R

Вебинары

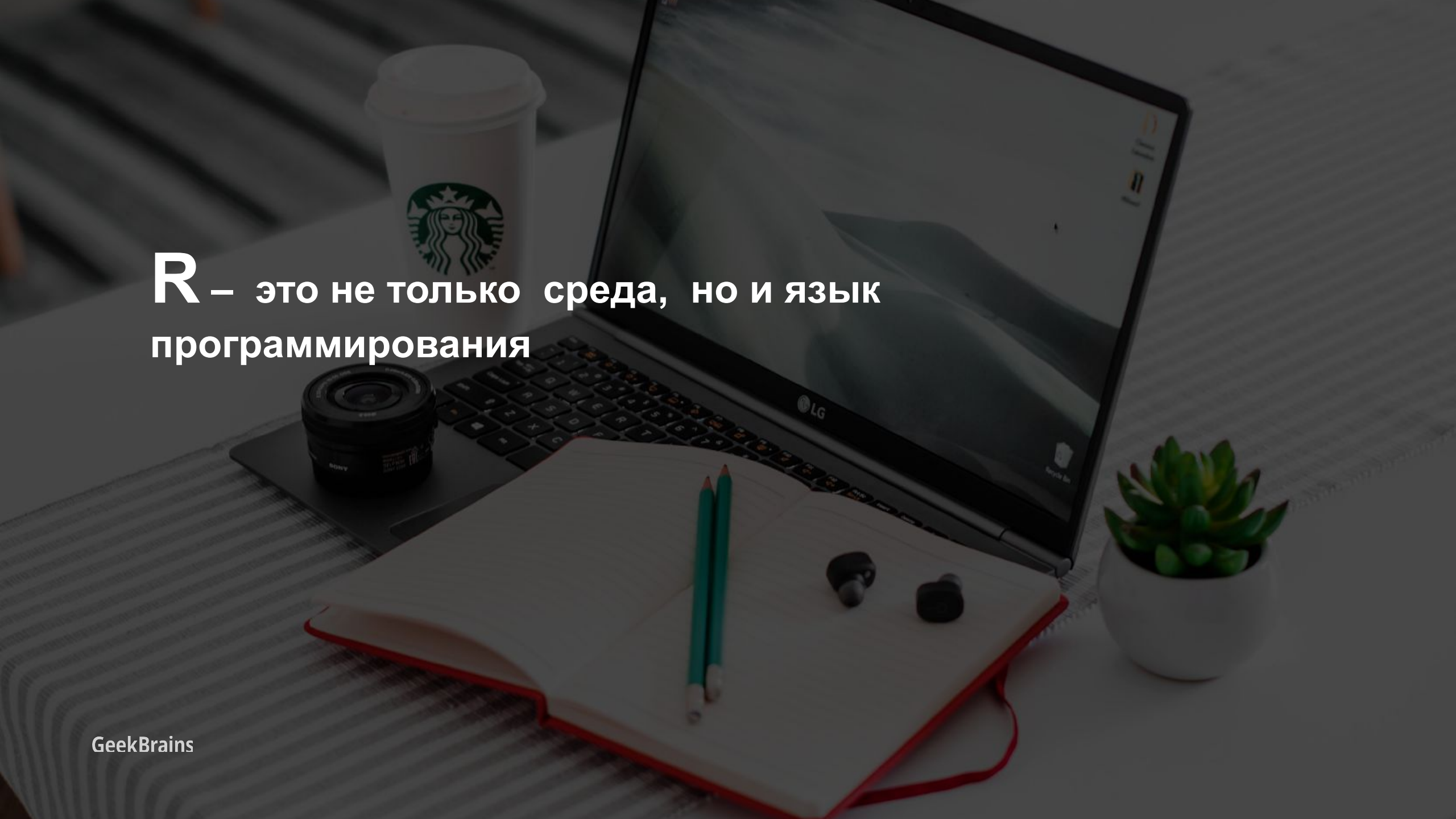
GeekBrains

Урок 1

ЗНАКОМСТВО И НАЧАЛО РАБОТЫ В R

На этом уроке мы изучим:

1. История R
2. Преимущества
3. Установка R и R Studio
4. Объекты R
5. Математические функции
6. Векторы и последовательности
7. Логические операторы
8. Построение матицы и dataframe`a



R – это не только среда, но и язык программирования

GeekBrains

1. ИСТОРИЯ R

Язык R является диалектом языка S, созданного десятки лет назад крупной американской компанией Bell Labs , занимающейся исследованиями в области промышленности и науки.

С 2016 года - NOKIA Bell Labs

Язык S предназначался для статистического анализа и графического представления данных

На сегодняшний день- R –это насыщенный и полезный

инструментарий для проведения статистического анализа, который в последнее время с большей силой набирает свою популярность в кругах Data Science

Штаб-квартира Bell Labs в США

(с 2016 **NOKIA Bell Labs**)

https://en.wikipedia.org/wiki/Bell_Labs

2 Преимущества R

Существует организация R Foundation, которая не является коммерческой и ее основными целями являются:

-обеспечение непрерывного развития языка;

-поиск новых методов статистических исследований;

- обучение & проведение конференций.

Конференции

useR! — International R User Conference

DSC — Directions in Statistical Computing

Изображения взяты с сайта https://www.r-project.org/conferences/

На сайте

https://www.r-project.org

можно найти огромное количество

информации, касающейся

любых аспектов, связанных с R

Board
Members
Donors
Donate

Help With R

Getting Help

Documentation

Manuals
FAQs
The R Journal
Books
Certification
Other

Links

Bioconductor
Related Projects
GSoC

https://www.r-project.org/conferences/

useR! 2020 will be held in St. Louis, Missouri from July 7 to July 10 at the Marriott St. Louis Grand. Follow '@useR2020stl' for further news.

This is the main meeting of the R user and developer community, its program consisting of both invited and user-contributed presentations:

- The invited keynote lectures cover a broad spectrum of topics ranging from technical and R-related computing issues to general statistical topics of current interest.
- The user-contributed presentations are submitted as abstracts prior to the conference and may be related to (virtually) any R-related topic. The presentations are typically organized in sessions of regular talks, lightning talks, and poster presentations.
- The RFCC actively invites potential local organisers to submit proposals regarding future useR! conferences. Groups thinking about hosting a useR! conference are welcome to contact the RFCC by email at R-conferences@r-project.org.

Usually, no proceedings are published for useR! conferences.

- useR! 2019, Toulouse, France.
- useR! 2018, Brisbane, Australia. (local copy).
- useR! 2017, Brussels, Belgium. (local copy).
- useR! 2016, Stanford, CA, USA. (local copy).
- useR! 2015, Aalborg, Denmark. (local copy)

Проект R	**https://www.r-project.org**
R Foundation - организация, занимающаяся развитием R	https://www.r-project.org/foundation/
Конференции по R	https://www.r-project.org/conferences/
UseR! 2019 (во Франции) здесь доступна ссылка на видео докладчиков конференции, также можно найти учебные пособия (tutorials)	https://user2019.r-project.org/
Огромный список полезной литературы в удобном для Вас формате, охватывающий огромный спектр тем: от установки R до машинного обучения	https://www.r-project.org/doc/bib/R-books.html
Различные руководства (целевая аудитория- от новичков до продвинутых пользователей)	https://cran.r-project.org/manuals.html
Скачать R	https://cran.r-project.org

ПОЧЕМУ R?

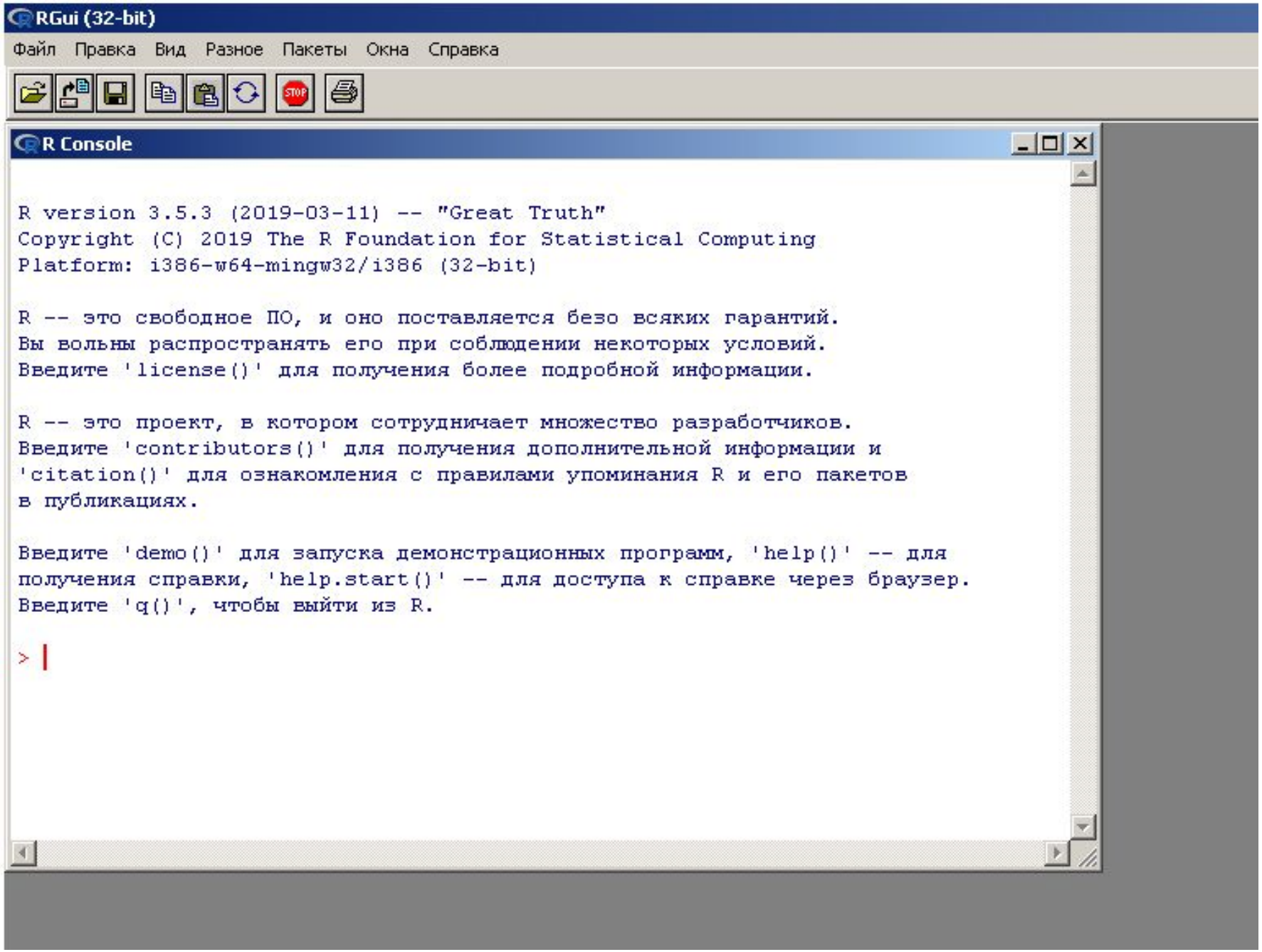
1. Бесплатный и открытый ресурс
2. Масса полезной литературы, записи конференций, обучающие пособия, доступная связь с разработчиками пакетов, форумы
3. Постоянно развивающийся язык. Также сообщество R занимается поиском новых статистических методов. Необходимый для них инструментарий в первую очередь появляется на языке R
4. Возможность совершенствовать язык, создавая свои функции и пакеты
5. R может работать Linux, MacOS, Windows
6. R & C++
7. Имеет мощный графический инструментарий
8. Позволяет проводить интерактивный анализ
9. Слово «среда» подразумевает полный комплекс гибких инструментов, «заточенных» под статистический анализ

