FITTING DISTRIBUTIONS WITH R

Release 0.4-21 February 2005

Vito Ricci
vito_ricci@yahoo.com

TABLE OF CONTENTS

1.0 Introduction

Fitting distributions consists in finding a mathematical function which represents in a good way a statistical variable. A statistician often is facing with this problem: he has some observations of a quantitative character x_1, x_2,… x_n and he wishes to test if those observations, being a sample of an unknown population, belong from a population with a pdf (probability density function) f(x,**θ**), where **θ** is a vector of parameters to estimate with available data.
We can identify 4 steps in fitting distributions:

1) Model/function choice: hypothesize families of distributions;
2) Estimate parameters;
3) Evaluate quality of fit;
4) Goodness of fit statistical tests.

This paper aims to face fitting distributions dealing shortly with theoretical issues and practical ones using the statistical environment and language R[1].
R is a language and an environment for statistical computing and graphics flexible and powerful. We are going to use some R statements concerning graphical techniques (§ 2.0), model/function choice (§ 3.0), parameters estimate (§ 4.0), measures of goodness of fit (§ 5.0) and most common goodness of fit tests (§ 6.0).
To understand this work a basic knowledge of R is needed. We suggest a reading of "*An introduction to R*"[2].
R statements, if not specified, are included in `stats` package.

2.0 Graphics

Exploratory data analysis can be the first step, getting descriptive statistics (mean, standard deviation, skewness, kurtosis, etc.) and using graphical techniques (histograms, density estimate, ECDF) which can suggest the kind of pdf to use to fit the model.
We can obtain samples from some pdf (such as gaussian, Poisson, Weibull, gamma, etc.) using R statements and after we draw a histogram of these data. Suppose we have a sample of size n=100 belonging from a normal population N(10,2) with mean=10 and standard deviation=2:

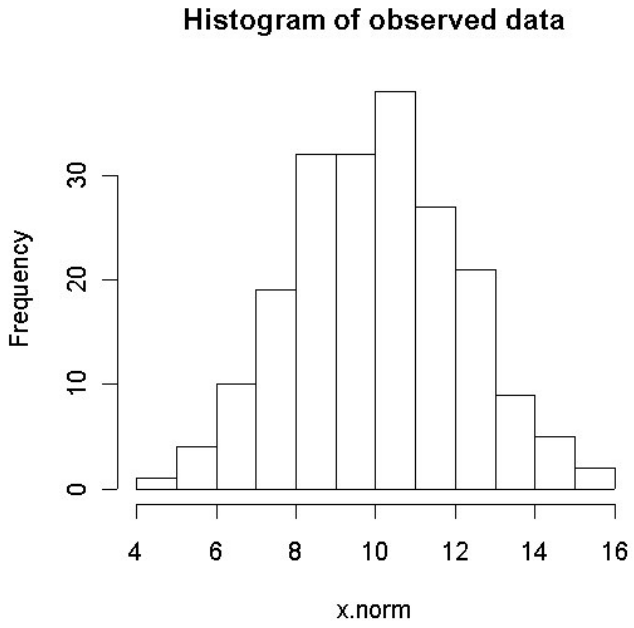
```
x.norm<-rnorm(n=200,m=10,sd=2)
```

We can get a histogram using `hist()` statement (Fig. 1):

```
hist(x.norm,main="Histogram of observed data")
```

[1] 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.
[2] R Development Core Team, An introdution to R, release 2.0.1, November 2004

[Fig. 1]



Histograms can provide insights on skewness, behavior in the tails, presence of multi-modal behavior, and data outliers; histograms can be compared to the fundamental shapes associated with standard analytic distributions.
We can estimate frequency density using `density()` and `plot()` to plot the graphic (Fig. 2):

```
plot(density(x.norm),main="Density estimate of data")
```

R allows to compute the empirical cumulative distribution function by `ecdf()` (Fig. 3):

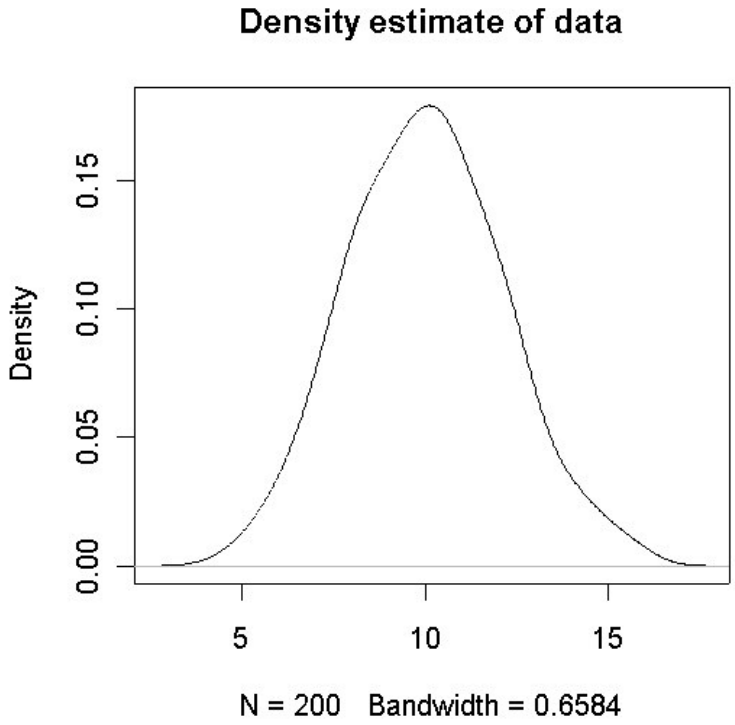
```
plot(ecdf(x.norm),main=" Empirical cumulative distribution function")
```

A Quantile-Quantile (Q-Q) plot[3] is a scatter plot comparing the fitted and empirical distributions in terms of the dimensional values of the variable (i.e., empirical quantiles). It is a graphical technique for determining if a data set come from a known population. In this plot on the y-axis we have empirical quantiles[4] e on the x-axis we have the ones got by the theorical model.
R offers to statements: `qqnorm()`, to test the goodness of fit of a gaussian distribution, or `qqplot()` for any kind of distribution. In our example we have (Fig. 4):

