R NOTE

R features

Basic Operation For R

Run

Debugging in R

Prepare

File directory settings

Load resources

Batch file and result redirection

Data processing

Input and output (read in and output data, files)

Data view

Drawing

Edit

Apply functions family

Plyr library

String processing

Control flow

Mathematical Statistics

Normal distribution

Distribution Function

Mean variance of a normal population

Independence test (Null hypothesis H0: X and Y are independent)

Rank

Significant difference test (analysis of variance, null hypothesis: same, correlation)

R language features

- Case sensitive
- Generally, numbers, letters,. And _ are all allowed. However, a name must start with. Or a letter, and if it starts with ., the second character is not allowed to be a number.
- Basic commands are either expressions or assignments.
- Basic commands can be combined with curly braces ({ and }) to form a compound expression.
- On a line, the sentences from the beginning of the pound sign (#) to the end of the sentence are comments.
- R is a dynamically typed, strongly typed language.
- The basic data types of R include numeric, character, complex, and logical. The object types include vectors, factors, arrays, matrices, data frames, lists, and time series.

Basic Operation For R

run

q() Exit the R program
tab auto completion
ctrl+L clear the console
ESC interrupt the current calculation

Debugging in R

browser() and debug() set breakpoints to proceed, you can browse and view after running
stop('your message here.') stop the program execution when the input parameters are incorrect
cat() view variables

Prepare

File directory settings

setwd (<dir>) set the working file directory
getwd() get the current working file directory
list.files() view the files in the current file directory

Load resources

search() Through the search() function, you can see that 7 core packages are loaded by default when R starts.
setRepositpries() Select the software library, and find the method of installing the package
(.packages()) List current packages
(.packages(all.available=TRUE)) list available packages
install.packages("<package>") installation package
library() and require() load R package (package) to the workspace
data() List the existing data sets that can be obtained (data sets of the base package)
data (<datasets>, package="nls") load the datasets of the nls package into the database

Batch file and result redirection

source("commands.R")	Execute commands.R (store batch commands) script file
cat(<Rcommond>,file="")	can output R commands to external files, and then call the source function for batch processing

do.call(<funcname>,<pars>)	call the function, the first parameter <funcnames> indicates the name, of the function string, the second parameter contains a list of parameters required for the call <pars>
sink(“record.lis")	redirect the subsequent output from the console to the external file record.lis
attach (<datafame>)	link the variables in the data frame <datafame> to the memory for easy data calling
detach()	corresponding to attach(<datafame>), cancel the link of the variable, there is no parameter in detach()!
Note:	Note: both attach() and detach() find the first matching variable name in the default variable search path table from front to back. Therefore, if there is a variable with the same name before, there may be problems! ! !

Data processing

Input and output (read in and output data, files)

assign("x",c(1,2,3)) and x <- c(1,2,3) and c(1,2,3)->x	Vector assignment
read.table("infantry.txt", sep="\t", header=TRUE)	The seq attribute is separated by other characters。 E.g. text files are separated by tab, and the header is set to the name of the table header that already exists in the file
read.csv("targets.csv")	Read in a csv (Comma Seperated Values) file, the attributes are separated by commas
read.csv(url("<link>"))	A combination of read.csv() and url(), read data stored on the Internet
x <- scan(file="")	Enter data manually, and scan can specify the data type of the input variable, which is suitable for large data files
scan("data.dat", what = list("", 0, 0))	what specifies the variable type list

write.table(Data, file="file.txt", row.names = FALSE, quote=FALSE)	output, quote is FALSE, remove the double quotes of string type, write.table(stasum, "stasum.csv" ,row.names = FALSE,col.name=FALSE,sep=",",append=TRUE)
save.image("./data.RData")	Dump the data that was originally active in the computer memory (work space) to the hard disk.
load("./RData")	load *.RData in the directory, load the document-term matrix from the disk to the memory

Data view

- Universal object

class(<object>) and data.class(object) — view the class or type of the object
unclass() — Eliminate the class of the object object