Инсталляция R

https://cran.r-project.org

1. Запуск инсталляции должен быть как от администратора
2. Открываем и запускаем скаченный файл
3. Не нужно менять никакие настройки

(рекомендация: выбрать английский язык)


RGui (32-bit)
Файл Правка Вид Разное Пакеты Окна Справка
R Console
R version 3.5.3 (2019-03-11) -- "Great Truth"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)
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.
>

Установка R Studio

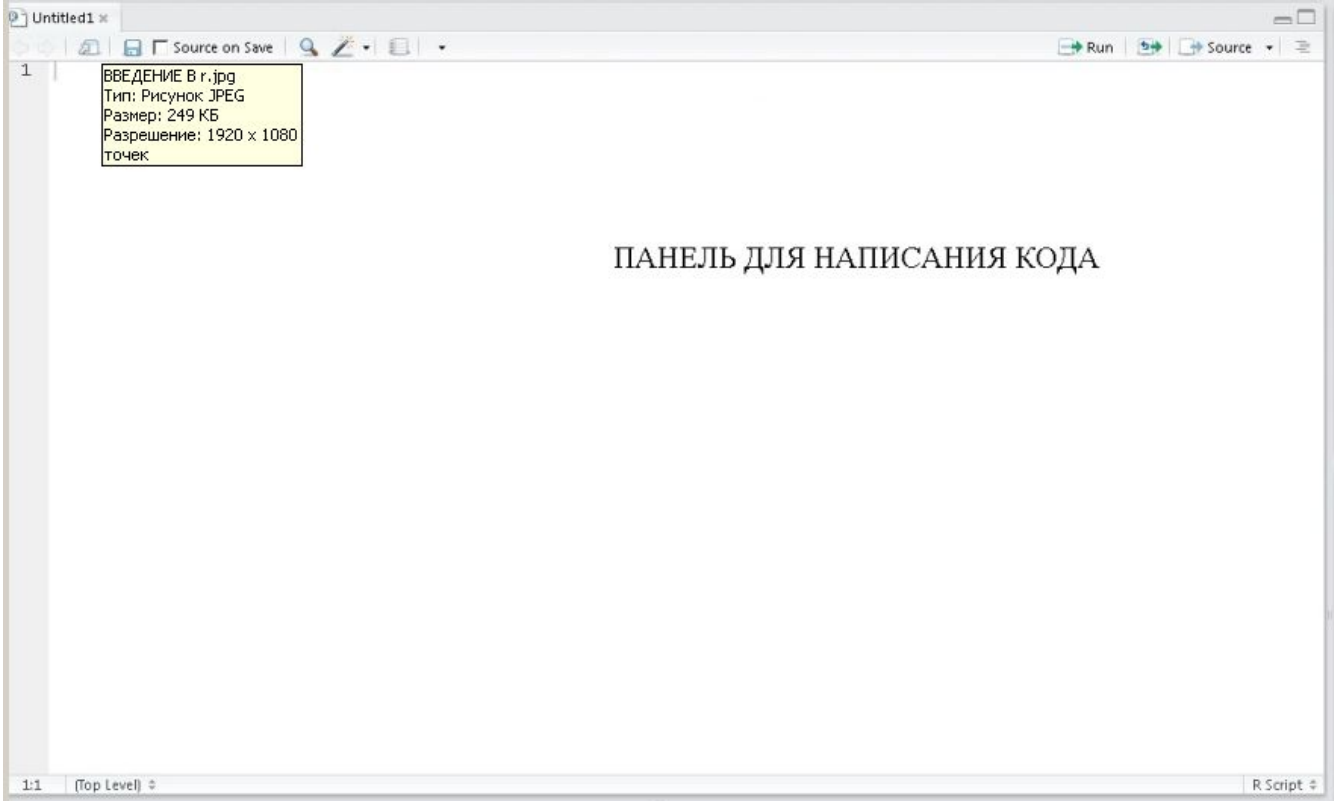
Зачем устанавливать R Studio ?

