Краткий справочник R

Оглавление

Начало работы

http://www.r-project.org. Загрузка R и всего остального.
http://stat.ethz.ch/R-manual/R-patched/library/base/html/Startup.html Старт системы

Способы работы в Windows

Rterm	вводим команды в ответ на запрос системы;
Rgui	графический интерфейс
изменение директории при старте	правая кнопка мыши по иконке R и изменить «свойства», указав директорию при старте.
q(()	Окончание работы
esc	прерывание работы

Установка пакетов

INSTALL package1	Установить пакет *package1*
installed.packages()	Список установленных пакетов
library(package1)	Загрузить пакет *package1*

Рабочая область

setwd("G:/Program Files/R-2.15.0/My_library")	установка рабочей директории
save.image()	сохраняет рабочую область в скрытом файле *.RData*
save.image("myimage.rda")	сохраняет рабочую область в конкретном файле
save(file="file_name",xxx)	сохранить на диск объект *xxx*
load("file_name")	загрузить с диска объект xxx
source("file_name")	загружается с диска файл с именем *file_name*, проверяется синтаксис, создаются объекты, выполняются команды
Например: *source("test.r")*	выполняет команды из файла *test.r*
savehistory(file=) loadhistory(file=) или (altF/altH)	сохранить/загрузить историю команд

Элементарные команды

ls()	Список всех ОБЪЕКТОВ в рабочей области
ls.str()	Перечисляет детали всех объектов
str(myobject)	Список подробностей объекта
myobject	печатает объект
list.files()	Список файлов в директории.
rm(xxx)	удалить объект с именем **xxx**
rm(list=ls(all=TRUE))	удалить ВСЕ объекты из рабочей области
Ctrl+L	Очистить консоль
myobject	Просто печатает объект
summary(mydata)	Печатает простую статистику для <mydata>
"стрелка вверх/низ"	Вернуться/перейти к следующей команде

Основы и помощь

?topic	Документация по теме
apropos('topic')	Имена всех объектов в списке поиска, удовлетворяющих правильному выражению topic'
attach(x)	База ***x*** для пути поиска R; ***x*** может быть списком, фреймом данных, или файлом данных R, созданным сохранением. Используй search() для показа пути поиска

detach(x)	***x*** отключить от пути поиска R; ***x*** может быть именем или строкой символов объектом, ранее attached (присоединенным), или пакетом
dir()	Показать файлы в текущей директории
example(command)	Примеры комманд
help(package=mva)	Справка, например, по пакету mva
help(topic)	Документация по теме
help.search('topic')	Поиск системы справки
help.start()	Начать справку в версии HTML
library(x)	Загрузить пакет; library(help=x) перечисляет наборы и функции в пакете ***x***
ls()	Показать объекты на пути поиска; укажи ='pat' для поиска по образцу
ls.str()	str() для каждой переменной на пути поиска
methods(a)	Gjrfpfnm vtnjls S3 lkz a
methods(class=class(a))	Список всех методов в обрабатываемых объектах класса a
options(...)	Установить или проверить глобальные опции; наиболее общие: width, digits, error
str(a)	Показать внутреннюю структуру объекта R
summary(a)	Итоговая информация для ***a***'summary' - обычно статистические итоги, но она является универсальной, что означает разные операции для разных классов объектов a

Данные (создание)

array(x, dim=)	Массив с данными x; указываются размерности подобно dim=c(3,4,2); элементы ***x*** рецеклично повторяются, если ***x*** не имеет досточной длины
c(...)	Универсальная функция с комбинированием аргументов в вектор по умолчанию; с recursive=TRUE descends through lists combining all elements into one vector
cbind(...)	Вид rbind(...) для колонок
data.frame(...)	Создает фрейм данных с именованными или без имени аргументами; data.frame(v=1:4, ch=c('a', 'b', 'c', 'd'), n=10); более короткие вектора рециклично дополняются для длины самых длинных
expand.grid()	Фрейм данных из всех комбинаций предоставленных векторов и факторов
factor(x, levels=)	Кодирует вектор ***x*** как фактор
from:to	Создает последовательность; ':' имеет приоритетность; 1:4 + 1 is '2, 3, 4, 5'
gl(n, k, length=n*k, labels=1:n)	Генерирует уровни (факторов) путем спецификации паттернов их уровней; k является числом уровней, и n является количеством повторений
list(...)	Создает список с именем или без имени аргументов; list(a=c(1, 2), b='hi', c=3i);

matrix(x, nrow=, ncol=)	Матрица; элементы ***x*** рецеклично повторяются
rbind(...)	Комбинируются аргументы рядами для матрицы, фрейма данных и других

Последовательности

x<- 0:7	в ***x*** содержится последовательность чисел от 0 до 7
rep(x, times)	Повторяет ***x*** раз; используй 'each=' для повторения 'each' элементов ***x*** раз; rep(c(1, 2, 3), 2) равно 1 2 3 1 2 3; rep(c(1, 2, 3), each=2) равно 1 1 2 2 3 3
seq(along=x)	Создает 1, 2, ..., length(x); полезно для циклов
seq(from, to)	Создает последовательность 'by=' указывает приращение; 'length=' указывает требуемую длину
seq(1,10,by=2)	генерирует последовательность нечётных чисел 1, 3, 5, 7, 9
seq(2,10,2)	генерирует последовательность чётных чисел 2, 4, 6, 8, 10
set.seed(125141)	задаёт начальное состояние ГСЧ
y<- x[order(x)]	y – вектор по возрастанию
z<- x[order(x,decreasing = TRUE)]	по убыванию

Строки

grep(pattern, x)	Ищет соответствие образцу в x; смотри ?regex
gsub(pattern, replacement, x)	Замена соответствий, определенных правильным выражением соответствия; sub() аналогичен, но лишь заменяет первое вхождение
match(x, table)	Вектор позиций первого соответствия для элементов ***x*** в таблице table
nchar(x)	Число символов
paste(...)	Слить вектора после преобразования в символы; 'sep=' is the string to separate terms (a single space is the default); 'collapse=' is an optional string to separate 'collapsed' results
pmatch(x, table)	Частичное соответствие для элементов ***x*** среди таблицы
strsplit(x, split)	Разделение ***x*** соответственно для разделяемой подстроки
substr(x, start, stop)	Подстрока в символьном векторе; также может присвоить как: substr(x, start, stop)=value
tolower(x)	Преобразовать в строчные буквы
toupper(x)	Преобразование в заглавные буквы
x %in% table	Аналогичен match(x,table), но возвращает логический вектор

