Initiation to R software Session III

Pierre Michel

Master AMSE 1st year, 2019

Import/Export text files

Import text files

To load text files stored in your computer, use the function `read.table()`.

`read.table(file, sep, header)`

- `file`: the name of the file (character)
- `sep`: separator used in file (" " by default)
- `header`: `TRUE` if file contains columns names, `FALSE` by default

```
tab = read.table("../data/age_gender.txt", header = T)
head(tab, 3)
```

```
##   age gender
## 1  28      F
## 2  36      H
## 3  45      F
```

Import text files: variants of `read.table()`

- `file.choose()`: choose a file through the GUI.
- `read.csv()`: read CSV files.
- `read.delim()`: read delimited text files.
- `read.fwf()`: read fixed-width-formatted files.

R can read files in specific software formats (Excel, SAS, SPSS, Stata) using the functions from the packages `foreign`.

```r
# Try this
install.packages("foreign")
library(foreign)
?read.dta
```

Export text files

To export a `data.frame` into text files on your working directory, use the function `write.table()`.

`write.table(x, file, append, col.names, row.names)`

- `x`: `data.frame`
- `file`: name of the file in which to write
- `append`: if `TRUE`, add to an (eventually existing) file, if `FALSE`, overwrite the existing file (default)
- `col.names` and `row.names`: if `TRUE`, write the columns/rows names

```
write.table(tab, "age_gender.txt", row.names = T,
            col.names = T)
```

`write()` is the same as `write.table()` with less options.

Save/Load R objects

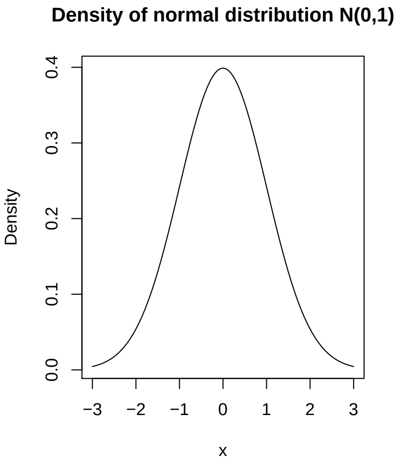
R objects can be saved in both ascii and binary formats:

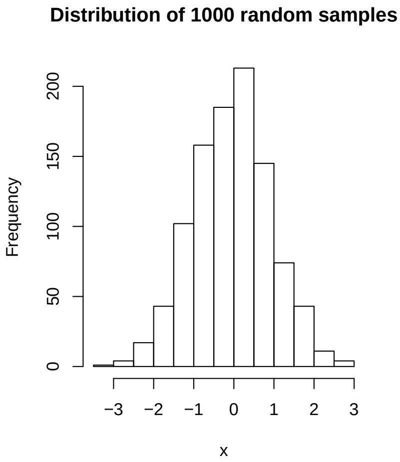
- `dump()`: save R objects in ascii format
- `source()`: load R objects saved with `dump()`
- `save()`: save R objects in binary format
- `load()`: load R objects saved with `save()`

```
# Try this
dump(ls(), file = "objects.txt")
source("objects.txt")
```

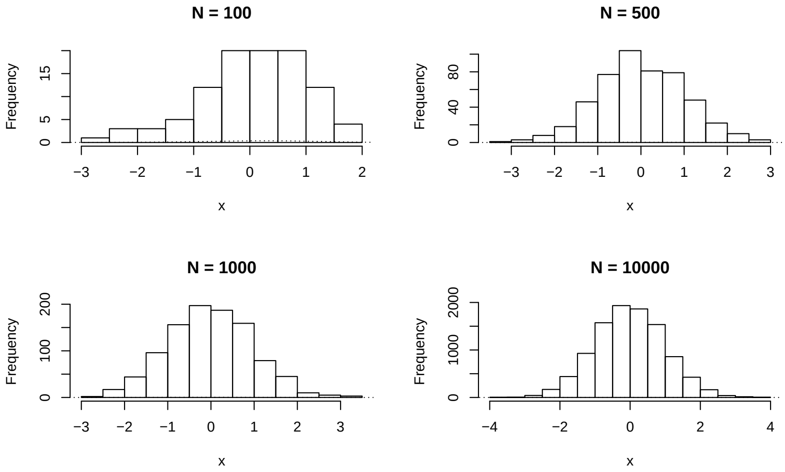
Probability distributions and sample simulation

