

# Основы программной среды «R»

ЦМФ, математические финансы

# Содержание

- операции с числами и векторами
- массивы и матрицы
- циклы и условия
- функции и графики
- загрузка пакетов и данных

# Операции с числами и векторами

# Присвоение значений

# создание скалярной переменной

```
x <- 5
```

```
x
```

```
[1] 5
```

# создание векторной переменной

```
x <- c(1,2,3)
```

```
[1] 1 2 3
```

```
x <- 1:5
```

```
[1] 1 2 3 4 5
```

# повторение числа (вектора) несколько раз

```
y <- rep(0,times=2)
```

```
[1] 0 0
```

```
y <- rep(x,times=2)
```

```
[1] 1 2 3 4 5 1 2 3 4 5
```

# Векторная арифметика

# операции с векторами выполняются покомпонентно

# вектора меньшей длины повторяются несколько раз

# размеры векторов должны быть кратными

```
x + 1
```

```
[1] 2 3 4 5 6
```

```
x + c(1:3, 4, 4)
```

```
[1] 2 4 6 8 9
```

# допустимые операции: +, -, \*, /, ^

# арифметические функции:

# log(), exp(), sqrt(), sin(), cos(), tan(), min(), max(), abs(), ...

# константа «пи»

```
pi
```

```
[1] 3.141593
```

# Статистические функции

# диапазон значений: `range(x) == c(min(x),max(x))`

```
range(x)
```

```
[1] 1 5
```

# длина вектора

```
length(x)
```

```
[1] 5
```

# сумма и произведение элементов

```
sum(x)
```

```
[1] 15
```

```
prod(x)
```

```
[1] 120
```

# среднее и дисперсия

```
mean(x)
```

```
[1] 3
```

```
var(x)
```

```
[1] 2.5
```

# Сортировка значений и последовательности

# сортировка

```
y <- c(10,-3,6,0)
sort(y,decreasing=FALSE)
[1] -3 0 6 10
sort(y,decreasing=TRUE)
[1] 10 6 0 -3
```

# последовательность с заданным шагом

```
seq(0,1,by=0.2)
[1] 0 0.2 0.4 0.6 0.8 1
```

# последовательность заданной длины

```
seq(0,9,length=4)
[1] 0 3 6 9
```

# Логические векторы

# операции сравнения: <, >, <=, >=, ==, !=

```
y <- x > 3
```

```
[1] FALSE FALSE FALSE TRUE TRUE
```

# логические операторы: &, |, !

```
y <- ( (x>=3) & (x<=4) )
```

```
[1] FALSE FALSE TRUE TRUE FALSE
```

```
y <- ( (x<3) | (x>4) )
```

```
[1] TRUE TRUE FALSE FALSE TRUE
```

```
!y
```

```
[1] FALSE FALSE TRUE TRUE FALSE
```

# в арифметических операциях TRUE == 1, FALSE == 0

```
sum(x>3)
```

```
[1] 2
```



# Обращение к элементам вектора

# с помощью логического вектора

```
x[y]
```

```
[1] 1 2 5
```

# с помощью набора положительных чисел

```
x[1:3]
```

```
[1] 1 2 3
```

# с помощью набора отрицательных чисел,

# убирая указанные элементы

```
x[-(1:3)]
```

```
[1] 4 5
```

# с помощью названий элементов

```
names(x) <- c("mon", "tue", "wed", "thu", "fri")
```

```
x[c("mon", "thu")]
```

```
mon thu
```

```
1 4
```

# Массивы и матрицы

# Создание массива

# изменение размерности вектора

```
z <- 1:1500
```

```
dim(z) <- c(3,5,100)
```

# в результате z — матрица 3 на 5 на 100

# произведение размерностей равно длине вектора

# порядок значений: z[1,1,1], z[2,1,1], ..., z[2,5,100], z[3,5,100]

# с помощью функций matrix() и array()

```
z <- matrix(1:20,nrow=5,ncol=4)
```

```
z <- array(1:20,dim=c(5,4))
```

	[,1]	[,2]	[,3]	[,4]
[1,]	1	6	11	16
[2,]	2	7	12	17
[3,]	3	8	13	18
[4,]	4	9	14	19
[5,]	5	10	15	20

# Обращение к элементам матрицы

```
z <- array(1:20,dim=c(5,4))
```

	[,1]	[,2]	[,3]	[,4]
[1,]	1	6	11	16
[2,]	2	7	12	17
[3,]	3	8	13	18
[4,]	4	9	14	19
[5,]	5	10	15	20

```
z[,1]
```

```
[1] 1 2 3 4 5
```

```
z[1,]
```

```
[1] 1 6 11 16
```

```
z[1:2,1:2]
```

	[,1]	[,2]
[1,]	1	6
[2,]	2	7

# Действия с матрицами

# транспонирование

```
tz <- t(z)
```

# обращение

```
inv.z <- solve(z)
```

# умножение

```
z.tz <- z %*% tz
```

# создание диагональной матрицы

```
y <- diag(1:5)
```

# собственные значения и вектора

```
e <- eigen(y)
```

```
e$vectors; e$values
```

# квадратный корень из матрицы

```
y.sqrt <- e$vectors %*% diag(sqrt(e$values)) %*% t(e$vectors)
```

# Циклы и условия

# Циклы и условия

## # цикл с предусловием

```
y <- numeric(); i <- 1
while (i <= 5) {
  y[i] <- i^2
  i <- i + 1
}
[1] 1 4 9 16 25
```

## # ЦИКЛ «ОТ-ДО»

```
y <- numeric()
for (i in 1:5) {
  y[i] <- i^2
}
[1] 1 4 9 16 25
```

## # оператор условия

```
if (y[1] == 1) {
  y <- y + 5
}
[1] 5 9 14 21 30
```

# Пользовательские функции и рисование графиков



# Пользовательские функции

# объявляем функцию  $f(x_1, x_2) = c - x_1^\alpha - x_2^\alpha$

```
f <- function(x1,x2,c,alpha) {  
  c - x1^alpha - x2^alpha  
}
```

```
f(0.5,0.5,1,2)
```

```
[1] 0.5
```