Данные (загрузка, чтение, запись и сохранение)

data(x)	Загружает указанный набор данных
data.entry()	Spreadsheet
download.file('url1')	Из интернета
load()	Сохраняет набор данных, записанный по сохранению (save)

read.csv('file', header=TRUE) Вид read.table(file), но с набором умолчаний для чтения: запятая-разделитель файлов

read.delim('file', header=TRUE) Вид read.table(file), но с набором умолчаний для чтения: табуляция-разделитель файлов

read.fwf('file', widths, header=FALSE, ...) Читает таблицу с форматом данных фиксированной ширины в 'data.frame'; ширина равна вектору целых, определяющих ширину полей фиксированной ширины

read.table('file') Читает файл в формате таблицы и создает фрейм данных из нее; Разделителем по умолчаниюthe sep=' ' является пробел; используй header=TRUE для чтения первой строки в качестве имен колонок; используй as.is=TRUE для блокирования конвертации вектора символов в факторы; используй comment.char=' ' для блокирования интерпретации '#' в качестве комментария; используй skip=n для пропуска n строк перед чтением данных; смотри справку для опции по именованию строк, обработки NA и других

read.table(file('clipboard'), header=T) Читает содержимое таблицы из кармана

save('file', ...) Сохраняет указанные объекты objects (...) в XDR в вдоичном формате, независимом от платформы

save.image('file') Сохраняет все объекты

write(object, 'file') Записывает объекты в файл с именем Name

write.table(dataFrame, 'file') Пишет таблицу

write.table(x, file=' ', row.names=TRUE, ...) Prints ***x*** after converting to a data frame; if quote is TRUE, character or factor columns are surrounded by quotes (' '); sep is the field separator; eol is the end-of-line separator; na is the string for missing values; Используй col.names=NA to add a blank column header to get the column headers aligned correctly for spreadsheet input

cat(..., file=' ', sep=' ') Печатает аргументы после преобразования в символы; sep является символом-разделителем между аргументами

format(x, ...) Форматировать объект R для красивой печати

print(a, ...) Печатает свои аргументы; функция универсальная, что означает, что имеет разные методы для разных объектов

read.table.url('url1') Удаленный ввод

scan(x) Читать вектор ***x***

sink(file) Вывод в файл пока sink()

source('file') Исполнить команду в файле

source(file('clipboard')) Исполнить команду в кармане

url.show('url') Удаленный ввод

Данные (выбор и манипуляция)

" Function="choose(n, k) Вычисляет комбинацию k событий среди n repetitions=n!/[(n-k)!k!]

cut(x, breaks) Делит ***x*** на интервалы (факторы); breaks is the number of cut intervals or a vector of cut points

match(x, y)	Возвращает вектор такой же длины как ***x*** с элементами ***x,*** которые входят в ***y*** (в противном NA)
na.fail(x)	Возвращает сообщение об ошибке, если ***x*** содержит, по крайней мере, хотя бы один NA
na.omit(x)	Подавляет наблюдения с пропущенными данными (NA) (suppresses the corresponding line if ***x*** is a matrix or a data frame)
prop.table(x, margin=)	Table entries as fraction of marginal table
rev(x)	Инверсия элементов ***x***
sample(x, size)	Случайная выборка без замены размера элементов в векторе ***x,*** опция replace=TRUE позволяет делать выборку с замещением
sort(x)	Сортирует элементы ***x*** по возрастанию; для сортировки по убыванию: rev(sort(x))
subset(x, ...)	Возвращает выборку из ***x*** в соответствии с критерием (..., типично сравнение); если ***x*** является фреймом данных, то опции выборки дают переменные для оставления или пропуск использование знака минуса
table(x)	Возвращает таблицу с набором разных значений ***x*** (типично для целых или факторов)
unique(x)	Если ***x*** является вектором или фреймом данный, то возвращает подобные объекты, но при этом подавляются дублирующие элементы
which(x==a)	Возвращает вектор индекса x, если верна операция сравнения (TRUE), в этом примере значение i ждя которого x[i]==a (аргумент этой функции должен быть переменной типа logical)
which.max(x)	Возвращает индекс максимального элемента ***x***
which.min(x)	Возвращает индекс минимального ***x***

Манипулирование данными

Массивы и матрицы

%*%	Умножение матриц
colMeans(x)	Быстрая версия средней колонок
colsum(x)	Сумма колонок для объекта, подобного матрице; colSums(x) является более быстрой версией
diag(x)	Диагональ
rowMeans(x)	Быстрая версия средней строки
rowsum(x)	Сумма строк для объекта, подобного матрице; rowSums(x) является более быстрой версией
solve(a)	Инверсия матрицы ***a***
solve(a, b)	Solves a %*% x=b for ***x***
t(x)	Транспонирование

Фреймы данных

aggregate(x, by, FUN)	Разделяет фрейм данных ***x*** на подмножества, вычисляет итоговую статистику для каждого из них, и возвращает результат в удобной форме; ***by*** является списком группирующих элементов, каждый из которых имеет такую же длину, как переменные в ***x***
str(lnu)	содержание фрейма данных
summary(lnu)	статистика по фрейму данных
lnu$logwage <- NULL	удаление переменной logwage из фрейма lnu

lnu.female <- subset(lnu, select=c(wage,female))

lnu.female <- data.frame(wage,female)

выбрать подмножество wage,female из фрейма lnu

lnux <- subset(lnu, select=-female)

выбрать все, кроме female

lnuxx <- subset(lnu, select=wage:public)

выбрать все элементы с wage по public

data3 <- data.frame(data1$wage, data1$female, data2$experience)

объединить части трех фреймов данных в одном

data4 <- merge(data1, data2)

объединить, если в data1 и data2 есть общий столбец

lnu <- na.omit(lnu)	удалить все наблюдения с пропущенными данными

highwage <- subset(lnu, wage > 90)

выбрать наблюдения, где wage > 90

a <- replace(a, is.na(a), 2)

a[is.na(a)] <- 2	замена на 2 значения NA в векторе
apply(x, INDEX, FUN=)	A vector or array or list of values obtained by applying a function FUN to margins (INDEX) of ***x***
by(data, INDEX, FUN)	Применяет FUN к данным фрейма данных, выделенных INDEX
lapply(x, FUN)	Применяет FUN к каждому элементу списка ***x***
merge(a, b)	Сливает два фрейма данных по общему имени колонки или строки
reshape(x, ...)	Reshapes a data frame between 'wide' format with repeated measurements in separate columns of the same record and 'long' format with the repeated measurements in separate records; используй (direction='wide') or (direction='long')
stack(x, ...)	Преобразует данные, доступные как отдельная колонка в фрейме данных или списке в отдельную колонку
tapply(x, INDEX, FUN=)	Apply FUN to each cell of a ragged array given by ***x*** with indexes INDEX
unstack(x, ...)	Инверсия stack()

xtabs(a, b, data=x)	A contingency table from cross-classifying factors
lag(ts, k)	сдвиг на лаги – это lead а не лаг
diff(x)	дифференцирование

Индексирование (фрейм данных)

x$name	Колонка с именем 'name'
x[['name']]	Колонка с именем 'name'

Индексирование (списков)

x$name	Элемент списка с именем 'name'
x[['name']]	Элемент списка с именем 'name'
x[[n]]	N-й элемент списка
x[n]	Список с n элементами"

Индексирование (Матрицы)

x[, c(1, 3)]	Колонки 1 и 3
x[, j]	Колонка j
x['name',]	Строка с именем 'name'
x[i,]	Строка i
x[i, j]	Элемент в строке i, колонке j"