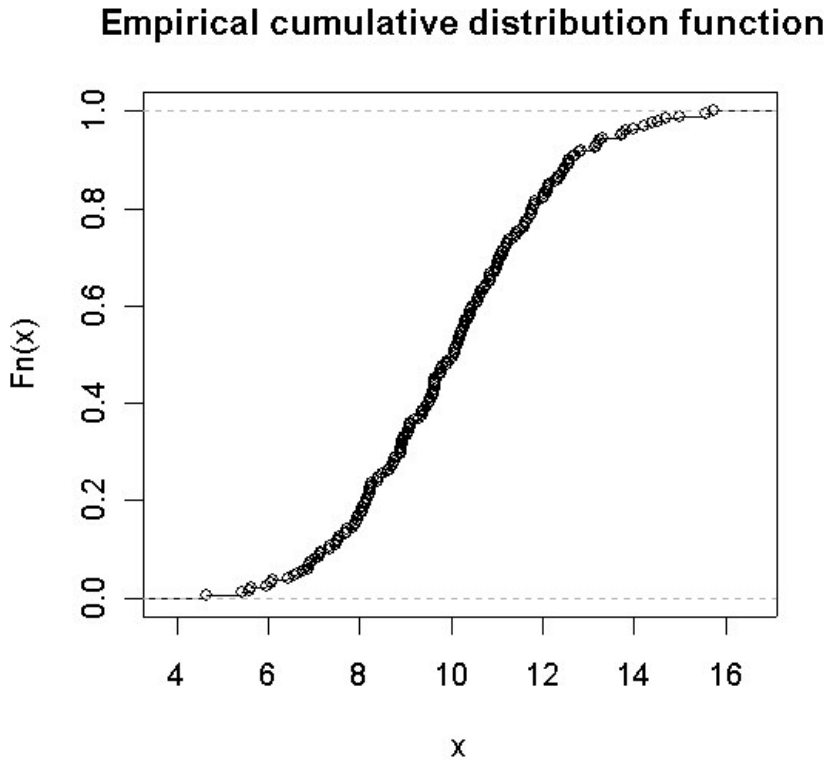
```
z.norm<-(x.norm-mean(x.norm))/sd(x.norm) ## standardized data
qqnorm(z.norm) ## drawing the QQplot
abline(0,1) ## drawing a 45-degree reference line
```

[3] See http://www.itl.nist.gov/div898/handbook/eda/section3/qqplot.htm [2005-01-11]
[4] By a quantile, we mean the fraction (or percent) of points below the given value. That is, the 0.3 (or 30%) quantile is the point at which 30% percent of the data fall below and 70% fall above that value.

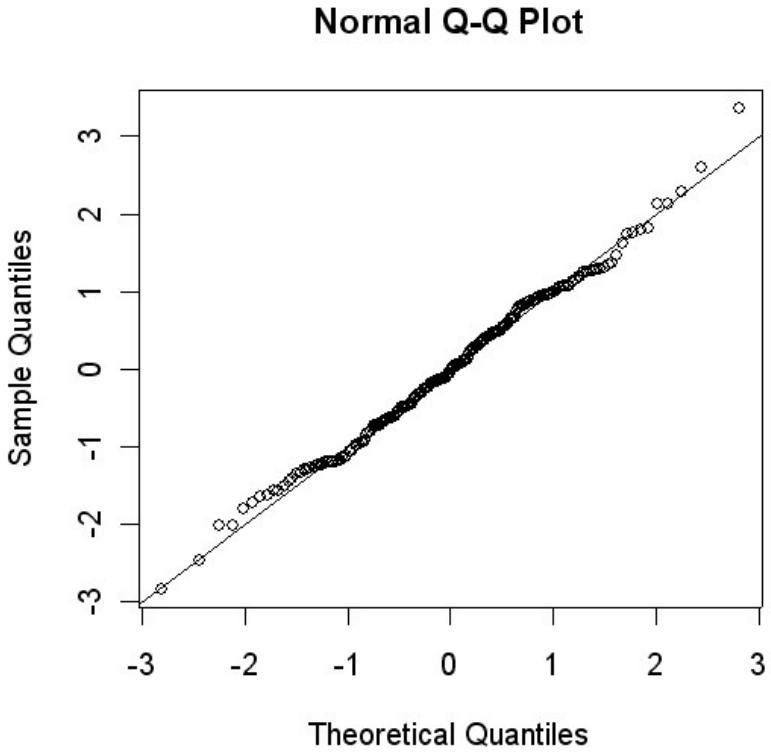
[Fig. 2]


Density estimate of data
Density
0.00
0.05
0.10
0.15
5
10
15
N = 200 Bandwidth = 0.6584

[Fig. 3]


Empirical cumulative distribution function
Fn(x)
0.0
0.2
0.4
0.6
0.8
1.0
4
6
8
10
12
14
16
x

[Fig. 4]

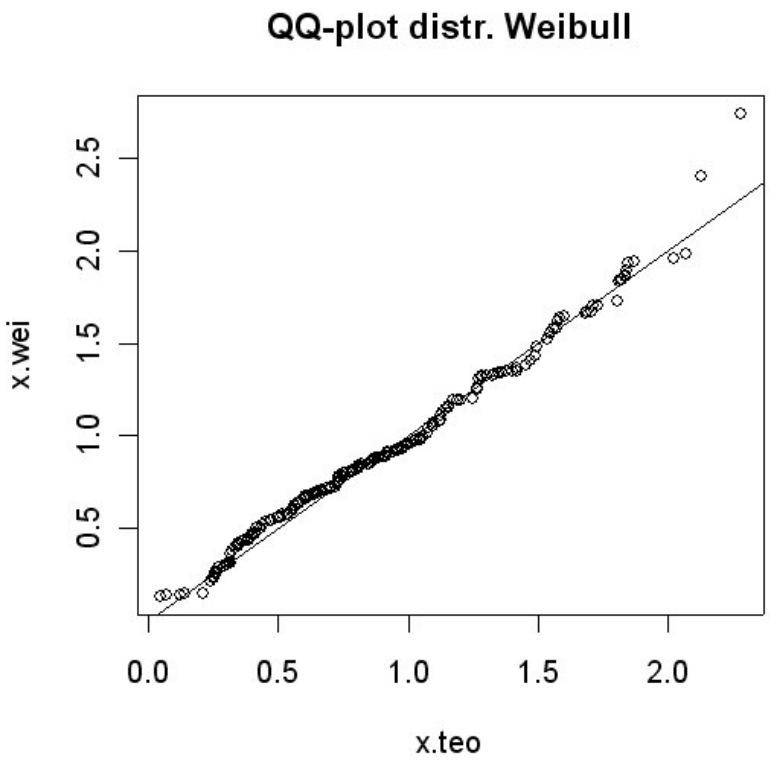


A 45-degree reference line is also plotted. If the empirical data come from the population with the choosen distribution, the points should fall approximately along this reference line. The greater the departure from this reference line, the greater the evidence for the conclusion that the data set have come from a population with a different distribution.

If data differ from a normal distribution (i.e. data belonging from a Weibull pdf) we can use `qqplot()` in this way (Fig. 5):

```
x.wei<-rweibull(n=200,shape=2.1,scale=1.1)  ## sampling from a Weibull
distribution with parameters shape=2.1 and scale=1.1
x.teo<-rweibull(n=200,shape=2,  scale=1)  ## theorical quantiles from a
Weibull population with known paramters shape=2 e scale=1
qqplot(x.teo,x.wei,main="QQ-plot distr. Weibull") ## QQ-plot
abline(0,1) ## a 45-degree reference line is plotted
```

[Fig. 5]



where `x.wei` is the vector of empirical data, while `x.teo` are quantiles from theorical model.