Probability distribution and sample simulation





Probability distribution and sample simulation



Usual probability distributions

Distributions for **continuous** variables:

- **Normal** distribution: $X \rightsquigarrow N(\mu, \sigma)$: $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}, \forall x \in \mathbb{R}$.
- **Uniform** distribution: $X \rightsquigarrow U(a, b)$: if $x \in [a, b]$, $f(x) = \frac{1}{b-a}$, elsewhere $f(x) = 0$.

Distributions for **discrete** variables:

- **Poisson** distribution: $X \rightsquigarrow P(\lambda)$: $P(X = x) = e^{-\lambda} \frac{\lambda^x}{x!}, \forall x \in \mathbb{N}$.
- **Binomial** distribution: $X \rightsquigarrow B(n, p)$: $P(X = x) = C_n^x p^x (1 - p)^{n-x}, \forall x \in [0, n]$.

Probability distributions in R

R can evaluate probabilistic quantitites following usual probability distributions.

Usual distributions: `*norm()`, `*binom()`, `*chisq()`, `*unif()`, `*pois()`, `*t()`, `*exp()`, ...

* should be replaced by one of the following:

- `p` distribution function, $F(x) = P(X <= x)$, use `pnorm()`, `pbinom()`, `pt()`, ...
- `d` density $P(X = x)$ or $f(x)$, use `dnorm()`, `dt()`, `dpois()`, ...
- `q` quantiles of order q, $\text{argmin}_x\{P(X <= x) > q\}$, use `qnorm()`, `qbinom()`, `qchisq()`, ...
- `r` sample simulation, use `runif()`, `rpois()`, ...

Probability distributions examples

```r
dbinom(3, 10, 0.2)
```

```
## [1] 0.2013266
```

```r
rbinom(10, 10, 0.2)
```

```
##  [1] 0 3 2 3 3 1 3 2 2 3
```

```r
pbinom(1, 10, 0.2)
```

```
## [1] 0.3758096
```

Probability distributions examples

```r
pbinom(2, 10, 0.2)
```

```
## [1] 0.6777995
```

```r
qbinom(0.5, 10, 0.2)
```

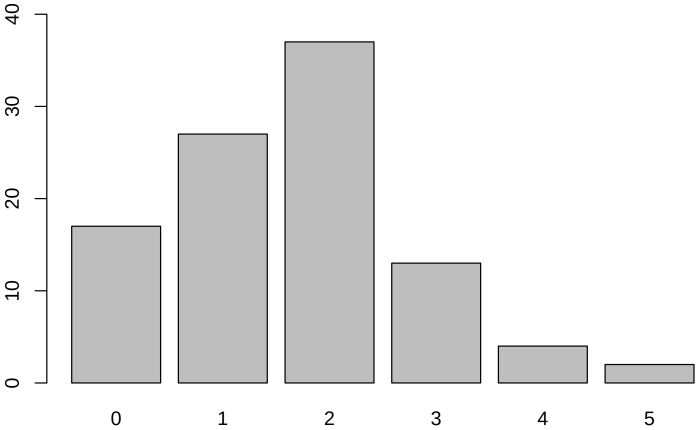
```
## [1] 2
```

```r
qchisq(0.1, 8)
```

```
## [1] 3.489539
```

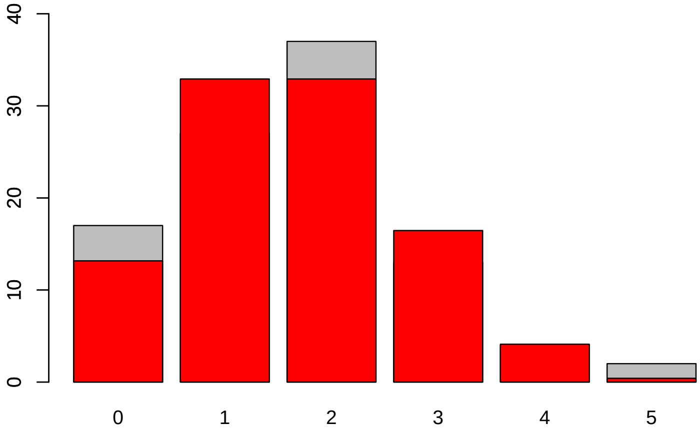
Probability distributions examples

```
x=rbinom(100, 5, 1/3)
par(mfrow = c(1,1))
barplot(table(x), ylim = c(0, 40))
```