КАК ОБРАТИТЬСЯ К СТРОКАМ ИЛИ СТОЛБЦАМ МАТРИЦЫ M

```
M[i,j]          # эдемент в i-той строке, j-том столбце
M[i,]           # i-тая строка
M[,j]           # j-тый столбец
M[c(1,3,5),]    # строки 1, 3 и 5
M[,c(1,3,5)]    # столбцы 1, 3 и 5
M[i:j,]         # строки с i по j
M[,i:j]         # столбцы с i по j
```

КАК УДАЛИТЬ УКАЗАННЫЕ СТРОКИ МАТРИЦЫ

```
        M[-c(1,3,5),]                   # удаляем строки 1, 3 и 5
        M[-3:5,]                        # удаляем строки с 3 по 5
        M[-i:j,]                        # удаляем строки с i по j
*       M[-(i+1):(j+1),]                # удаляем строки с i+1 по j+1
*       M[-i:j+1,]                      # удаляем строки с i+1 по j+1
```

КАК УДАЛИТЬ УКАЗАННЫЕ СТОЛБЦЫ МАТРИЦЫ

```
        M[,-c(1,3,5)]                   # удаляем столбцы 1, 3 и 5
        M[,-3:5]                        # удаляем столбцы с 3 по 5
        M[,-i:j]                        # удаляем столбцы с i по j
*       M[,-(i+1):(j+1)]                # удаляем столбцы с i+1 по j+1
*       M[,-i:j+1]                      # удаляем столбцы с i+1 по j+1
```

Индексирование (vectors)

x['name']	Элемент с именем 'name'
x[-(1:n)]	Элемент с n+1 до конца
x[1:n]	Первые n элементов
x[c(1, 4, 2)]	Определенный элемент
x[-n]	Все, кроме n-го элемента

x[n]	N-1 элемент
x[x %in% c('a', 'and', 'the')]	Элемент в данном наборе
unlist(список)	Преобразование списка в вектор (важно для функций, работающих с векторами)

Переменные (преобразование)

as.array(x)	Преобразовать тип; для полного списка используй methods(as)
as.character(x)	Преобразовать тип; для полного списка используй methods(as)
as.complex(x)	Преобразовать тип; для полного списка используй methods(as)
as.data.frame(x)	Преобразовать тип; для полного списка используй methods(as)
as.logical(x)	Преобразовать тип; для полного списка используй methods(as)
as.numeric(x)	Преобразовать тип; для полного списка используй methods(as)

Переменные (информация)

attr(x,which)	Получить или устанавливать атрибуты ***x***
attributes(obj)	Получить или установить список атрибутов obj
class(x)	Получить или установить класс x; class(x)='myclass'
dim(x)	Поиск или установка размерности объекта; dim(x)=c(3,2)
dimnames(x)	Поиск или установка имени размерности объекта
is.array(x)	Тест типа; для полного списка используй methods(is)
is.character(x)	Тест типа; для полного списка используй methods(is)
is.complex(x)	Тест типа; для полного списка используй methods(is)
is.data.frame(x)	Тест типа; для полного списка используй methods(is)
is.na(x)	Тест типа; для полного списка используй methods(is)
is.null(x)	Тест типа; для полного списка используй methods(is)
is.numeric(x)	Тест типа; для полного списка используй methods(is)
length(x)	Число элементов в ***x***
ncol(x)	Число колонок; NCOL(x) аналогичен, но обрабатывает вектор как матрицу из одной строки
nrow(x)	Число строк; NROW(x) аналогичен, но обрабатывает вектор как матрицу из одной строки
str(object)	Печатать полезную информацию об объекте

Переменные (управление)

ls()	Показать объекты на пути поиска; укажи pat='pat' для поиска по образцу
rm(object)	Удалить объект
unclass(x)	Удалить атрибут класса для ***x***

Даты и время

as.Date(s)	Конвертирует в соответствующий класс; format(dt) конвертирует в вид строк. Формат строки по умолчанию равен '2001-02-21'.

	Этим принимается второй аргумент для указания формата конвертации. Некоторые общие форматы: %a, %A сокращенное и полное название дня недели; %b, %B сокращенное и полное название месяца; %d день месяца (01-31); %H час (00-23); %I часы (01-12); %j день года (001-366); %m месяц (01-12); %M Минута (00-59); %p указатель AM/PM; %S секунда как десятичное число(00-61); %U Week (00-53); the first Sunday as day 1 of week 1; %w Weekday (0-6, Sunday is 0); %W Week (00-53); the first Monday as day 1 of week 1; %y Year without century (00-99). Don't используй; %Y Year with century; %z (output only.) Offset from Greenwich; -0800 is 8 hours west of; %Z (output only.) Time zone as a character string (empty if not available)
as.POSIXct(s)	Конвертирует в соответствующий класс; format(dt) конвертирует в вид строк. Формат строки по умолчанию равен '2001-02-21'. Этим принимается второй аргумент для указания формата конвертации. Некоторые общие форматы: %a, %A сокращенное и полное название дня недели; %b, %B сокращенное и полное название месяца; %d день месяца (01-31); %H час (00-23); %I часы (01-12); %j день года (001-366); %m месяц (01-12); %M Минута (00-59); %p указатль AM/PM; %S секунда как десятичное число (00-61); %U Week (00-53); the first Sunday as day 1 of week 1; %w Weekday (0-6, Sunday is 0); %W Week (00-53); the first Monday as day 1 of week 1; %y Year without century (00-99). Don't используй; %Y Year with century; %z (output only.) Offset from Greenwich; -0800 is 8 hours west of; %Z (output only.) Time zone as a character string (empty if not available)
z0 <- strptime("12.10.11", "%m.%d.%y")	
	z0 соответствует 10 декабря 2011 года
z1 <- strptime("12.10.11", "%d.%m.%Y")	
	z1 соответствует 12 октября 2011года
as.numeric(z1-z0)	число дней, прошедших между z0 и z1

Временные ряды

library(zoo)	
ts <- zoo(x, date)	Временной ряд в пакете **zoo**
library(xts)	
ts <- xts(x, date)	Временной ряд в пакете **xts**

Операторы

Группа "Math":

- *abs, sign, sqrt,*
  floor, ceiling, trunc,
  round, signif
- *exp, log, expm1, log10,*
  cos, sin, tan,
  acos, asin, atan
  cosh, sinh, tanh,
  acosh, asinh, atanh
- *lgamma, gamma, digamma, trigamma*

- *cumsum, cumprod, cummax, cummin*

Члены этой группы зависят от ***x***. Большинство членов принимает только один аргумент, но члены *log, round* и *signif* принимают один или два аргумента, *trunc* принимает один или более.