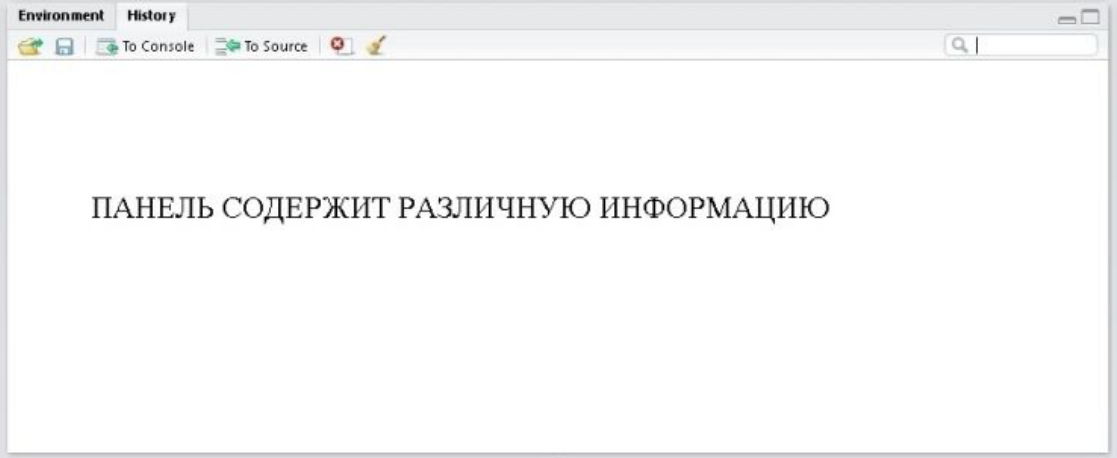
R Studio упрощает работу с R

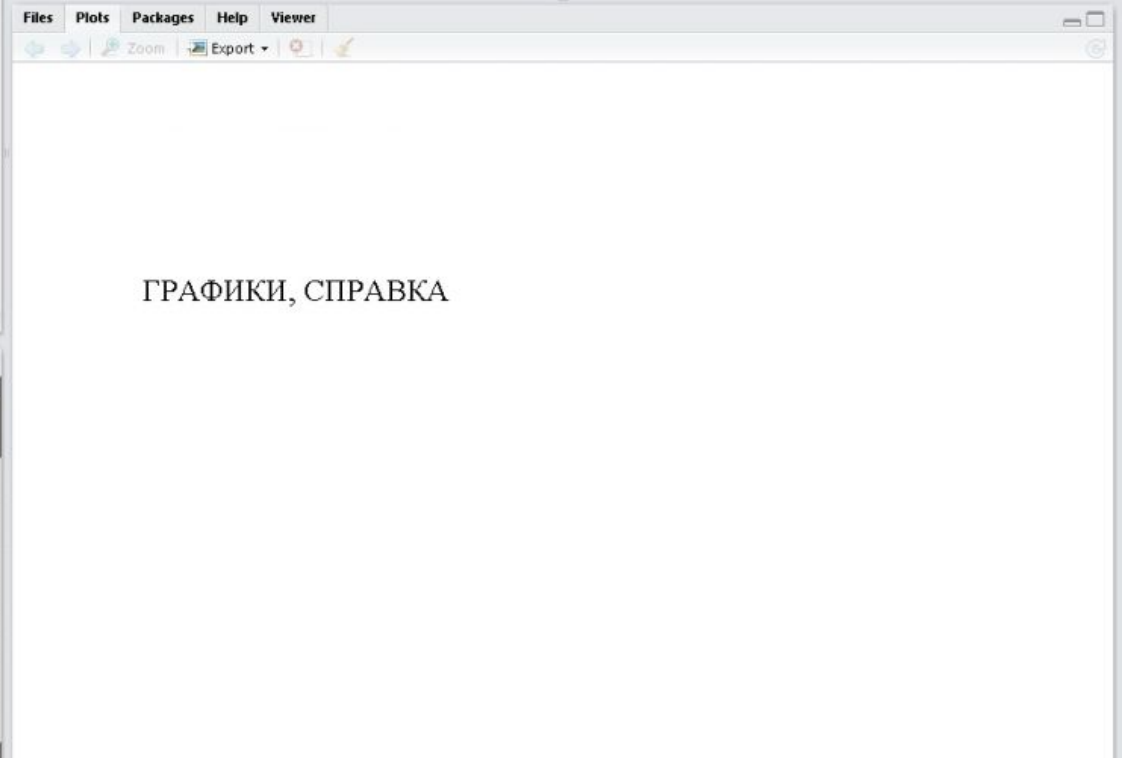
Ссылка для скачивания файла

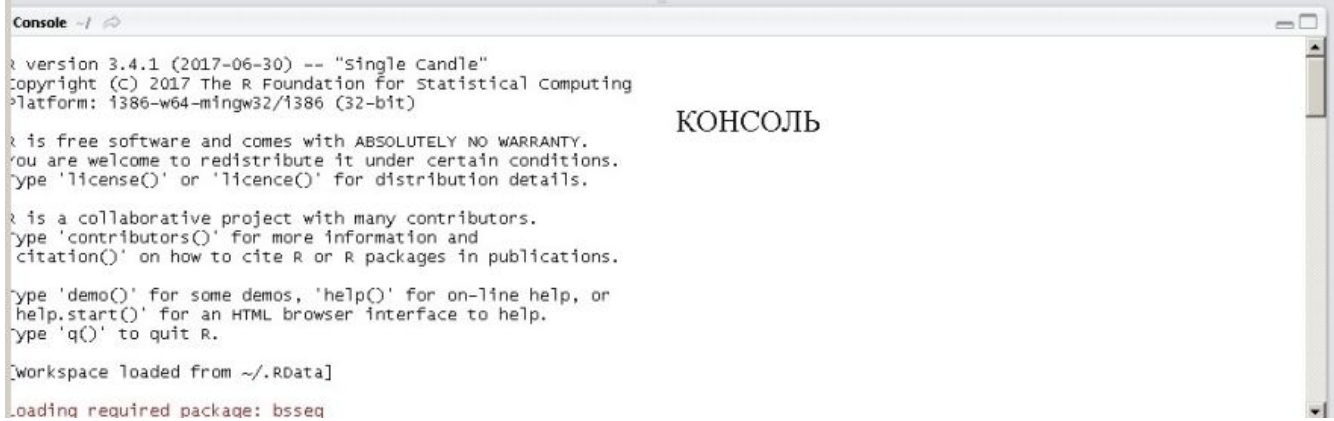
- https://www.rstudio.com

- При инсталляции файла
  не нужно менять никаких настроек


Untitled1
Source on Save
Run
Source
1
ВВЕДЕНИЕ В r.jpg
Тип: Рисунок JPEG
Размер: 249 КБ
Разрешение: 1920 x 1080
точек
ПАНЕЛЬ ДЛЯ НАПИСАНИЯ КОДА
1:1
(Top Level)
R Script


Environment
History
To Console
To Source
ПАНЕЛЬ СОДЕРЖИТ РАЗЛИЧНУЮ ИНФОРМАЦИЮ


Files
Plots
Packages
Help
Viewer
Zoom
Export
ГРАФИКИ, СПРАВКА


Console ~/
R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from ~/.RData]
Loading required package: bsseq
КОНСОЛЬ

СПИСОК БЫСТРЫХ КЛАВИШ В RStudio

Чтобы отобразить список быстрых клавиш нажмите:

«Alt»+ «Sift»+ «K»

Keyboard Shortcut Quick Reference

Tabs

Ctrl+Shift+.	Switch to Tab...
Ctrl+Tab	Next Tab
Ctrl+Shift+Tab	Previous Tab
Ctrl+Shift+F11	First Tab
Ctrl+Shift+F12	Last Tab

Panes

Ctrl+1	Move Focus to Source
Ctrl+2	Move Focus to Console
Shift+Alt+T	Move Focus to Terminal
Ctrl+3	Move Focus to Help
Ctrl+4	Show History
Ctrl+5	Show Files
Ctrl+6	Show Plots
Ctrl+7	Show Packages
Ctrl+8	Show Environment
Ctrl+9	Show Viewer
Ctrl+F1	Show Vcs
Ctrl+F2	Show Build
Ctrl+F5	Show Connections
Ctrl+Shift+Alt+O	Show All Panes

Add Shift to zoom (maximize) pane.

Files

Ctrl+S	Save
Ctrl+Alt+S	Save All
Ctrl+Shift+N	New R Script
Ctrl+O	Open File...
Ctrl+W	Close
Ctrl+Shift+Alt+W	Close All Except Current
Ctrl+Shift+W	Close All
Ctrl+Shift+F	Find in Files...

Source Navigation

Ctrl+F9	Back
Ctrl+F10	Forward
Ctrl+Alt+U	Find Usages
Ctrl+F3	Use Selection for Find
Ctrl+F	Find...
F3	Find Next
Shift+F3	Find Previous
Ctrl+Shift+J	Replace and Find
Ctrl+.	Go To File/Function...
Shift+Alt+G	Go to Line...
Shift+Alt+J	Jump To...
Ctrl+P	Jump To Matching
Ctrl+Shift+Alt+E	Expand To Matching
Ctrl+Shift+O	Show Document Outline
Ctrl+Alt+Up	Add Cursor Above Current Cursor
Ctrl+Alt+Down	Add Cursor Below Current Cursor
Ctrl+Shift+Up	Expand Selection
Ctrl+Shift+Down	Shrink Selection
Ctrl+PgDn	Go to Next Section
Ctrl+PgUp	Go to Previous Section
Ctrl+Alt+A	Split Into Lines
Ctrl+Shift+Alt+Up	Move active cursor up
Ctrl+Shift+Alt+Down	Move active cursor down

Execute

Ctrl+Shift+S	Source Active File
Ctrl+Shift+Enter	Source with Echo
Ctrl+Alt+G	Source a File...
Ctrl+Shift+P	Re-Run Previous
Ctrl+Enter	Run Selected Line(s)
Alt+Enter	Run Line(s) without moving cursor
Ctrl+Alt+R	Run All
Ctrl+Alt+B	Run From Beginning To Line
Ctrl+Alt+E	Run From Line to End
Ctrl+Alt+F	Run Function Definition
Ctrl+Alt+T	Run Code Section
Ctrl+Alt+P	Run All Chunks Above
Ctrl+Shift+Enter	Run Current Chunk
Ctrl+Alt+N	Run Next Chunk

Source Editor

Ctrl+Alt+I	Insert Chunk
Ctrl+Shift+R	Insert Section...
Ctrl+Alt+X	Extract Function
Ctrl+Alt+V	Extract Variable
Ctrl+Shift+C	Comment/Uncomment Lines
Ctrl+I	Reindent Lines
Ctrl+Shift+/	Reflow Comment
Ctrl+Shift+A	Reformat Code
Ctrl+Shift+Alt+D	Show Diagnostics (Project)
Alt+L	Collapse Fold
Shift+Alt+L	Expand Fold
Alt+O	Collapse All Folds
Shift+Alt+O	Expand All Folds
Alt+Up	Move Lines Up
Alt+Down	Move Lines Down
Ctrl+D	Delete Line
Ctrl+U	Yank Line Up to Cursor
Ctrl+K	Yank Line After Cursor
Ctrl+Y	Insert Yanked Text
Alt+-	Insert Assignment Operator
Ctrl+Shift+M	Insert Pipe Operator
Ctrl+Shift+Alt+M	Rename in Scope
Ctrl+Shift+Alt+R	Insert Roxygen Skeleton
Shift+Tab	Insert Snippet

Debug

Shift+F9	Toggle Breakpoint
F10	Execute Next Line
Shift+F4	Step Into Function
Shift+F6	Finish Function/Loop
Shift+F5	Continue
Shift+F8	Stop Debugging

Source Control

Ctrl+Alt+D	Diff Files
Ctrl+Alt+M	Commit...

Build

Ctrl+Shift+K	Compile PDF
Ctrl+Shift+K	Preview
Ctrl+Shift+K	Knit Document
Ctrl+Shift+B	Install and Restart
Ctrl+Shift+L	Load All
Ctrl+Shift+E	Check Package
Ctrl+Shift+T	Test Package
Ctrl+Shift+D	Document

Console

Ctrl+L	Clear Console
Ctrl+Up	Popup Command History

Terminal

Shift+Alt+R	New Terminal
Ctrl+Alt+F11	Previous Terminal
Ctrl+Alt+F12	Next Terminal

Other

F1	Show Function Help
F2	Go To Function / File
Tab	Complete Code
Ctrl+Q	Quit Session...
Ctrl+Shift+F10	Restart R
Ctrl+Alt+F11	Previous Plot
Ctrl+Alt+F12	Next Plot
Ctrl+`	Request Log
Ctrl+Shift+`	Log focused element
Ctrl+Shift+H	Choose Directory...
Ctrl+F8	Sync PDF View to Editor
F7	Check Spelling...
Shift+Alt+K	Keyboard Shortcuts Help

Действия перед началом работы:

1.Создаем новый файл (File-> New File-> R Script) (рис1, сл16)

2. Создаем папку на своем компьютере, куда будем Помещать созданные файлы, скаченные дата фреймы

3 Устанавливаем рабочую директорию (Session -> Set Working Directory) . Выбираем путь к папке ,созданную в пункте 2