- Basic data type

mode() — view basic data types
length() — view length
as.<data type> — change the data type of the object

- Special attributes

attributes(<object>) — view a list of various attributes of the object
attr(<object>, "name") — access to the attribute named name of the object

- Mixed type

Logical type + numeric type = numeric type
Logical type + character type = character type
Numerical type + character type = character type

ls() and objects()	view the objects (variables) existing in the current workspace
rm(list=ls())	delete all objects in the workspace
methods(x)	View the source code of the x function, some of the built-in functions can be directly seen by entering the name x, some need to call the methods method to view the source code of the function x, there are multiple names, just enter the corresponding name
str()	View the overall information of the data in the data (box) (such as the number of samples, the number of variables, the name of the attribute variable, and the type)

nrow(dataframe)	View the overall information of the data in the data (box) (such as the number of samples, the number of variables, the name of the attribute variable, and the type)
NROW (vector)	view the number of rows of the vector, equal to length(x)
head(dataframe)	View the first 6 rows of data in the data set
tail(dataframe)	View the last 6 rows of data in the data set

Drawing

plot()	draw an image
plot(<vecter_horizontal>, <vector_vertical>, pch=as.integer(<factors>), col, xlab, ylab)	use factors to distinguish the type of image point pch (circle, triangle, cross), col is the color category , Xlab or ylab corresponds to the title of the horizontal and vertical axis
legend(<location="topright">,legend=<vector_labelname>,pch=1:3,cex=1,col)	Legend, <location> is the location (such as the upper right), <vector_labelname> legend category label name, pch is the category id (vector) of the label corresponding to the legend, <cex> adjusts the font scale size, color setting, legend("topright", levels(<factors>), pch=1:length(levels(factors)))
text(X,Y,labels=c(1,2,3),adj=1.2)	Add label, X,Y are the vector corresponding to the coordinate, labels are the label value, adj adjusts the label position
abline(h = <int>, lty=2)	low-level drawing adds a horizontal line h or regression model straight line, vertical line v; lty is 2 means drawing a dashed line
points(x,y)	draw a line with y=a+bx
points(x,y)	low-level drawing, draw a point, the coordinates are the vector x,y
lines (x, y)	low-level drawing, draw a line, the coordinates are the vector x, y
axis(side=1, at=seq(from=0.7, by=1.2, length.out=7), labels=c(...))	drawing axis, low-level drawing, side=2 is the ordinate

barchart()	lattice package needs to summarize the data in advance
barplot(<vector>)	Draw a histogram, vector can increase its name. You can also draw a histogram, which is not the same as the hist() evenly dividing the data. You need to use table() to count the number of samples in each sub-segment and then draw the graph.
mosaicplot (x~y, main, color=T, xlab, ylab)	column correspondence graph
contour(<matrix>)	create contour lines
persp(<matrix>, expand=0.2)	create a 3D image, set the expand expansion value to 0.2, otherwise it is a full-screen expansion
image (volcano)	load raster (matrix) image
par(mfrow=c(1,2),oma,mar)	mfrow sets the graph output window to 1 row and 2 columns, add car package? oma is the distance of all images from the border (bottom, left, top, right), mar is the distance from each image to the border, the default is c(5, 4, 4, 2) + 0.1.
lines(data)	(low-level) draw lines in the original image, data is composed of scattered points (x, y)
rug(jitter(<data>), side=2)	draw lines in the original image, data is composed of scattered points (x, y)
pairs(data)	scatter diagram of each variable in the data frame
coplot(y~x|a+b)	The scatter plot of multiple variables, the scatter plot of y and x under the division of a, b (vector or factor)