acos	
Arg(x)	Угол в радианах комплексного числа
asin	арксинус
atan	арккосинус
atan2	
Conj(x)	Сопряженное комплексное число
convolve(x, y)	Вычислить несколько видов свертки двух последовательностей
cos	Косинус
cov(x, y)	Ковариация между ***x*** и ***y,*** или между двумя колонками ***x*** и соответствующими ***y,*** если они матрицы или фреймы данных
cummax(x)	Вектор, в котором i-й элемент равен максимуму из x[1] до x[i]
cummin(x)	Вектор, в котором i-й элемент равен минимуму из x[1] до x[i]
cumprod(x)	Вектор, в котором i-й элемент равен произведению из x[1] до x[i]
cumsum(x)	Вектор, в котором i-й элемент равен сумме от x[1] до x[i]
diff(x)	Лаговые и повторные разности вектора ***x***
exp	Экспонента
fft(x)	Быстрое преобразование Фурье для массива
filter(x, filter)	Применение линейной фильтрации для одномерного временного ряда или для каждого ряда отдельно в многомерном временном ряду
Im(x)	Мнимая часть
intersect(x, y)	'set' function
is.element(el, set)	'set' function
log	Логарифм
log(x, base)	вычисляет логарифм ***x*** по основанию base
log10	Десятичный логарифм
max(x)	Максимум из элементов ***x***
min(x)	Минимум из элементов ***x***
Mod(x)	Модуль; abs(x) аналогичен
mvfft(x)	FFT каждой колонки матрицы
pmax(x, ***y,*** ...)	Вектор, каждый i-й элемент которого равен максимуму из x[i], y[i], . . .
pmin(x, ***y,*** ...)	Вектор, каждый i-й элемент которого равен минимуму из x[i], y[i], . . .

prod(x)	Произведение элементов ***x***
range(x)	Аналогично c(min(x), max(x))
rank(x)	Разрядность элементов ***x***
Re(x)	Действительная часть комплексного числа
round(x, n)	Округление элементов ***x*** до n знаков
scale(x)	Если ***x*** является матрицей, то центрирует и масштабирует данные; to center only используй the option scale=FALSE, to scale only center=FALSE (by default center=TRUE, scale=TRUE)
setdiff(x, y)	'set' function
setequal(x, y)	'set' function
sin	Синус
sum(x)	Сумма элементов ***x***
tan	Тангенс
union(x, y)	'set' function

Группа "Ops":

- "+", "-", "*", "/", "^" - степень, "%%" - модуль, "%/%" – деление нацело
- "&" – «и», "|" – «или», "!" – «нет»
- "==" - равно, "!=" – не равно, "<", "<=", ">=", ">"

Группа "Summary":

- *all – в логическом, any-что-либо TRUE?*
- *sum-сумма значений, prod-произведение*
- *min, max*
- *range-мин и мах в векторе*

Члены этой группы зависят от первого предоставленного аргумента.

Группа "Complex":

Arg, Conj, Im, Mod, Re

Другие полезные операторы

~	Тильда, используемая для формул модели, может быть или унарным или бинарным
?	Справка
:	Последовательность, двоичная (в формулах модели взаимодействие)
%x%	Специальные бинарные операторы, ***x*** могут быть заменены любым допустимым именем
% * %	Матричное произведение, бинарное
%o%	Внешнее произведение, бинарное
%x%	Кронекерово умножение, бинарное
%in%	Соответствие оператора, бинарного (в формулах модели: гнездованное)
<-	Левое присвоение, бинарное
->	Правое присвоение, бинарное
<<-	Слева глобальная переменная (глобальное присвоение)
$	Подмножество списка, бинарное

Управление последовательностью выполнения

Операторы условия

if(cond) expr

if(cond) cons.expr else alt.expr

ifelse(test, yes, no)	A value with the same shape as test filled with elements from either yes or no

if (statement1)

statement2 else

statement3	Если else перенести на следующую строчку, то получим завершенный оператор if.
switch(n,a1,...,ak)	значение ***n*** дает номер оператора (любые конструкции R)

Операторы цикла

for (name in vector) statement1,

while (statement1) statement2,

repeat statement,	Повторение
next и *break*	Прерывание цикла

Функции

fun1 <- function(data, data.frame, graph, limit) { [тело функции опущено]}

Эквивалентные вызовы функции:

ans <- fun1(d, df, TRUE, 20)	Вызов по месту аргумента
ans <- fun1(d, df, graph=TRUE, limit=20)	Вызов по месту и имени
ans <- fun1(d, df)	Вызов по умолчанию
"%!%" <- function(X, y) { ... }	Новый бинарный оператор
X%!%y	Использование

return(выражение)	Возвращает значение выражения, или последнее значение в функции (значение может быть функцией)

Обработка ошибок

warnings()	Вызов сообщений
suppressWarnings(adf.test(x))	Подавление сообщений *adf*
stop("сообщение", filename)	Вывод сообщения в файл и переход на уровень выше

result <- try(nonlinear_modeling());

if(class(result) == "try-error") {};

Пример:

```
result = tryCatch(
        { expr},
        warning = function(w) { warning-handler-code},
        error = function(e) { error-handler-code},
        finally { cleanup-code} )
```

class(result)

"simpleError" "error" "condition"

http://stackoverflow.com/questions/2158780/r-catching-an-error-and-then-branching-logic

Статистика

Основные статистики

acf	возвращает коррелограмму (или набор аргументов для автоковариационной функции).
anova(fit, ...)	Анализ таблицы дисперсий (или отклонений) для одного или более объектов подогнанных моделей
aov(formula)	Анализ дисперсии модели
binom.test()	Используй help.search('test')
chisq.test(x)	Тест хи-квадрат для матрицы ***x***
cor(x)	Корреляция матрицы ***x,*** если она является матрицей или фреймом данных (1 если ***x*** является вектором)
cor(x, y)	Линейная корреляция между ***x*** и ***y,*** или корреляция матриц, если они являются матрицами или фреймами данных
cor.test(a, b)	Тест корреляции
z$estimate	коэффициент корреляции между векторами ***x*** и ***y***
z$conf.int	95% доверительный интервал для оценки коэффициента корреляции
z$p.value	*p-value*, достигнутое при проверке гипотезы о нулевом значении коэффициента корреляции
cov(x)	Ковариация элементов ***x*** (вычислено на on n-1); если ***x*** является матрицей или фреймом данных, то вычисляется дисперсия-ковариация матрицы
density(x)	Оценка ядерной плотности ***x***
fisher.test()	Точный тест Fisher
friedman.test()	Тест Friedman
mean(x)	Средняя элементов ***x***
median(x)	Медиана элементов ***x***
pairwise.t.test()	Используй help.search('test')
power.t.test()	Используй help.search('test')
prop.test()	Тест значимости
quantile(x, probs=)	Выборки квантилей, соответствующих данной вероятности (по умолчанию это 0, 0.25, 0.5, 0.75, 1)
sd(x)	Стандартное отклонениеf ***x***
t.test()	Используй help.search('test')
var(x)	Дисперсия элементов ***x*** (выцчислена для n-1); если ***x*** является матрицей или фреймом данных, то вычисляется дисперсия-ковариация матрицы

var(x, y)	Ковариация между ***x*** и ***y,*** или между колонками ***x*** и соответствующими ***y*** если они являются матрицами или фреймами данных
weighted.mean(x, w)	Средняя ***x*** с весом w

