}^n (x_i - M)^4}{[\sum_{i=1}^n (x_i - M)^2]^2} \quad (19)$$

$$\widehat{VI} = \frac{\sqrt{n \sum_{i=1}^n (x_i - M)^2}}{\sum_{i=1}^n |x_i - M|} \quad (20)$$

$$\hat{I} = \frac{1}{\widehat{VI}} \quad (21)$$

$$\hat{\beta}_p = \frac{n \sum_{i=1}^n |x_i - M|^{2\hat{p}}}{\left(\sum_{i=1}^n |x_i - M|^{\hat{p}}\right)^2} = \hat{p} + 1. \quad (22)$$

where M is the arithmetic mean. A first solution to this problem has been given by Lunetta (1967), which observed how the kurtosis indices could be used for estimating p .

Afterwards, A. Mineo (1980) proposed the solution of an equation based on a suitable combination of sampling indices of kurtosis. Again, A. Mineo (1986) shows by simulation studies the lower efficiency of some well-known robust estimators, compared to the maximum likelihood estimator of the location parameter μ . In the same paper, the author estimates p by using an inverse interpolation procedure. In a following paper, A. Mineo (1989), by implementing an iterative procedure, obtains estimators for a simple linear regression model with errors distributed as a normal distribution of order p and an estimate of p by solving the equation:

$$n \sum_{i=1}^n |x_i - M_p|^{2\hat{p}} - (\hat{p} + 1) \left(\sum_{i=1}^n |x_i - M_p|^{\hat{p}} \right)^2 = 0 \quad (23)$$

that derives from the theoretical value of the *generalized index of kurtosis* β_p . In this formula, M_p is an estimate of the location parameter μ , obtained by solving the equation (4): this is necessary to make possible the iterative process of estimation of all the three parameters μ , σ_p and p .

An estimation method based on the index of kurtosis VI has been proposed by A.M. Mineo (1994), by comparing the theoretical value and the empirical value computed on the sample. The index VI has been preferred to β_2 , because it is less affected by extreme values on data, in order to obtain better estimates, even when small samples ($n \leq 50$) are considered. Moreover, the equation (23), obtained from β_p , has not been considered since its expression is more difficult computationally. The validity of the proposed procedure has been provided by simulation studies. Actually, the obtained results seem to be very interesting and better than the ones obtained by methods based on the likelihood function (A.M. Mineo, 2003b).

4. The normalp package

The **normalp** package (A.M. Mineo, 2003a) is an R (R Development Core Team 2004) package containing a collection of tools related to the exponential power distribution. The implemented functions generalize some commands already available in the **base** package, related to observations distributed as a normal distribution. Moreover, useful commands deal with estimation problems for linear regression models when the random errors are drawn from an exponential power distribution. The **normalp** package is available on the *Comprehensive R Archive Network* (CRAN¹).

4.1. Computation of the exponential power distribution

The **normalp** package has four functions, `dnormp()`, `pnormp()`, `qnormp()` and `rnormp()`, to compute the typical quantities of a probability distribution, i.e.:

¹URL: <http://CRAN.R-project.org/>.

- `dnormp()`, that gives the density function of an exponential power distribution;
- `pnormp()`, that gives the probability function of an exponential power distribution;
- `qnormp()`, that gives the quantiles of an exponential power distribution;
- `rnormp()`, that generates a vector of n pseudo-random numbers from an exponential power distribution.

The use of these functions is the following:

```
dnormp(x, mu = 0, sigmap = 1, p = 2, log = FALSE)}
```

with *arguments*

`x` vector of quantiles;
`mu` vector of location parameters;
`sigmap` vector of scale parameters;
`p` shape parameter;
`log` logical; if TRUE the density is given as a *log(density)*.

```
pnormp(q, mu = 0, sigmap = 1, p = 2, lower.tail = TRUE, log.pr = FALSE)
```

with *arguments*

`q` vector of quantiles;
`mu` vector of location parameters;
`sigmap` vector of scale parameters;
`p` shape parameter;
`lower.tail` logical; if FALSE gives $P[X > x]$, otherwise (default) gives $P[X \leq x]$;
`log.pr` logical; if TRUE probabilities *pr* are given as *log(pr)*.

```
qnormp(pr, mu = 0, sigmap = 1, p = 2, lower.tail = TRUE, log.pr = FALSE)
```