earth.count(na.omit(x),number=4,overlap=1/5)	discretization of continuous variable x, converting x into factor type; number sets the number of intervals, overlap sets two intervals close to the boundary Coincidence? Observations in each interval are equal
palette()	The color corresponding to the col value, "black" "red" "green3" "blue" "cyan" "magenta" "yellow" "gray"
scatterplotMatr()	scatter plot matrix, car package
identify (<data>)	interactively click, click on the point in the graph, the row number of the corresponding data will be output, right click to end the interaction
stem(x,scale=1,width=80,atom=1e-08)	stem and leaf map, scale controls the length of the stem and leaf map, 2 means a group of 0~4, and a group of 5~9 Divide the ones digit into two parts, width is the drawing width, atom is the tolerance
boxplot(y~f,notch=TRUE,col=1:3,add=TRUE)	y is data, f is composed of factors, notch is a box plot with cutouts, add=T is superimposed on the previous one picture.
bwplot (<factor> ~ <y>, data, ylab)	box plot of lattice package, draw box plots of y under different factors (conditional plotting, the change of y value under a certain factor value set)
bwplot(size~a1,data,panel=panel.bpplot,prob=seq(.01,.49,by=.01),datadensity=TRUE,ylab='')	The bin plot of the Hmisc package
stripplot(x1~y|x2)	complex box plot of lattice package, there are two factors x1, x2 controlled by y, x2 are arranged in the order from left to right, from bottom to top, the value of x2 at the bottom left is lower small
colors()	List the corresponding color array

qcc ()	qcc package, a quality monitoring chart (P control chart) for monitoring conversion rate indicators, monitoring abnormal points, provided that the binomial distribution is large enough and tends to a normal distribution
mosaic (<tab>, shade=T, legend=T)	draw a three-level contingency table, <tab> is a three-level contingency table or formula, vcd package
curve(sapply(x,<func>),<from>,<to>)	draw a curve graph, from and to set the value range of the abscissa

Edit

curve(sapply(x,<func>),<from>,<to>)	draw a curve graph, from and to set the value range of the abscissa
sample(length(x),<size>,replace=F)	sample, generate a new vector of size <size> in random order of vector x; replace is False for non-repetitive sampling, and True for repeated sampling
Round	rounding. accurate
ceiling()	rounded, biased towards small values
floor()	rounded, biased towards larger values
colnames(Data)[4]="value"	Replace a column name
edit ()	edit the data table
fix()	rm (x, y)-remove objects (variables) x and y
rm (x, y)	remove objects (variables) x and y
na.exclude(<data>)	remove the entire row of missing data
na.omit(<data>)	delete missing data
attr(na.omit(<data>),"na.action")	returns the subscript of NA whose element in vector a

na.fail()	If the vector contains at least 1 NA value, it returns an error; if it does not include any NA, it returns the original vector
merge(x = targets, y = infanty)	merge data frames, x and y are the data frames to be merged, and the same attribute fields will also be merged together
merge(x, y, by = intersect(names(x), names(y)),by.x = by, by.y = by, all = FALSE, all.x = all, all.y = all,sort = TRUE, suffixes = c(".x",".y"),incomparables = NULL, ...)	Description of merge function parameters:
	1 x,y: two data frames used for merging
	2 by, by.x, by.y: Specify which rows to merge the data frame according to, and the default value is the column with the same column name.
	3 all, all.x, all.y: Specify whether the lines of x and y should all be in the output file.
	4 sort: Whether the column specified by by is to be sorted.
	5 suffixes: Specify the suffix of the same column name except by.
	6 incomparables: Specify which units in by are not to be merged.
scale(x, center = TRUE, scale = TRUE)	centralization and standardization, center is centralization, scale is standardization.
cut(x,breaks=c(0,10,30),labels,ordered_result=F)	cut(x,breaks=c(0,10,30),labels,ordered_result=F)——discretization of continuous data, the vector is divided into factor vectors according to the interval of breaks. Labels sets the horizontal label value of the returned factor vector, and the factor vector generated by ordered_result is False has no size meaning, otherwise it has size meaning

