

Description	Command in R
Package installation	Menu Packages/Installpackage
Loading package	<code>library(package name)</code>
Assignment operator	<code>=</code> or <code><=</code>
Comment	A line starting with <code>#</code>
Square root	<code>sqrt(number)</code>
Sine	<code>sin(argument)</code>
Cosine	<code>cos(argument)</code>
e number	<code>exp(1)</code>
The natural logarithm of x	<code>log(x, base=exp(1))</code>
Vector x	<code>x=c(a, b, ...)</code>
Values: minimum, 1st, 2nd, and 3rd quartile, maximum	<code>quantile(vektor)</code>
i-th element of the vector x	<code>x[i]</code>
Interquartile distance	<code>IQR(vector)</code>
Generating numbers from a given range	<code>:</code>
Generating numbers into the vector x from a given interval and with a certain step	<code>x=seq(start, end, by=step)</code>
Generating numbers into the vector x from a given interval and with a certain length	<code>x=seq(start, end, length=length)</code>
Generating a vector x that repeats another	<code>x=rep(vector, number of repetitions)</code>
Generating a vector x by repeating each element of a different vector	<code>x=rep(vector, each= number of repetitions)</code>
All elements of the vector x except the indicated one	<code>x[-number of element]</code>
All the elements of the vector x satisfying the condition	<code>x[x>2]</code>
Grouping of data	<code>data.frame(vector1, vector2, vector3)</code>
The sum of the vector elements	<code>sum(vector)</code>
Matrix consisting of the elements from a vector with n rows and m columns	<code>matrix(vector, ncol=m, nrow=n)</code>
Matrix transposition	<code>t(matrix)</code>
Determinant of the matrix	<code>det(matrix)</code>
Matrix product	<code>matrix1%*%matrix2</code>
Diagonal matrix / Vector of elements on the diagonal	<code>diag(matrix)</code>
Inverse matrix	<code>solve(matrix)</code>
Eigenvalues and eigenvectors of matrix	<code>eigen(matrix)</code>
Creating a matrix from vectors (columnar)	<code>cbind(vector1, vector2,...)</code>
Create a matrix from vectors (in line)	<code>rbind(vector1, vector2,...)</code>
Removal of the i-th row from the matrix M	<code>M[-i,]</code>
Removal of the j-th column from the matrix M	<code>M[, -j]</code>
Conditional statement	<pre>if (condition) { do if true } else{ do if false }</pre>
FOR loop	<pre>for {i in variable that the loop is executing on } { expressions }</pre>

WHILE loop	<code>while {condition} { expressions }</code>
Plot	<code>curve</code>
Plot	<code>plot (x,y)</code>
- marker symbol	<code>pch=marker symbol (from 0 to 25)</code>
- X axis range (Y axis)	<code>xlim= (ylim=)</code>
- X axis title (Y axis)	<code>Xlab= (ylab=)</code>
- plot title	<code>main=</code>
- line thickness	<code>lwd=</code>
- color of the line	<code>col="color"</code>
Loading csv / txt files	<code>data=read.csv("Name file.csv", dec=".") data=read.csv2("Name file.csv", dec=".") data=read.table("Name file.txt", dec=".")</code>
Saving the results to files	<code>write.table(dane, file="Name file.txt") write.csv(dane, file="Name file.csv") write.csv2(dane, file="Name file.csv")</code>
Histogram	<code>hist()</code>
- sum of the column heights is 1	<code>prob="true"</code>
- no gap to the i axis	<code>yaxs="i"</code>
Boxplot	<code>boxplot()</code>
Variance	<code>var(vector)</code>
Standard deviation	<code>sd(vector)</code>
Correlation	<code>cor(vector1, vector2), cor(matrix)</code>
Covariance	<code>Cov(vector1, vector2)</code>
Confidence interval for the mean with unknown sigma	<code>t.test(vector, conf.level=confidence level)</code>
Confidence interval for the mean with known sigma	<code>z.test(vector, sigma.x=sigma, conf.level=confidence level)</code>
Confidence interval for the mean (large sample)	<code>zsum.test(sample mean, standard deviation, sample size, conf.level = confidence level)</code>
Test hypothesis about mean mu (unknown sigma)	<code>t.test(data, mu=mu, alternative='greater/less/two.sided')</code>
Test hypothesis about mean mu (known sigma)	<code>library(PASWR) or library(BSDA) z.test(data,mu=mu,sigma.x=sigma)</code>
Test hypothesis about mean mu (large sample)	<code>library(PASWR) or library(BSDA) zsum.test(mean, standard deviation, sample size, mu=mu, conf.level= 1- significance level, alternative=' greater/less/two.sided')</code>
Test hypothesis about variance	<code>library(TeachingDemos) sigma.test(data, sigma=sigma, alternative=" greater/less/two.sided")</code>
Test for two variances	<code>var.test(data1, data2)</code>
Test for two means (homogeneity of variance)	<code>t.test(data1, data2)</code>

Distribution	CDF	PDF	Quantile	Generator	Remarks
Binomial	pbinom	dbinom	qbinom	rbinom	size, prob
Poisson	ppois	dpois	qpois	rpois	lambda
Negative binomial	pnbinom	dnbinom	qnbinom	rnbinom	
Geometric	pgeom	dgeom	qgeom	rgeom	prob
Hypergeometric	phyper	dhyper	qhyper	rhyper	
Polynomial		dmultinom		rmultinom	
Uniform	punif	dunif	qunif	runif	min, max
Beta	pbeta	dbeta	qbeta	rbeta	
Exponential	pexp	dexp	qexp	rexp	rate
Gamma	pgamma	dgamma	qgamma	rgamma	
Normal	pnorm	dnorm	qnorm	rnorm	mean,sd
Weibull	pweibull	dweibull	qweibull	rweibull	
Chi-square	pchisq	dchisq	qchisq	rchisq	
T	pt	dt	qt	rt	df
F	pf	df	qf	rf	