Probability distributions examples

```
# add a barplot to an existing barplot
barplot(table(x), ylim = c(0, 40))
barplot(dbinom(0:5, 5, 1/3)*100, add=T, col=2)
```



Descriptive statistics

Characteristics of a statistical serie

Let (x, y) be a pair of statistical series (`vector` or `data.frame`).

- `table(x)` returns the value counts of x (for x discrete or character).
- `summary(x)` returns a summary of descriptives statistics of x (mninimum, 1st quartile, mean, median, 3rd quartile, maximum), for x numeric.
- `mean(x)`, `median(x)`, `var(x)`, `sd(x)` for mean, median, variance and standard-deviation (x numeric).
- `quantile(x, probs)` returns the quantiles of x (numeric) corresponding to parameter vector of probabilities (% of the population)/ Returns quartiles by default.
- `cor(x,y)`, `cov(x,y)` return the correlation/covariance matrix between x and y (numeric).

Characteristics of a statistical serie: examples

```r
# Try this
age = c(18, 15, 12, 16, 20, 17)
weight = c(55, 57, 46, 54, 60,57)
name = c("a", "b", "c", "a")
table(weight); table(name)
summary(weight); summary(name)
mean(weight)
quantile(age)
quantile(weight)
cor(age, weight)
quantile(weight, probs = 0.25)
quantile(age, probs = c(0.1, 0.4))
```

Visualization of a statistical serie

- `hist(x)` plot the histogram of x.
- `density(x)` computes the kernel density estimator of x..
- `ecdf(x)` computes the empirical distribution function of x.
- `barplot(x)` bar chart of x (discrete).
- `stem(x)` tree of values of x (discrete).
- `boxplot(x)` boxplot of values of x (discrete).
- `qqnorm(x)` plot the quantiles of x in function of the quantiles of the normal distribution.
- `qqplot(x,y)` plot the quantiles of x in function of the quantiles of y.
- `plot(x)` plot the values of x.
- `plot(x,y)` scatter plot of coordinates (x, y).

Examples: estimation of the distribution function using the empirical cumulative distribution function (`ecdf()`)

The **empirical distribution function** of a series of observations $x_1, ..., x_n$ is the **step function** between points $(x_i, \frac{i}{n})$, it is defined as follows:

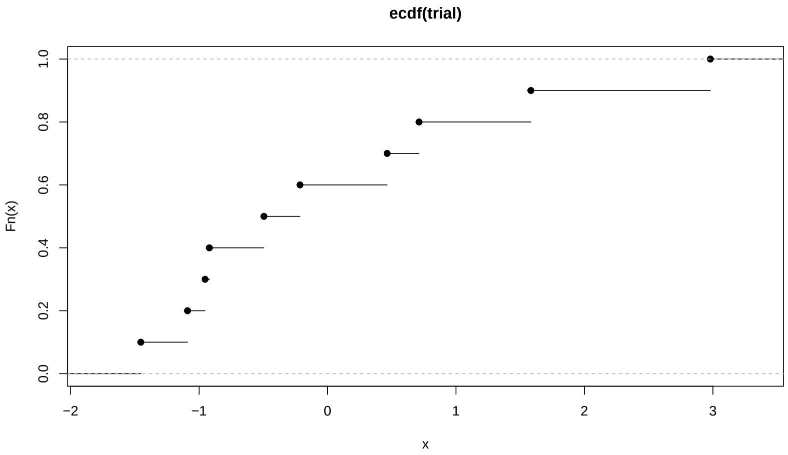
$$F_n(x) = \frac{\#\{i : x_i <= x\}}{n} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{x_i <= x\}}$$

If $x_1, ..., x_n$ are realizations of random variable X, $F_n(x)$ is an estimation of the distribution function of X: $F(x) = P(X <= x)$.

