Exploratory Data Analysis with R

Rasoul Behboudi
Northeastern University

Exploratory Data Analysis

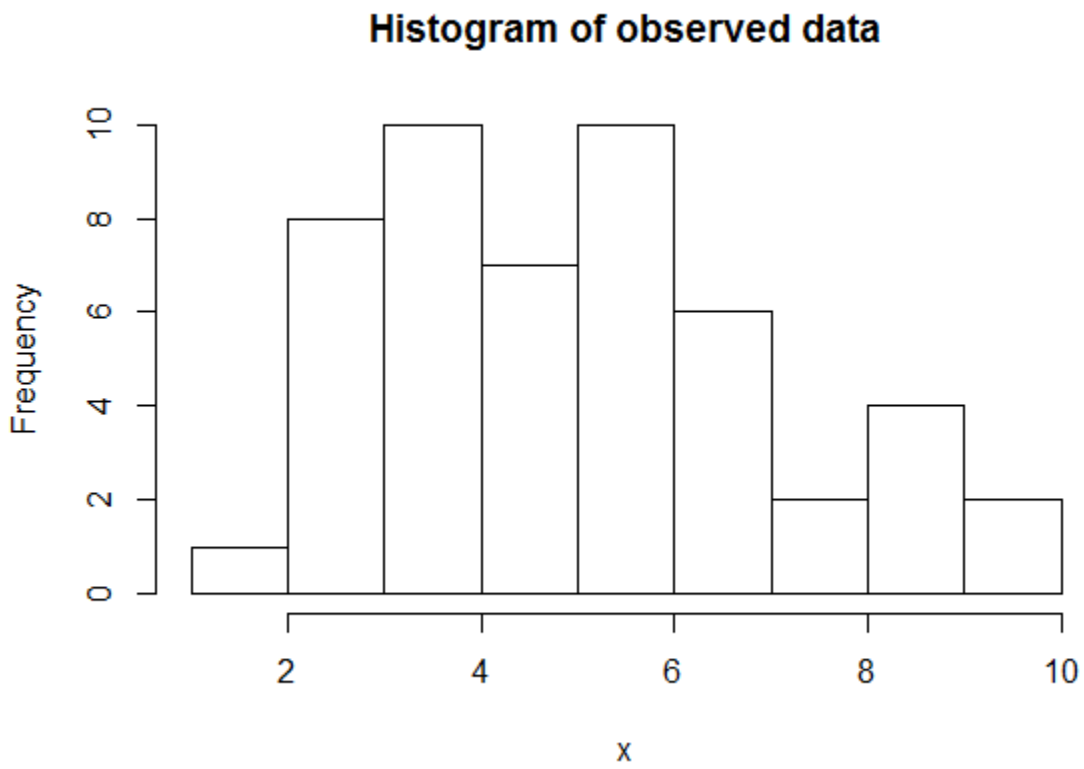
Exploratory data analysis is often the first step in analyzing data. This involves calculating numerical characteristics of data (such as the mean, median, standard deviation, skewness, …, etc.). Using graphical techniques (histograms, density estimate, ECDF) will suggest the kind of pdf to use to fit the model.

We can obtain samples from some pdf (such as normal, Poisson, Weibull, gamma, etc.) using R statements. Suppose we have a sample of size n=50 belonging to a normal population $N(5, 4)$; that is, with mean=5 and standard deviation=2:

```
x <-rnorm(n=50,5,2)
```

We can get a histogram using **`hist()`** statement:

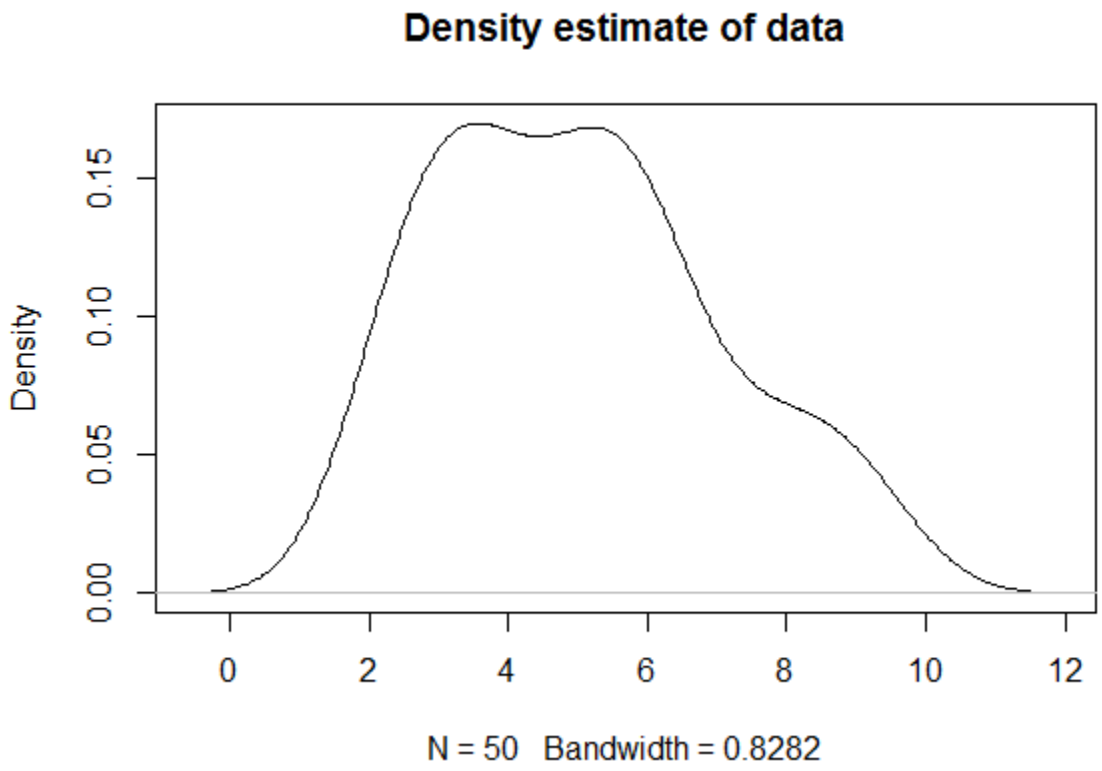
```
hist(x, main="Histogram of observed data")
```



Histograms can provide insights on outliers, skewness, behavior in the tails, and presence of multi-modal behavior. They can also be compared to the fundamental shapes of standard analytic distributions.