Apply functions family

apply(A, MARGIN, FUN,...)	The processing object A is a matrix or an array, MARGIN sets the dimension to be calculated, and FUN is some functions, such as mean, sum
lappy(dataframe, FUN, list(median,sd))	The processing object is a vector, a list or other objects, and the output format is a list list
sapply(dataframe$Filed, FUN)	similar to lapply(), the output format is matrix (or data frame)

lappy(dataframe, FUN, list(median,sd))	The processing object is a vector, a list or other objects, and the output format is a list list
sapply(dataframe$Filed, FUN)	similar to lapply(), the output format is matrix (or data frame)
tapply(X, INDEX, FUN, simplify = TRUE)	processing grouped data, INDEX and X are factors with the same length, simplify is a logical variable (the amount is defaulted to T)
aggregate(x~y+z, data, FUN) and by()	similar to tapply function

Plyr library

ddply(Data, .(user_id, item_id), summarize, liulan=sum(liulan))	an integrated function of split-apply-combine; .(user_id, item_id) is used as a pair of identification IDs (factors) for each row, The "." in front of the data frame name is omitted; summrize is a function fun; liulan is a variable, and the finally generated data frame has only three columns: user_id, item_id, and liulan. For details, see the example R language tool ddply
transform(x, y)	transform the columns of x and y into a data frame.
reshape library (reshape2)	
melt (data, id.vars)	the converted data melts. Modify the data organization structure, create a data matrix, use id.var as the number of each row, the remaining column data value is only used as a column value, and use the original column name as the classification mark of the new value.
cast(data, userid~itemid,value="rattings",fill=0)	Statistics and conversion data, generate matrix, formula ~ the left side as the row table name, the right side as the list name. Then you can use cor() to calculate the correlation coefficient between each column of data and calculate the distance.
acast and dcast (data, userid~itemid, value.var="rattings")	Same as above, reshape2 package, acast finally generates an array, dcase generates a data frame. See R language advanced 4: data reshape

String processing

nchar()	Get the length of a string, it can get the length of a string, and it also supports string vector operations.

paste("a", "b", sep="")	string gluing, responsible for concatenating several strings and returning them to a single string. The advantage is that even if some processing objects are not character types, they can be automatically converted to character types.
strsplit(A,split='[,.]')	string split, responsible for dividing the string according to a certain form of splitting, it is the inverse operation of paste().
substr(data,start,stop)	string interception, can take out a subset of a given string object, and its parameters are the starting and ending positions of the subset. The subset is the subscript interval from start to stop
grep()	string matching, is responsible for searching for a specific expression in a given string object and returning its position index. The grepl() function is similar, but the "l" behind it means that the return will be a logical value
regexpr (pattern, text)	extract the subscript position of a specific string from the string text
gregexpr()	only query the subscript position of the first specific string that matches
gsub("a",1,<vector>)	string substitution, responsible for searching the specific expression of the string and replacing it with new content.
chartr()	string replacement function
toupper( ), tolower() and casefold()	case conversion functions

Control flow

if—else	branch statement
switch(index,case1,case2,casen)	index indicates to jump to the i-th casei
while	loop statement, by setting loop range
repeat—break	loop statement, infinite loop, jump out of break

Mathematical Statistics

Normal distribution

dnorm (x, mean=0, sd=1, log=FALSE)	probability density function of normal distribution

pnorm(x, mean=0, sd=1)	Returns the distribution function of the normal distribution·
rnorm (n, mean=0.sd=1)	generate a vector of n normal distribution random numbers
qnorm()	Divided into point function
qqnorm (data)	draw a scatter plot of qq
qqline (data)	low-level drawing, use the scatter points of the qq chart to draw lines
qq.plot (<x>, main='')	qq plot to test whether the variable is normally distributed

Distribution function

shapiro.test(data)	normal W test method, when p value is greater than a, it is normal distribution
ks.test(x,y	KS test method of empirical distribution, compare whether the distribution of x and y are the same, y is the data vector compared with x or the name of a certain distribution, ks.test(x, rnorm(length(x), mean(x), sd(x))), or ks.test(x,"pnorm",mean(x),sd(x))
chisq.test(x,y,p)	Pearson goodness-of-fit X2 (chi-square) test, x is the frequency of each interval, p is the theoretical probability that the null hypothesis falls between the cells, the default value means uniform distribution, to To test other distributions, such as normal distribution, first construct the small area and calculate the probability value of each interval.