Распределения

rbeta(n, shape1, shape2)	Beta
rbinom(n, size, prob)	Binomial
rcauchy(n, location=0, scale=1)	Cauchy
rchisq(n, df)	Pearson
rexp(n, rate=1)	Exponential
rf(n, df1, df2)	Fisher-Snedecor
rgamma(n, shape, scale=1)	Gamma
rgeom(n, prob)	Geometric
rhyper(nn, m, n, k)	Hypergeometric
rlnorm(n, meanlog=0, sdlog=1)	Lognormal
rlogis(n, location=0, scale=1)	Logistic
rnbinom(n, size, prob)	Negative binomial
rnorm(n, mean=0, sd=1)	Gaussian (normal)
rpois(n, lambda)	Poisson
rsignrank(nn, n)	Wilcoxon's Статистика
rt(n, df)	Student (t)
runif(n, min=0, max=1)	Uniform
rweibull(n, shape, scale=1)	Weibull
rwilcox(nn, m, n)	Wilcoxon's Статистика

Регрессии

y ~ x, **y** ~ 1 + **x**	Регрессия **y** на **x**. У 1-й неявный параметр смещения, а у 2-й - явный.
y ~ 0 + x, **y** ~-1 + x, **y ~ x - 1**	Простая линейная регрессия **y** на **x** без смещения
log (y) ~ x1 + x2	Множественная регрессия log(y) на **x1** и **x2** с неявным смещением
y ~ poly (x, 2)	
y ~ 1 + **x** + I(x^2)	Параболическая регрессия **y** на **x** степени 2. Первая форма использует ортогональные полиномы, и вторая использует явную степень, как основание. I(M) - изолированное **M.** Внутри **M** все операторы имеют свое нормальное арифметическое значение, и этот параметр появляется в матрице модели.

full.model <- lm(y ~ x1 + x2 + x3 + x4)

reduced.model <- step(full.model, direction="backward")

поиск минимальной модели путем уменьшения

```
min.model <- lm(y ~ 1)
fwd.model <- step(min.model,
+ direction="forward",
+ scope=( ~ x1 + x2 + x3 + x4),
+ trace=0 )
```

поиск минимальной модели путем расширения

OLS (МНК) подгонка линейной модели

y.lm=lm(y~x,data=XXX)	строит регрессионную модель вида *y=a+b*x* по набору данных (data.frame), содержащему переменные ***x*** и ***y***
y.lm =lm(y~x1+x2,data=XXX)	строит регрессионную модель вида *y=a+b*x1+c*x2* по набору данных (*data.frame*), содержащему переменные ***x1, x2*** и ***y***
abline(y.lm)	добавляет на график регрессионную линию
equation1 <- formula(y.lm)	извлекает формулу модели
anova(y.lm)	таблица ANOVA
coef(y.lm)	извлекает коэффициенты модели
conrint(y.lm)	доверительные интервалы для коэф. регрессии
effects(y.lm)	вектор ортогонального влияния
fitted(y.lm)	вектор значений подгонки **y**
sqrt(diag(vcov(y.lm)))	извлекает стандартную ошибку из диагонали
coef(summary(y.lm))[, 2]	Извлекает стандартные ошибки из диагональной матрицы.
coef(summary(y.lm))[, 3]	Извлекает t-значения
deviance(y.lm)	Сумма квадратов остатков
residuals(y.lm)	Остаток модели
fitted.values(y.lm)	
summary(y.lm)$r.squared	
summary(y.lm)$adj.r.squared	
summary(y.lm)$sigma	
summary(y.lm)$fstatistic	
acf(resid(y.lm))	
pacf(resid(y.lm))	коррелограммы остатков

Проверка пригодности OLS

library(lmtest)	
bptest(y.lm)	
dwtest(y.lm)	
library(car)	
ncv.test(y.lm)	проверка на гетероскедастичность
durbin.watson	проверка на автокорреляцию
singular.ok=FALSE	проверка на сингуляроность (вырожденность) – вставляем в подгонку lm

Менее чувствительные к выбросам и гетероскедастичности регрессии

Стойкие (resistant) регрессии

```
library(MASS)
## S3 method for class 'formula':
lqs(formula, data, ...,
method = c("lts", "lqs", "lms", "S", "model.frame"),
subset, na.action, model = TRUE,
x.ret = FALSE, y.ret = FALSE, contrasts = NULL)
## Default S3 method:
lqs(x, y, intercept = TRUE, method = c("lts", "lqs", "lms", "S"),
quantile, control = lqs.control(...), k0 = 1.548, seed, ...)
```

Робастные регрессии

```
## S3 method for class 'formula':
rlm(formula, data, weights, ..., subset, na.action,
method = c("M", "MM", "model.frame"),
wt.method = c("inv.var", "case"),
model = TRUE, x.ret = TRUE, y.ret = FALSE, contrasts = NULL)
## Default S3 method:
rlm(x, y, weights, ..., w = rep(1, nrow(x)),
init = "ls", psi = psi.huber,
scale.est = c("MAD", "Huber", "proposal 2"), k2 = 1.345,
method = c("M", "MM"), wt.method = c("inv.var", "case"),
maxit = 20, acc = 1e-4, test.vec = "resid", lqs.control = NULL)


library(robust)
lmRob(formula, data, weights, subset, na.action, model = TRUE, x = FALSE,
y = FALSE, contrasts = NULL, nrep = NULL,
control = lmRob.control(...), genetic.control = NULL, ...)
```

Сравнение регрессий

```
library(nutshell)
data(schiller.index)
hpi.lm <- lm(Index~Year,data=schiller.index)
hpi.rlm <- rlm(Index~Year,data=schiller.index)
hpi.lqs <- lqs(Index~Year,data=schiller.index)


plot(hpi,pch=19,cex=0.3)
abline(reg=hpi.lm,lty=1)
```

abline(reg=hpi.rlm,lty=2)

abline(reg=hpi.lqs,lty=3)

legend(x=1900,y=200,legend=c("lm","rlm","lqs"),lty=c(1,2,3))

Пошаговый выбор переменных

step(object, scope, scale = 0,

direction = c("both", "backward", "forward"),

trace = 1, keep = NULL, steps = 1000, k = 2, ...)

Анализ коллинеарности

library(MASS)

lm.ridge(formula, data, subset, na.action, lambda = 0, model = FALSE,

x = FALSE, **y** = FALSE, contrasts = NULL, ...)

Уменьшение величины коэффициентов

library(lars)

lars(x, y, type = c("lasso", "lar", "forward.stagewise", "stepwise"),

trace = FALSE, normalize = TRUE, intercept = TRUE, Gram,

eps = .Machine$double.eps, max.steps, use.Gram = TRUE)

Универсальные функции для извлечения информации о модели

add1	
deviance	Возвращает отклонение подогнанной модели объекта
formula	
predict	
step	Выбирает основанную на формуле модель по AIC
alias	Находит линейно зависимые параметры в указанной линейной модели по формуле
drop1	
kappa	Вычисляет или оценивает условный номер матрицы
print	
summary	

anova	effects	labels	proj	vcov
coef	family	plot	residuals	

Обобщенные наименьшие квадраты (GL)

library(nlme)

x.gls <- gls(x ~ Time, cor = corAR1(0.8))

plot(x, type = "l")

abline(0, 0)

glm(formula, family = gaussian, data, weights, subset,

na.action, start = NULL, etastart, mustart,

offset, control = glm.control(...), model = TRUE,

method = "glm.fit", **x** = FALSE, **y** = TRUE, contrasts = NULL,

...)