# заданные при объявлении функции значения параметров

# воспринимаются как значения «по умолчанию»

```
f <- function(x1,x2,c=1,alpha=2) c - x1^alpha - x2^alpha
```

```
f(0.5,0.5)
```

```
[1] 0.5
```

# обратиться к функции также можно, явно задав параметры

```
f(alpha=2,x2=0.5,x1=0.5,c=1)
```

```
[1] 0.5
```

# Двумерные графики

```
g <- function(x,c=1) c*abs(x)^0.5  
x <- seq(-5,5,length=101)
```

# команда «**plot**» рисует новый график

```
plot(x,g(x),type="l",lty="solid",xlim=c(-5,5),ylim=c(0,2.5),  
main="График",xlab="Аргумент",ylab="Функция")
```

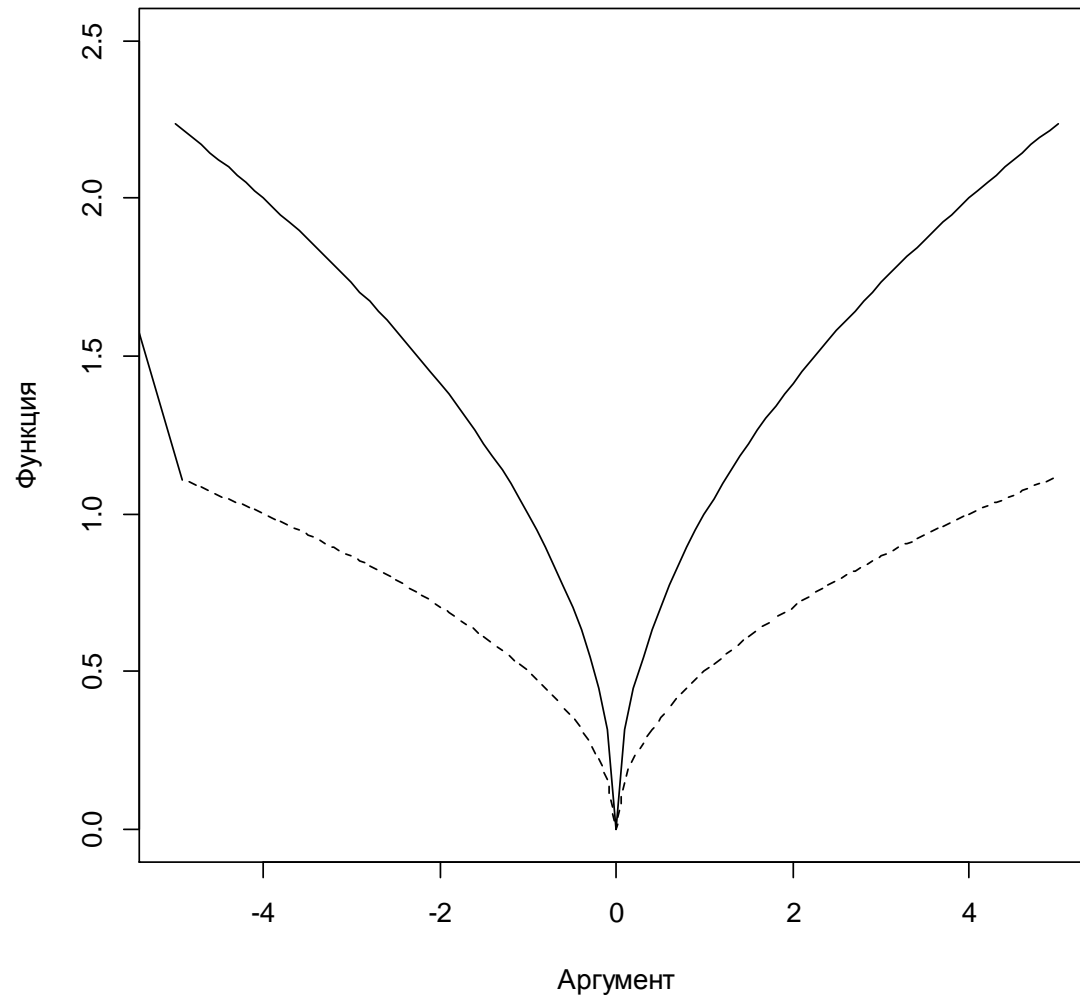
- **type** — вид графика: "l" — линии, "p" — точки, ...
- **lty** — вид линии: "solid" — сплошная, "dashed" — пунктирная, "dotted" — точками, ...
- **xlim, ylim** — границы графика по осям x и y
- **main, xlab, ylab** — подписи графика и осей