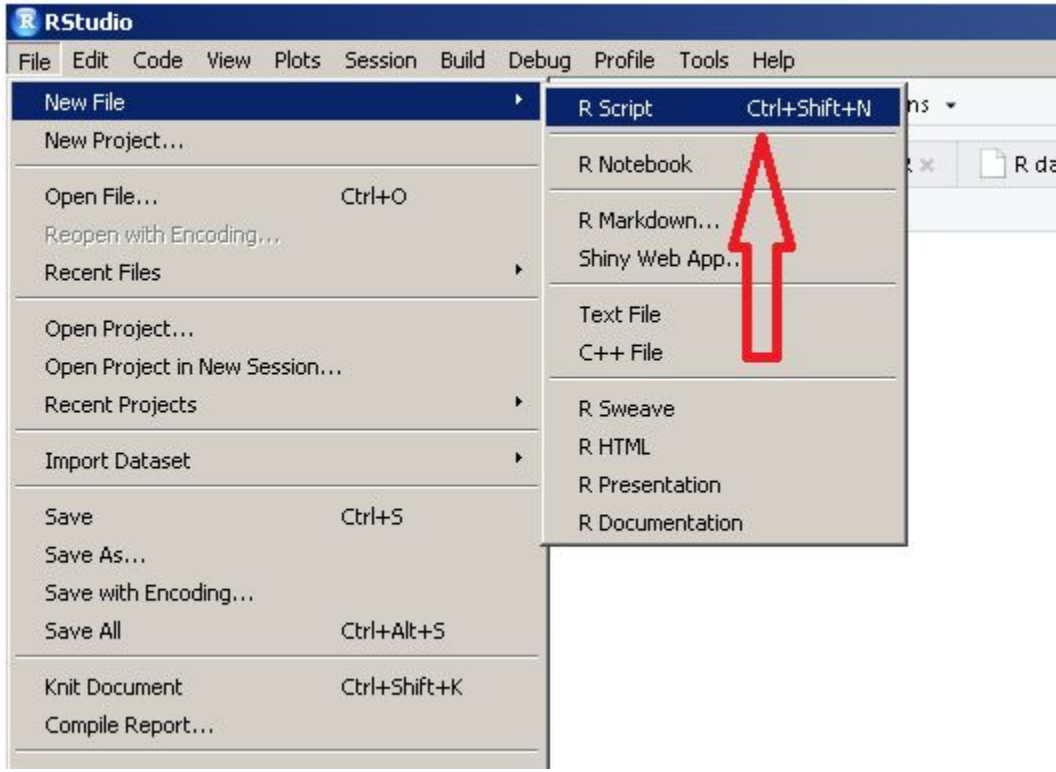


Рис.1, слайд 16

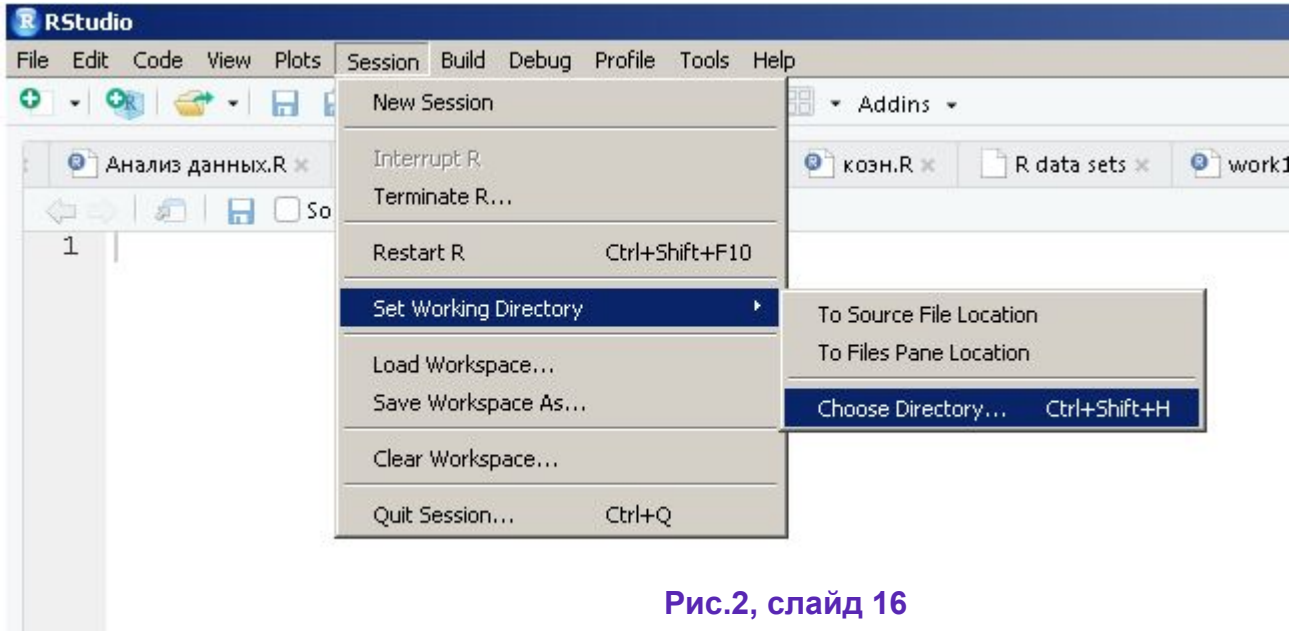


Рис.2, слайд 16

4. **Проверим, путь к рабочей директории с помощью функции**

getwd() Чтобы запустить код в консоли нажмите CTRL+Enter *

* **В зависимости от ОС эти клавиши могут меняться , проверить, какие клавиши запускают код у Вас , можно, как показано на слайде 15:**

«Alt»+ «Sift»+ «K» Execute -> Run Selected line

```
> getwd()
[1] "C:/Users/AD/Documents/R"
```

При запуске R в консоли появляется:

```
R version 3.5.3 (2019-03-11) -- "Great Truth"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]
```

Правила написания кода

- Наименование переменных начинать с буквы
- Вместо пробела в названии переменной использовать «.» или нижнее подчеркивание (например, x.1 или x_1)
- **Не использовать имена для переменных, которые совпадают с именами уже существующих объектов в R ,если это произошло используйте функцию rm()**
- Чтобы код был максимально читаемый использовать # для описания своих действий

Пример: (x<-10 # наименование переменной)

Основные объекты R

1. Переменная

```
> x.1 <- 17
> x.1
[1] 17
> rm(x.1)
> x.1
Error: object 'x.1' not found
>
```

2. Функции

2.1 Основные математические функции «+» , «-», «:», « * », степень «^», квадратный корень sqrt(), log2(), factorial (), exp ()

```
> 1+6
[1] 7
> 1-6
[1] -5
> 10:2
[1] 10  9  8  7  6  5  4  3  2
> 10*2
[1] 20
> sqrt(25)
[1] 5
> log2(8)
[1] 3
> log10(100)
[1] 2
> 10^3
[1] 1000
> exp(1)
[1] 2.718282
> log(exp(1))
[1] 1
```

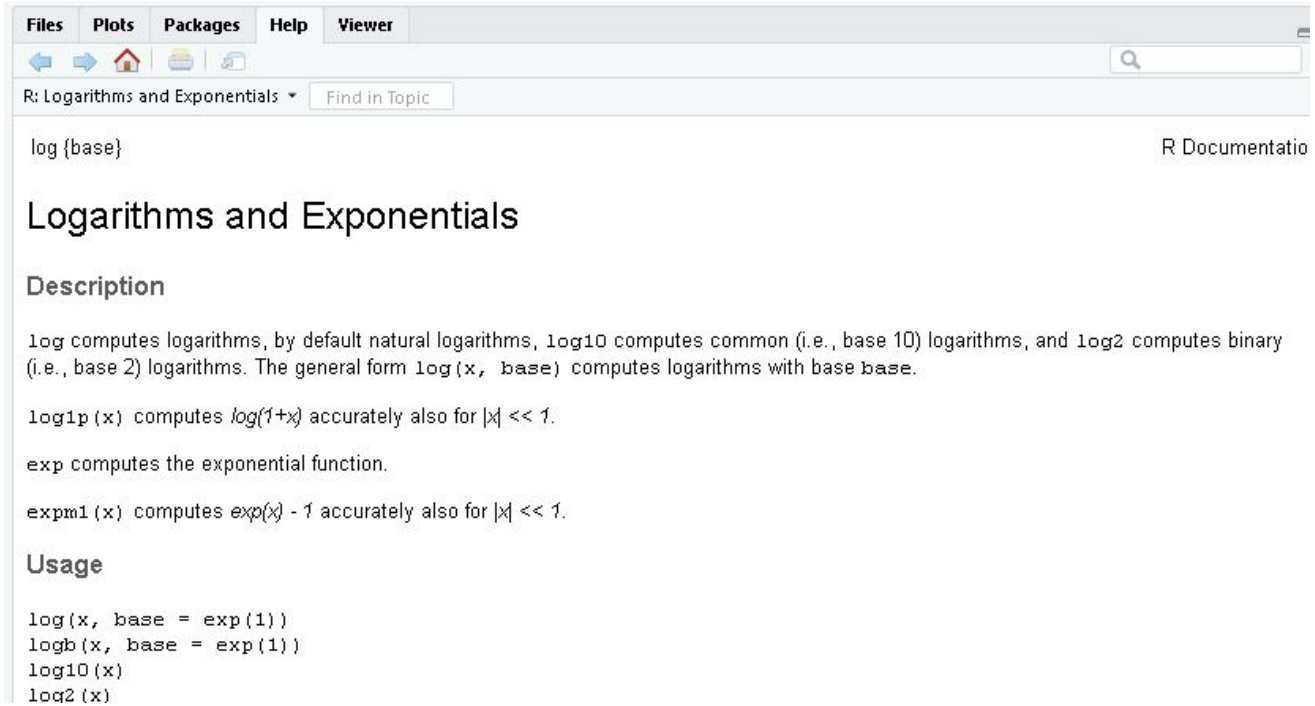
2.2 Как выполняется функция

sqrt (log2(16))

```
> log(exp(1))
[1] 1
> sqrt(log2(16))
[1] 2
```

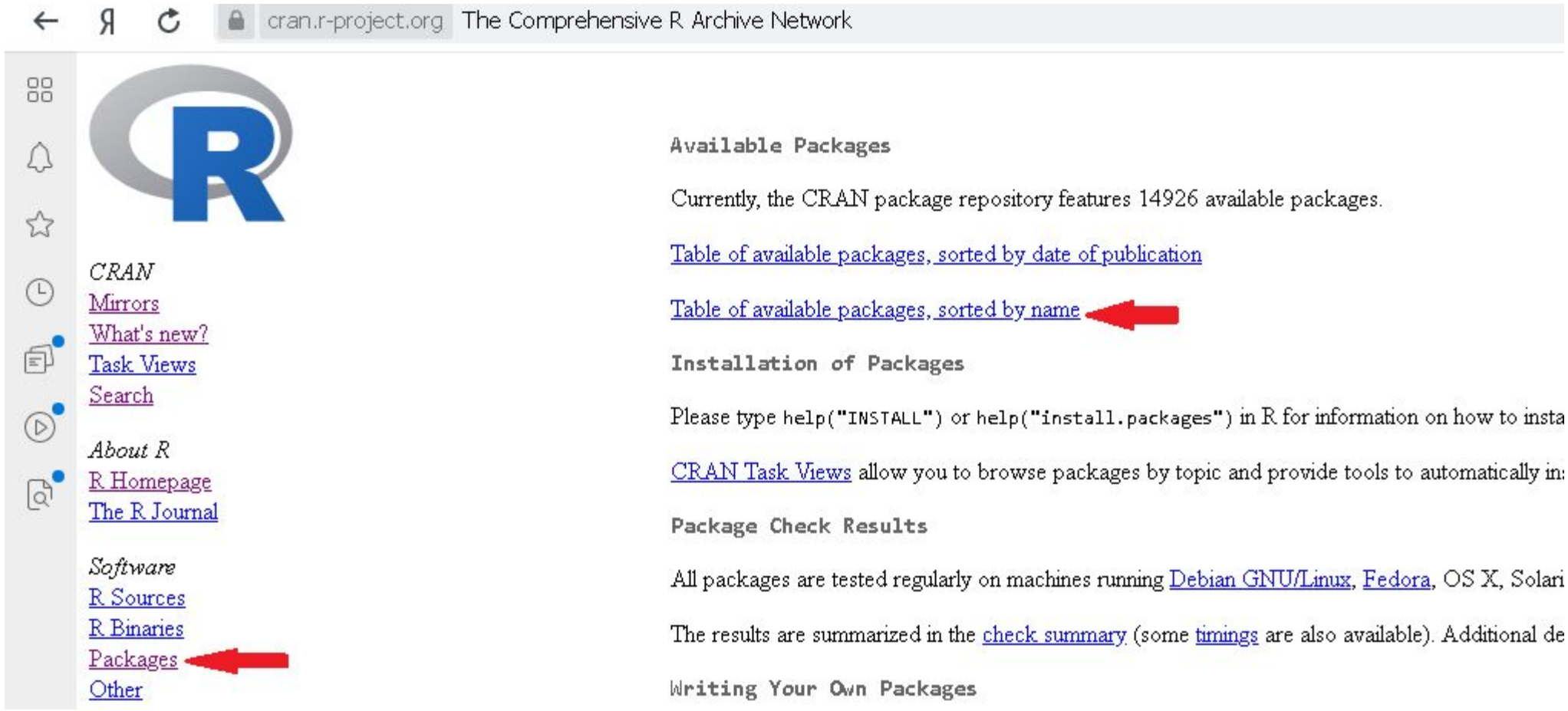
2.3 С помощью функции help() или “?” можно вызвать справку

```
> ?log
> help(log)
```