Нелинейные модели

x.nls <- nls(x ~ exp(alp0 + alp1 * Time), start = list(alp0 = 0.1,alp1 = 0.5))

Оцениваем параметры alp0, alp1

Функции прогоноза

`predict.glm`

`predict.lm`

`predict.loess`

`predict.nls`

`predict.poly`

`predict.princomp`

`predict.smooth.spline`.

SafePrediction Для прогноза от полиномов и сплайнов

`predict.ar`

`predict.Arima`

`predict.arima0`

`predict.HoltWinters`

`predict.StructTS`. Для прогноза авторегрессий

Сглаживание (детрендирование)

Сплайны

library(stats)

smooth.spline(x, **y** = NULL, w = NULL, df, spar = NULL,

cv = FALSE, all.knots = FALSE, nknots = NULL,

keep.data = TRUE, df.offset = 0, penalty = 1,

control.spar = list())

Подгонка поверхностей полиномами

loess(formula, data, weights, subset, na.action, model = FALSE,

span = 0.75, enp.target, degree = 2,

parametric = FALSE, drop.square = FALSE, normalize = TRUE,

family = c("gaussian", "symmetric"),

method = c("loess", "model.frame"),

control = loess.control(...), ...)

Ядерное сглаживание

library(KernSmooth)

locpoly(x, y, drv = 0L, degree, kernel = "normal",

bandwidth, gridsize = 401L, bwdisc = 25,

range.x, binned = FALSE, truncate = TRUE)

Фильтры

library(mFilter)

hp.eur<-hpfilter(eur.sub, freq= 1000, type="lambda")

Фильтр HP. freq – это значение лямбды

Оптимизация и модели подгонки

AIC(fit)	Вычисляет информационный критерий Akaike или AIC
approx(x, y=)	Линейная интерполяция указанных точек данных; ***x*** can be an xy plotting structure
coef(fit)	Возвращает оцененные коэффициенты (иногда с их стандартными ошибками)
deviance(fit)	Возвращает отклонение
df.residual(fit)	Возвращает число степеней свободы остатка
fitted(fit)	Возвращает подогнанные значения
glm(formula, family=)	Подгонка обобщенной линейной модели, указанной данным символьным описанием линейного предиктора и описанием распределения ошибки; family является описанием распределения ошибки и связывает функцию подлежащую использованию в моделе; смотрие ?family
lm(formula)	Подгонка линейной модели; formula типично выглядит как response termA + termB + ...; используй I(x*y) + I(x^2) для создания параметров нелинейных компонент
loess(formula)	Подгонка полиномиальной поверхности с использованием локальной подгонки
logLik(fit)	Вычисляет логарифм правдоподобия и число параметров
nlm(f, p)	Минимизирует функцию f с использованием алгоритма типа Newton со стартовым значением p
nls(formula)	Оценка нелинейными наименьшими квадратами параметров нелинейной модели
optim(par, fn, method=c('Nelder', ...), ...)	Универсальная оптимизация; par является первоначальным значением, fn является функцией, подлежащей оптимизации (обычно минимизации)
predict(fit, ...)	Предсказание из подгонки, основанной на входных данных
residuals(fit)	Возврат остатка
spline(x, y=)	Кубическая сплайн интерполяция

Стационарные модели

Моделируется остаток после удаления из ряда тренда и сезонных колебаний. Удаление можно сделать с помощью регрессии.

x.ar <- ar(x, method = "mle")

x.ar$order — Подгонка модели ar с автоматическим выбором параметра по AIC

x.ma <- arima(x, order = c(0, 0, 3)) — Подгонка MA(3)

x.ar <- arima(x.ts, order = c(1, 0, 0)) Подгонка AR(1)

x.arma <- arima(x.ts, order = c(1, 0, 1)) — Подгонка ARMA(1,0,1)

Выбор оптимальных параметров

best.order <- c(0, 0, 0) — Начальные значения параметров ARMA

best.aic <- Inf — Максимальный AIC

```
for (i in 0:2) for (j in 0:2) {
fit.aic <- AIC(arima(resid(Elec.lm), order = c(i, 0,
j))) Моделируем остаток от регрессии
if (fit.aic < best.aic) {
    best.order <- c(i, 0, j)
    best.arma <- arima(resid(Elec.lm), order = best.order)
    best.aic <- fit.aic
        }
best.order
[1]  2  0  0
acf(resid(best.arma))
```

Нестационарные модели

Несезонные модели ARIMA

```
arima(x, order = c(1, 1, 1))
```

Сезонные модели ARIMA

```
arima(x.ts, order = c(p,d,q),
      seas = list(order = c(P,D,Q),
frequency(x.ts)), method = "CSS")
```

Модели ARCH

```
library(tseries)
a.garch <- garch(a, trace = FALSE)
```
Квадрат ошибок **a** был коррелирован

```
a.res  <- a.garch$res[-1]
acf(a.res)
```
Должна отсутствовать корреляция в остатке

```
acf(f.res^2)
```
Должна отсутствовать корреляция

```
confint(a.garch)
```
Доверительные интервалы для коэффициента

Процессы долгосрочной памяти

Величина интегрированности от -0.5 до 0.5. От 0 до 0.5 – долгосрочная память, между стационарной моделью и нестационарным случайным блужданием. Удаляем тренда и для остатков делаем подгонку FARIMA

library(fracdiff)

fracdiff(x, nar = 1)	nar и nam – это порядок ar и ma

Многомерные модели

library(tseries)

po.test(cbind(x, y))	Тест на коинтеграцию (**x** и **y** в матрице)

Phillips-Ouliaris Cointegration Test

data: cbind(x, y)

Phillips-Ouliaris demeaned = -1020, Truncation lag parameter = 9,

p-value = 0.01	вероятность того, что **не** коинтегрированы

Коинтеграция

eur.Cycle <- eur.sub-hp.eur$trend

gbp.Cycle <- gbp.sub-hp.gbp$trend

po.test(cbind(eur.Cycle, gbp.Cycle))

Модели пространства состояний

Анализ остатков

library(tseries)

adf.test(x)	Тест единичного корня DF
pp.test(x)	Тест Phillips-Perron
po.test(cbind(eur.Cycle, gbp.Cycle))	тест на коинтеграцию: Но: вероятность отсутствия коинтеграции. По cbind из двух векторов получаем матрицы.