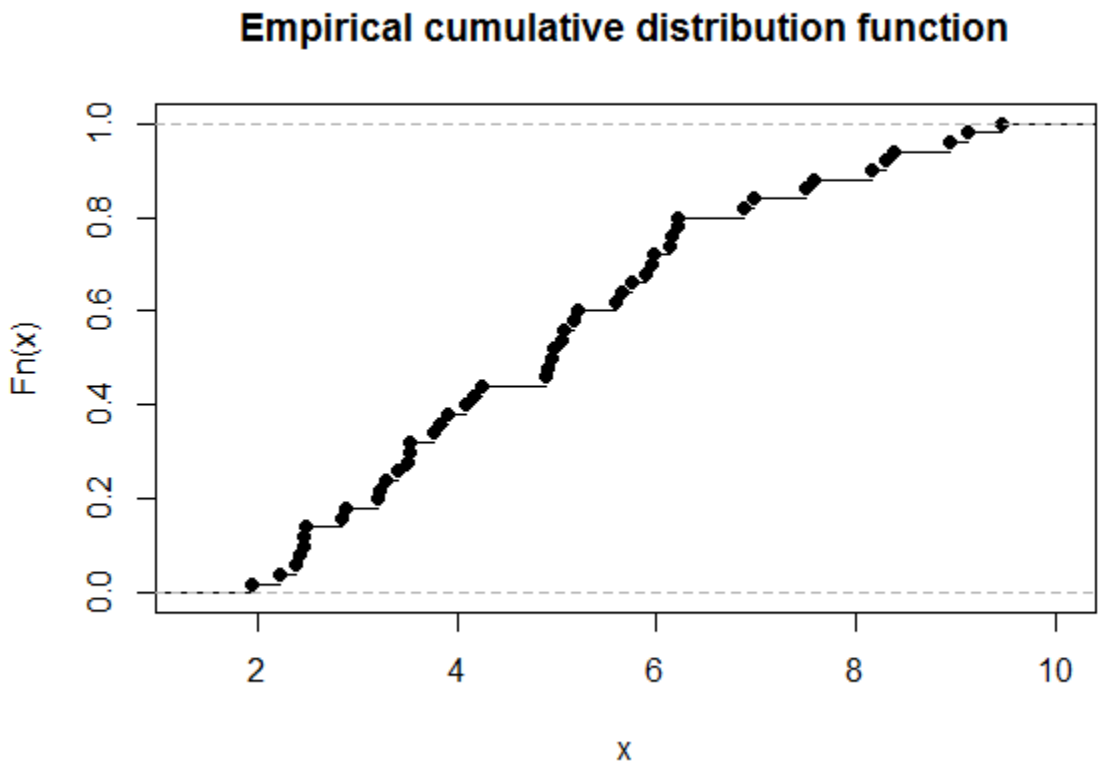
The frequency density of data can be estimated by using **`density()`** and **`plot()`** functions:

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



R also allows to compute the empirical cumulative distribution function by using the **`ecdf()`** function:

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



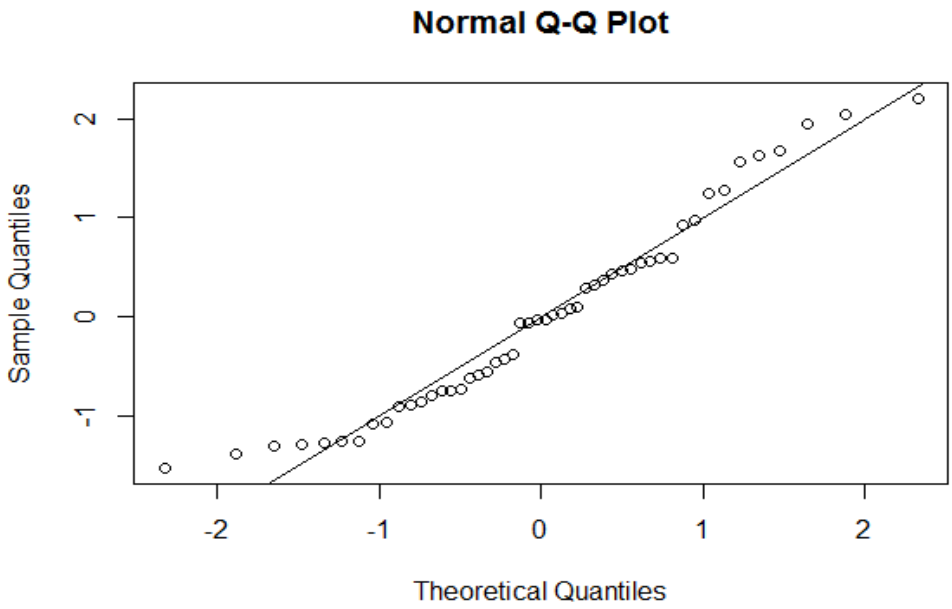
A good graphical technique for determining if a data set has come from a known popul ation, is to use a Quantile-Quantile plot (**qqplot()**). A Q-Q plot is a scatter plot that is us ed for comparing the fitted and the empirical distributions.