3.0 Model choice

The first step in fitting distributions consists in choosing the mathematical model or function to represent data in the better way.
Sometimes the type of model or function can be argued by some hypothesis concerning the nature of data, often histograms and other graphical techniques can help in this step (see § 2.0), but graphics could be quite subjective, so there are methods based on analytical expressions such us the Pearson's K criterion. Solving a particular differential equation we can obtain several families of function able to represent quite all empirical distributions. Those curves depend only by mean, variability, skewness and kurtosis. Standardizing data, the type of curve depends only by skewness and kurtosis[5] measures as shown in this formula:

$$K = \frac{\gamma_1^2(\gamma_2 + 6)^2}{4(4\gamma_2 - 3\gamma_1^2 + 12)(2\gamma_2 - 3\gamma_1^2)}$$

where:

$\gamma_1 = \frac{\sum_{i=1}^{n}(x_i - \mu)^3}{n\sigma^3}$ is Pearson's skewness coefficient

[5] http://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm [2005-02-04]

$\gamma_2 = \frac{\sum_{i=1}^{n}(x_i - \mu)^4}{n\sigma^4} - 3$ is Pearson's kurtosis coefficient.

According to the value of K, obtained by available data, we have a particular kind of function.
Here are some examples of continuous and discrete distributions[6], they will be used afterwards in this paper.
For each distribution there is the graphic shape and R statements to get graphics.
Dealing with discrete data we can refer to Poisson's distribution[7] (Fig. 6) with probability mass function:

$f(x,\lambda) = e^{-\lambda}\frac{\lambda^x}{x!}$ where x=0,1,2,…

```
x.poi<-rpois(n=200,lambda=2.5)
hist(x.poi,main="Poisson distribution")
```

As concern continuous data we have:

normal (gaussian) distribution[8] (Fig. 7): $f(x,\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$ with $x \in R$

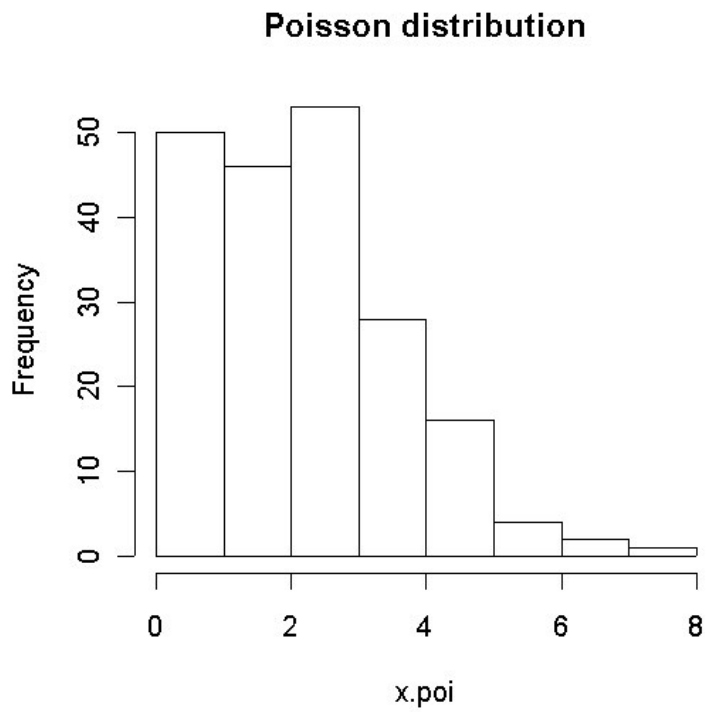
```
curve(dnorm(x,m=10,sd=2),from=0,to=20,main="Normal distribution")
```

gamma distribution[9] (Fig. 8): $f(x,\alpha,\lambda) = \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x}$ with $x \in R^+$

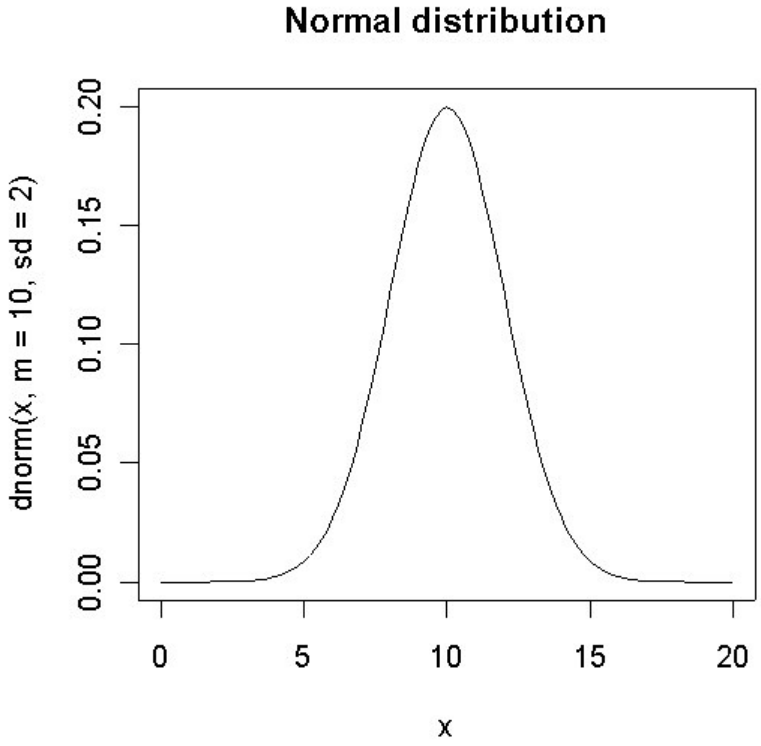
```
curve(dgamma(x, scale=1.5, shape=2),from=0, to=15, main="Gamma
distribution")
```

[6] See these websites for an overview on several kinds of distributions existing in statistical literature: http://www.xycoon.com/continuousdistributions.htm,,http://www.itl.nist.gov/div898/handbook/eda/section3/eda366.htm and http://www.statsoft.com/textbook/stdisfit.html [2005-01-11]
[7] See: http://www.itl.nist.gov/div898/handbook/eda/section3/eda366j.htm [2005-02-04]
[8] See: http://www.xycoon.com/normal.htm [2005-01-12]
[9] See: http://www.xycoon.com/gamma.htm [2005-01-11]

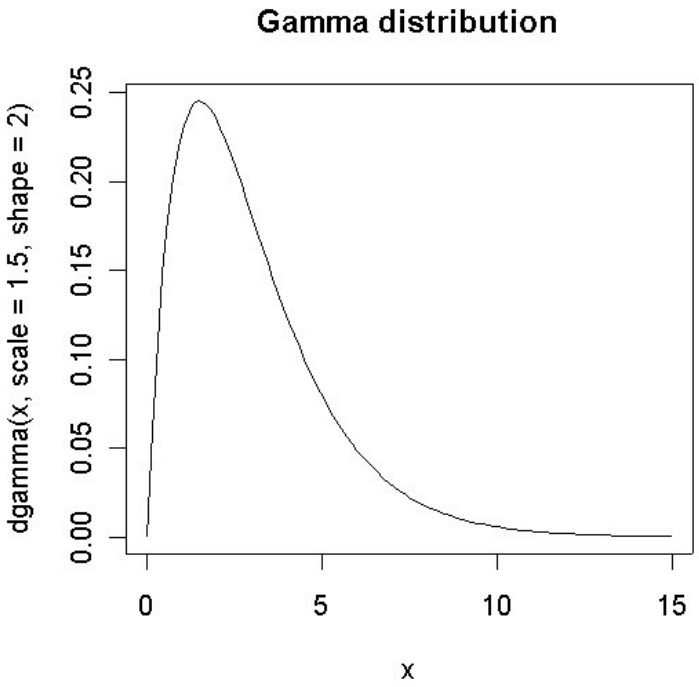
[Fig. 6]