with *arguments*

`pr` vector of probabilities;
`mu` vector of location parameters;
`sigmap` vector of scale parameters;
`p` shape parameter;
`lower.tail` logical; if FALSE gives $P[X > x]$, otherwise (default) gives $P[X \leq x]$;
`log.pr` logical; if TRUE probabilities *pr* are given as *log(pr)*.

```
rnormp(n, mu = 0, sigmap = 1, p = 2, method = "def")
```

with *arguments*

`n` vector of observations;
`mu` vector of location parameters;
`sigmap` vector of scale parameters;
`p` shape parameter;
`method` if set to `chiodi` it uses an algorithm based on a generalization of the Marsaglia (1964) formula, suggested by Chiodi (1986), otherwise it uses a faster method based on the relationship linking an exponential power distribution and a Gamma distribution (Lunetta 1963).

4.2. Graphical functions

Besides the above described functions, some useful graphical functions have been implemented. In particular, these functions are:

- `graphnp()`, that visualizes from one to five different exponential power distributions, each one plotted with a different color;
- `qqnormp()`, that gives an exponential power distribution Q-Q plot;
- `qqlinep()`, that sketches a line through the first and the third quartile in an exponential power distribution Q-Q plot;

Then, the functions `qqnormp()` and `qqlinep()` allow to check graphically if a vector of observations can be considered drawn from an exponential power distribution.

The use of these functions is, respectively:

```
graphnp(p = c(1, 2, 3, 4, 5), mu = 0, sigmap = 1,
        title="Exponential power distributions")
```

with *arguments*

`p` vector of p values. It has to contain at most five elements. If one or more values of p are ≥ 50 , the plot will give an uniform distribution;

`mu` value of the location parameter;

`sigmap` value of the scale parameter;

`title` the plot title.

```
qqnormp(y, ylim, p = 2, main = "Exponential power distribution Q-Q plot",
        xlab = "Theoretical Quantiles",
        ylab = "Sample Quantiles", ...)
qqlinep(y, p = 2, ...)
```

with *arguments*

`y` vector of observations;

`p` shape parameter;

`main, xlab, ylab` plot labels;

`ylim, ...` graphical parameters.

4.3. Functions to estimate the exponential power distribution parameters

Another set of functions is related to the estimation of the parameters of an exponential power distribution. In particular, we have the following functions:

- `estimatep()`, that estimates the shape parameter p from a vector of observations by means of the index of kurtosis VI (A.M. Mineo, 1994);
- `paramp()`, that estimates the location parameter μ and the scale parameter σ_p by means of the maximum likelihood method, by considering the two cases when p is known

and when p is unknown; in the latter case, an estimate of p is obtained from the `estimatep()` function; this function returns a list of elements: the arithmetic mean, the standard deviation, the estimated values of location, scale and shape parameters of an exponential power distribution. Moreover, if the message `iter=1` prints out, then problems on convergence have arisen in the computation;

- `kurtosis()`, that computes the theoretical and the empirical values of the kurtosis indices VI , β_2 and β_p ;
- `simul.mp()`, that performs a Monte Carlo simulation to verify the behavior of the estimators of the parameters μ , σ_p and p , behavior that can be seen graphically by using the function `plot.simul.mp()`, that returns an histogram for each set of the estimates produced by `simul.mp()`. The function `simul.mp()` returns a *list* containing the following components
 - `dat`: a matrix $m \times 5$ reporting the result of `paramp()` for each sample;
 - `table`: a matrix 2×5 of the means and the variances of the five estimator values.

The use of these functions is described below:

```
estimatep(x, mu = 0, p = 2, method = c("inverse", "direct"))
```

with *arguments*

`x` vector of observations;
`mu` the location parameter;
`p` starting value of the shape parameter;
`method` the user can choose an inverse interpolation (faster) or a
 direct solution of the equation $VI = \widehat{VI}$.

```
paramp(x, p = NULL, ...)
```

with *arguments*

`x` vector of observations;
`p` if specified, the algorithm does not use an estimate of p , but
 the specified value;
`...` other arguments passed to or from other methods.

```
kurtosis(x, p = 3, value = "parameter")
```

with *arguments*

`x` a sample of observations; if it is not specified, theoretical indices are computed;
`p` the shape parameter;
`value` if set to parameter, the value specified in `p` is used, otherwise
 an estimate of p is computed and estimates of the three indices are given.

```
simul.mp(n, m, mu = 0, sigmap = 1, p = 2)
```

with *arguments*

`n` sample size;
`m` number of samples;
`mu` the location parameter;
`sigmap` the scale parameter;
`p` the shape parameter.