2.4 R содержит множество встроенных функция для статистического анализа

Также доступ к огромному количеству дополнительных функций можно получить после установки необходимых пакетов, которые хранятся на сайте https://cran.r-project.org/



Загрузка и установка пакетов

```
install.packages("ИМЯ_ПАКЕТА") # ИМЯ В ""
library( ИМЯ_ПАКЕТА) # БЕЗ " "
```

3. Множества и действия с ними

```
> a<-c(0,1,2,3)
> a[2]
[1] 1
> b<-c(rep(2,time=3))
> b
[1] 2 2 2
> b.1<-rep(2,4)
> a+b.1
[1] 2 3 4 5
> a
[1] 0 1 2 3
> b.1
[1] 2 2 2 2
> seq(2,8,by=2) # четные
[1] 2 4 6 8
> seq(1,9, by=2)#нечетные
[1] 1 3 5 7 9
> seq(1,10,length.out = 5)
[1]  1.00  3.25  5.50  7.75 10.00
> s<-1:10
> s
 [1]  1  2  3  4  5  6  7  8  9 10
> rnorm(n=50,mean=0,sd=1) # задаем нормальное распеделение
 [1] -1.19568940  0.38345390 -0.82137052 -0.45849240 -0.65319504  0.01894129 -1.46996829  0.34808740  1.28532810
[10] -1.15069195 -0.60169971  0.52078337 -0.23368330  0.36036533 -1.18349242 -1.00304384 -0.28676327  0.06462923
[19]  0.89833023  0.10230306  0.10422599  1.08995167 -0.68732123  0.98880655  0.01128064 -0.86978533  0.35993318
[28] -0.65246617  0.19107954  0.03878777  1.06212786  0.88307130  2.02883359 -1.44752991  0.23115790 -0.13923298
[37]  0.22410651  1.70352091 -0.46614243  1.27191456 -1.76937943 -0.52221922 -0.24999959 -0.12848811  1.52593587
[46] -1.36266494 -0.92678491 -1.30562983  0.07837400 -0.43599946
> rpois(100,10)# распеделение Пуассона
  [1] 16  8 12  9  8  9  8 14 14 10 12  7  9 10  8  9  6  8 13  9 12  6 13  9  7  9 16  9  6  8 11  3  7 13 11  8 12  9 11
 [40] 10 12  6 13  9  7 12 11  6  6 18  7 12  7  6  6 15 11 13 13 10 11 12 12  5  7 10 10 12 10  8 15  8 12  9  8 11  6 11
 [79] 10 16 10  8 12 12  5 11 12 13 13 15  7  9 11 11  9 15 10  8  7 13
> rbinom(100,10,0.5) # биномиальное распределение
  [1] 4 6 8 8 4 5 4 4 3 4 6 5 2 3 3 6 5 7 6 7 7 8 4 5 7 5 4 4 5 4 4 6 7 4 4 5 5 9 4 4 7 6 5 5 5 4 8 4 6 6 3 5 4 6 3 4 5 6 5
 [60] 6 2 5 4 4 4 7 6 4 2 8 5 4 6 5 4 6 7 5 6 5 5 4 5 7 6 7 4 6 5 1 5 5 4 7 9 7 8 5 5 8
> rep(c(0,2),time=2)
[1] 0 2 0 2
> rep(c(0,2),each=2)
[1] 0 0 2 2
```

Текстовый вектор

```
> letters
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
> LETTERS
 [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
> paste(letters,set="_",seq(1,26))
 [1] "a _ 1"  "b _ 2"  "c _ 3"  "d _ 4"  "e _ 5"  "f _ 6"  "g _ 7"  "h _ 8"  "i _ 9"  "j _ 10" "k _ 11" "l _ 12" "m _ 13"
[14] "n _ 14" "o _ 15" "p _ 16" "q _ 17" "r _ 18" "s _ 19" "t _ 20" "u _ 21" "v _ 22" "w _ 23" "x _ 24" "y _ 25" "z _ 26"
> ?ISOdate
> format(ISOdate(2019, 9, 1:30), "%d")
 [1] "01" "02" "03" "04" "05" "06" "07" "08" "09" "10" "11" "12" "13" "14" "15" "16" "17" "18" "19" "20" "21" "22" "23"
[24] "24" "25" "26" "27" "28" "29" "30"
> format(ISOdate(2019, 9, 1:30), "%b")
 [1] "сен" "сен" "сен" "сен" "сен" "сен" "сен" "сен" "сен" "сен" "сен" "сен" "сен" "сен" "сен" "сен" "сен" "сен" "сен"
[20] "сен" "сен" "сен" "сен" "сен" "сен" "сен" "сен" "сен" "сен" "сен"
> paste0(format(ISOdate(2019, 9, 1:30), "%d"),set="_", rep("сен",30))
 [1] "01_сен" "02_сен" "03_сен" "04_сен" "05_сен" "06_сен" "07_сен" "08_сен" "09_сен" "10_сен" "11_сен" "12_сен" "13_сен"
[14] "14_сен" "15_сен" "16_сен" "17_сен" "18_сен" "19_сен" "20_сен" "21_сен" "22_сен" "23_сен" "24_сен" "25_сен" "26_сен"
[27] "27_сен" "28_сен" "29_сен" "30_сен"
```

ISOdatetime {base}

Date-time Conversion Functions from Numeric Representations

Description

Convenience wrappers to create date-times from numeric representations.

Usage

```
ISOdatetime(year, month, day, hour, min, sec, tz = "")
ISOdate(year, month, day, hour = 12, min = 0, sec = 0, tz = "GMT")
```

Прочие основные и востребованные функции

Class(), length() , sum(), mean(), table()

```
> a<-c(rep(1,10))
> a
 [1] 1 1 1 1 1 1 1 1 1 1
> class(a)
[1] "numeric"
> b<-letters
> b
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
> class(b)
[1] "character"
> class(2!=0)
[1] "logical"
> class(factorial)
[1] "function"
> d<- c(rep("a",4),rep("b",6),(rep("c",2)))
> sample(d)
 [1] "b" "c" "b" "b" "a" "a" "c" "b" "a" "b" "b" "a"
> d.table<-table(d)
> r<-  seq(10,1)
> r
 [1] 10  9  8  7  6  5  4  3  2  1
> sum(r)
[1] 55
> mean.1<-sum(r)/length(r)
> class(d.table)
[1] "table"
> d
 [1] "a" "a" "a" "a" "b" "b" "b" "b" "b" "b" "c" "c"
> mean.1
[1] 5.5
> mean.2<-mean(r)
> mean.2
[1] 5.5
```

Функции set.seed(), sort()

```
> set.seed(1)
> popul<-rnorm(100)
> popul
  [1] -0.626453811  0.183643324 -0.835628612  1.595280802  0.329507772 -0.820468384  0.487429052  0.738324705  0.575781352
 [10] -0.305388387  1.511781168  0.389843236 -0.621240581 -2.214699887  1.124930918 -0.044933609 -0.016190263  0.943836211
 [19]  0.821221195  0.593901321  0.918977372  0.782136301  0.074564983 -1.989351696  0.619825748 -0.056128740 -0.155795507
 [28] -1.470752384 -0.478150055  0.417941560  1.358679552 -0.102787727  0.387671612 -0.053805041 -1.377059557 -0.414994563
 [37] -0.394289954 -0.059313397  1.100025372  0.763175748 -0.164523596 -0.253361680  0.696963375  0.556663199 -0.688755695
 [46] -0.707495157  0.364581962  0.768532925 -0.112346212  0.881107726  0.398105880 -0.612026393  0.341119691 -1.129363096
 [55]  1.433023702  1.980399899 -0.367221476 -1.044134626  0.569719627 -0.135054604  2.401617761 -0.039240003  0.689739362
 [64]  0.028002159 -0.743273209  0.188792300 -1.804958629  1.465554862  0.153253338  2.172611670  0.475509529 -0.709946431
 [73]  0.610726353 -0.934097632 -1.253633400  0.291446236 -0.443291873  0.001105352  0.074341324 -0.589520946 -0.568668733
 [82] -0.135178615  1.178086997 -1.523566800  0.593946188  0.332950371  1.063099837 -0.304183924  0.370018810  0.267098791
 [91] -0.542520031  1.207867806  1.160402616  0.700213650  1.586833455  0.558486426 -1.276592208 -0.573265414 -1.224612615
[100] -0.473400636
> popul.1<-round(popul,3)
> popul.1
  [1] -0.626  0.184 -0.836  1.595  0.330 -0.820  0.487  0.738  0.576 -0.305  1.512  0.390 -0.621 -2.215  1.125 -0.045
 [17] -0.016  0.944  0.821  0.594  0.919  0.782  0.075 -1.989  0.620 -0.056 -0.156 -1.471 -0.478  0.418  1.359 -0.103
 [33]  0.388 -0.054 -1.377 -0.415 -0.394 -0.059  1.100  0.763 -0.165 -0.253  0.697  0.557 -0.689 -0.707  0.365  0.769
 [49] -0.112  0.881  0.398 -0.612  0.341 -1.129  1.433  1.980 -0.367 -1.044  0.570 -0.135  2.402 -0.039  0.690  0.028
 [65] -0.743  0.189 -1.805  1.466  0.153  2.173  0.476 -0.710  0.611 -0.934 -1.254  0.291 -0.443  0.001  0.074 -0.590
 [81] -0.569 -0.135  1.178 -1.524  0.594  0.333  1.063 -0.304  0.370  0.267 -0.543  1.208  1.160  0.700  1.587  0.558
 [97] -1.277 -0.573 -1.225 -0.473
> sort(popul.1)
  [1] -2.215 -1.989 -1.805 -1.524 -1.471 -1.377 -1.277 -1.254 -1.225 -1.129 -1.044 -0.934 -0.836 -0.820 -0.743 -0.710
 [17] -0.707 -0.689 -0.626 -0.621 -0.612 -0.590 -0.573 -0.569 -0.543 -0.478 -0.473 -0.443 -0.415 -0.394 -0.367 -0.305
 [33] -0.304 -0.253 -0.165 -0.156 -0.135 -0.135 -0.112 -0.103 -0.059 -0.056 -0.054 -0.045 -0.039 -0.016  0.001  0.028
 [49]  0.074  0.075  0.153  0.184  0.189  0.267  0.291  0.330  0.333  0.341  0.365  0.370  0.388  0.390  0.398  0.418
 [65]  0.476  0.487  0.557  0.558  0.570  0.576  0.594  0.594  0.611  0.620  0.690  0.697  0.700  0.738  0.763  0.769
 [81]  0.782  0.821  0.881  0.919  0.944  1.063  1.100  1.125  1.160  1.178  1.208  1.359  1.433  1.466  1.512  1.587
 [97]  1.595  1.980  2.173  2.402
> sort(popul.1,decreasing = TRUE)
  [1]  2.402  2.173  1.980  1.595  1.587  1.512  1.466  1.433  1.359  1.208  1.178  1.160  1.125  1.100  1.063  0.944
 [17]  0.919  0.881  0.821  0.782  0.769  0.763  0.738  0.700  0.697  0.690  0.620  0.611  0.594  0.594  0.576  0.570
 [33]  0.558  0.557  0.487  0.476  0.418  0.398  0.390  0.388  0.370  0.365  0.341  0.333  0.330  0.291  0.267  0.189
 [49]  0.184  0.153  0.075  0.074  0.028  0.001 -0.016 -0.039 -0.045 -0.054 -0.056 -0.059 -0.103 -0.112 -0.135 -0.135
 [65] -0.156 -0.165 -0.253 -0.304 -0.305 -0.367 -0.394 -0.415 -0.443 -0.473 -0.478 -0.543 -0.569 -0.573 -0.590 -0.612
 [81] -0.621 -0.626 -0.689 -0.707 -0.710 -0.743 -0.820 -0.836 -0.934 -1.044 -1.129 -1.225 -1.254 -1.277 -1.377 -1.471
 [97] -1.524 -1.805 -1.989 -2.215
```