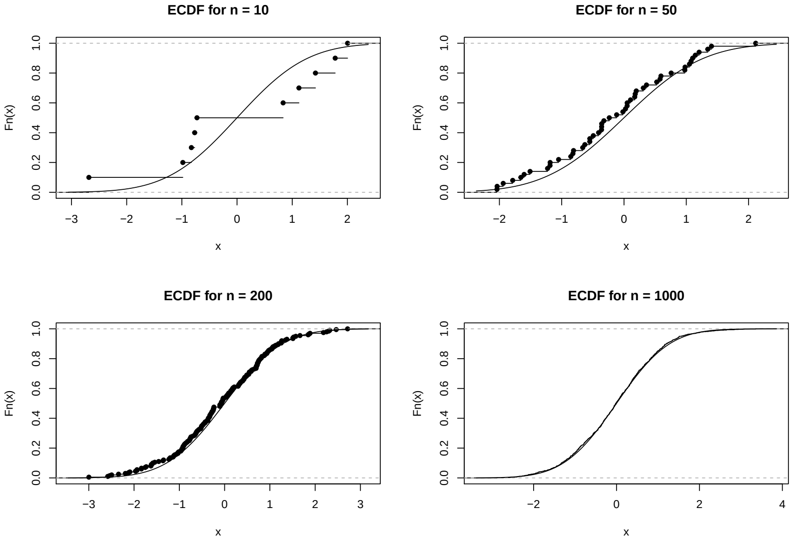
$\forall x, F_n(x) \to F$ when $n \to +\infty$.

Examples: estimation of the distribution function of a probability distribution

```
trial = rnorm(10); plot(ecdf(trial))
```



Examples: estimation of the distribution function of a probability distribution



Examples: estimation of a discrete probability distribution

For a discrete series $x_1, ..., x_n$, consider the proportion of observations that take each value x of the series (equivalent to a bar plot).

$$p_n(x) = \frac{\#\{i : x_i = x\}}{n} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{x_i = x\}}$$

If $x_1, ..., x_n$ are realizations of a discrete random variable X, $p_n(x)$ is an approximation of the probability distribution of X, $P(X = x)$.

The bar plot of the series estimates the graph of the probability distribution of X. Greater is n, better is the estimation.

Examples: estimation of a discrete probability distribution

```r
trial = rbinom(10, 10, 0.3);
t = table(trial); t
```

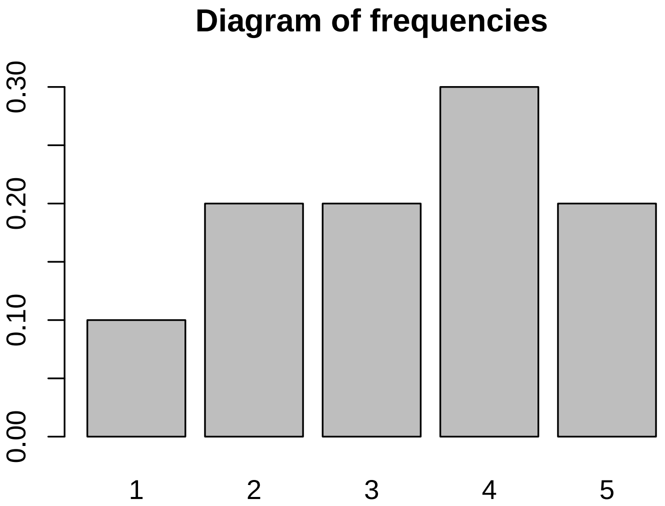
```
## trial
## 1 2 3 4 5
## 1 2 2 3 2
```

```r
stem(trial)
```