`plot.simul.mp(x, ...)`

with *arguments*

`x` a `simul.mp` object, i.e. a result of `simul.mp()`;
`...` further arguments passed to or from other methods.

4.4. Functions to deal with linear regression problems

The **normalp** package contains also a set of functions to fit linear regression models, when random errors are distributed according to an exponential power distribution. The implemented functions are:

- `lmp()` that evaluates the coefficients of the linear regression model by means of the maximum likelihood method, either when p is known, or when p is unknown (A.M. Mineo, 1995); in the latter case an estimate of p is obtained from `estimatep()`. The `lmp()` function returns a list containing the following components:
 - `coefficients`: the coefficient estimates;
 - `residuals`: the residuals;
 - `fitted.values`: the fitted values;
 - `rank`: the numeric rank of the fitted linear model;
 - `df.residual`: the residual degrees of freedom computed as in `lm()`;
 - `call`: the matched call;
 - `terms`: the terms object used;
 - `p`: the p estimate computed on residuals;
 - `knp`: a logical parameter used by `summary()`;
 - `model`: the model frame used;
 - `iter`: its value is 1 if difficulties on convergence arise.
- `plot.lmp()` that performs an analysis of residuals by returning a set of plots:
 - a plot of residuals against fitted values;
 - a Normal Q-Q plot;
 - an Exponential Power Distribution Q-Q plot;
 - a Scale-Location plot.

- `summary.lmp()` that produces a list of summary statistics related to the fitted linear model given by `lmp()`; among them, *the power deviation of order p* computed on residuals:

$$S_p = \left[\frac{\sum e_i^p}{n - q} \right]^{\frac{1}{p}} \quad (24)$$

where q is the number of the estimated regression coefficients if p is known, otherwise is the number of the estimated regression coefficients plus 1.

- `simul.lmp()` that allows to verify the behavior of the regression parameter estimators and the estimators of σ_p and p by means of a Monte Carlo simulation study. This function returns a *list* containing a matrix reporting the result of `lmp()` for all the samples and a matrix of means and variances of the m estimates for each related parameter. The regressors are drawn from an Uniform distribution.
- `summary.simul.lmp()` that summarizes the results of `simul.lmp`
- `plot.simul.lmp()` returns a histogram for each set of the estimates computed by `simul.lmp()`.

The use of these functions is the following (it is very similar to that of `lm()` and related functions of the **stats** package):

```
lmp(formula, data = list(), p = NULL)
```

with *arguments*

`formula` the model to be fitted;
`data` an optional data frame containing the variables in the model.
 By default the variables are taken from the environment;
`p` the shape parameter. It is estimated if is not specified.

```
plot.lmp(x, ...)
```

with *arguments*

`x` is a `lmp` object, i.e. a result of `lmp()`;
`...` are further arguments passed to or from other methods.

```
summary.lmp(object, ...)
```

with *arguments*

`object` a `lmp` object, i.e. a result of `lmp()`;
`...` further arguments passed to or from other methods.

```
simul.lmp(n, m, q, data, int = 0, sigmap = 1, p = 2,  
          lp = FALSE)
```

with *arguments*

n sample size;
m number of samples;
q number of regressors;
data a vector of coefficients;
int the intercept value;
sigmap the scale parameter;
p the shape parameter;
lp if FALSE p is estimated, otherwise the specified value of p is considered.

`summary.simul.lmp(x, ...)`

with *arguments*

x a `simul.lmp` object, i.e. a result of `simul.lmp()`;
 ... further arguments passed to or from other methods.

`plot.simul.lmp(x, ...)`

with *arguments*

x a `simul.lmp` object, i.e. a result of `simul.lmp()`;
 ... further arguments passed to or from other methods.

5. Some examples of use

In this section different examples of use of the **normalp** package will be showed. All the examples have only the purpose to show some possible ways of use of the implemented functions.