Poisson distribution
Frequency
0
10
20
30
40
50
0
2
4
6
8
x.poi

[Fig. 7]


Normal distribution
dnorm(x, m = 10, sd = 2)
0.00
0.05
0.10
0.15
0.20
0
5
10
15
20
x

[Fig. 8]



Weibull distribution[10] (Fig. 9): $f(x,\alpha,\beta) = \alpha\beta^{-\alpha}x^{\alpha-1}e^{[-(\frac{x}{\beta})^{\alpha}]}$ with $x \in R^{+}$

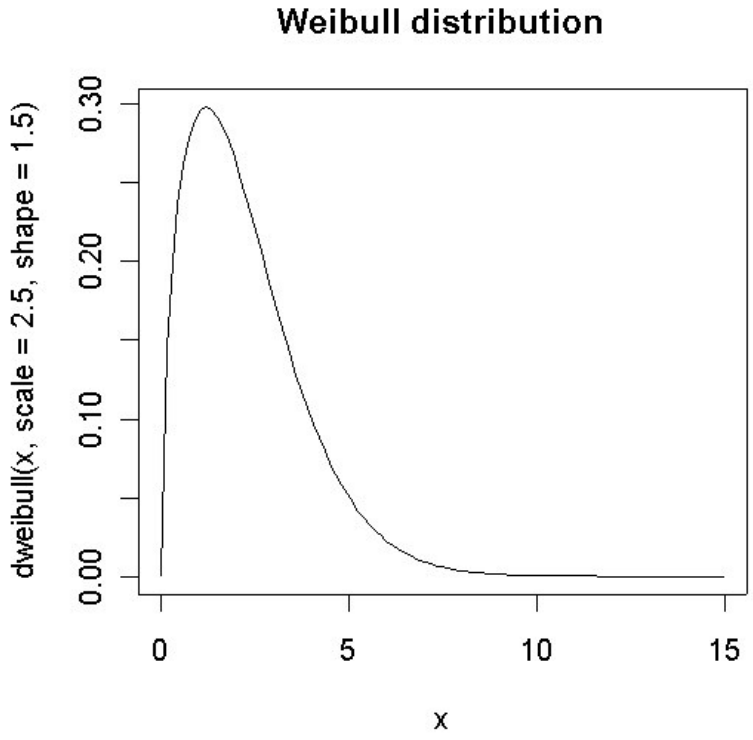
```
curve(dweibull(x, scale=2.5, shape=1.5),from=0, to=15, main="Weibull
distribution")
```

To compute skewness and kurtosis index we can use those statements: `skewness()` and `kurtosis()` included in `fBasics` package (you need to download this package from CRAN website):

```
library(fBasics) ## package loading
skewness(x.norm) ## skewness of a normal distribution
[1] 0.1242952
kurtosis(x.norm) ## kurtosis of a normal distribution
[1] 0.01372539

skewness(x.wei) ## skewness of a Weibull distribution
[1] 0.7788843
kurtosis(x.wei) ## kurtosis of a Weibull distribution
[1] 0.4331281
```

[10] See: http://www.xycoon.com/Weibull.htm [2005-01-12]

[Fig. 9]



4.0 Parameters' estimate

After choosing a model that can mathematically represent our data we have to estimate parameters of such model. There are several estimate methods in statistical literature, but in this paper we are focusing on these ones:

1) analogic
2) moments
3) maximum likelihood

Analogic method consists in estimating model parameters applying the same function to empirical data. I.e., we estimate the unknown mean of a normal population using the sample mean:

```
mean.hat<-mean(x.norm)
mean.hat
[1] 9.935537
```

The method of moments[11] is a technique for constructing estimators of the parameters that is based on matching the sample moments with the corresponding distribution moments. This method equates sample moments to population (theorical) ones. When moment methods are available, they have the advantage of simplicity. We define sample (empirical) moments in this way:

- *t-th sample moment about 0*: $m_t = \sum_{i=1}^{n} x_i^t y_i$ t=0,1,2…

[11] See http://www.itl.nist.gov/div898/handbook/eda/section3/eda3651.htm [2005-02-08]

- *t-th sample moment about mean*: $m_t^{'} = \sum_{i=1}^{n} (x_i - \mu)^t \, y_i$ t=0,1,2…

while theorical (population) ones:

- *t-th population moment about 0*: $m_t^{*} = \int_{\beta}^{\alpha} x^t f(x,\theta)\,\mathrm{dx}$ t=0,1,2…
- *t-th population moment about mean:* $m_t^{'*} = \int_{\beta}^{\alpha} (x-\mu)^t f(x,\theta)\,\mathrm{dx}$ t=0,1,2…

where $\beta-\alpha$ is the range where f(x,**θ**) is defined, μ is the mean of the distribution, and y_i are emprirical relative frequencies. I.e., we shall estimate parameters of a gamma distribution using the method of moments considering the first moment about 0 (mean) and the second moment about mean (variance):

$$\frac{\alpha}{\lambda} = \bar{x}$$

$$\frac{\alpha}{\lambda^2} = s^2$$

where on the left there mean and variance of gamma distribution and on the right sample mean and sample corrected variance. Solving we can get parameters' estimates:

$$\hat{\lambda} = \frac{\bar{x}}{s^2}$$

$$\hat{\alpha} = \frac{\bar{x}^2}{s^2}$$

```
x.gam<-rgamma(200,rate=0.5,shape=3.5)   ##   sampling   from   a   gamma
distribution with λ=0.5 (scale parameter[12]) and α=3.5 (shape parameter)

med.gam<-mean(x.gam) ## sample mean
var.gam<-var(x.gam) ## sample variance
l.est<-med.gam/var.gam ## lambda estimate (corresponds to rate)
a.est<-((med.gam)^2)/var.gam ## alfa estimate

l.est
[1] 0.5625486
a.est
[1] 3.916339
```

The method of maximum likelihood[13] is used in statistical inference to estimate parameters. We have a random variable with a known pdf f(x,**θ**) describing a quantitative character in the population. We should estimate the vector of costant and unknown parameters **θ** according to sampling data: $x_1, x_2, \ldots x_n$.
Maximum likelihood estimation begins with the mathematical expression known as a likelihood function of the sample data.. Loosely speaking, the likelihood of a set of data is the probability of obtaining that

[12] In `rgamma()` we can assign a value to `rate` or to `scale=1/rate`; `rate` argument corresponds to λ
[13] See: http://www.weibull.com/LifeDataWeb/maximum_likelihood_estimation_appendix.htm, and http://www.itl.nist.gov/div898/handbook/eda/section3/eda3652.htm [2005-01-13]

particular set of data given the chosen probability model. This expression contains the unknown parameters. Those values of the parameter that maximize the sample likelihood are known as the maximum likelihood estimates (MLE). We define the likelihood function as:

$$L(x_1, x_2, ...x_n, \theta) = \prod_{i=1}^{n} f(x_i, \theta)$$

MLE consist in finding **θ** which maximizes $L(x_1, x_2, ...x_n, \theta)$ or its logarithmic function.