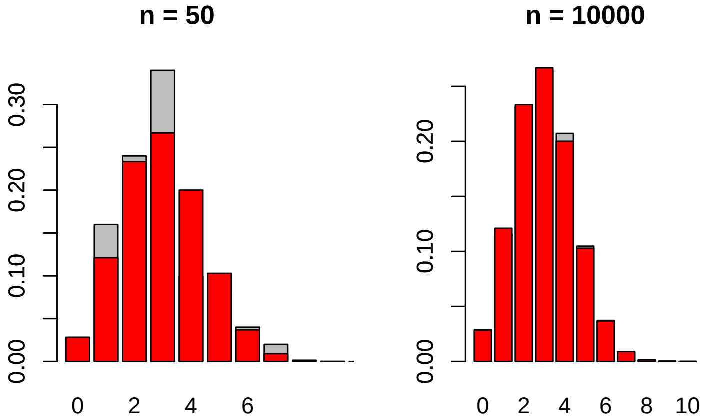
```
##
##   The decimal point is at the |
##
##   1 | 0
##   2 | 00
##   3 | 00
##   4 | 000
##   5 | 00
```

Examples: estimation of a discrete probability distribution

```
par(mfrow = c(1,1))
barplot(t/length(trial), main = "Diagram of frequencies")
```



Examples: estimation of a discrete probability distribution



Examples: density estimation of a continuous probability distribution

The **density** f of a random variable X can be estimated with a **histogram**: let I be the interval of a series' values and $(I_j)_{j \leq k}$ a partition of I in k classes. The histogram based on this partition is the step function which is the proportion of observations in each class, normalized by class amplitude. Class amplitudes l_j should be adjusted to better match with the real distribution of observations.

$$\hat{f}_n(x) = \sum_{j=1}^{k} \left(\frac{\#\{i : x_i \in I_j\}}{n l_j}\right) 1_{\{x \in I_j\}}$$

In general, classes have the same amplitude, determined by the number of classes k (you can also use `cut()`).

Greater is n, more $\hat{f}_n(x)$ is similar to f.

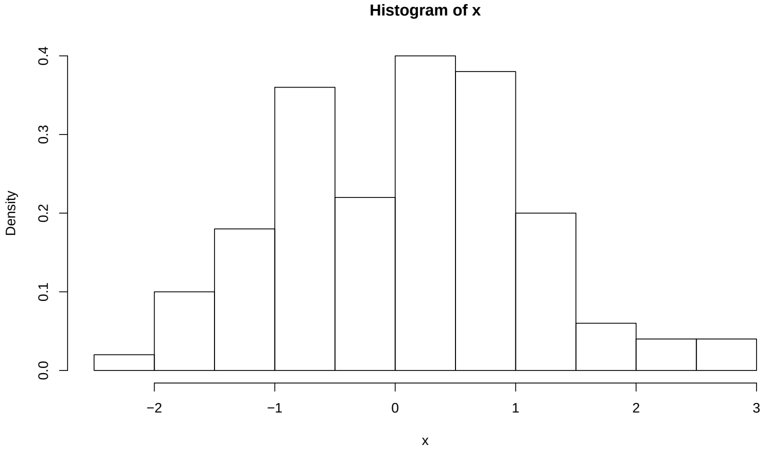
Examples: density estimation of a continuous probability distribution

```
hist(x, breaks, prob, right, col, main, xlab, ylab,
...)
```

- `x` is a statistical series
- `breaks` is a vector of class breakpoints, or a number corresponding to $k+1$, or a function to compute the number of classes. Default uses the Sturge rule ($k = 1 + 1.322 \log_{10} n$)
- `prob` returns the histogram of counts if `prob = FALSE`, returns the histogram of relative counts (frequencies) if `prob = TRUE`. Default is `FALSE`.
- `right` is `TRUE` if the classes of the histogram are right-closed and left-opened, default is `FALSE`
- `col` is the color
- `main`, `xlab` and `ylab` are the main title, x-axis title and y-axis title

Examples: density estimation of a continuous probability distribution

```
x = rnorm(100); hist(x, prob=T, breaks=12)
t = hist(x, prob = T, breaks = 12)
```



Examples: density estimation of a continuous probability distribution

```r
names(t)
```

```
## [1] "breaks"   "counts"   "density"  "mids"     "xname"    "equidist"
```

```r
t$breaks
```

```
##  [1] -2.5 -2.0 -1.5 -1.0 -0.5  0.0  0.5  1.0  1.5  2.0  2.5  3.0
```

```r
t$density
```