5.1. Dealing with an exponential power distribution

Let's compute the density for a x value with $\mu = 0$, $\text{sigmap} = 1$ and $p = 1.5$:

```
R> dnormp(2, p = 1.5)
```

```
[1] 0.06413529
```

Let's compute the distribution function for a q value with $\mu = 1$, $\text{sigmap} = 2$ and $p = 1.5$:

```
R> pnormp(0.7, mu = 1, sigmap = 2, p = 1.5)
```

```
[1] 0.4375686
```

Let's compute the quantile for a probability value $pr = 0.3$ with $\mu = 3$, $\text{sigmap} = 2$ and $p = 1.5$:

```
R> qnormp(0.3, mu = 3, sigmap = 2, p = 1.5)
```

```
[1] 1.956552
```

In the following example, we generate a random sample of size $n = 10$ from an exponential power distribution with $\mu = 2$, $\sigma = 3$ and $p = 1.5$:

```
R> rnormp(10, 2, 3, 1.5)
```

```
[1] 4.0685043 3.3570330 3.6317871 -0.9743280 3.3015496 1.2702016
[7] 2.6252958 9.5512745 3.1040467 0.8221423
```

With the following command, we plot five different distributions with $p=1, 2, 3, 4, 50$ (the last one will be the uniform distribution):

```
R> graphnp(c(1 : 4, 50))
```

(see Figure 1)

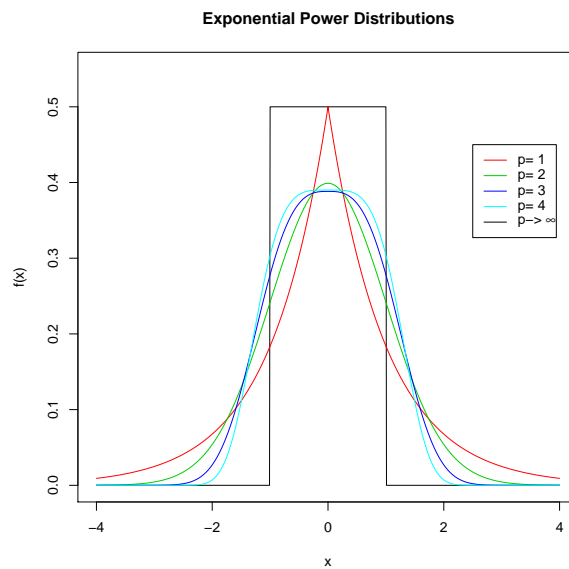


Figure 1: Exponential power distributions.

and an exponential power distribution Q-Q plot for a sample of size $n = 100$:

```
R> x <- rnormp(100, p = 3)
```

```
R> qqnormp(x, p = 3)
```

```
R> qqlinep(x, p = 3)
```

(see Figure 2)

5.2. Estimating the exponential power distribution parameters

Let's estimate the shape parameter p from a vector of observations:

```
R> x <- rnormp(100, 1, 2, 4)
```

```
R> p <- estimatep(x, 1, 2)
```

```
R> p
```

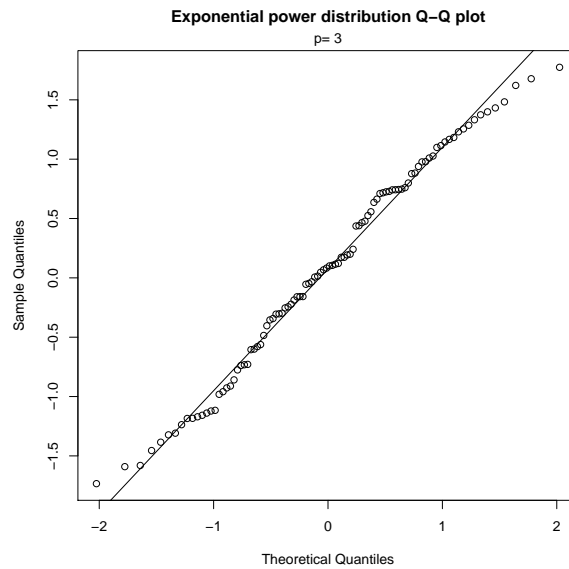


Figure 2: Exponential power distribution Q-Q plot for a sample of size $n = 100$.

```
[1] 3.255839
```

Let's estimate μ , σ_p and p from a vector of observations:

```
R> x <- rnormp(100, 1, 2, 3)
R> parameters <- paramp(x)
R> parameters
```

Mean	Mp	Sd	Sp	p
0.8752648	1.1033518	1.8194269	2.3706584	4.5758560

Let's compute the values of theoretical and empirical indices of kurtosis, with p known or estimated:

```
R> kurtosis(p = 2)
```