sample(), data.frame(), head(), order(),$

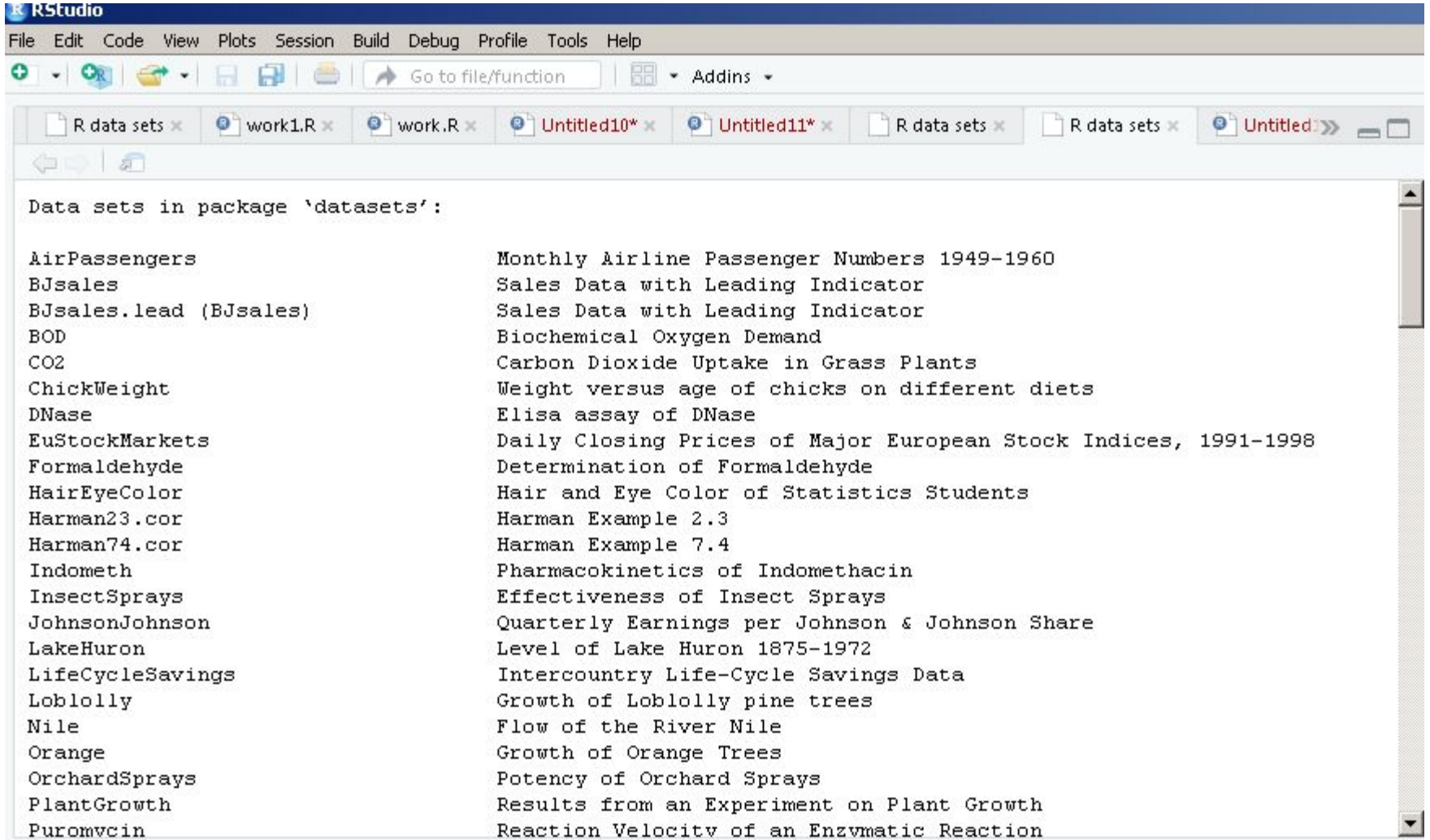
```
> set.seed(1)
> lets<- sample(letters,100,replace = TRUE)
> lets
  [1] "g" "j" "o" "x" "f" "x" "y" "r" "q" "b" "f" "e" "r" "j" "u" "m" "s" "z" "j" "u" "y" "f" "q" "d" "g" "k" "a" "j" "w"
 [30] "i" "m" "p" "m" "e" "v" "r" "u" "c" "s" "k" "v" "q" "u" "o" "n" "u" "a" "m" "t" "s" "m" "w" "l" "g" "b" "c" "i" "n"
 [59] "r" "k" "x" "h" "l" "i" "q" "g" "m" "t" "c" "w" "i" "v" "j" "i" "m" "x" "w" "k" "u" "y" "l" "s" "k" "i" "t" "f" "s"
 [88] "d" "g" "d" "g" "b" "q" "w" "u" "u" "l" "k" "v" "p"
> df<-data.frame(lets,popul.1)
> head(df)
  lets popul.1
1    g  -0.626
2    j   0.184
3    o  -0.836
4    x   1.595
5    f   0.330
6    x  -0.820
> #нужно разместить по возрастанию, но теперь каждому значению еще соответствует буква
> ind<-order(df$popul.1)
> ind
  [1]  14  24  67  84  28  35  97  75  99  54  58  74   3   6  65  72  46  45   1  13  52  80  98  81  91  29 100  77  36
 [30]  37  57  10  88  42  41  27  60  82  49  32  38  26  34  16  62  17  78  64  79  23  69   2  66  90  76   5  86  53
 [59]  47  89  33  12  51  30  71   7  44  96  59   9  20  85  73  25  63  43  94   8  40  48  22  19  50  21  18  87  39
 [88]  15  93  83  92  31  55  68  11  95   4  56  70  61
> df.new<-df[ind,]
> head(df.new,10)
   lets popul.1
14    j  -2.215
24    d  -1.989
67    m  -1.805
84    i  -1.524
28    j  -1.471
35    v  -1.377
97    l  -1.277
75    m  -1.254
99    v  -1.225
54    g  -1.129
```