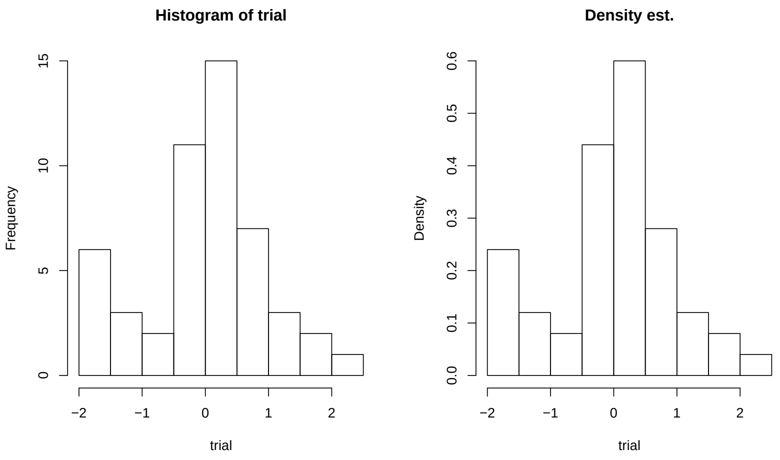
```
##  [1] 0.02 0.10 0.18 0.36 0.22 0.40 0.38 0.20 0.06 0.04 0.04
```

```r
t$counts
```

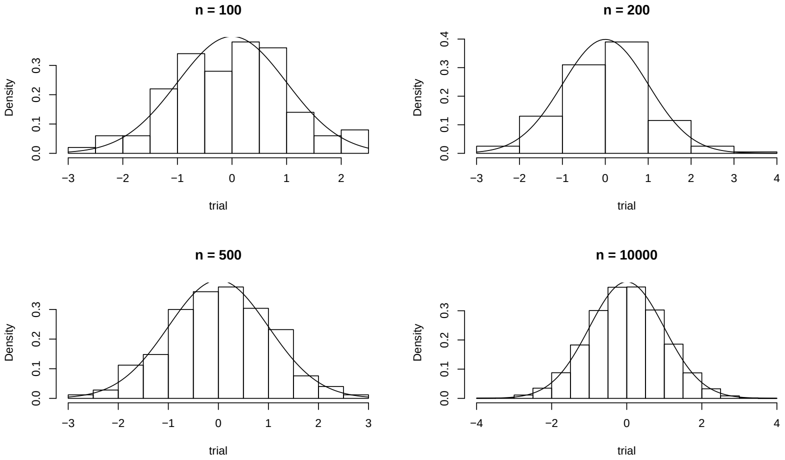
```
##  [1]  1  5  9 18 11 20 19 10  3  2  2
```

Examples: density estimation of a continuous probability distribution

```
trial = rnorm(50); par(mfrow = c(1,2))
hist(trial); hist(trial, prob=T, main="Density est.")
```



Examples: density estimation of a continuous probability distribution



Examples: density estimation of a continuous probability distribution

The **density** f of a random variable X can be estimated with a **kernel estimator**: the "derivative" of the empirical cumulative distribution function is a classical estimator of the distribution function. $\forall x \in I$, and $\epsilon > 0$ (a small value), the "derivative" is:

$$\frac{\hat{F}_n(x+\epsilon) - \hat{F}_n(x-\epsilon)}{2\epsilon} = \frac{\#\{i : x_i \in [-\epsilon x, \epsilon x]\}}{2n\epsilon} = \frac{1}{2n\epsilon}\sum_{i=1}^{n} 1_{-1 \leq \frac{x_i - x}{\epsilon} \leq 1}$$

This estimator assumes that the observations are uniformly drawn around each x_i. One can use a smoother distribution, of density K, with k is the smoothness parameter. We thus consider the following estimator:

$$\hat{f}_n(x) = \frac{1}{n\epsilon}\sum_{i=1}^{n} K\left(\frac{x - x_i}{\epsilon}\right)$$

Examples: density estimation of a continuous probability distribution

```
density(x, bw, kernel, ...)
```

