

Введение в компьютерный и интеллектуальный анализ данных

Воротницкая Т.И.

4. Введение в статистический пакет R.

- R —свободно распространяемая программная среда с открытым кодом и язык программирования для статистической обработки данных и работы с графикой.
- В августе 1993 новозеландские учёные Robert Gentleman и Ross Ihaka (Statistics Department, Auckland University) анонсировали свою разработку под названием R. Это была новая реализация языка S, отличающаяся от S-PLUS рядом деталей, например, работой с памятью, обращением с глобальными и локальными переменными.
- Дополнительную популярность R принесло создание центральной системы хранения и распространения пакетов — CRAN (Comprehensive R Archive Network — <http://cran.r-project.org>).

4. Введение в статистический пакет R.

| Sep 2018 | Sep 2017 | Change | Programming Language | Ratings | Change |
|----------|----------|--------|----------------------|---------|--------|
| 1 | 1 | | Java | 17.436% | +4.75% |
| 2 | 2 | | C | 15.447% | +8.06% |
| 3 | 5 | ▲ | Python | 7.653% | +4.67% |
| 4 | 3 | ▼ | C++ | 7.394% | +1.83% |
| 5 | 8 | ▲ | Visual Basic .NET | 5.308% | +3.33% |
| 6 | 4 | ▼ | C# | 3.295% | -1.48% |
| 7 | 6 | ▼ | PHP | 2.775% | +0.57% |
| 8 | 7 | ▼ | JavaScript | 2.131% | +0.11% |
| 9 | - | ▲▲ | SQL | 2.062% | +2.06% |
| 10 | 18 | ▲▲ | Objective-C | 1.509% | +0.00% |
| 11 | 12 | ▲ | Delphi/Object Pascal | 1.292% | -0.49% |
| 12 | 10 | ▼ | Ruby | 1.291% | -0.64% |
| 13 | 16 | ▲ | MATLAB | 1.276% | -0.35% |
| 14 | 15 | ▲ | Assembly language | 1.232% | -0.41% |
| 15 | 13 | ▼ | Swift | 1.223% | -0.54% |
| 16 | 17 | ▲ | Go | 1.081% | -0.49% |
| 17 | 9 | ▼▼ | Perl | 1.073% | -0.88% |
| 18 | 11 | ▼▼ | R | 1.016% | -0.80% |
| 19 | 19 | | PL/SQL | 0.850% | -0.63% |
| 20 | 14 | ▼▼ | Visual Basic | 0.682% | -1.07% |

<http://www.tiobe.com/tpci.htm>

The TIOBE Programming Community index gives an indication of the popularity of programming languages. The index is updated once a month. The ratings are based on the world-wide availability of skilled engineers, courses and third party vendors. The popular search engines Google, MSN, and Yahoo! are used to calculate the ratings.