# «**lines**» добавляет линии на существующий график

```
lines(x,g(x,c=0.5),lty="dashed")
```

# Двумерный график

График



# Трёхмерные графики и линии уровня

```
x <- y <- seq(-5,5,length=51)
z <- outer(x,y,f)
```

# команда «**persp**» рисует трёхмерный график

```
persp(x,y,z,theta=30,phi=10,col="white",ticktype="detailed")
```

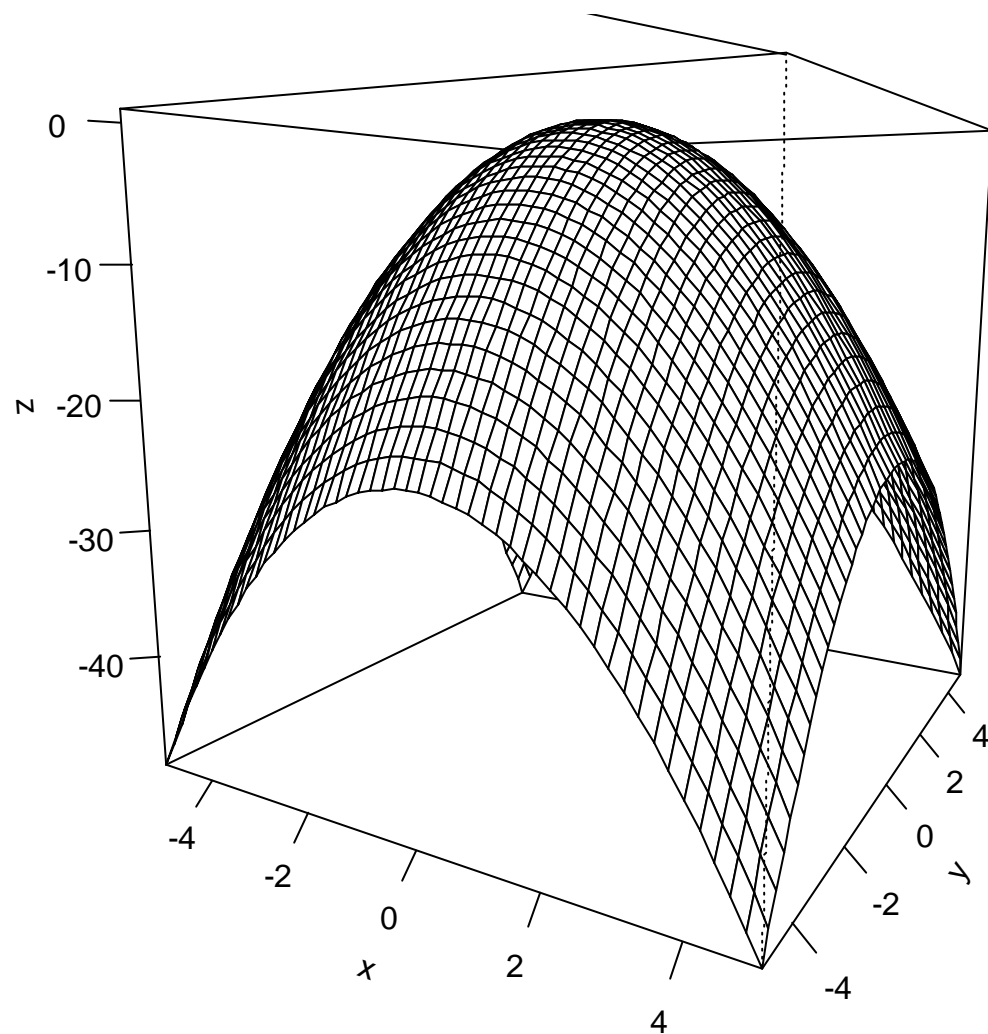
- **theta, phi** — углы обзора по горизонтали и вертикали
- **col** — цвет графика (см. функцию `colors()`)
- **ticktype** — маркировка осей: "detailed" — точные значения, "simple" — только стрелки
- **xlim, ylim, main, xlab, ylab** — аналогично команде «**plot**»