VI	B2	Bp
1.253314	3.000000	3.000000

```
R> kurtosis(x, p = 2, value = "parameter")
```

VI	B2	Bp
1.199764	2.513143	2.513143

```
R> kurtosis(x)
```

VI	B2	Bp
1.200588	2.501471	4.019306

Let's perform a simulation plan for $m = 100$ samples of size $n = 30$, with $\mu = 0$, $\sigma = 1$, $p = 3$, by showing the histogram for each set of estimates:

```
R> sim <- simul.mp(30, 100, p = 3)
R> sim
```

	Mean	Mp	Sd	Sp	p
Mean	-0.02374401	-0.02288745	0.84925812	0.93616566	3.264425
Variance	0.02009107	0.02643337	0.01095461	0.05394765	8.107726
Number of samples with a difficult convergence: 1					

```
R> par(mfrow = c(3, 2))
R> plot(sim)
R> par(mfrow = c(1, 1))
```

(see Figure 3)

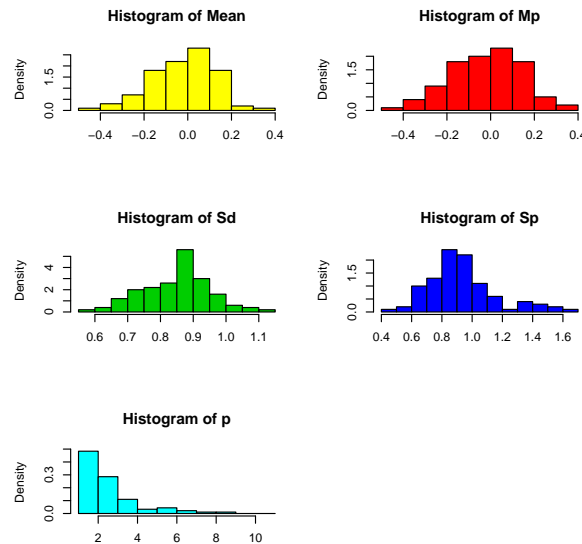


Figure 3: Histograms of the estimates obtained by `simul.mp()`.

5.3. Fitting a linear regression model with random errors distributed according to an exponential power distribution

In this subsection we consider an example that has only the purpose to show the use of the implemented functions to fit a linear regression model when we have to assume that the random errors are distributed according to an exponential power distribution. Let's consider the following data set taken from [Levine, Krehbiel, and Berenson \(2000\)](#); this data set reports the profits at the box office in million of dollars (**Gross**) and the number of sold home videos in thousands (**Videos**) for a sample of 30 movies; the idea is that the number of sold home videos is related to the profits at the box office:


```
R> dataset <- read.table("movie.txt", header = TRUE)
R> dataset
```

```
      Gross Videos
1    1.10  57.18
2    1.13  26.17
3    1.18  92.79
4    1.25  61.60
5    1.44  46.50
6    1.53  85.06
7    1.53 103.52
8    1.69  30.88
9    1.74  49.29
10   1.77  24.14
11   2.42 115.31
12   5.34  87.04
13   5.70 128.45
14   6.43 126.64
15   8.59 107.28
16   9.36 190.80
17   9.89 121.57
18  12.66 183.30
19  15.35 204.72
20  17.55 112.47
21  17.91 162.95
22  18.25 109.20
23  23.13 280.79
24  27.62 229.51
25  37.09 277.68
26  40.73 226.73
27  45.55 365.14
28  46.62 218.64
29  54.70 286.31
30  58.51 254.58
```

```
R> attach(dataset)
R> res.lmp <- lmp(Videos ~ Gross)
R> res.lmp
```

```
Call:
lmp(formula = Videos ~ Gross)
```

```
Coefficients:
(Intercept)      Gross
      77.411      4.368
```

Let's report a summary of the main results with a plot showing the two straight lines derived by using the OLS methods and the `lmp()` method:

```
R> summary(res.lmp)
```

Call:

```
lmp(formula = Videos ~ Gross)
```

Residuals:

Min	1Q	Median	3Q	Max
-78.43	-36.83	-3.35	27.03	102.34

Coefficients:

(Intercept)	Gross
77.411	4.368

Estimate of p

2.388677

Power deviation of order p: 50.95

```
R> plot(Videos ~ Gross)
```

```
R> abline(lm(Videos ~ Gross))
```

```
R> abline(res.lmp, col = "red")
```