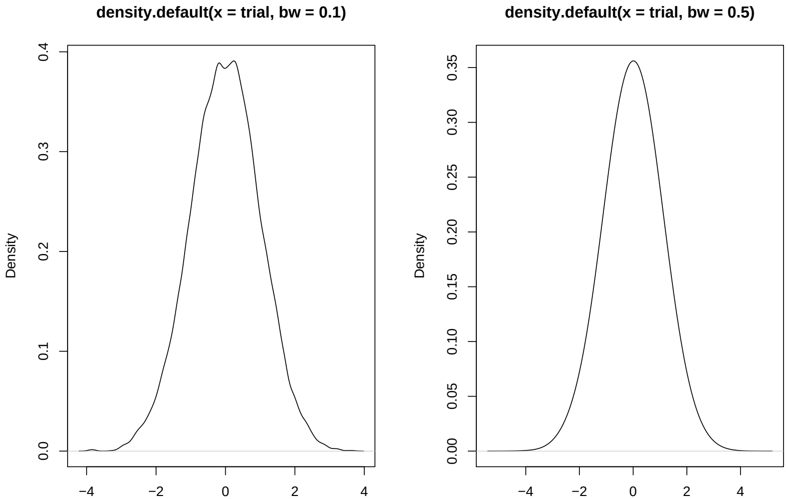
- `x` is a series
- `bw` is the size of the smoothing window. It is defined by a number. Default is automatic.
- `kernel` is a character string giving the kernel used (gaussian, rectangular, ...). Default is gaussian.

The output of `density()` corresponds to descriptive statistics of x and the density estimated on values of x.

Use `plot(density())` to plot the density curve.

Examples: density estimation of a continuous probability distribution

The smoothing window is important, greater it is, smoother is the estimation. Lower it is, noisier is the estimation.



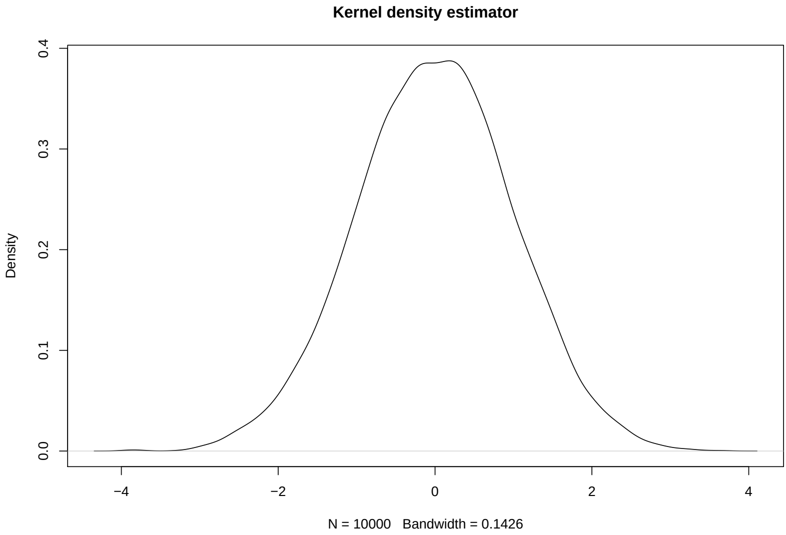
Examples: density estimation

```r
density(trial)
```