Mean variance of a normal population

t.test(x,y,alternative=c("two.sided","less","greater"),var.equal=FALSE)	The mean of a single normal population μ or the difference between the mean of two normal populations μ1 -interval estimation of μ2; alternative means alternative hypothesis: two.side (default) is a two-sided test, less means H1: μ<μ0, greater means H1: μ>μ0 one-sided test (μ0 represents the null hypothesis); when var When .equal=TRUE, the two-sample variance is the same, and the default is different

var.test(x,y)	interval estimation of the two-sample variance ratio
Independence test (Null hypothesis H0: X and Y are independent)	
chisq.test(x,correct=FALSE)	chi-square test, x is a matrix, dim(x)=c(2,2), for large samples (frequency greater than 5)
fisher.test()	The unit frequency is less than 5, and the contingency table is 2*2
cor.test(x,y,method=c("pearson","kendall","spearman"))	Correlation test, observe that the p-value is less than 0.05 to be correlated. method selects the correlation test method
Rank	
rank()	rank statistics
cor.test()	rank correlation test: Spearman, Kendall
wilcox.test(x,y=NULL,mu,alternative,paired=FALSE,exact=FALSE,correct=FALSE,conf.int=FALSE)	rank significance test (a sample from the overall test, significant difference Test), Wilcoxon rank sum test (rank sum test of unpaired samples), mu is the parameter to be tested, such as the median, paired logical variables, indicating whether the variables x and y are paired data, and exact says whether the people are accurate Calculate the P value, correct is a logical variable, indicating whether to use continuity correction for the p value, and conf.int is a logical variable, giving the corresponding confidence interval.
uniroot(f, interval=c(1,2))	A function to find the roots of a one-variable equation, f is the equation, interval is the interval for solving the roots, and the return value root is the solution
nlm(f, p)	Solve unconstrained problems, solve the minimum value, f is the minimum objective function, p is the initial value of all parameters, the Newton type algorithm is used to find the minimum, the return value of the function is a list containing extremes The minimum value, the estimated value of the minimum point, the gradient at the minimum point, the Hesse matrix, and the number of iterations required for the solution, etc.

Significant difference test (analysis of variance, null hypothesis: same, correlation)	
mcnemar.test(x,y,correct=FALSE)	Two tests on the same individual to test the significance of the change in the frequency ratio of the two correlation distributions of the two metadata, that is, the null hypothesis is that the correlation distributions are the same. y is an object composed of factors. When x is a matrix, this value is invalid.
binom.test(x,n,p, alternative=c("two.sided","less","greater"),conf. level=0.95)	binomial distribution, sign test (a sample comes from the overall Test, significant difference test
aov(x~f)	Calculate the analysis of variance table, x is the value of the factor level corresponding to (factor) f, use the summary() function to view the information
aov (x~A+B+A:B)	two-factor variance, where A and B in X~A+B are the level factors of different factors (interaction is not considered), and A:B represents the factor generated by the interaction
p.adjust()	P value adjustment function
pairwise.t.test(x,g,p.adjust.method="holm")	Multiple t test, p.adjust.method is the adjustment method of P value, the method is given by p.adjust(), default The value is adjusted according to the Holm method ("holm"). If it is "none", it means that the P value will not be adjusted. For two-factor interaction, g=A: B
shapiro.test(x)	the normal W test of the data
bartlett.test (x~f, data)	Bartlett test, test for homogeneity of variance
kruskal.test(x~f, data)	Kruskal-Wallis rank sum test, non-parametric test method, does not satisfy normal distribution
friedman.test(x, f1, f2, data)	Friedman rank sum test, which does not satisfy normal distribution and homogeneity of variance, f1 is a factor of different levels, and f2 is a factor of the number of trials