We can employ mathematical analysis methods (partial derivates equal to zero) when the likelihood function is rather simple, but very often we optimise $L(x_1, x_2, ...x_n, \theta)$ using iterative methods. MLE have several statistical properties and advantages.

I. e., in case of a gamma distribution, the likelihood function is[14]::

$$L(x_1, x_2, ...x_n, \alpha, \lambda) = \prod_{i=1}^{n} f(x_i, \alpha, \lambda) = \prod_{i=1}^{n} \frac{\lambda^\alpha}{\Gamma(\alpha)} x_i^{\alpha-1} e^{-\lambda x_i} = (\frac{\lambda^\alpha}{\Gamma(\alpha)})^n (\prod_{i=1}^{n} x_i)^{\alpha-1} e^{-\lambda \sum_{i=1}^{n} x_i}$$

while its logarithmic is:

$$\log(L) = n\alpha \log(\lambda) - n \log(\Gamma(\alpha)) + (\alpha - 1) \sum_{i=1}^{n} \log x_i - \lambda \sum_{i=1}^{n} x_i$$

In R environment we can get MLE by two statements:

1) `mle()` included in package `stats4`
2) `fitdistr()` included in package `MASS`

`mle()` allows to fit parameters by maximum likelihood method using iterative methods of numerical calculus to minimize the negative log-likelihood (which is the same of maximizing the log-likelihood). You have to specify the negative log-likelihood analytical expression as argument and giving some starting parameters estimates. In case of a gamma distribution:

```
library(stats4) ## loading package stats4
ll<-function(lambda,alfa) {n<-200
                           x<-x.gam
-n*alfa*log(lambda)+n*log(gamma(alfa))-(alfa-
1)*sum(log(x))+lambda*sum(x)} ## -log-likelihood function
est<-mle(minuslog=ll, start=list(lambda=2,alfa=1))
summary(est)
Maximum likelihood estimation

Call:
mle(minuslogl = ll, start = list(lambda = 2, alfa = 1))

Coefficients:
        Estimate Std. Error
lambda 0.5290189 0.05430615
alfa   3.6829126 0.35287672

-2 log L: 1044.282
```

We supply as starting values of parameters estimates arbitrary ones, but, we could use estimates got by the methods of moments. The statement `mle()` allows to estimate parameters for every kind of pdf, it needs only to know the likelihood analytical expression to be optimised.

[14] See http://www-mtl.mit.edu/CIDM/memos/94-13/subsection3.4.1.html [2005-01-12]

In MASS package is available fitdistr() for maximum-likelihood fitting of univariate distributions without any information about likelihood analytical expression. It is enough to specify a data vector, the type of pdf (densfun) and eventually the list of starting values for iterative procedure (start).

```
library(MASS) ## loading package MASS
fitdistr(x.gam,"gamma") ## fitting gamma pdf parameters
     shape          rate
  3.68320097   0.52910229
 (0.35290545) (0.05431458)

fitdistr(x.wei,densfun=dweibull,start=list(scale=1,shape=2))##    fitting
Weibull pdf parameters
     scale         shape
  1.04721828   2.04959159
 (0.03814184) (0.11080258)

fitdistr(x.norm,"normal") ## fitting gaussian pdf parameters
     mean          sd
  9.9355373   2.0101691
 (0.1421404) (0.1005085)
```

5.0 Measures of goodness of fit

A goodness of fit measure is useful for matching empirical frequencies with fitted ones by a theorical model. We have absolute and relative measures. Among the absolute ones we can choice:

$$\xi = \frac{\sum_{i=1}^{n} | y_i - y_i^* |}{n}$$

$$^2\xi = \sqrt{\frac{\sum_{i=1}^{n} (y_i - y_i^*)^2}{n}}$$

where y_i ore emprical frequencies and y_i^* are fitted ones.

Some relative measures are below:

$$\delta = \frac{\xi}{\sum_{i=1}^{n} y_i / n} = \frac{\sum_{i=1}^{n} | y_i - y_i^* |}{\sum_{i=1}^{n} y_i}$$

$$^2\delta = \frac{^2\xi}{\sum_{i=1}^{n} y_i / n} = \frac{\sqrt{\sum_{i=1}^{n} (y_i - y_i^*)^2 / n}}{\sum_{i=1}^{n} y_i / n}$$

$$^2_2\delta = \frac{^2\xi}{\sqrt{\sum_{i=1}^{n} y_i^2 / n}} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - y_i^*)^2}{\sum_{i=1}^{n} y_i^2}}$$

Usually those indexes are expressed by a percent measure of the corresponding mean.

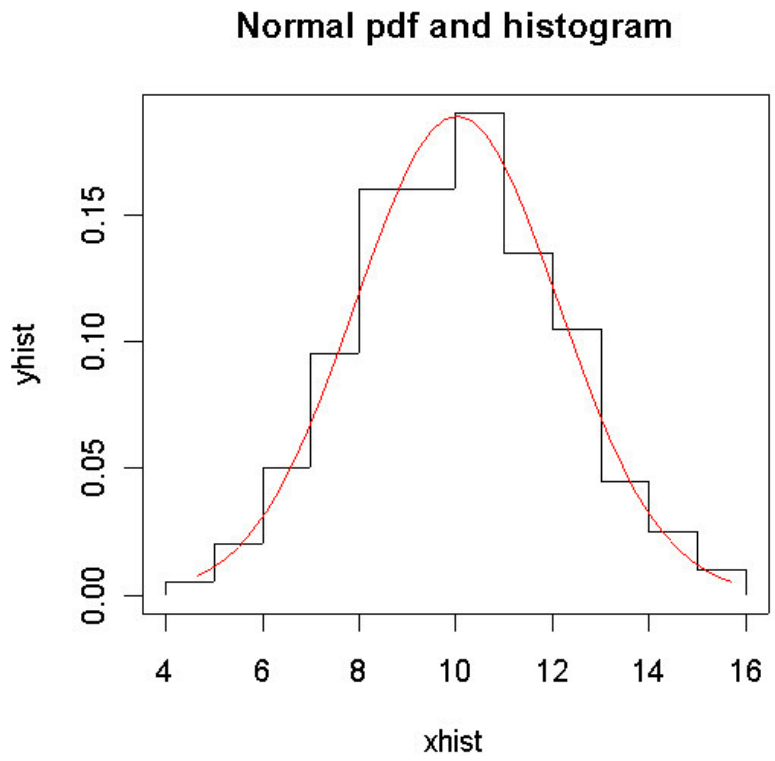
Here is an example using R for count data (Poisson distribution):

```
lambda.est<-mean(x.poi) ## estimate of parameter lambda
tab.os<-table(x.poi)## table with empirical frequencies
tab.os
x.poi
 0  1  2  3  4  5  6  7  8
21 29 46 53 28 16  4  2  1

freq.os<-vector()
for(i in 1: length(tab.os)) freq.os[i]<-tab.os[[i]] ## vector of emprical
frequencies
freq.ex<-(dpois(0:max(x.poi),lambda=lambda.est)*200) ## vector of fitted
(expected) frequencies
freq.os
[1] 21 29 46 53 28 16  4  2  1

freq.ex
[1] 15.0040080 38.8603808 50.3241931 43.4465534 28.1316433 14.5721912
6.2903292
[8]  2.3274218  0.7535028

acc<-mean(abs(freq.os-trunc(freq.ex))) ## absolute goodness of fit index
acc
[1] 2.111111

acc/mean(freq.os)*100 ## relative (percent) goodness of fit index
[1] 17
```