(see Figure 4)

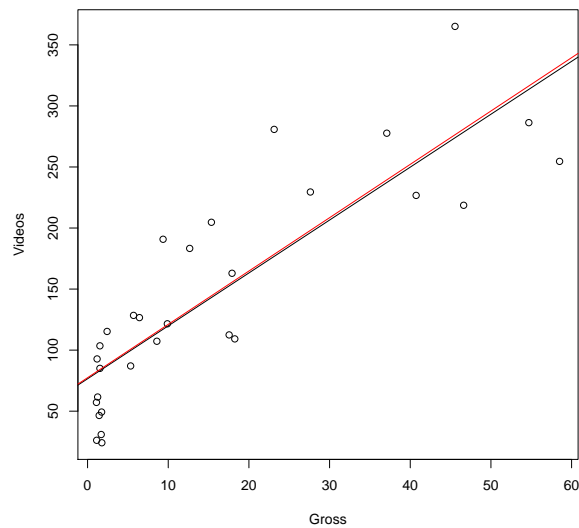


Figure 4: Plot of the linear regression models.

Let's perform an analysis of residuals:

```
R> par(mfrow = c(2, 2))
R> plot(res.lmp)
```

(see Figure 5)

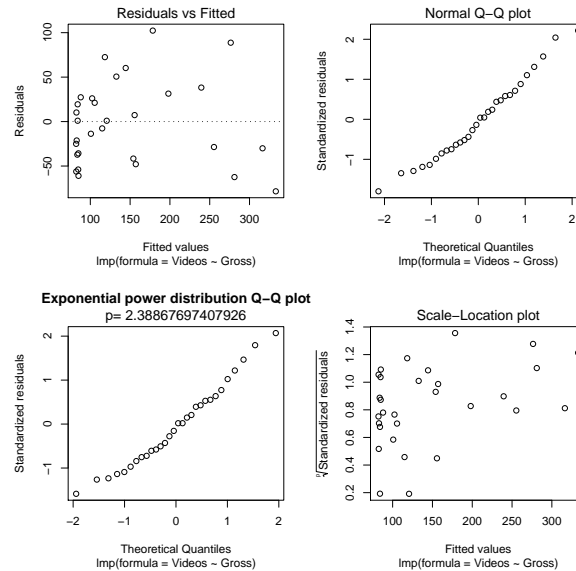


Figure 5: Residuals plots obtained by the function `plot.lmp()`.

Let's perform a simulation plan for $m = 100$ samples of size $n = 20$ for a linear regression model with only one regressor, by showing the histogram for each set of estimates:

```
R> sim <- simul.lmp(20, 100, 1, 1.5, 1, 1, 3)
R> sim
```

```
      (intercept)      x1      Sp      p
Mean      0.9734609 1.527420 0.92207680  3.735001
Variance  0.1947046 0.715167 0.03944582 10.998811
Number of samples with a difficult convergence: 1
```

```
R> summary(sim)
```

Results:

```
      (intercept)      x1      Sp      p
Mean      0.9734609 1.527420 0.92207680  3.735001
Variance  0.1947046 0.715167 0.03944582 10.998811
Coefficients:
(intercept)      x1
      1.0      1.5
```

Formula:

$y \sim x_1$

Number of samples:

100

Value of p:

3

Number of samples with problems on convergence

1

```
R> par(mfrow = c(2, 2))
```

```
R> plot(sim)
```

(see Figure 6)

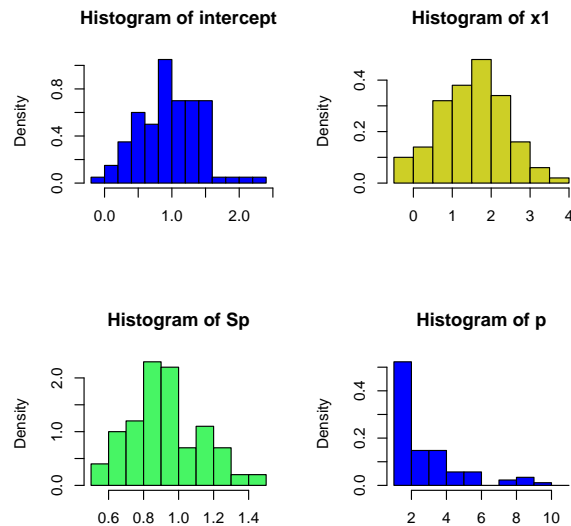


Figure 6: Histograms of the estimates produced by `simul.lmp()`.