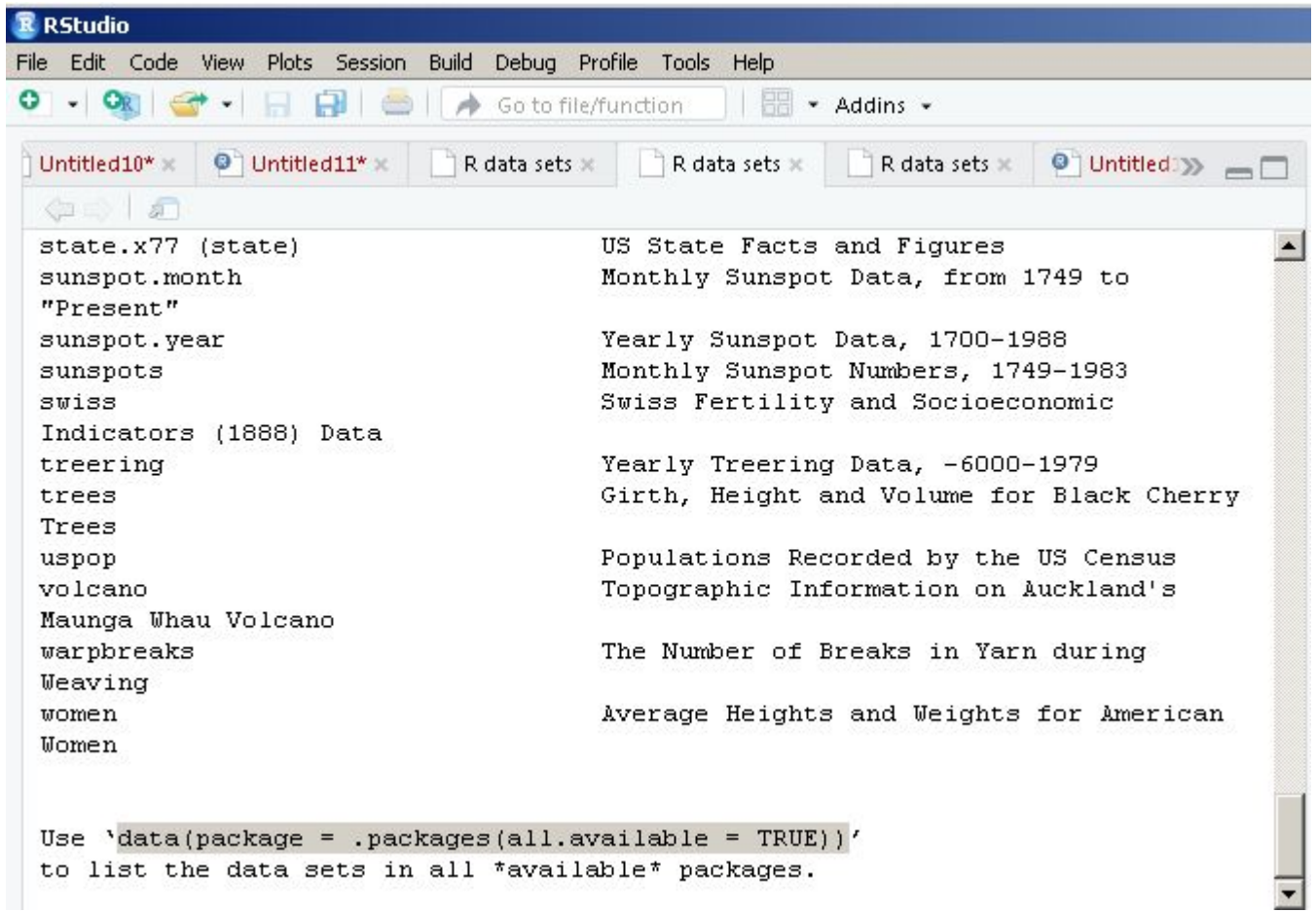
Что еще можно делать с векторами?

```
> g<-seq(31,45)
> g
 [1] 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
> length(g)
[1] 15
> g[1]
[1] 31
> g[c(1,5,15)]
[1] 31 35 45
> g[-c(1,5,15)]
 [1] 32 33 34 36 37 38 39 40 41 42 43 44
> g[1:5]
[1] 31 32 33 34 35
> h<-1:15
> g+h
 [1] 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60
> g*2
 [1] 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90
```

В пакете «datasets» находится множество наборов, на которых можно совершенствовать знания в области R

```
data()
```




RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Untitled10*
Untitled11*
R data sets
R data sets
R data sets
Untitled
state.x77 (state) US State Facts and Figures
sunspot.month Monthly Sunspot Data, from 1749 to
"Present"
sunspot.year Yearly Sunspot Data, 1700-1988
sunspots Monthly Sunspot Numbers, 1749-1983
swiss Swiss Fertility and Socioeconomic
Indicators (1888) Data
treering Yearly Treering Data, -6000-1979
trees Girth, Height and Volume for Black Cherry
Trees
uspop Populations Recorded by the US Census
volcano Topographic Information on Auckland's
Maunga Whau Volcano
warpbreaks The Number of Breaks in Yarn during
Weaving
women Average Heights and Weights for American
Women
Use 'data(package = .packages(all.available = TRUE))'
to list the data sets in all *available* packages.

R data sets | R data sets | R data sets | R data sets | R data sets

```
                        Data
oilType (oil)           Fatty acid composition of commercial oils
potteryClass (pottery)
                        Pottery from Pre-Classical Sites in Italy
scat                    Morphometric Data on Scat
scat_orig (scat)        Morphometric Data on Scat
segmentationData        Cell Body Segmentation

Data sets in package 'cluster':

agriculture             European Union Agricultural Workforces
animals                 Attributes of Animals
chorSub                 Subset of C-horizon of Kola Data
flower                  Flower Characteristics
plantTraits             Plant Species Traits Data
pluton                  Isotopic Composition Plutonium Batches
ruspini                 Ruspini Data
```

Console | Terminal

```
> library(cluster)
> ?animals
> animals
    war fly ver end gro hai
ant   1   1   1   1   2   1
bee   1   2   1   1   2   2
cat   2   1   2   1   1   2
cpl   1   1   1   1   1   2
chi   2   1   2   2   2   2
cow   2   1   2   1   2   2
duc   2   2   2   1   2   1
eag   2   2   2   2   1   1
ele   2   1   2   2   2   1
fly   1   2   1   1   1   1
fro   1   1   2   2  NA   1
her   1   1   2   1   2   1
lio   2   1   2  NA   2   2
liz   1   1   2   1   1   1
lob   1   1   1   1  NA   1
man   2   1   2   2   2   2
rab   2   1   2   1   2   2
sal   1   1   2   1  NA   1
```

Environment | History | Connections

Import Dataset | List

Global Environment

g	int [1:15] 31 32 33 34 35 36 37 38 39 40 ...
g.1	num [1:4] 1 2 3 4
h	int [1:15] 1 2 3 4 5 6 7 8 9 10 ...
height	Named num [1:4] 50 65 67 59
heigt	num [1:4] 50 65 67 59
i	10L
imena	chr [1:5] "X1" "X2" "X3" "X4" "X5"
inclass	Factor w/ 2 levels "Female","Male": 1 1 2 1 2 2 2 2 1 1 ...
ind	int [1:100] 14 24 67 84 28 35 97 75 99 54 ...
k	num [1:4] 1 3 5 7
lets	chr [1:100] "g" "j" "o" "x" "f" "x" "y" "r" "q" "b" "f" "e...
m	69.2561797752809
mean.1	5.5

Files | Plots | Packages | Help | Viewer

R: Attributes of Animals | Find in Topic

animals {cluster} R Documentation

Attributes of Animals

Description

This data set considers 6 binary attributes for 20 animals.

Usage

```
data(animals)
```

Format

A data frame with 20 observations on 6 variables:

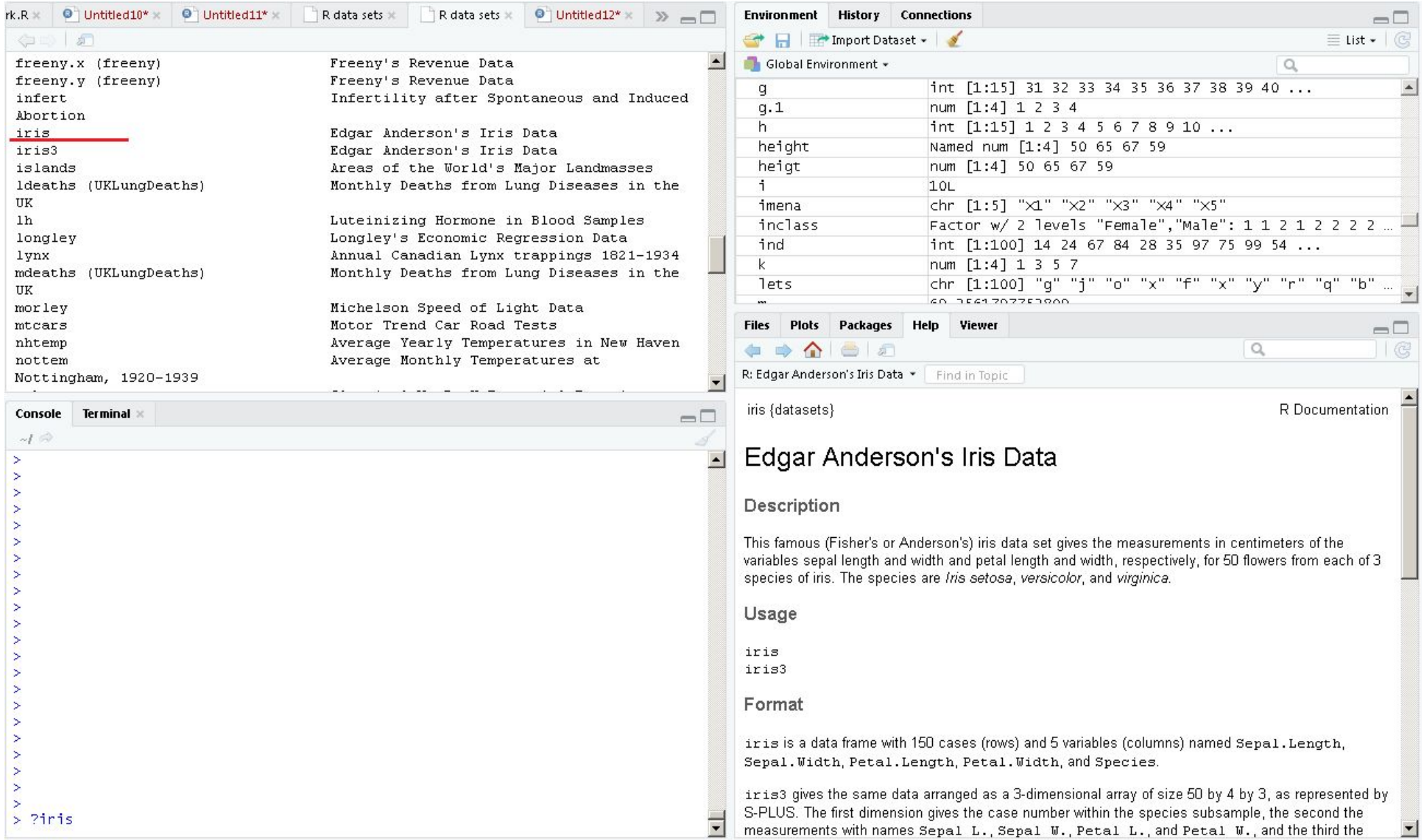
[, 1] war warm-blooded
[, 2] fly can fly