R offers two statements: **qqnorm() -** to analyze the goodness of fit of a normal distributi on **-** and **qqplot()** to analyze the goodness of fit of any kind of distribution. To obtain t hese plots, we must first convert our ***x*** measurements to their corresponding standardiz ed (*z*) values:

```
z <-(x-mean(x))/sd(x)
```

We can then draw the **qqplot()** and a 45-degree reference line for analyzing the goodness of the fit:

```
qqnorm(z)
abline(0,1)
```



If the empirical data come from the population with the chosen 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.

`qqplot()` can also be used to indicate that the selected distribution is the correct selection, however specifications are not accurate. For example, for the case of the normal population, our sample specifications (mean and standard deviation) may not truly reflect those of the population.

Suppose we sample 200 values from a normal population with a mean of 10 and a standard deviation of 2:

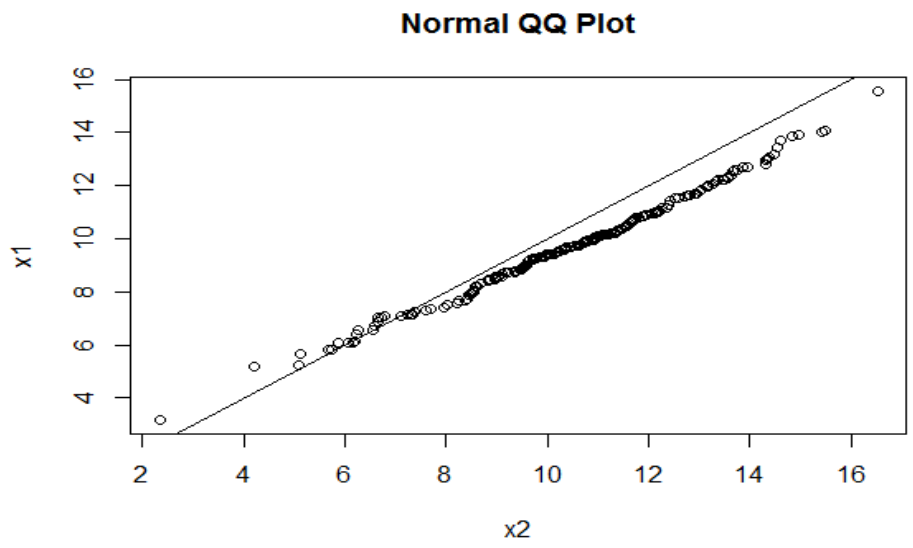
```
x1 <-rnorm(n=200,10,2)
```

Now let x2 denote the theoretical quantiles from a normal distribution with a mean of 10.5 and a standard deviation of 2.5:

```
x2 <-rnorm(n=200,10.5,2.5)
```

We can now use the `qqplot()` in the following way:

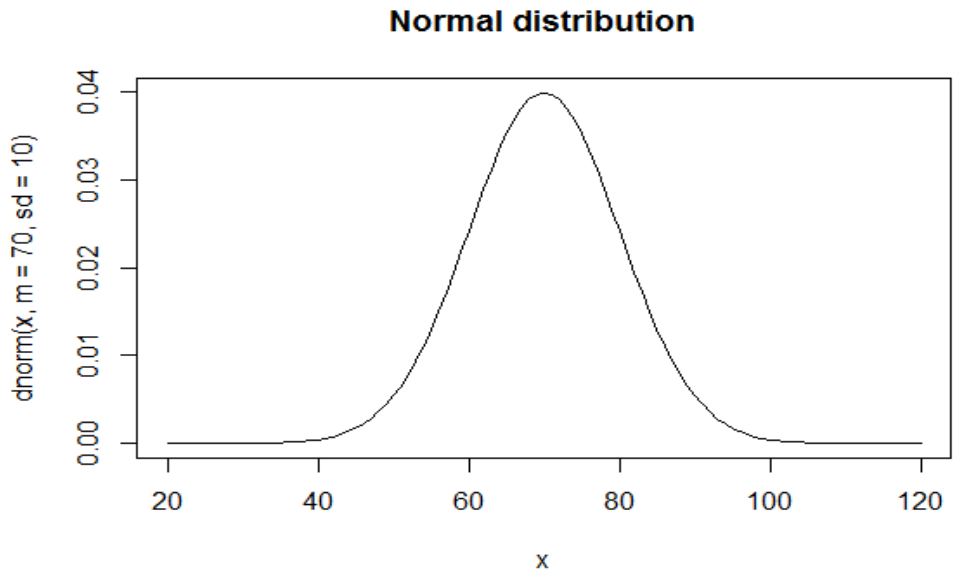
```
qqplot(x2, x1, main="Normal QQ Plot")
abline(0,1)
```



A Few Important Continuous Distributions:

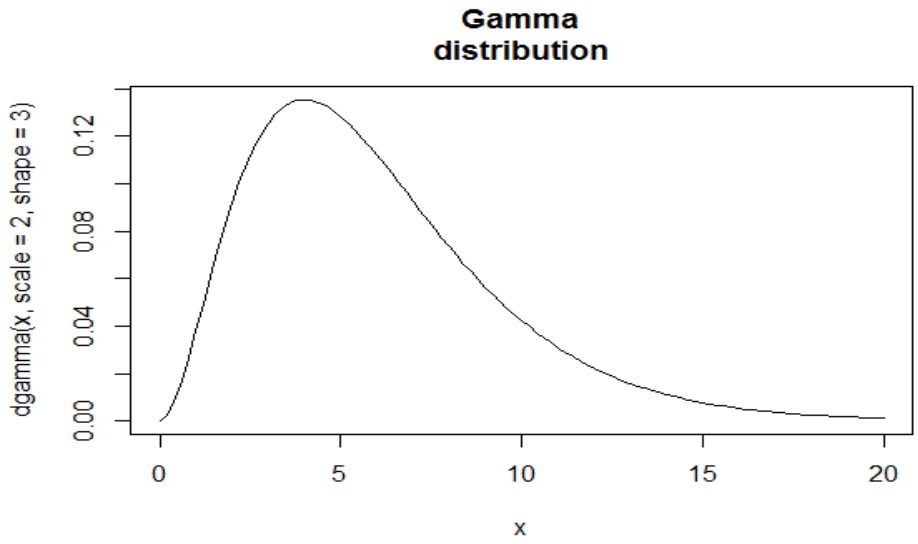
1. Normal Distribution: A normal distribution is characterized by the values of its mean and its standard deviation. The `curve()` function can be used to plot a normal distribution:

```
curve(dnorm(x,m=70,sd=10),from=20,to=120,main="Normal distribution")
```



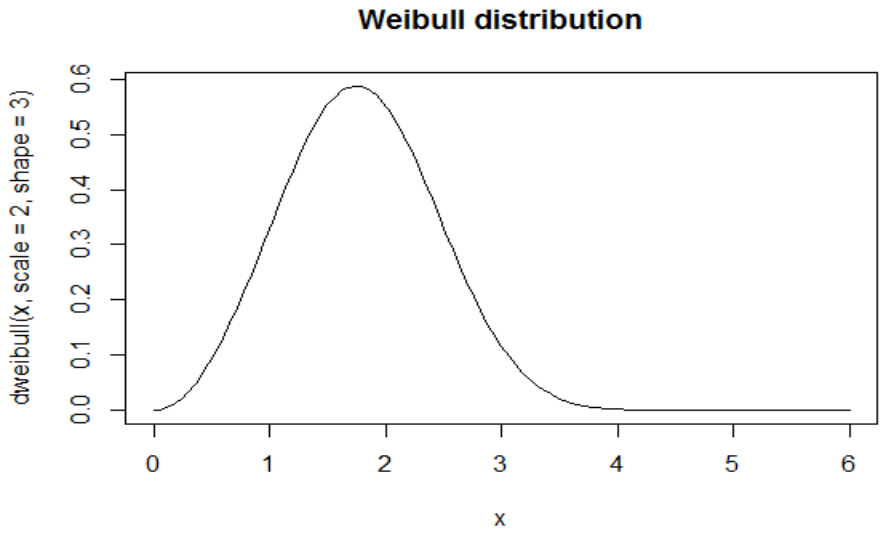
2. Gamma Distribution: A gamma distribution is characterized by two parameters: scale (α) (or rate) and shape (β).

```
curve(dgamma(x, scale=2, shape=3),from=0, to=20, main="Gamma distribution")
```



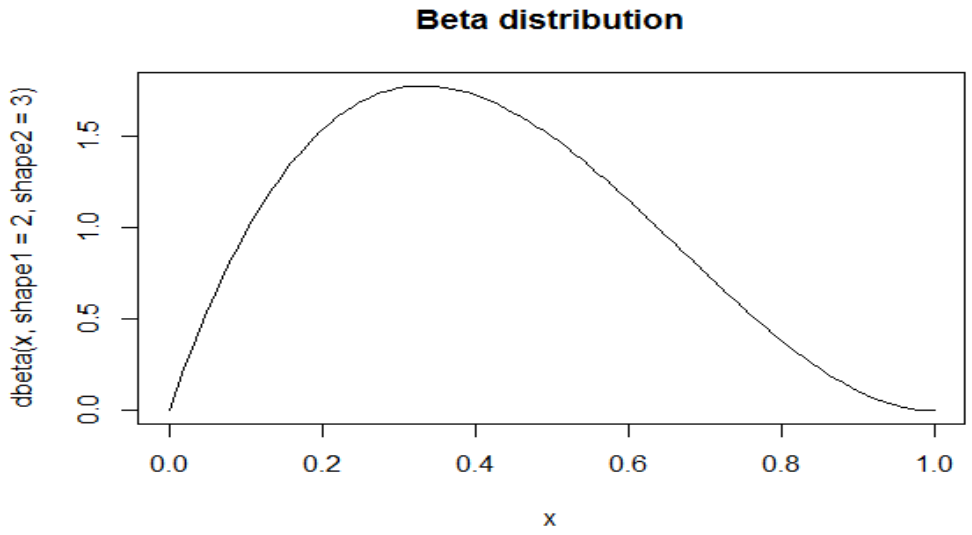
3. Weibull Distribution: As in gamma distribution, a Weibull distribution is characterized by two parameters: scale (α) and shape (β).

```
curve(dweibull(x, scale=2,shape=3),from=0,to=6,main="Weibull distribution")
```



4. Beta Distribution: A beta distribution is characterized by two shape parameters: shape1 (a) and shape2 (b).

```
curve(dbeta(x, shape1=2,shape2=3),from=0,to=1,main="Beta distribution")
```



Skewness and Kurtosis:

To compute skewness and kurtosis index we can use those statements: **`skewness()`** and **`kurtosis()`** included in **`fBasics`** package (you need to install this package)

```
install.packages('fBasics')
library(fBasics)
```

We now create a random vector of length 1000 with a scale parameter of 2 and a shape parameter of 3:

```
y<-rgamma(n=1000, scale=2, shape=3)
```

We can now calculate the Skewness and Kurtosis of this variable:

```
skewness(y)
[1] 1.316736  # this is a positively (right) skewed distribution
```

```
kurtosis(y)
[1] 3.055233  # this distribution is taller than the standard normal
```

Estimating the Parameters of a Distribution:

After choosing a model that can mathematically represent our data, we have to estimate the parameters of that model. There are several estimate methods in statistics that allow for estimating parameters. The following are a few examples of such methods:

1. Analogic Methods: Allow for estimating parameters based on empirical data (sample).
2. Method of Moments[1]: Estimation is based on matching the moments of the sample data with those of the theoretical distribution.
3. Maximum Likelihood Method: Estimation is based on maximizing a certain function called the Likelihood Function.