A graphical technique to evaluate the goodness of fit can be drawing pdf curve and histogram together (Fig. 10):

```
h<-hist(x.norm,breaks=15)
xhist<-c(min(h$breaks),h$breaks)
yhist<-c(0,h$density,0)
xfit<-seq(min(x.norm),max(x.norm),length=40)
yfit<-dnorm(xfit,mean=mean(x.norm),sd=sd(x.norm))
plot(xhist,yhist,type="s",ylim=c(0,max(yhist,yfit)), main="Normal pdf and
histogram")
lines(xfit,yfit, col="red")
```

[Fig. 10]



6.0 Goodness of fit tests

Goodness of fit tests indicate whether or not it is reasonable to assume that a random sample comes from a specific distribution. They are a form of hypothesis testing where the null and alternative hypotheses are:

H_0: Sample data come from the stated distribution
H_A: Sample data do not come from the stated distribution

These tests are sometimes called as *omnibus test* and they are *distribution free*, meaning they do not depend according the pdf. We shall point out our attention to normality tests.
The chi-square test[15] is the oldest goodness of fit test dating back to Karl Pearson (1900). The test may be thought of as a formal comparison of a histogram with the fitted density.
An attractive feature of the chi-square (χ^2) goodness of fit test is that it can be applied to any univariate distribution for which you can calculate the cumulative distribution function. The chi-square goodness of fit test is applied to binned data (i.e., data put into classes). This is actually not a restriction since for non-binned data you can simply calculate a histogram or frequency table before generating the chi-square test. However, the value of the chi-square test statistic is dependent on how the data is binned. Another disadvantage of this test is that it requires a sufficient sample size in order for the chi square approximation to be valid. The chi-square goodness of fit test can be applied either to discrete distributions or continuous ones while the Kolmogorov-Smirnov and Anderson-Darling tests are restricted to continuous distributions. Estimating model parameters with sample is allowed with this test. The chi-square test is defined for the hypothesis:

H_0: the data follow a specified distribution
H_A: the data do not follow the specified distribution

[15] See: http://www.itl.nist.gov/div898/handbook/eda/section3/eda35f.htm and http://www.itl.nist.gov/div898/handbook/prc/section2/prc211.htm [2005-01-14]

For the chi-square goodness of fit computation, the data are divided into k bins and the test statistic is defined in this way:

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

where O_i is the observed frequency for bin i and E_i is the expected frequency for bin i. The expected frequency is calculated by cumulative distribution function. This statistic is distributed as a χ^2 random variable with k-p-1 degrees of freedom (p is the number of parameters estimated by sample data). The hypothesis that the data are from a population with the specified distribution is accepted if χ^2 is lower than the chi-square percent point function with k - p-1 degrees of freedom and a significance level of α. The chi-square test is sensitive to the choice of bins.

In R environment there are three ways to perform a chi-square test.
In case of count data we can use `goodfit()` included in `vcd` package (to be downloaded from CRAN website):

```
library(vcd)## loading vcd package

gf<-goodfit(x.poi,type= "poisson",method= "MinChisq")
summary(gf)

           Goodness-of-fit test for poisson distribution

              X^2 df  P(> X^2)
Pearson 8.378968  7 0.3003653

plot(gf,main="Count data vs Poisson distribution")
```

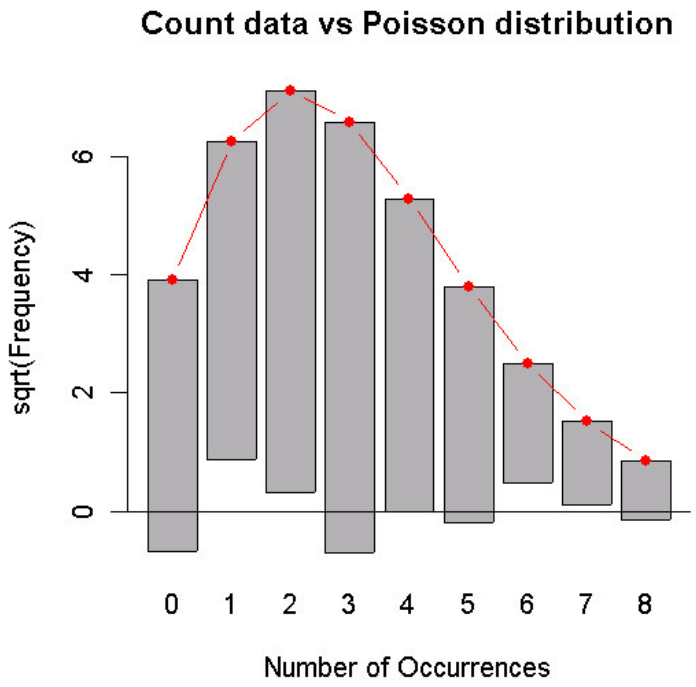
In case of a continuous variable, such as a gamma distribution as in the following example, with parameters estimated by sample data:

```
x.gam.cut<-cut(x.gam,breaks=c(0,3,6,9,12,18)) ##binning data
table(x.gam.cut) ## binned data table
x.gam.cut
  (0,3]   (3,6]   (6,9]  (9,12] (12,18]
     26      64      60      27      23

## computing expected frequencies
(pgamma(3,shape=a.est,rate=l.est)-pgamma(0,shape=a.est,rate=l.est))*200
[1] 19.95678

(pgamma(6,shape=a.est,rate=l.est)-pgamma(3,shape=a.est,rate=l.est))*200
[1] 70.82366

(pgamma(9,shape=a.est,rate=l.est)-pgamma(6,shape=a.est,rate=l.est))*200
[1] 60.61188

(pgamma(12,shape=a.est,rate=l.est)-pgamma(9,shape=a.est,rate=l.est))*200
[1] 30.77605

(pgamma(18,shape=a.est,rate=l.est)-pgamma(12,shape=a.est,rate=l.est))*200
[1] 16.12495

f.ex<-c(20,71,61,31,17) ## expected frequencies vector
```

```
f.os<-vector()
for(i in 1:5) f.os[i]<- table(x.gam.cut)[[i]] ## empirical frequencies

X2<-sum(((f.os-f.ex)^2)/f.ex) ## chi-square statistic

gdl<-5-2-1 ## degrees of freedom
1-pchisq(X2,gdl) ## p-value
[1] 0.07652367
```

H_0 is accepted as the p-value is grater than of a significance level fixed al least in 5%.