6. Conclusions

In this paper, we have described some results obtained on the Exponential Power Distribution and implemented in a package, **normalp**, that is a contributed package of the statistical environment R. The implemented functions concern essentially estimation problems for linear regression models, besides some graphical commands that generalize graphical tools already implemented in the package **graphs** of R, related to observations distributed as a normal distribution. At the end of the paper, several examples of use have been showed. People that are used to working with R can find very easy the syntax of the functions implemented in this package. We think that this package could be very useful for anyone has to treat with this

kind of probability distribution. Our intention is to work more on this package to improve its flexibility.

Acknowledgments

The authors are grateful to two anonymous referees for thorough and constructive comments which greatly improved this paper.

References

- Achcar De Araujo Pereira JA (1999). “Use of Exponential Power Distributions for Mixture Models in the Presence of Covariates.” *Journal of Applied Statistics*, **26**, 669–679.
- Agrò G (1992). “Maximum Likelihood and L_p -norm Estimators.” *Statistica Applicata*, **4**(2), 171–182.
- Agrò G (1995). “Maximum Likelihood Estimation for the Exponential Power Distribution.” *Communications in Statistics (simulation and computation)*, **24**(2), 523–536.
- Agrò G (1999). “Parameter Orthogonality and Conditional Profile Likelihood: The Exponential Power Function case.” *Communications in Statistics (theory and methods)*, **28**(8), 1759–1768.
- Barndorff-Nielsen OE (1988). *Parametric Statistical Models and Likelihood*. Lecture Notes in Statistics, No. 50, Springer-Verlag.
- Box GEP, Muller ME (1958). “A Note on the Generalization of Random Normal Deviates.” *Annals of Mathematical Statistics*, **29**, 610–611.
- Box GEP, Tiao GC (1992). *Bayesian Inference in Statistical Analysis*. J. Wiley, New York.
- Capobianco R (2000). “Robustness Aspects of the Generalized Normal Distribution.” *Quaderni di Statistica, Centro per la formazione in Economia e Politica dello sviluppo rurale, Dipartimento di Scienze Statistiche dell’Università di Napoli “Federico II”, Dipartimento di Scienze Economiche dell’Università di Salerno*, **2**, 127–145.
- Chiodi M (1986). “Procedures for Generating Pseudo-Random Numbers from a Normal Distribution of Order p .” *Statistica Applicata*, **19**, 7–26.
- Chiodi M (1988a). “Due Tests per la Verifica di Ipotesi di Omogeneità ed Omoscedasticità per Campioni Provenienti da Distribuzioni Normali di Ordine p .” *Technical report*, Istituto di Statistica, Facoltà di Economia e Commercio di Palermo.
- Chiodi M (1988b). “Sulle Distribuzioni di Campionamento delle Stime di Massima Verosimiglianza dei Parametri delle Curve Normali di Ordine p .” *Technical report*, Istituto di Statistica, Facoltà di Economia e Commercio di Palermo.
- Chiodi M (1994). “Approssimazioni Saddle-point alle Distribuzioni Campionarie degli Stimatori di Massima Verosimiglianza dei Parametri delle Curve Normali di Ordine p per Piccoli Campioni.” In “Atti della XXXVII Riunione Scientifica della SIS,” pp. 139–146. San Remo.