Analogic methods are the simplest methods for estimating population parameters. Here, one simply calculates the sample parameter and uses it as an estimate for the corresponding population parameter. This method is useful for when the functions for estimating the parameters are available. For example, in the case of a normal distribution, the parameters are the mean and the standard deviation, and the "mean" and "sd" functions can be used to calculate these statistics and use them to estimate the corresponding population parameters.

The method of moments on the other hand become significant when functions for estimating the parameters are not available. For example, in the case of the gamma distribution, the two parameters α (scale, rate) and β (shape) cannot be directly calculated by using the corresponding sample statistics. However, when equating the first and the second moments of the sample and the population (theoretical moments), one obtains the following two formulas for estimating α and β:

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

[1] By definition, the t-th sample moment about zero is: $m_t = \sum_i^n x_i^t\ f_i\ ; i = 1, 2, \ldots, n$; and the t-th sample moment about the mean is: $m_t = \sum_i^n (x_i - \mu)^t\ f_i\ ; i = 1, 2, \ldots, n$. In both formulas f_i represents the relative frequency corresponding to x_i and can be replaced with $\frac{1}{n}$ where n is the sample size. Note that the first moment ($t = 1$) is the sample mean and the second moment ($t = 2$) is the variance.

For example, suppose that we have obtained a sample data of size 200 from a gamma distribution with $\alpha = 0.5$ and $\beta = 3$:

```
x<-rgamma(200,rate=0.5,shape=3)
```

We can then estimate the population parameters α and β by using the sample mean and sample variance in the above two formulas:

```
m<-mean(x)
v<-var(x)
alpha.est<-m/v


beta.est<-m^2/v
data.frame("alpha estimate"=alpha.est,"beta estimate"=beta.est)
  alpha.estimate beta.estimate
1      0.4241133      2.692433
```

The Maximum likelihood estimation begins with the mathematical expression known as a likelihood function of the sample data. The likelihood of a set of data (sample) is the conditional probability of obtaining that particular sample 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).

In R, we can obtain MLE by two statements:

1. `mle()` included in the package **stats4**
2. `fitdist()` 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 as maximizing the log-likelihood). Here, we have to specify the negative log-likelihood analytical expression as argument and also provide some starting values for the parameters estimates. For example, for the gamma distribution,

```
library(stats4)
```

Suppose that we have the following sample:

```
x<-rgamma(200,rate=0.5,shape=2)
```

We first create a function to define the negative log-likelihood for the gamma distribution:

```
likegamma<-function(alpha,beta) {
     n<-200
     x<-x
     -n*beta*log(alpha)+n*log(gamma(beta))-(beta-1)*sum(log(x))+alpha*sum(x)
 }
```

We can now use the **mle()** function to estimate the parameters of the gamma distribution:

```
mle(minuslogl = likegamma, start = list(alpha = 1, beta = 1))

Call:
mle(minuslogl = likegamma, start = list(alpha = 1, beta = 1))

Coefficients:
    alpha      beta
0.4746478 1.9130568
```

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

```
library(MASS)
fitdistr(x,"gamma")

shape         rate
  1.91294978   0.47462323
 (0.17714486) (0.05020469)
```

(The values in parentheses are the standard errors for the two estimations).

Analyzing the Goodness of Fit:

There are several goodness of fit measures that allow for measuring the difference between the empirical frequencies with those obtained from a theoretical distributions. Below are a few examples. In this list, f_i describes the i-th observed frequency and $\hat{f}_i$ is corresponding predicted frequency obtained from the theoretical model:

(1) $d_1 = \frac{\Sigma| f_i - \hat{f}_i |}{n}$ *mean absolute deviation (MAD)*

(2) $d_2 = \frac{\Sigma| f_i - \hat{f}_i |}{\Sigma| f_i |}$ *mean absolute percentage deviation (MAPD)*

(3) $d_3 = \sqrt{\frac{\Sigma(f_i - \hat{f}_i)^2}{n}}$ *mean square deviation (MSD)*

(4) $d_3 = \sqrt{\frac{\Sigma(f_i - \hat{f}_i)^2}{\Sigma(f_i)^2}}$ *mean square percentage deviation (MSPD)*

For example, suppose a random sample of length 1000 has been selected from a Poison distribution with a mean (lambda) of 3:

```
x<-rpois(n=1000,lambda=3)
```

The analogic method can be used to estimate the population mean:

```
lambdaest<-mean(x)
```

A table with empirical frequencies can now be generated:

```
table.observed<-table(x)
table.observed
x
  0   1   2   3   4   5   6   7   8   9  10
 45 168 203 243 162 105  46  22   3   2   1
```

(Note that the top row indicates the x values and the bottom row shows the corresponding frequencies).