# «**contour**» рисует линии уровня функции f

```
contour(x,y,z,nlevels=5)
```

- **nlevels** — количество линий уровня

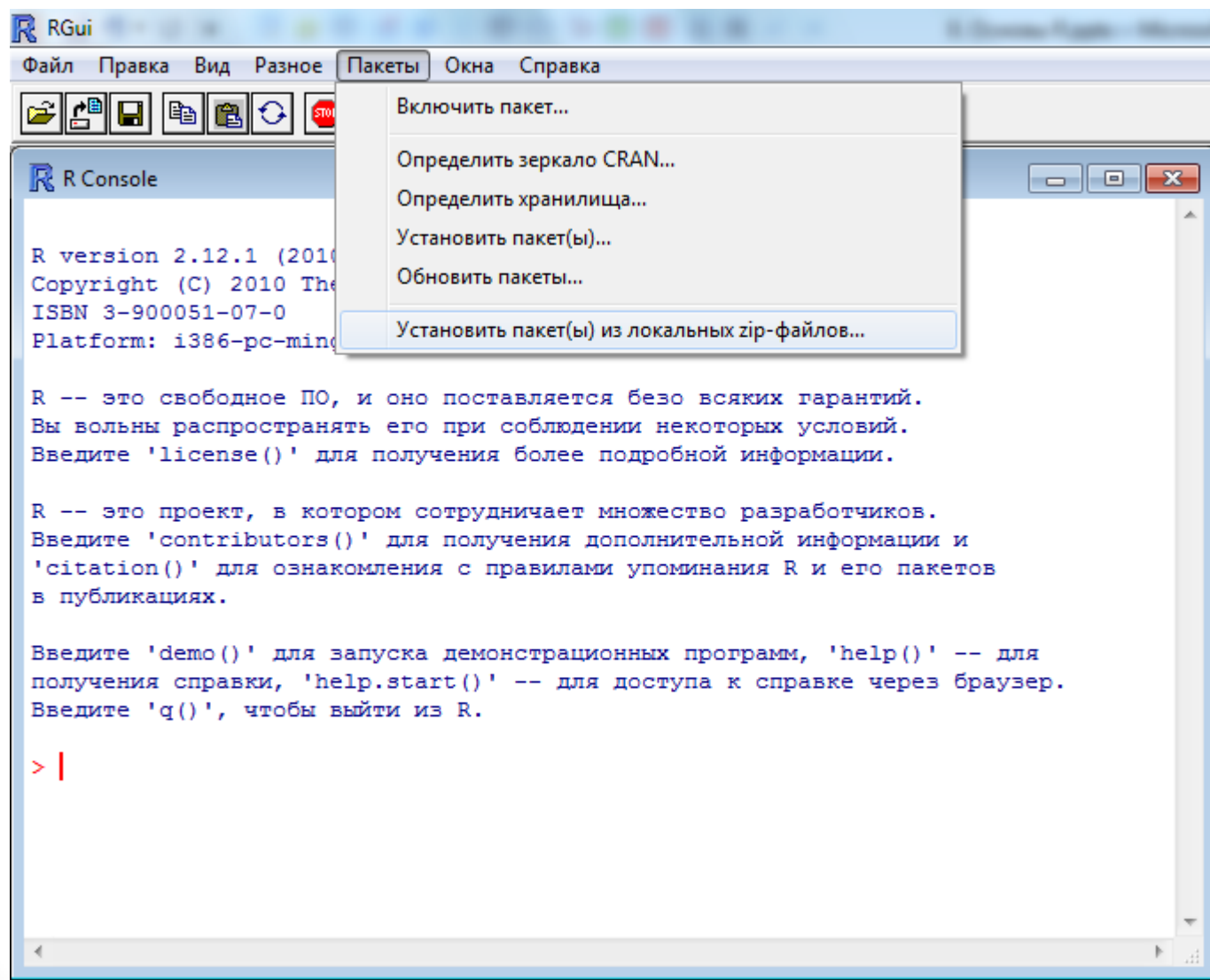
# Трёхмерный график



# Установка пакетов, загрузка и сохранение данных

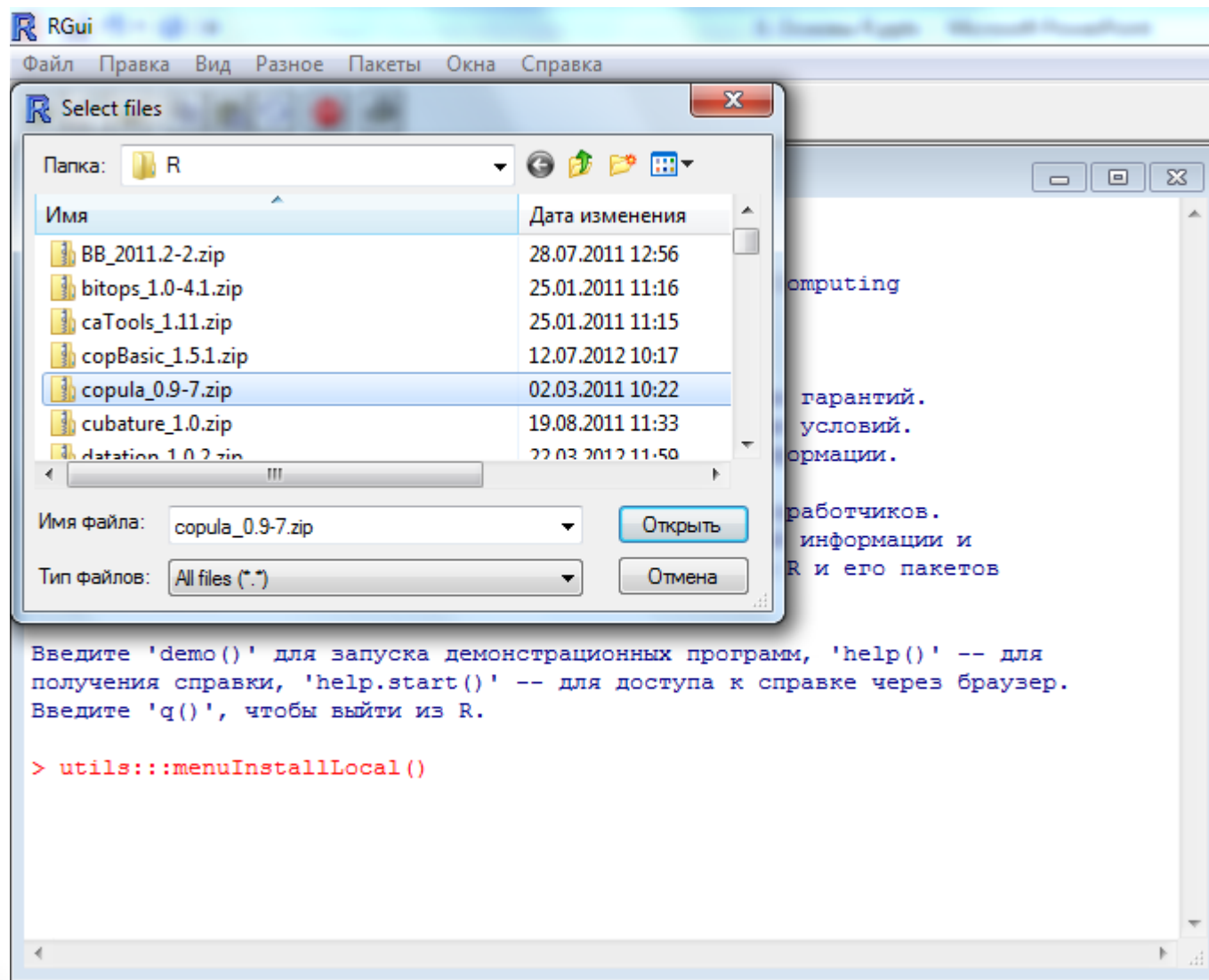
# Установка пакетов из локального диска

## Шаг 1. Выбор команды из меню



# Установка пакетов из локального диска

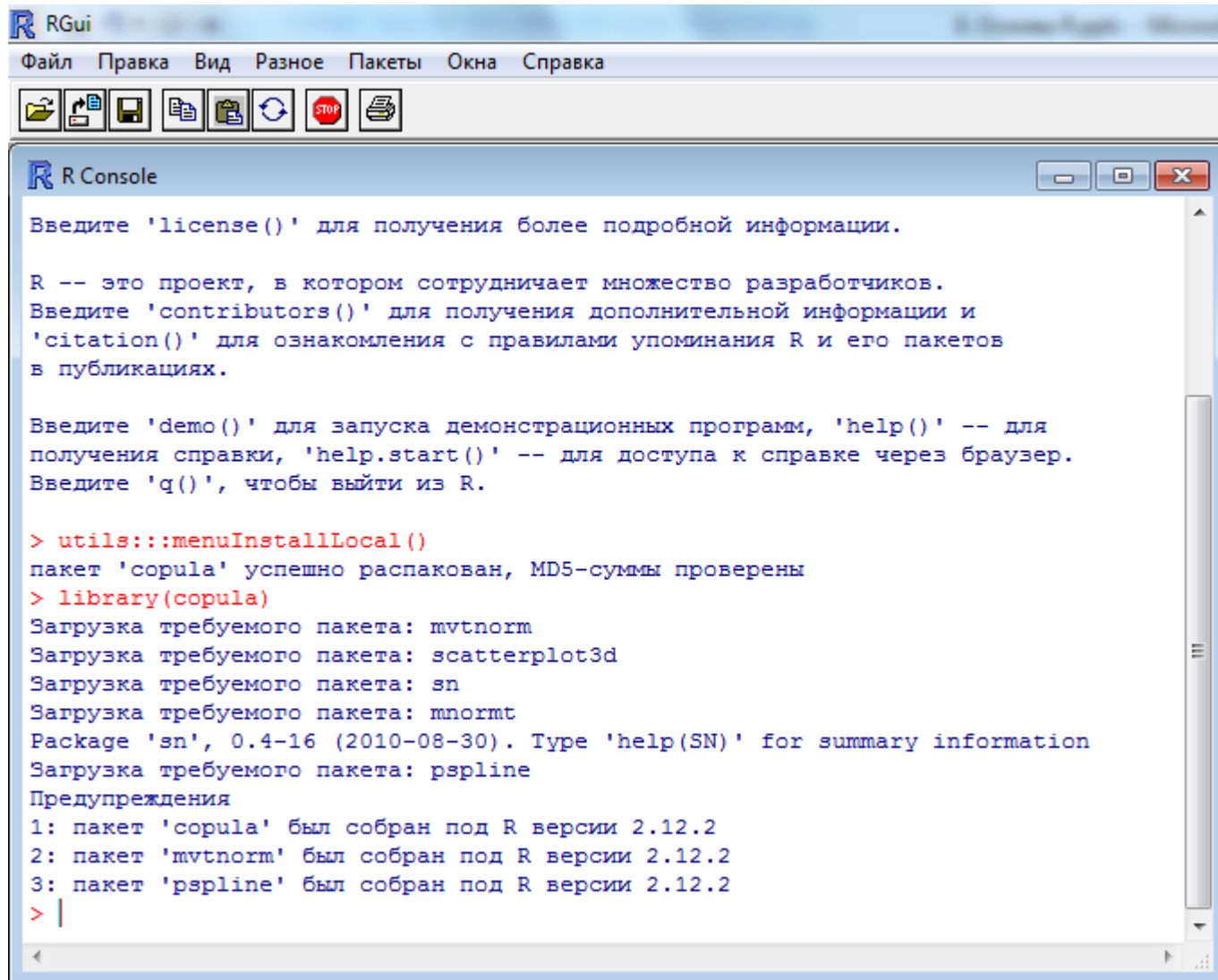
## Шаг 2. Выбор пакетного zip-файла





# Установка пакетов из локального диска

## Шаг 3. Объявление библиотеки

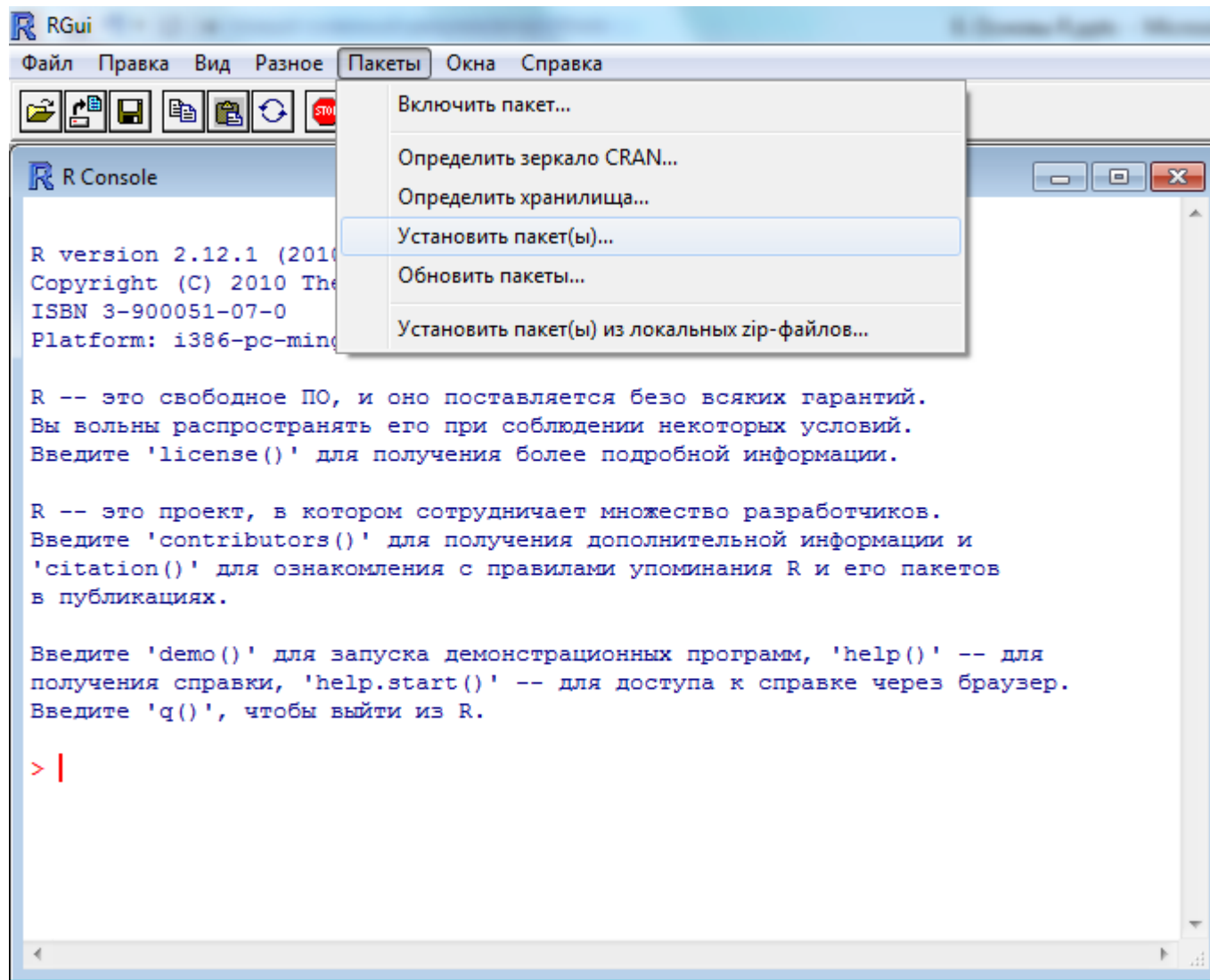