[Fig. 11]



Whether we are dealing with a continuous variable and all its pdf parameters are known we can use `chisq.test()`:

```
## computing relative expected frequencies
p<-c((pgamma(3,shape=3.5,rate=0.5)-pgamma(0,shape=3.5,rate=0.5)),
(pgamma(6,shape=3.5,rate=0.5)-pgamma(3,shape=3.5,rate=0.5)),
(pgamma(9,shape=3.5,rate=0.5)-pgamma(6,shape=3.5,rate=0.5)),
(pgamma(12,shape=3.5,rate=0.5)-pgamma(9,shape=3.5,rate=0.5)),
(pgamma(18,shape=3.5,rate=0.5)-pgamma(12,shape=3.5,rate=0.5)))

chisq.test(x=f.os,p=p) ## chi-square test

        Chi-squared test for given probabilities

data:  f.os
X-squared = 2.8361, df = 4, p-value = 0.5856
```

We can't reject null hypothesis as p-value is rather high, so probably sample data belong from a gamma distribution with shape parameter=3.5 and rate parameter=0.5.

The Kolmogorov-Smirnov[16] test is used to decide if a sample comes from a population with a specific distribution. I can be applied both for discrete (count) data and continuous binned (even if some Authors do not agree on this point) and both for continuous variables. It is based on a comparison between the empirical distribution function (ECDF) and the theoretical one defined as $F(x)=\int_{\alpha}^{x} f(y,\theta)dy$, where f(y,θ) is the pdf. Given n ordered data points X_1, X_2, ..., Xn, the ECDF is defined as:

$$F_n(X_i)=N(i)/n$$

where N(i) is the number of points less than X_i (X_i are ordered from smallest to largest value). This is a step function that increases by 1/n at the value of each ordered data point.
The test statistic used is:

$$D_n = \sup_{1 \le i \le n} |F(x_i) - F_n(x_i)|$$

that is the upper extreme among absolute value differencies between ECDF and theoretical CDF.
The hypothesis regarding the distributional form is rejected if the test statistic, D_n, is greater than the critical value obtained from a table, or, which is the same, if the p-value is lower than the significance level.
Kolmogorov-Smirnov test is more powerful than chi-square test when sample size is not too great. For large size sample both the tests have the same power. The most serious limitation of Kolmogorov-Smirnov test is that the distribution must be fully specified, that is, location, scale, and shape parameters can't be estimated from the data sample. Due to this limitation, many analysts prefer to use the Anderson-Darling goodness-of-fit test. However, the Anderson-Darling test is only available for a few specific distributions.
In R we can perform Kolmogorov-Smirnov test using the function `ks.test()` and apply this test to a sample belonging from a Weibull pdf with known parameters (shape=2 and scale=1):

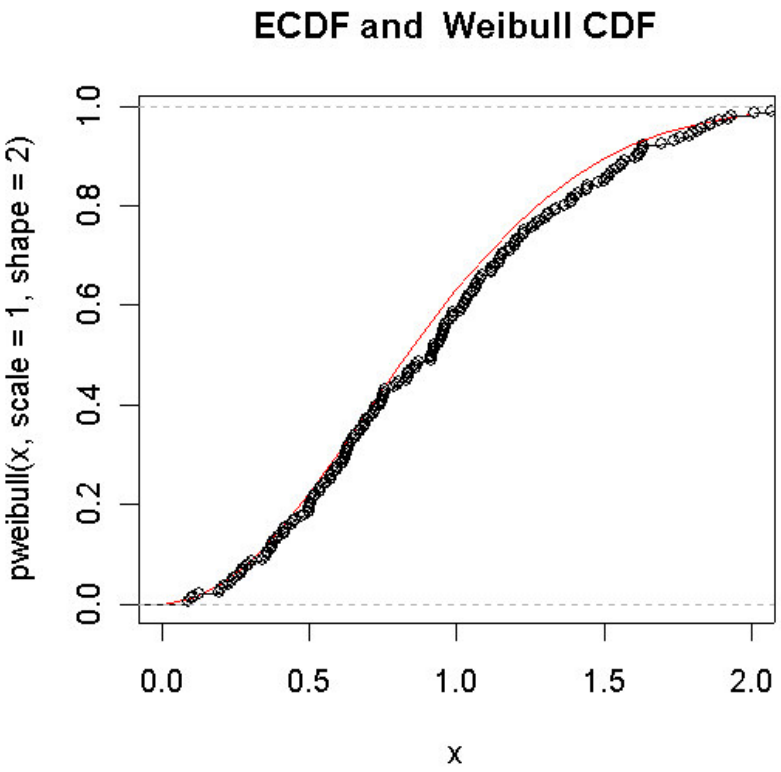
```
ks.test(x.wei,"pweibull", shape=2,scale=1)

        One-sample Kolmogorov-Smirnov test

data:  x.wei
D = 0.0623, p-value = 0.4198
alternative hypothesis: two.sided
```

We accept null hypothesis that the data follow a Weibull distribution because the p-value is enough higher than significance levels usually referred in statistical literature.
In Fig.12 is drawn both ECDF and theoretical one in the same plot:

```
x<-seq(0,2,0.1)
plot(x,pweibull(x,scale=1,shape=2),type="l",col="red",   main="ECDF   and
Weibull CDF")
plot(ecdf(x.wei),add=TRUE)
```

[16] http://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm [2005-01-14]

[Fig. 12]



6.1 Normality tests

Very often a statistician is called to test if data collected come or not from a normal population, we shall examine the main normality tests[17].
There are in statistical literature some tests useful for testing only skewness or only kurtosis (or both the two at the same time) of a distribution based on the well known b3 e b4 (or gamma3 e gamma4).[18]
Shapiro-Wilk test[19] is one of the most powerful normality tests, especially for small samples. Normality is tested by matching two alternative variance estimates: a non-parametric estimator got by a linear combination of ordered sample values and the usual parametric estimator. The weights (a_i) are available in a statistical table:

$$W = \frac{(\sum_{i=1}^{n} a_i x_{(i)})^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

The statement performing Shapiro-Wilk test is `shapiro.test()` and it supplies W statistic and the p-value:

```
shapiro.test(x.norm)

        Shapiro-Wilk normality test
```

[17] E. Seier, Testing for normality and E. Seier, Comparison of tests for univariate normality [2005-02-18]
[18] See: http://www.xycoon.com/skewness_test_1.htm and http://www.xycoon.com/skewness_small_sample_test_2.htm [2005-02-10]
[19] See: http://www.itl.nist.gov/div898/handbook/prc/section2/prc213.htm [2005-01-14]

```
data:  x.norm
W = 0.9938, p-value = 0.5659
```

The p-value is higher than significance levels usually used to test statistical hypotheses, we accept null hypothesis that is sample data belong form a gaussian distribution.