We will now create a vector whose components are the observed frequencies (the second row):

```
freq.observed<-vector()
for(i in 1: length(table.observed)) freq.observed[i]<-table.observed[[i]]
freq.observed
 [1]  45 168 203 243 162 105  46  22   3   2   1
```

The expected (theoretical) frequencies can now be calculated as follows:

```
freq.expected<-(dpois(0:max(x),lambda=lambdaest)*1000)
freq.expected
[1]  51.92265 153.58722 227.15550 223.97532 165.62975  97.98656  48.30737
[8]  20.41331   7.54782   2.48071   0.73379
```

The mean absolute deviation (formula 1) can now be calculated as follows:

```
MeanAD<-mean(abs(freq.observed-trunc(freq.expected)))
MeanAD
[1] 7.727273
```

The mean absolute percentage deviation MAPD (formula 2) is calculated as follows:

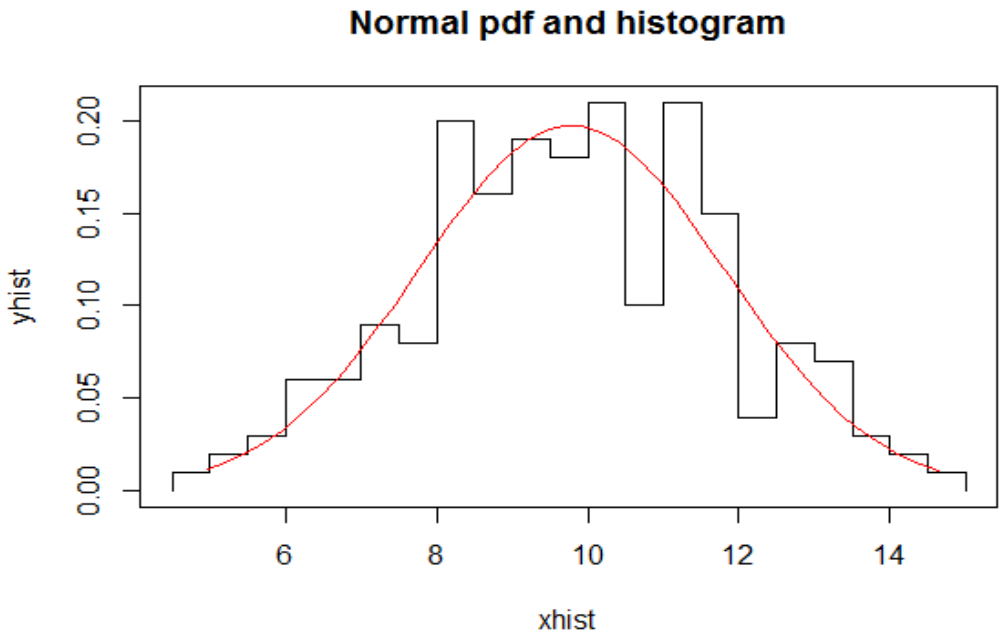
```
MAPD<-sum(abs(freq.observed-trunc(freq.expected)))/sum(abs(freq.observed))
paste(100*MAPD,"%")
[1] "8.5 %"
```

For continuous distributions, the goodness of fit can also be examined by using a graphical method. For example, suppose that **x.norm** is a random sample of size 200 obtained from a normal population with a mean of 10 and a standard deviation of 2:

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

We can examine the goodness of fit by drawing a histogram and a PDF together:

```
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="Histogram & PDF")
lines(xfit,yfit,col="red")
```



Goodness of Fit Tests:

Goodness of fit tests are used to determine whether it is reasonable to assume that a sample data belongs to a specified distribution. The procedure is that of hypothesis testing where the Null and Alternative Hypotheses are stated as follows:

Null	H_0: Sample data come from the specified distribution
Alternative	H_a: Sample data do not come from the specified distribution

The goodness of fit tests are *distribution free*, meaning they do not depend on a particular probability density type. The three most important tests of goodness of fit are:

1. ***The Chi-squared*** (χ^2) test can be applied to both discrete and continuous distributions
2. ***Kolmogorov-Smirnov*** test can be applied to continuous distributions[2]
3. ***Anderson-Darling*** test can be applied to continuous distributions

[2] Many analysts believe that the Kolmogorov-Smirnov test can be applied to both discrete and continuous distributions.

The Chi-squared (χ^2) Test:
The chi-square test is the oldest goodness of fit test dating back to Karl Pearson (1900). This test is basically a formal comparison of a histogram with the fitted density, and can be applied to any univariate distribution for which we can calculate the cumulative distribution function. An important feature of the chi-square goodness of fit test is that it is applied to data that have been put into bins (classes). An important restriction of this test is that it requires a sufficient size sample for the test to work.

Suppose that a sample data have been put into *K* bins(classes). Then the ***Chi-squared test statistic*** $\boldsymbol{\chi^2}$is calculated by the following formula:

$$\boldsymbol{\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}},$$

where $\boldsymbol{O_i}$ and $\boldsymbol{E_i}$ respectively represent the observed and the expected (theoretical) frequencies of the i-th bin. The degrees of freedom (*df*) of the $\boldsymbol{\chi^2}$ test statistic is:

$$\boldsymbol{df = K - p - 1},$$

where $\boldsymbol{K}$ is the number of bins and $\boldsymbol{p}$ is the number of parameters estimated from the sample. For example, for the case of the normal distribution p is equal to 2 since two parameters (mean and standard deviation) are estimated from a given sample.

Another important feature of the Chi-squared test is that it is always a right-tailed test; that is:

$$\boldsymbol{P.value = P(X^2 \geq \chi^2)}$$

The above P-value is compared with a prescribed level of significance ($\boldsymbol{\alpha}$) – such as 0.05- and the null hypothesis is rejected if $\boldsymbol{P.value \leq \alpha}$.

In R, there are a number of different ways to perform a Chi-squared test.