Графика

Команды низкого уровня

abline(a, b)	Рисует линию с наклоном b и смещением a
abline(h=y)	Рисует горизонтальную линию с ординатой ***y***
abline(lm.obj)	Рисует линию регрессии, определенную lm.obj
abline(v=x)	Рисует вертикальную линию с абсциссой ***x***
arrows(x0, y0, x1, y1, angle=30, code=2)	Вид segments(x0, y0, x1, y1), но со стрелкой в точке (x0, y0); если code=2, то точка (points) (x1, y1); если code=1, или обе, если if code=3; angle управляет углом from the shaft of the arrow to the edge of the arrow head
axis(side)	Добавляет ось внизу (side=1), слева (2), сверху 3), или справа (4); at=vect (опционально) дает абсциссу (или ординату) где рисуются маркеры тиков

box()	Рисует рамку вокруг текущего рисунка
identify()	Получить сведения о точке, отмеченной мышью
legend(x, ***y,*** legend)	Добавляет легенду в точке (x,y) с символами, указанными в легенде
lines(x, y)	Добавляет линию (может использоваться опция 'type=')
locator(n, type='n', ...)	Возвращает координату (x, y) после щелчка мышью n раз на рисунке; также рисует символ (type='p') или линию (type='l') в соответствии с указанными графическими параметрами (...); по умолчанию ничего не рисуется (type='n')
mtext(text, side=3, line=0, ...)	Добавляет текс, указанный text в промежутке, указанном стороной (смотри axis()); line указывает линию из области рисования
points(x, y)	Добавляет точки (может использоваться опция 'type=')
polygon(x, y)	Рисует многоугольник, связывающий точки с координатами, данными ***x*** и ***y***
rect(x1, y1, x2, y2)	Рисует прямоугольник, который слева, справа, сверху и снизу ограничен x1, x2, y1 и y2, соответственно
rug(x)	Рисует данные ***x*** на оси ***x*** как маленькие вертикальные линии
segments(x0, y0, x1, y1)	Рисует линию из точки (x0, y0) до точки (x1, y1)
text(x, ***y,*** labels, ...)	Добавляет текст, данный labels в координате (x, y); типичное использование: plot(x, ***y,*** type='n'); text(x, ***y,*** names)
title()	Добавляет заголовок и возможно подзаголовок

Параметры

adj	Управляет выравниванием текста (0 выровнено влева, 0.5 центрировано, 1 выровнено вправо)
bg	Указывает цвет фона (например: bg='red', bg='blue', . . . список 657 доступных цветов показывается с colors())
bty	Управляет типом рамки вокруг рисунка, разрешенными значениями являются: 'o', 'l', '7', 'c', 'u' ou ']' (рамка состоит из соответствующего символа); если bty='n', то рамка не рисуется
cex	Значения, управляющие размером текста и символов в отношении к умолчанию; следующие параметры имеют аналогичное управление для чисел на осях cex.axis, метки осей cex.lab, заголовок cex.main, и подзаголовок cex.sub
col	Управляет цветом символов и линий; Используй имена цветов: 'red', 'blue'. Смотри colors() или '#RRGGBB'; смотри rgb(), hsv(), gray() и rainbow(); в отношении cex существуют: col.axis, col.lab, col.main, col.sub
font	Целое, которое управляет стилем текста (1: normal, 2: italics, 3: bold, 4: bold italics); в отношении cex существуют: font.axis, font.lab, font.main, font.sub
las	Целое, которое управляет ориентацией меток осей (0: параллельно оси, 1: горизонтально, 2: перпендекулярно к оси, 3: вертикально)

lty	Управляет типом линий, может быть целым или строкой (1: 'solid', 2: 'dashed', 3: 'dotted', 4: 'dotdash', 5: 'longdash', 6: 'twodash', or a string of up to eight characters (between '0' and '9') which specifies alternatively the length, in points or pixels, of the drawn elements and the blanks, for example lty='44' will have the same effect than lty=2
lwd	Число, которое управляет шириной линии, по умолчанию 1
mar	Вектор из 4 числовых значений, которые управляют пространством между осями и границей графика в форме c(bottom, left, top, right), значения по умолчанию равны c(5.1, 4.1, 4.1, 2.1)
mfcol	Вектор в форме c(nr, nc), который делит окно графика как матрицу на nr строк и nc колонок, затем рисуется по колонками
mfrow	Подобно mfcol, но рисуется по строкам
pch	Управляет типом символов, либо целое между 1 и 25, или отдельный символ в кавычках ' '
ps	Целое, которое управляет размером в точках текстов и символов
pty	Символ, который указывает тип рисуемой области, 's': квадрат, 'm': максимальный
tck	Значение, которое специфицирует длину штриха на оси как часть наименьшей ширины или высоты рисунка; если tck=1, то рисуется сетка
tcl	Значение, которое специфицирует длину штриха на оси как часть высоты линии текста (по умолчанию tcl=-0.5)
xaxs	Стиль вычисления интервала между осями; по умолчанию 'r' для дополнительного пространства; 'i' без дополнительного пространства
xaxt	Если xaxt='n', то x-ось устанавливается, но не рисуется (полезно в соединении с axis(side=1, ...))
yaxs	Стиль вычисления интервала между осями; по умолчанию 'r' для дополнительного пространства; 'i' без дополнительного пространства
yaxt	Если yaxt='n', y-ось устанавливается, но не рисуется (полезно с соединении с axis(side=2, ...))

Рисование

add=FALSE	Общие параметры для многих функций рисования, если TRUE, то график накладывается на предыдущий (если он существует)
assocplot(x)	График Cohen-Friendly, показывающий отклонение от независимости строк и колонок в двухмерной таблице сопряженности
axes=TRUE	Общие параметры для многих функций рисования, Если FALSE, то не рисуются оси и рамка
barplot(x)	Гистограмма значений x; используй horiz=FALSE для горизонтальных баров

boxplot(x)	Box-and-whiskers plot
contour(x, ***y,*** z)	Контурный график (данные интерполируются для рисования кривых), ***x*** и ***y*** должны быть векторами, а z должна быть матрицей так чтобы dim(z)=c(length(x), length(y)) (x and ***y*** may be omitted)
coplot(x~y | z)	Двумерный график ***x*** и ***y*** для каждого значения или интервала значений z
dotchart(x)	Если ***x*** является фреймом данных, то рисуется точечный график Cleveland (stacked plots line-by-line and column-by-column)
filled.contour(x, ***y,*** z)	Подобно contour(x, ***y,*** z), но области между контурами окрашены, и нарисованы надписи на цвете
fourfoldplot(x)	Визуализирует четвертями кругов зависимость между двумя дихотомическими переменными для различных совокупностей (x должен быть массивом с dim=c(2, 2, k), или матрицей с dim=c(2, 2) если k=1)
hist(x)	Гистограмма частот ***x***
image(x, ***y,*** z)	Подобно contour(x, ***y,*** z), но цветное (рисуются реальные данные)
interaction.plot(f1, f2, y)	Если f1 и f2 являются факторами, то рисует средние ***y*** (на y-оси) против значения f1 (на x-оси) и f2 (разные кривые); опция fun позволяет выбрать итоговую статистику ***y*** (по умолчанию fun=mean).
main=	Общие параметры для многих функций рисования, основной заголовок, должны быть переменными типа символьный
matplot(x, y)	Двумерный график первой колонки ***x*** против первой колонки ***y,*** вторая ***x*** против второй ***y,*** и т.д.
mosaicplot(x)	Мозаичный график остатков от log-линейной регрессия таблицы сопряженности
pairs(x)	Если ***x*** является матрицей или фреймом данных, рисует все возможные двумерные графики между колонками ***x***
persp(x, ***y,*** z)	Подобно contour(x, ***y,*** z), но перспектива (рисуются актуальные данные)
pie(x)	Circular pie-chart
plot(x)	Рисует значения ***x*** (по оси y), упорядоченными по оси ***x***
plot(x, y)	Двумерный график ***x*** (с x-осью) и ***y*** (с y-осью)
plot.ts(x)	Если ***x*** является объектом класса 'ts', то рисует ***x*** в соответствии со временем, ***x*** может быть многомерным рядом, но должен иметь одинаковую частоту и даты
qqnorm(x)	Квантили ***x*** в соотношении со значениями, ожидаемыми под нормальным законом
qqplot(x, y)	Квантили ***y*** в соотношении с квантилями ***x***
stars(x)	Если ***x*** является матрицей или фреймом данных, то рисуется график с сегментами или звездой, где каждая строка ***x*** представлена звездой, а столбцы являются длинами сегментов