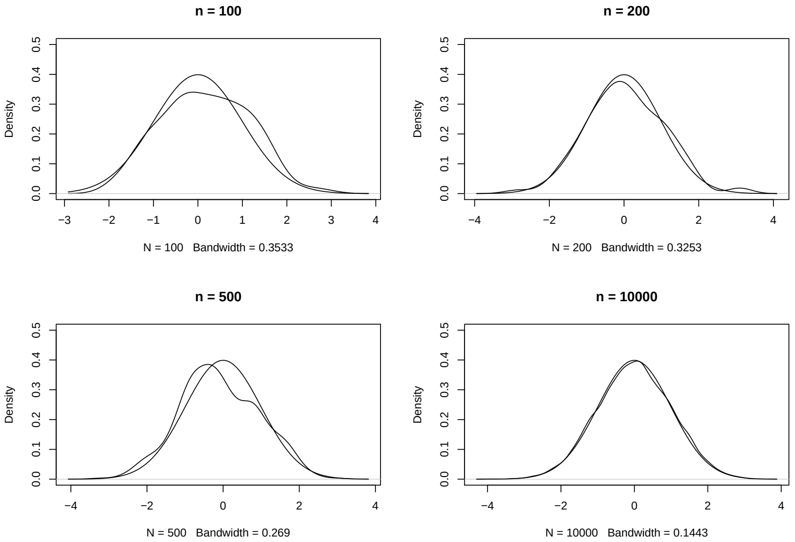
```
##
## Call:
##  density.default(x = trial)
##
## Data: trial (10000 obs.);    Bandwidth 'bw' = 0.1426
##
##        x                 y
##  Min.   :-4.3469   Min.   :0.0000033
##  1st Qu.:-2.2336   1st Qu.:0.0021840
##  Median :-0.1204   Median :0.0439594
##  Mean   :-0.1204   Mean   :0.1181849
##  3rd Qu.: 1.9929   3rd Qu.:0.2254036
##  Max.   : 4.1062   Max.   :0.3875813
```

Examples: density estimation

```
plot(density(trial), main="Kernel density estimator")
```



Examples: density estimation



Examples: characteristics of a series

`boxplot()` plots a **boxplot**, which aims to visualize the characteristics of a series (outliers, symmetry, dispersion).

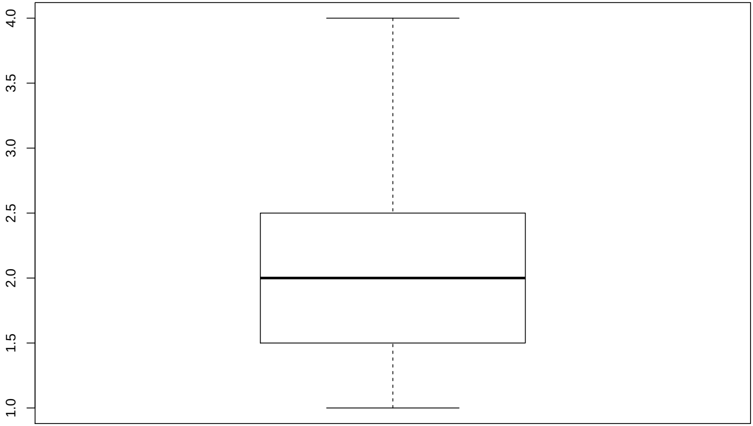
`qqnorm()` aims to compare a series' distribution to a standard gaussian distribution.

```
x=c(1,1,2,2,2,3,4); summary(x)
```

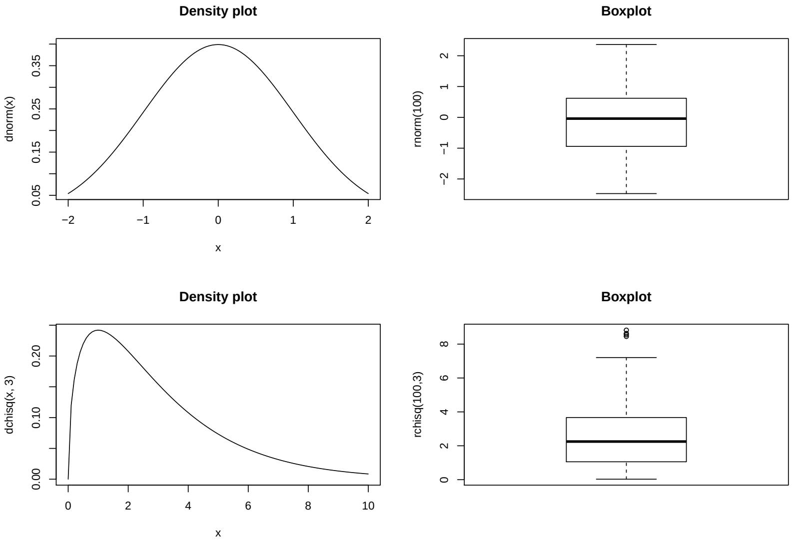
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.500   2.000   2.143   2.500   4.000
```

Examples: characteristics of a series

```
boxplot(x)
```



Examples: characteristics of a series



Examples: characteristics of a series