The screenshot shows the RGui application window. The menu bar includes 'Файл', 'Правка', 'Вид', 'Разное', 'Пакеты', 'Окна', and 'Справка'. The toolbar contains icons for file operations and execution. The R Console window is active, displaying the following text:

```
Введите 'license()' для получения более подробной информации.  
  
R -- это проект, в котором сотрудничает множество разработчиков.  
Введите 'contributors()' для получения дополнительной информации и  
'citation()' для ознакомления с правилами упоминания R и его пакетов  
в публикациях.  
  
Введите 'demo()' для запуска демонстрационных программ, 'help()' -- для  
получения справки, 'help.start()' -- для доступа к справке через браузер.  
Введите 'q()', чтобы выйти из R.  
  
> utils:::menuInstallLocal()  
пакет 'copula' успешно распакован, MD5-суммы проверены  
> library(copula)  
Загрузка требуемого пакета: mvtnorm  
Загрузка требуемого пакета: scatterplot3d  
Загрузка требуемого пакета: sn  
Загрузка требуемого пакета: mnormt  
Package 'sn', 0.4-16 (2010-08-30). Type 'help(SN)' for summary information  
Загрузка требуемого пакета: pspline  
Предупреждения  
1: пакет 'copula' был собран под R версии 2.12.2  
2: пакет 'mvtnorm' был собран под R версии 2.12.2  
3: пакет 'pspline' был собран под R версии 2.12.2  
> |
```