stem(x)	Производит рисунок stem-and-leaf для значений в 'x'
stripplot(x)	Рисует значение ***x*** на линии (альтернатива boxplot() для выборок небольших размеров)
sub=	Общие параметры для многих функций рисования, подзаголовок (пишется меньшим шрифтом)
sunflowerplot(x, y)	Подобно plot() но точки с одинаковыми координатами рисуются как цветы с лепестками, в которых ппредставлено число точек
symbols(x, ***y,*** ...)	На координатах, указанных ***x*** и ***y*** рисует символы (круги, квадраты, треугольники, звезды, термометры или 'свечи'), размер, цвет .....которых указывается дополнительными аргументами
termplot(mod.obj)	Рисует (частично) эффекты регрессионной модели (mod.obj)
ts.plot(x)	Подобно plot.ts(x), но если ***x*** является многомерным рядом, то может иметь разные даты и должен иметь одинаковую частоту
type='p'	Общие параметры для многих функций рисования, специфицирует тип рисования:
'p':	точки,
'l':	линии,
'b':	точки, соединенные линиями,
'o':	аналогично, но линии поверх точек,
'h':	вертикальные линии,
's':	шаги, данные представлены верхним концом вертикальных линий,
'S':	аналогично, но данные представлены нижним концом вертикальных линий
xlab=	Общие параметры для многих функций рисования, надписывает оси, должны быть переменными типа символьный
xlim=	Общие параметры для многих функций рисования, указывает нижний и верхний предел осей, например, xlim=c(1, 10) или xlim=range(x)
ylab=	Общие параметры для многих функций рисования, надписывает оси, должны быть переменными типа символьный
ylim=	Общие параметры для многих функций рисования, указывает нижний и верхний предел осей, например с ylim=c(1, 10) или ylim=range(x)

Устройства

bitmap	Смотри ?Devices
dev.set(3)	делает активным устройство с номером 3
dev.off()	Закрыть указанное графическое окно (по умолчанию текущее); также смотри dev.cur, dev.set; также смотри dev.cur, dev.set
graphics.off()	Закрыть все графические устройства

jpeg	Смотри ?Devices
pdf	Смотри ?Devices
pictex	Смотри ?Devices
png	Смотри ?Devices
postscript(file)	Запустить графическое устройство для производства графики PostScript; используй horizontal=FALSE, onefile=FALSE, paper='special' for EPS files; 'family=' specifies the font (AvantGarde, Bookman, Courier, Helvetica, Helvetica-Narrow, NewCenturySchoolbook, Palatino, Times, or ComputerModern); 'width=' and 'height=' specifies the size of the region in inches (for paper='special', these specify the paper size)
ps.options()	Установить и обозреть (если вызвано без аргументов) значения по умолчанию для аргументов postscript
windows()	Открыть графическое окно
x11()	Открыть графическое окно
xfig	Смотри ?Devices

3-D поверхности (пакет rgl)

open3d()	открываем графическое окно
persp3d(z)	***z*** - матрица, значения элементов которой задают высоту поверхности
persp3d(z, color='green')	поверхность будет зелёного цвета (по умолчанию цвет чёрный)
persp3d(z, color=cl)	***cl*** - вектор цвета для элементов матрицы ***z*** (расположенных по столбцам)
cl <- rainbow(7)[6(max(z)-z)/(max(z)-min(z))+1]*	располагает 7 цветов радуги так, чтобы поверхность раскрасилась согласно значению её высоты

Перечень пакетов эконометрики

afmtools	оценка прогноз и диагностика ARFIMA
AICcmodavg	выбор моделей на основе критерия AIC
BootPR	The package provides alternative bias-correction methods for univariate autoregressive model parameters; and generate point forecats and prediction intervals for economic time series.
CADFtest	Тест единичного корня по Hansen
car	Трансформация данных, подгонка линейных моделей, подгонка обобщенных линейных моделей, диагностика проблем, рисование
chron	Хронологические объекты, представленные датой и временем
debug	Отладчик в R
devtools	Набор утилит для создания библиотек (package) в R
dlm	Байесовский анализ динамических линейных моделей
dyn	Регрессия ВР
EvalEst	Оценка по Монте-Карло методов оценки моделей
fGarch	GARCH моделирование
FitAR	Подгонка моделей AR
FitARMA	Подгонка моделей ARMA
FKF	Многомерные модели в пространстве состояний
forecast	Одномерные прогнозы временного ряда
fracdiff	Подгонка дробных моделей ARFIMA
fractal	Фрактальный анализ и моделирование ВР

fts	Неправильные временные ряды на POSIXct
fUnitRoots	Тест единичного корня с удобной выдачей
fxregime	Анализ режима обменного курса
IBrokers	Интерфейс к брокеру
KFAS	Быстрый многомерный фильтр Кальмана
kza	Адаптивные фильтры
lmtest	Тестирование моделей линейной регрессии
ltsa	Линейный анализ ВР
mAr	Анализ много вариантных регрессий
meboot	Бутстрэпинг с максимально энтропией ВР
mFilter	Фильтры для извлечения тренда и циклической составляющей
mvbutils	Организация рабочего пространства, кодирования и документирования
PairTrading	Парный трейдинг
robfilter	Устойчивые фильтры ВР
Rssa	Сингулярный анализ спектра для разложения ВР
sspir	Модели пространства состояний
tframe	Устанавливает периоды времени в разных форматах
timeDate	Хронологические и календарные объекты
TimeSeries	Временной ряд с отметками времени timeDate
timsac	Анализ и управление ВР
tis	обеспечивает класс ***ti*** для времени и даты
TSA	Анализ ВР
TSAgg	Агрегация неполных данных временного ряда
tseries	Неправильные временные ряды на POSIXct
urca	Тест единичного корня и коинтеграции
wavelets	Фильтры, трансформация и анализ
waveslim	Основные вейвлет-программы для ВР
wavethresh	Локально стационарные модели вейвлета для нестационарного ВР
wmtsa	Software to book Wavelet Methods for Time Series Analysis, Percival, Walden
xts	Обработка основанных на времени классов данных R
Zelig	Регрессии для наблюдений с разными законами распределений
zoo	Инфраструктура для регулярным и нерегулярных ВР