Jarque-Bera test[20] is used a lot to test normalità in Econometric. It is based on skewness and kurtosis measures of a distribution considering the asymptotic distribution of b3 e b4 which, under null hypothesis, is a chi-square with 2 degrees of freedom.
In R such test is available in `tseries` package (it should be downloaded from CRAN website) using this statement: `jarque.bera.test()` which supplies the value of statistic, the degrees of freedom and the p-value:

```
library(tseries) ## package tseries loading
jarque.bera.test(x.norm)

        Jarque Bera Test

data:  x.norm
X-squared = 0.539, df = 2, p-value = 0.7638
```

A test proposed by Cucconi (an Italian statistician) allows to test normality without the problem of estimating parameters by sample data. Let be $x_1 \geq x_2 \geq \ldots \geq x_n$ a sample from a continuous variable and $\zeta_1, \zeta_2, \ldots \zeta_n$ a set of standardized random normal numbers of size n; let be:

$$r = \zeta_n \quad \text{e} \quad q = \sqrt{\frac{\sum_{i=1}^{n-1} \zeta_i^2}{n-1}}$$

we consider a transformation of x_i: $y_i = q\frac{x_i - \bar{x}}{\hat{\sigma}} + \frac{r}{\sqrt{n}}$ where $\bar{x}$ is the sample mean and $\hat{\sigma}$ is the square root of the corrected sample variance. It could be demonstered that, if x_i come from a normal population, y_i have a normal standardized distribution. We can employ Kolmogorov-Smirnov test to check this hypothesis. Here is an example of R code:

```
zz<-rnorm(n=200,m=0,sd=1) ## sampling random numbers from N(0,1)
r<-zz[200]
q<-sd(zz[-200])
m<-mean(x.norm)
s<-sqrt(var(x.norm))
y<-q*((x.norm-m)/s)+(r/sqrt(200))
ks.test(y,"pnorm",m=0,sd=1)

        One-sample Kolmogorov-Smirnov test

data:  y
D = 0.0298, p-value = 0.9943
alternative hypothesis: two.sided
```

A package called `nortest` (it should be downloaded from CRAN website) allows to perform 5 different normality test:

[20] See: http://homepages.uel.ac.uk/D.A.C.Boyd/JARQUE-B.PDF [2005-01-14]

1) `sf.test()` performs Shapiro-Francia test:

```
library(nortest) ## package loading
sf.test(x.norm)

        Shapiro-Francia normality test

data:  x.norm
W = 0.9926, p-value = 0.3471
```

2) `ad.test()` performs Anderson-Darling[21] test:

it is a modification of Kolmogorov-Smirnov test and it makes use of the specific distribution in calculating critical values. Currently, tables of critical values are available for the normal, lognormal, exponential, Weibull, extreme value type I, and logistic distributions. Anderson-Darling test is based on this statistic:

$$A^2=-n-S$$

where:

$S=\sum_{i=1}^{n}\frac{(2i-1)}{n}[\ln F(x_i)+\ln(1-F(x_{n+i+1}))]$, n is the sample size and F(x) is the theoretical CDF (in our case is the normal CDF). In R environment this test can be used only to check normality:

```
library(nortest) ## package loading
ad.test(x.norm)

        Anderson-Darling normality test

data:  x.norm
A = 0.4007, p-value = 0.3581
```

3) `cvm.test()` performs Cramer-Von Mises test based on the statistic:

$$W^2=\int_{-\infty}^{+\infty}(F_n(x)-F(x))^2 f(x)dx$$

```
library(nortest) ## package loading
cvm.test(x.norm)

        Cramer-von Mises normality test

data:  x.norm
W = 0.0545, p-value = 0.4449
```

4) `lillie.test()` performs Lilliefors[22] test:

It is a modification of Kolmogorov-Smirnov test which can't be used normality when the mean and standard deviation of the hypothesized normal distribution are not known (i.e., they are estimated from the sample data), it is particularly useful in case of small samples. The Lilliefors test evaluates the hypothesis that X has a normal distribution with unspecified mean and variance, against the alternative that X does not have a normal distribution. This test compares the empirical distribution of X with a normal distribution having the same mean and variance as X. It is similar to the Kolmogorov-Smirnov test, but it adjusts for the fact that the parameters of the normal distribution are estimated from X rather than specified in advance.

```
library(nortest) ## package loading
```

[21] http://www.itl.nist.gov/div898/handbook/eda/section3/eda35e.htm [2005-01-18]

```
lillie.test(x.norm)
Lilliefors (Kolmogorov-Smirnov) normality test
data:  x.norm
D = 0.0414, p-value = 0.5509
```

5) `pearson.test()` performs Pearson's chi-square test:
it is the same χ^2 test previously treated used to test the goodness of fit of a normal distribution:

```
library(nortest) ## package loading
pearson.test(x.norm)

        Pearson chi-square normality test

data:  x.norm
P = 10.12, p-value = 0.753
```

Appendix

List of R statements useful in fitting distributions. The package including statement is written in parenthesis.

`ad.test()`: Anderson-Darling test for normality (`nortest`)
`chisq.test()`: chi-squared test (`stats`)
`cut`: divides the range of data vector into intervals
`cvm.test()`: Cramer-von Mises test for normality (`nortest`)
`ecdf()`: computes an empirical cumulative distribution function (`stats`)
`fitdistr()`: Maximum-likelihood fitting of univariate distributions (`MASS`)
`goodfit()`: fits a discrete (count data) distribution for goodness-of-fit tests (`vcd`)
`hist()`: computes a histogram of the given data values (`stats`)
`jarque.bera.test()`: Jarque-Bera test for normality (`tseries`)
`ks.test()`: Kolmogorov-Sminorv test (`stats`)
`kurtosis()`: returns value of kurtosis (`fBasics`)
`lillie.test()`: Lilliefors test for normality (`nortest`)
`mle()`: estimate parameters by the method of maximum likelihood (`stats4`)
`pearson.test()`: Pearson chi-square test for normality (`nortest`)
`plot()`: generic function for plotting of R objects (`stats`)
`qqnorm()`: produces a normal QQ plot (`stats`)
`qqline()`, `qqplot()`: produce a QQ plot of two datasets (`stats`)
`sf.test()`: test di Shapiro-Francia per la normalità (`nortest`)
`shapiro.test()`:Shapiro-Francia test for normalità (`stats`)
`skewness()`: returns value of skewness (`fBasics`)
`table()`: builds a contingency table (`stats`)

References

D.M. BATES, "Using Open Source Software to Teach Mathematical Statistics", 2001 http://www.stat.wisc.edu/~bates/JSM2001.pdf
R CORE DEVELOPMENT TEAM, An introcution to R, Release 2.0.1, November 2004 http://cran.r-project.org/doc/manuals/R-intro.pdf
E. SEIER, Testing for normality http://www.etsu.edu/math/seier/2050/normtest.doc
E. SEIER, Comparison of tests for univariate normality http://interstat.stat.vt.edu/InterStat/ARTICLES/2002/articles/J02001.pdf
SYRACUSE RESEARCH CORPORATION, ENVIRONMENTAL SCIENCE CENTER
Selecting and Parameterizing Data-Rich Distributions PRA Center Short Course October 20-21, 2004 http://esc.syrres.com/pracenter/esf2004/downloads/classnotes/_2_Select%20Parameterize.ppt

Statistics - Econometrics - Forecasting: http://www.xycoon.com/
NIST/SEMATECH e-Handbook of Statistical Methods: http://www.itl.nist.gov/div898/handbook/
Reliability Engineering and Weibull Analysis Resources: http://www.weibull.com/