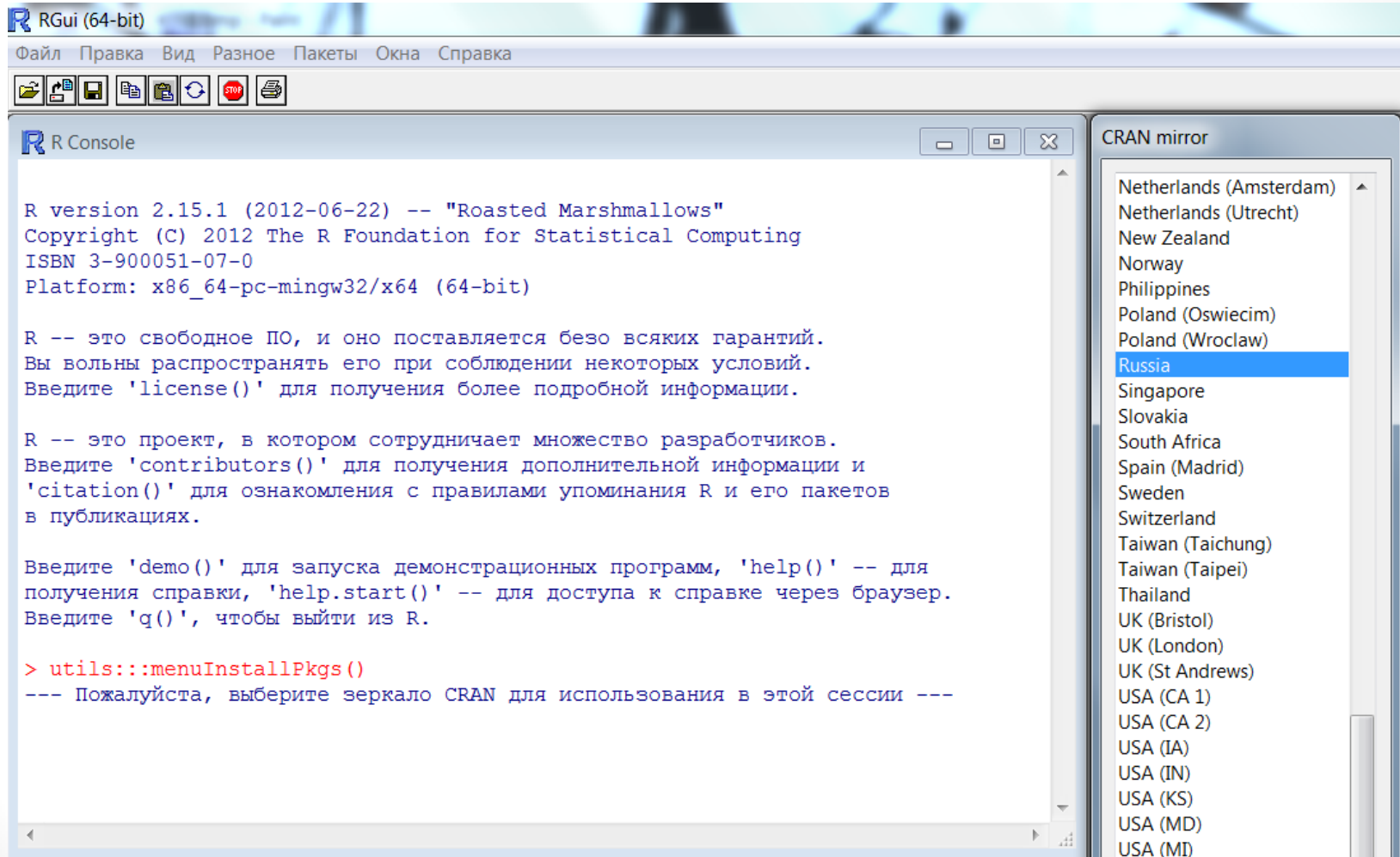
# Установка пакетов с сайта R

## Шаг 1. Выбор команды из меню



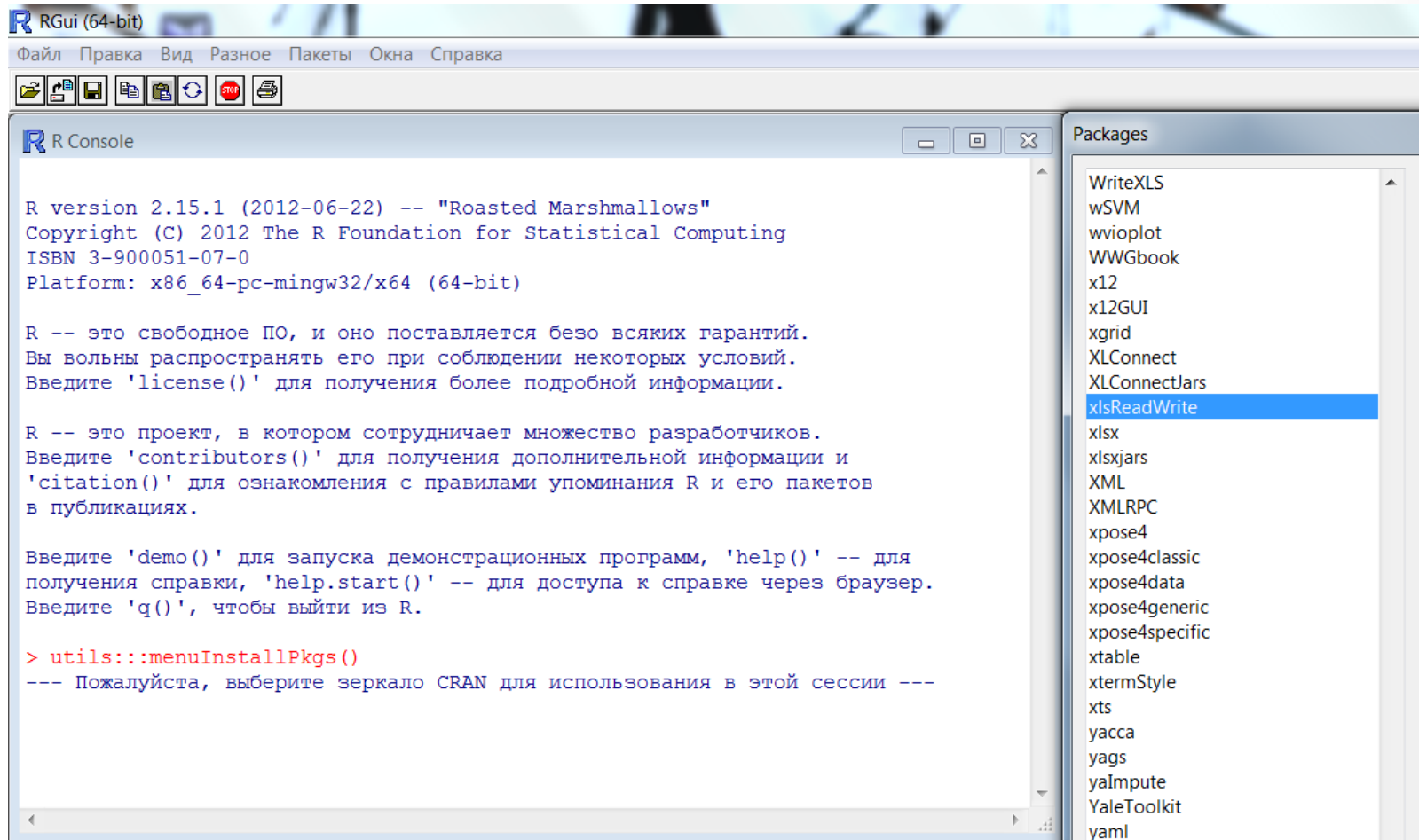
# Установка пакетов с сайта R

## Шаг 2. Выбор зеркала



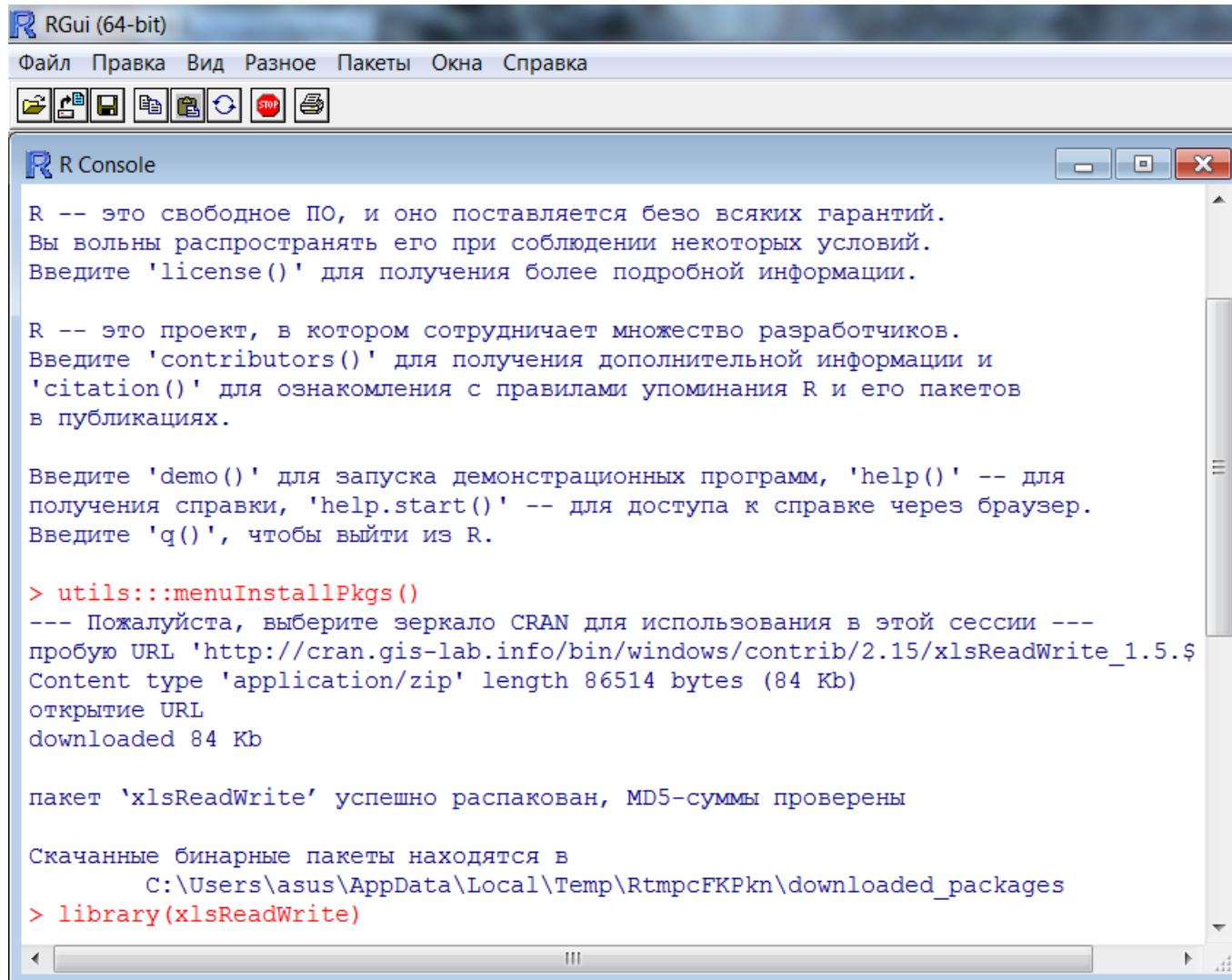
# Установка пакетов с сайта R

## Шаг 3. Выбор пакета



# Установка пакетов с сайта R

## Шаг 4. Объявление библиотеки



```
RGui (64-bit)
Файл  Правка  Вид  Разное  Пакеты  Окна  Справка

R Console

R -- это свободное ПО, и оно поставляется безо всяких гарантий.
Вы вольны распространять его при соблюдении некоторых условий.
Введите 'license()' для получения более подробной информации.

R -- это проект, в котором сотрудничает множество разработчиков.
Введите 'contributors()' для получения дополнительной информации и
'citation()' для ознакомления с правилами упоминания R и его пакетов
в публикациях.

Введите 'demo()' для запуска демонстрационных программ, 'help()' -- для
получения справки, 'help.start()' -- для доступа к справке через браузер.
Введите 'q()', чтобы выйти из R.

> utils::menuInstallPkgs()
--- Пожалуйста, выберите зеркало CRAN для использования в этой сессии ---
пробую URL 'http://cran.gis-lab.info/bin/windows/contrib/2.15/xlsReadWrite_1.5.$
Content type 'application/zip' length 86514 bytes (84 Kb)
открытие URL
downloaded 84 Kb

пакет 'xlsReadWrite' успешно распакован, MD5-суммы проверены

Скачанные бинарные пакеты находятся в