Логические функции и операторы

```
> 5>6
[1] FALSE
> 61 < 100
[1] TRUE
> 5 > 6
[1] FALSE
> 61 < 100
[1] TRUE
> 2!= 2
[1] FALSE
> 3==3
[1] TRUE
> 10 >= 10
[1] TRUE
> 12<= 12
[1] TRUE
> 6 > 7 | 8 > 9
[1] FALSE
> 6 > 7 | 10 > 9
[1] TRUE
> 6> 7 & 10 > 9
[1] FALSE
> 200!= 100 | 10 <= 6
[1] TRUE
> ((TRUE== FALSE)& (1 == 1)) & 100 == 100
[1] FALSE
> ((TRUE== FALSE)& (1 == 1)) | 100 == 100
[1] TRUE
```

```
> sum(is.na(animals))
[1] 5
> is.na(animals)
      war   fly   ver   end   gro   hai
ant FALSE FALSE FALSE FALSE FALSE FALSE
bee FALSE FALSE FALSE FALSE FALSE FALSE
cat FALSE FALSE FALSE FALSE FALSE FALSE
cpl FALSE FALSE FALSE FALSE FALSE FALSE
chi FALSE FALSE FALSE FALSE FALSE FALSE
cow FALSE FALSE FALSE FALSE FALSE FALSE
duc FALSE FALSE FALSE FALSE FALSE FALSE
eag FALSE FALSE FALSE FALSE FALSE FALSE
ele FALSE FALSE FALSE FALSE FALSE FALSE
fly FALSE FALSE FALSE FALSE FALSE FALSE
fro FALSE FALSE FALSE FALSE  TRUE FALSE
her FALSE FALSE FALSE FALSE FALSE FALSE
lio FALSE FALSE FALSE  TRUE FALSE FALSE
liz FALSE FALSE FALSE FALSE FALSE FALSE
lob FALSE FALSE FALSE FALSE  TRUE FALSE
man FALSE FALSE FALSE FALSE FALSE FALSE
rab FALSE FALSE FALSE FALSE FALSE FALSE
sal FALSE FALSE FALSE FALSE  TRUE FALSE
spi FALSE FALSE FALSE  TRUE FALSE FALSE
wha FALSE FALSE FALSE FALSE FALSE FALSE
> head(animals,3)
    war fly ver end gro hai
ant   1   1   1   1   2   1
bee   1   2   1   1   2   2
cat   2   1   2   1   1   2
> animals$gro
 [1]  2  2  1  1  2  2  2  1  2  1 NA  2  2  1 NA  2  2 NA  1  2
> animals$gro[!is.na(animals$gro)]
 [1] 2 2 1 1 2 2 2 1 2 1 2 2 1 2 2 1 2
> animals$gro[!is.na(animals[,5])]
 [1] 2 2 1 1 2 2 2 1 2 1 2 2 1 2 2 1 2
```

Продолжим работу на наборе “iris”. С помощью “?” вызываем справку с описанием набора “iris”



Извлекаем поднаборы из готовых датасэтов

```
> head(iris,10)
   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
51          7.0         3.2          4.7         1.4 versicolor
52          6.4         3.2          4.5         1.5 versicolor
53          6.9         3.1          4.9         1.5 versicolor
54          5.5         2.3          4.0         1.3 versicolor
55          6.5         2.8          4.6         1.5 versicolor
56          5.7         2.8          4.5         1.3 versicolor
57          6.3         3.3          4.7         1.6 versicolor
58          4.9         2.4          3.3         1.0 versicolor
59          6.6         2.9          4.6         1.3 versicolor
60          5.2         2.7          3.9         1.4 versicolor
> dim(iris)
[1] 100   5
> str(iris)
'data.frame':   100 obs. of  5 variables:
 $ Sepal.Length: num  7 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 ...
 $ Sepal.Width : num  3.2 3.2 3.1 2.3 2.8 2.8 3.3 2.4 2.9 2.7 ...
 $ Petal.Length: num  4.7 4.5 4.9 4 4.6 4.5 4.7 3.3 4.6 3.9 ...
 $ Petal.Width : num  1.4 1.5 1.5 1.3 1.5 1.3 1.6 1 1.3 1.4 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 2 2 2 2 2 2 2 2 2 2 ...
> unique(iris[,5])
[1] versicolor virginica
Levels: setosa versicolor virginica
> levels(iris$Species)
[1] "setosa"     "versicolor" "virginica"
```

```
> head(iris)
   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
51          7.0         3.2          4.7         1.4 versicolor
52          6.4         3.2          4.5         1.5 versicolor
53          6.9         3.1          4.9         1.5 versicolor
54          5.5         2.3          4.0         1.3 versicolor
55          6.5         2.8          4.6         1.5 versicolor
56          5.7         2.8          4.5         1.3 versicolor
> iris[1,1]
[1] 7
> traindat<-iris[1:5,c(2,4,5)]
> traindat
   Sepal.Width Petal.Width    Species
51         3.2         1.4 versicolor
52         3.2         1.5 versicolor
53         3.1         1.5 versicolor
54         2.3         1.3 versicolor
55         2.8         1.5 versicolor
> traindat[traindat$Sepal.Width>3.0,] # не забудьте о "," в конце!
   Sepal.Width Petal.Width    Species
51         3.2         1.4 versicolor
52         3.2         1.5 versicolor
53         3.1         1.5 versicolor
> traindat
   Sepal.Width Petal.Width    Species
51         3.2         1.4 versicolor
52         3.2         1.5 versicolor
53         3.1         1.5 versicolor
54         2.3         1.3 versicolor
55         2.8         1.5 versicolor
> traindat[traindat$Sepal.Width>3.0 & traindat$Petal.Width > 1.4,]
   Sepal.Width Petal.Width    Species
52         3.2         1.5 versicolor
53         3.1         1.5 versicolor
```

Построим свой dataframe

```
> weight<- c(78,56,67,48,69,90)
> height<-c(170,160,165,159,170,185)
> sex<-(c(rep("F",3),rep("M",3)))
> sex
[1] "F" "F" "F" "M" "M" "M"
> df.1<-data.frame(weight,height,sex)
> df.1
  weight height sex
1     78    170   F
2     56    160   F
3     67    165   F
4     48    159   M
5     69    170   M
6     90    185   M
> str(df.1)
'data.frame':   6 obs. of  3 variables:
 $ weight: num  78 56 67 48 69 90
 $ height: num  170 160 165 159 170 185
 $ sex   : chr  "F" "F" "F" "M" ...
> df.2<-data.frame(weight,height,sex,stringsAsFactors = TRUE)
> df.2
  weight height sex
1     78    170   F
2     56    160   F
3     67    165   F
4     48    159   M
5     69    170   M
6     90    185   M
> str(df.2)
'data.frame':   6 obs. of  3 variables:
 $ weight: num  78 56 67 48 69 90
 $ height: num  170 160 165 159 170 185
 $ sex   : Factor w/ 2 levels "F","M": 1 1 1 2 2 2
> sex
[1] "F" "F" "F" "M" "M" "M"
> factor(sex)
[1] F F F M M M
Levels: F M
```

Стоим матрицу

```
> m<-1:30
> m
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
> z<-c(10,3)
> z
[1] 10  3
> dim(m)
NULL
> dim(m)<-z
> dim(m)
[1] 10  3
> m
      [,1] [,2] [,3]
 [1,]    1   11   21
 [2,]    2   12   22
 [3,]    3   13   23
 [4,]    4   14   24
 [5,]    5   15   25
 [6,]    6   16   26
 [7,]    7   17   27
 [8,]    8   18   28
 [9,]    9   19   29
[10,]   10   20   30
> class(m)
[1] "matrix"
> ## другой способ
> y<-1:50
> mt<-matrix(y,10,5)
> mt
      [,1] [,2] [,3] [,4] [,5]
 [1,]    1   11   21   31   41
 [2,]    2   12   22   32   42
 [3,]    3   13   23   33   43
 [4,]    4   14   24   34   44
 [5,]    5   15   25   35   45
 [6,]    6   16   26   36   46
 [7,]    7   17   27   37   47
 [8,]    8   18   28   38   48
 [9,]    9   19   29   39   49
[10,]   10   20   30   40   50
```

```
> imena<-LETTERS[1:10]
> imena
 [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J"
> cbind(imena,mt)
      imena
 [1,] "A"   "1"  "11" "21" "31" "41"
 [2,] "B"   "2"  "12" "22" "32" "42"
 [3,] "C"   "3"  "13" "23" "33" "43"
 [4,] "D"   "4"  "14" "24" "34" "44"
 [5,] "E"   "5"  "15" "25" "35" "45"
 [6,] "F"   "6"  "16" "26" "36" "46"
 [7,] "G"   "7"  "17" "27" "37" "47"
 [8,] "H"   "8"  "18" "28" "38" "48"
 [9,] "I"   "9"  "19" "29" "39" "49"
[10,] "J"   "10" "20" "30" "40" "50"
> class(cbind(imena,mt))
[1] "matrix"
> cbind(imena,mt)[2,]
imena
  "B"   "2"  "12"  "22"  "32"  "42"
```

```
> rownames(mt)
NULL
> colnames(mt)
NULL
> ?rownames
> rownames(mt)
NULL
> colnames(mt)
NULL
> rownames(mt,do.NULL = FALSE, prefix = "row")
 [1] "row1"  "row2"  "row3"  "row4"  "row5"  "row6"  "row7"  "row8"  "row9"  "row10"
> rownames(mt)
NULL
> rownames(mt)<-imena
> mt
  [,1] [,2] [,3] [,4] [,5]
A    1   11   21   31   41
B    2   12   22   32   42
C    3   13   23   33   43
D    4   14   24   34   44
E    5   15   25   35   45
F    6   16   26   36   46
G    7   17   27   37   47
H    8   18   28   38   48
I    9   19   29   39   49
J   10   20   30   40   50
> colnames(mt)<-paste("day",1:5)
> mt
  day 1 day 2 day 3 day 4 day 5
A     1    11    21    31    41
B     2    12    22    32    42
C     3    13    23    33    43
D     4    14    24    34    44
E     5    15    25    35    45
F     6    16    26    36    46
G     7    17    27    37    47
H     8    18    28    38    48
I     9    19    29    39    49
J    10    20    30    40    50
```


Files | Plots | Packages | Help | Viewer

R: Row and Column Names — Find in Topic

row+colnames {base}

Row and Column Names

Description

Retrieve or set the row or column names of a matrix-like object.

Usage

```
rownames(x, do.NULL = TRUE, prefix = "row")
rownames(x) <- value

colnames(x, do.NULL = TRUE, prefix = "col")
colnames(x) <- value
```

Arguments

x	a matrix-like R object, with at least two dimensions for colnames.
do.NULL	logical. If FALSE and names are NULL, names are created.
prefix	for created names.
value	a valid value for that component of dimnames(x). For a matrix or arra zero length equal to the appropriate dimension.

Итоги

1.Узнали об истории происхождения и о предназначении R

2 Обсудили многочисленные преимущества

3. Узнали массу дополнительных источников, где можно брать информацию

4. Научились устанавливать R и R Studio

5. Познакомились со средой R и ее объектами

6. Узнали множество функций для работы с данными

7 Научились задавать последовательности и множества

8. Научились извлекать необходимую информацию из готовых фреймов и выбирать поднаборы данных

9 Научились строить свои набор данных и матрицу