- Chiodi M (1995). "Generation of Pseudo-Random Variates from a Normal Distribution of Order p ." *Statistica Applicata*, **7**, 401–416.
- Chiodi M (2000). "Le Curve Normali di Ordine p come Distribuzioni di Errori Accidentali: una Rassegna dei Risultati e Problemi Aperti per il Caso Univariato e per quello Multivariato." In "Atti della XL Riunione Scientifica della SIS," pp. 289–300. Firenze.
- Choy STB, Smith AFM (1997). "On Robust Analysis of a Normal Location Parameter." *Journal of the Royal Statistical Society, B*, **59**, 463–474.
- Cox DR, Reid N (1987). "Parameter Orthogonality and Approximate Conditional Inference." *Journal of the Royal Statistical Society (B)*, **49**, 1–39.
- De Simone S (1968). "Su una Estensione dello Schema delle Curve Normali di Ordine r alle Variabili Doppie." *Statistica*, **37**, 447–474.
- Fang KT, Kotz S, Ng KW (1989). *Symmetric Multivariate and Related Distributions*. Chapman and Hall, New York.
- Johnson ME (1979). "Computer Generation of the Exponential Power Distributions." *Journal of Statistical Computation and Simulation*, **9**, 239–240.
- Krzanowski WJ, Marriott FHC (1994). *Multivariate Analysis (Part I)*. Edward Arnold, London.
- Levine DM, Krehbiel TC, Berenson ML (2000). *Business Statistics: A First Course*. Prentice-Hall.
- Lunetta G (1963). "Di una Generalizzazione dello Schema della Curva Normale." *Annali della Facoltà di Economia e Commercio di Palermo*, **17**, 237–244.
- Lunetta G (1966). "Di Alcune Distribuzioni Deducibili da una Generalizzazione dello Schema della Curva Normale." *Annali della Facoltà di Economia e Commercio di Palermo*, **20**, 119–143.
- Lunetta G (1967). "Su una Estensione del Coefficiente di Correlazione Lineare." *Annali della Facoltà di Economia e Commercio di Palermo*, **21**, 187–201.
- Marsaglia G, MacLaren MD, Bray TA (1964). "A Fast Procedure for Generating Normal Random Variables." *Communications of the ACM*, **7**, 4–10.
- Mineo A (1980). "La Stima dei Parametri delle Curve Normali di Ordine r per Intervalli Finiti e di quelle per Intervalli Infiniti." In CLEUB (ed.), "Studi in Onore di Paolo Fortunati," pp. 565–577. Bologna.
- Mineo A (1986). "La Migliore Combinazione delle Osservazioni per la Stima dei Parametri di Intensità e di Dispersione." In "Studi in Onore di Silvio Vianelli," pp. 517–549. Palermo.
- Mineo A (1989). "The Norm- p Estimation of Location, Scale and Simple Linear Regression Parameters." In Springer-Verlag (ed.), "Lectures Notes in Statistics (57)," pp. 222–233. Heidelberg.

- Mineo AM (1994). “Un Nuovo Metodo di Stima di p per una Corretta Valutazione dei Parametri di Intensità e di Scala di una Curva Normale di Ordine p .” In “Atti della XXXVII Riunione Scientifica della SIS,” pp. 147–154. San Remo.
- Mineo AM (1995). “Stima dei Parametri di Regressione Lineare Semplice quando gli Errori Seguono una Distribuzione Normale di Ordine p (p incognito).” *Annali della Facoltà di Economia di Palermo (area statistico-matematica)*, **49**, 161–186.
- Mineo AM (2003a). *The normalp package*. URL <http://CRAN.R-project.org/src/contrib/PACKAGES.html#normalp>.
- Mineo AM (2003b). “On the Estimation of the Structure Parameter of a Normal Distribution of Order p .” *Statistica*, **63**, 109–122.
- Pearson K (1895). “Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material.” *Philosophical Transactions of the Royal Society of London (A)*, **186**, 343–414.
- R Development Core Team (2004). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org/>.
- Subbotin MT (1923). “On the Law of Frequency of Errors.” *Matematicheskii Sbornik*, **31**, 296–301.
- Tadikamalla PR (1980). “Random Sampling from the Exponential Power Distribution.” *Journal of the American Statistical Association*, **75**, 683–686.
- Taguchi T (1978). “On a Generalization of Gaussian Distribution.” *Annals of the Institute of Statistical Mathematics*, **30**, 211–242.
- Vianelli S (1963). “La Misura della Variabilità Condizionata in uno Schema Generale delle Curve Normali di Frequenza.” *Statistica*, **23**, 447–474.
- Vitrano S, Baragona R (2004). “The Genetic Algorithm Estimates for the Parameters of Order p Normal Distributions.” In Springer-Verlag (ed.), “Advances in Multivariate Data Analysis,” pp. 133–143. Heidelberg.
- Zeckhauser R, Thompson M (1970). “Linear Regression with Non-Normal Error Terms.” *The Review of Economics and Statistics*, **52**, 280–286.

Affiliation:

Angelo M. Mineo

Dipartimento di Scienze Statistiche e Matematiche "Silvio Vianelli"

Università di Palermo

90128 Palermo, Italy

Telephone: +39/091/6626-239

Fax: +39/091/485726

E-mail: elio.mineo@dssm.unipa.it

URL: <http://dssm.unipa.it/elio/>