C:\Users\asus\AppData\Local\Temp\RtmpcFKPkn\downloaded_packages
> library(xlsReadWrite)
```

# Загрузка и сохранение данных

# загрузка данных из Excel

# (\*.xlsx-файл должен существовать)

```
library(xlsx)
```

```
dat <- read.xlsx("C:/R/input.xlsx", sheetIndex=1, header=TRUE)
```

```
dat <- as.matrix(dat)
```

	V1	V2	V3
1	1	0	0
2	0	1	1
3	1	1	0

# экспорт данных в Excel

# (\*.xlsx-файл может не существовать)

```
write.xlsx(dat, "C:/R/output.xlsx", sheetName="Таблица1",  
           col.names=TRUE, row.names=FALSE, append=FALSE)
```

# Загрузка и сохранение данных

# загрузка данных из \*.csv

```
dat <- read.csv("C:/R/input.csv", header=TRUE, sep=",")
```

- **header** == TRUE, если в файле есть заголовки столбцов
- **sep** — разделитель столбцов ("," для \*.csv)

# загрузка данных из \*.txt

```
dat <- read.table("C:/R/input.txt",  
  header=TRUE, sep="," ,dec=".")
```

- **dec** — разделитель целой и дробной частей
- **header, sep** — аналогично **read.csv**

# Домашнее задание

- загрузить в R котировки трёх акций или индексов
- перейти от цен финансовых инструментов к их доходностям
- построить графики цен и доходностей
- написать пользовательскую функцию, при подстановке в которую вектора доходностей актива, возвращается средняя доходность и её стандартное отклонение

Источник данных — сайты [finam.ru](http://finam.ru), [finance.yahoo.com](http://finance.yahoo.com) и др.