1. We may manually calculate the χ^2 *test statistic*, and then use the formula below to calculate the P-value:

```
pvalue<-1-pchisq(χ², df)
```

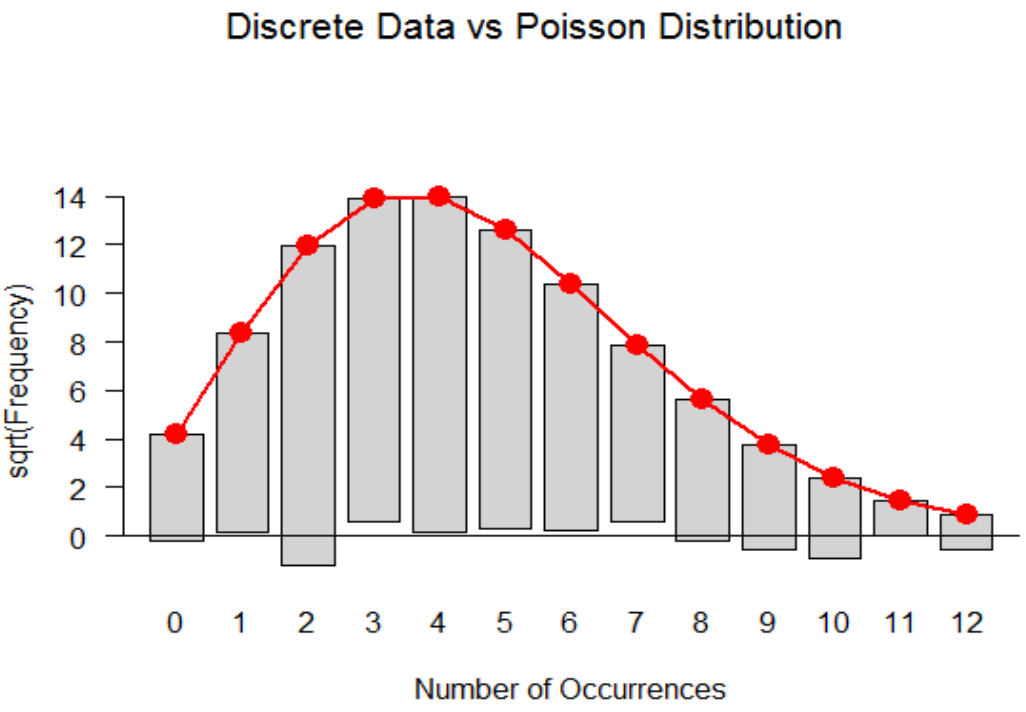
2. For discrete distributions, we can use the **goodfit()** function included in the **vcd** package. For example, suppose a sample **x** has been obtained from a Poisson distribution:

```
x<-rpois(1000,4)
library(vcd)
gf<-goodfit(x,type= "poisson",method= "MinChisq")
summary(gf)

 Goodness-of-fit test for poisson distribution

       X^2      df  P(> X^2)
Pearson 17.73987 11 0.08781614


plot(gf,main="Discrete Data vs Poisson Distribution")
```



3. For continuous variables, there are two cases:
   (i) The distribution parameters are estimated by sample data:
   For example, suppose **g** is a sample of 200 measurements from a Gamma distribution:

```
g<-rgamma(200,rate=0.5,shape=3.5)
```

The rate and the shape parameters are estimated by using the sample data (along with the Maximum Likelihood Principle)

```
rate.est<-mean(g)/var(g)
shape.est<-mean(g)^2/var(g)
```

We will next divide the sample data into classes (bins). To do this, we must first calculate the range of the sample data:

```
max(g)-min(g)
[1] 22.23497
```

Therefore, we can decide on 5 bins, and each bin having a width of 5 units:

```
g.cut<-cut(g,breaks=c(0,5,10,15,20,25))

table(g.cut)
g.cut
(0,5]  (5,10] (10,15] (15,20] (20,25]
76      87      32       4       1
```

We can now calculate the expected frequencies of the Gamma distribution:

```
        (pgamma(5,shape=shape.est,rate=rate.est)-pgamma(0,shape=shape.
 est,rate=rate.est))*200
 [1] 72.09209
        (pgamma(10,shape=shape.est,rate=rate.est)-pgamma(5,shape=shape
.est,rate=rate.est))*200
 [1] 94.43609
```

```
        (pgamma(15,shape=shape.est,rate=rate.est)-pgamma(10,shape=shap
e.est,rate=rate.est))*200
  [1] 27.73208
        (pgamma(20,shape=shape.est,rate=rate.est)-pgamma(15,shape=shap
e.est,rate=rate.est))*200
  [1] 4.948529
        (pgamma(25,shape=shape.est,rate=rate.est)-pgamma(20,shape=shap
e.est,rate=rate.est))*200
  [1] 0.6955848
```

The expected and observed frequencies can now be formed as vectors:

```
exp.freq<-c(72,94,28,5,1)

obs.freq<-vector()
for(i in 1:5) obs.freq[i]<- table(g.cut)[[i]]
```

The Chi-squared test statistic and the corresponding Pvalue can now be calcula ted:

```
test.statistic<-sum(((obs.freq-exp.freq)^2)/exp.freq)
df<-5-2-1
Pvalue<-1-pchisq(test.statistic,df)
Pvalue
[1] 0.4688541
```

(ii) The distribution parameters are known. In this case, we may use the method described above in (i), with the difference that $\boldsymbol{df = 5 - 0 - 1 = 4}$; or we may use the **chisq.test()** function as described below:

Suppose that we are given that the shape and the rate parameters of the Gamma distribution are 3.5 and 0.5 respectively. The expected relative frequencies are then calculated according to the above parameters and are put into a vector called **exp.rel.freq**:

```
exp.rel.freq<-c((pgamma(5,shape=3.5,rate=0.5)-pgamma(0,shape=3.5,rate=0.5)),
      (pgamma(10,shape=3.5,rate=0.5)-pgamma(5,shape=3.5,rate=0.5)),
      (pgamma(15,shape=3.5,rate=0.5)-pgamma(10,shape=3.5,rate=0.5)),
      (pgamma(20,shape=3.5,rate=0.5)-pgamma(15,shape=3.5,rate=0.5)),
      (1-pgamma(20,shape=3.5,rate=0.5)))
```

We can then perform the **chisq.test()**:

```
chisq.test(x=obs.freq,p=exp.rel.freq)

Chi-squared test for given probabilities


data:  obs.freq
X-squared = 2.3001, df = 4, p-value = 0.6808
```

Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov test is used to test whether a sample has come from a population with a specific distribution. It can be applied to both for discrete and continuous binned data (even if some analysts do not agree on its applicability to discrete distributions).

The K-S test is based on a comparison between the empirical cumulative distribution function (ECDF) and the theoretical cumulative function $F(x)$. Given an ordered sample data (sorted in the ascending order), the empirical cumulative distribution $F_n(x_i)$ at each point x_i is defined by:

$$F_n(x_i) = \frac{N(i)}{n},$$

where n is the sample size and $N(i)$ is the number of data points less than x_i. The KS test statistic is:

$$D_n = \max \mid F(x_i) - F_n(x_i) \mid$$

The hypothesis regarding the distributional form is rejected if the test statistic, D_n, is greater than the critical value, or equivalently, 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.

In R we can perform Kolmogorov-Smirnov test using the function **ks.test()**.
Suppose that **w** is a sample belonging from a Weibull pdf with known parameters (shape=2 and scale=1):

```
w<-rweibull(n=200,shape=2,scale=1)
```

The KS test can be applied as follows. Note that KS test is a two-tailed test.

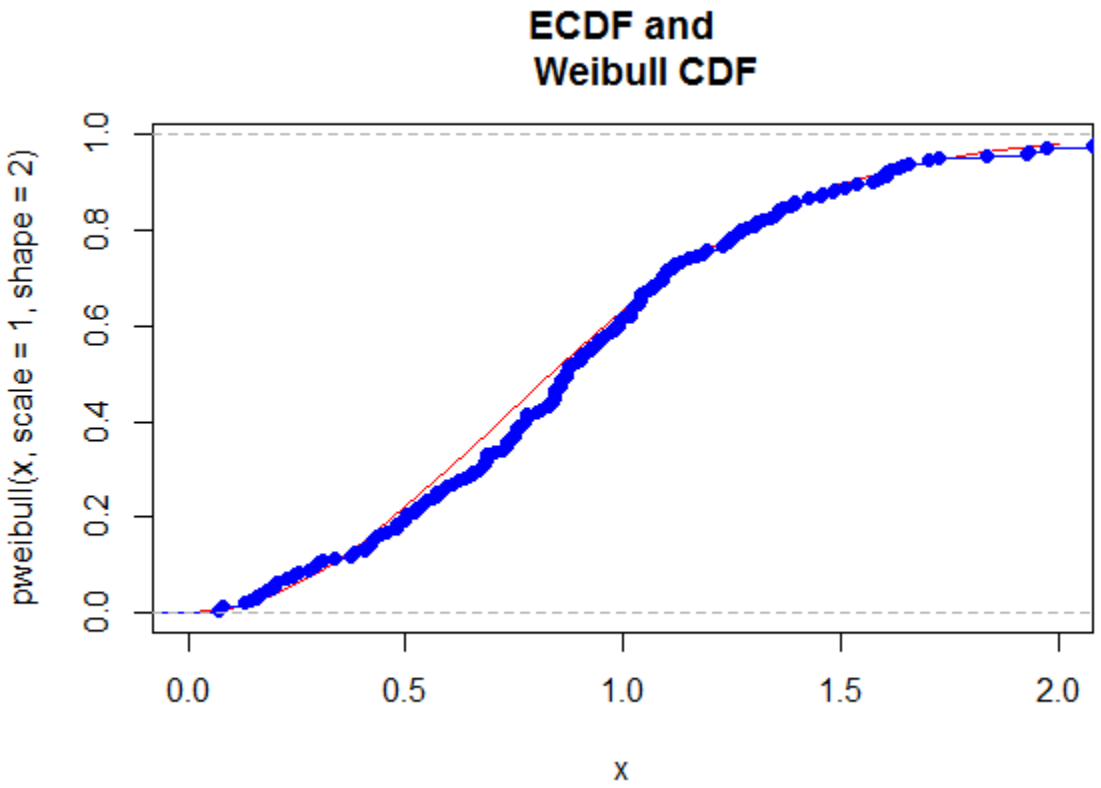
```
ks.test(w,"pweibull", shape=2,scale=1)

One-sample Kolmogorov-Smirnov test

data:  w
D = 0.072059, p-value = 0.2501
alternative hypothesis: two-sided
```

Below, we plot the sample ECDF and the theoretical CDF in the same graph.

```
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(w),col="blue",add=TRUE)
```



References

1. Devore, Jay L. & Berk, Kenneth N. (2012). *Modern Mathematical Statistics with Applications*. New York, NY: Springer.
2. Ricci, Vito (2005). *Fitting Distributions with R*.
3. R Development Core Team (2004). R: *A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL: http://www.r-project.org.
4. Seier, E. (2005) *Testing for normality and Comparison of tests for univariate normality* [2005-02-18]