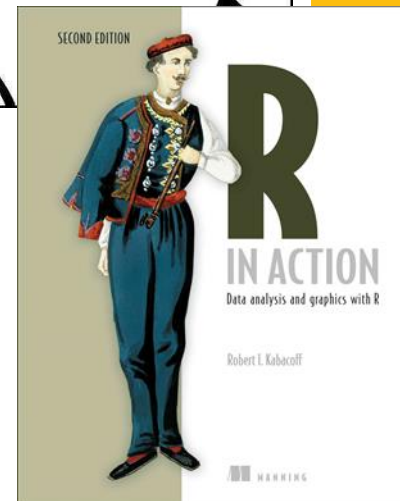
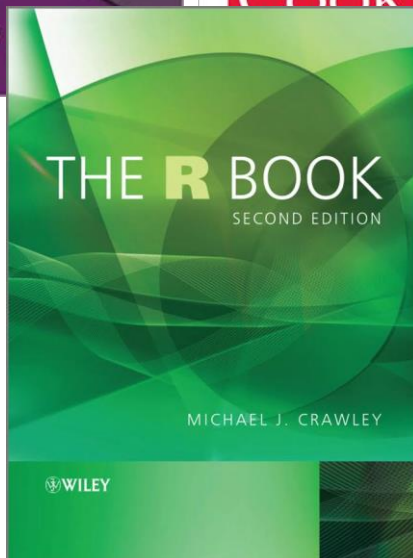
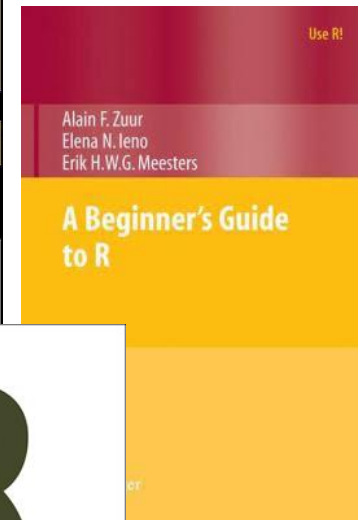
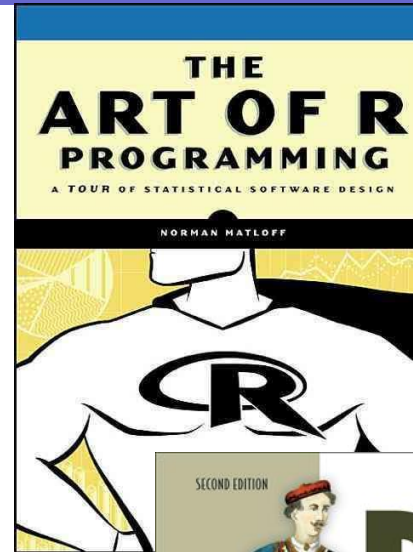
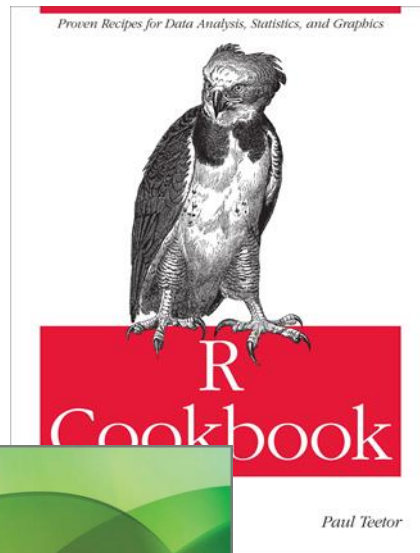
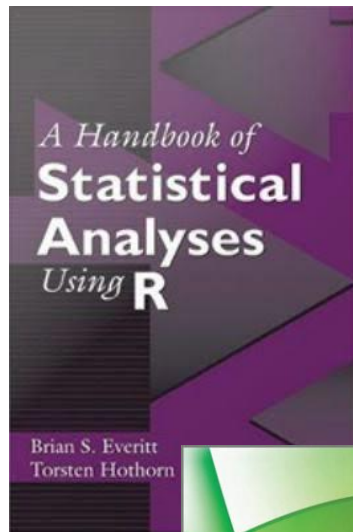
4. Введение в статистический пакет R.

| Oct 2017 | Oct 2016 | Change | Programming Language | Ratings | Change |
|----------|----------|--------|----------------------|---------|--------|
| 1 | 1 | | Java | 12.431% | -6.37% |
| 2 | 2 | | C | 8.374% | -1.46% |
| 3 | 3 | | C++ | 5.007% | -0.79% |
| 4 | 4 | | C# | 3.858% | -0.51% |
| 5 | 5 | | Python | 3.803% | +0.03% |
| 6 | 6 | | JavaScript | 3.010% | +0.26% |
| 7 | 7 | | PHP | 2.790% | +0.05% |
| 8 | 8 | | Visual Basic .NET | 2.735% | +0.08% |
| 9 | 11 | ⬆ | Assembly language | 2.374% | +0.14% |
| 10 | 13 | ⬆ | Ruby | 2.324% | +0.32% |
| 11 | 15 | ⬆ | Delphi/Object Pascal | 2.180% | +0.31% |
| 12 | 9 | ⬇ | Perl | 1.963% | -0.53% |
| 13 | 19 | ⬆ | MATLAB | 1.880% | +0.26% |
| 14 | 23 | ⬆ | Scratch | 1.819% | +0.69% |
| 15 | 18 | ⬆ | R | 1.684% | -0.06% |
| 16 | 12 | ⬇ | Swift | 1.668% | -0.34% |
| 17 | 10 | ⬇ | Objective-C | 1.513% | -0.75% |
| 18 | 14 | ⬇ | Visual Basic | 1.420% | -0.57% |

4. Введение в статистический пакет R.

- homepage: <http://www.r-project.org/> <http://cran.r-project.org>
- `help.start()`
- R-bloggers (www.r-bloggers.com)
- Quick-R (www.statmethods.net)

4. Введение в статистический пакет R.



4. Введение в статистический пакет R.

Преимущества R

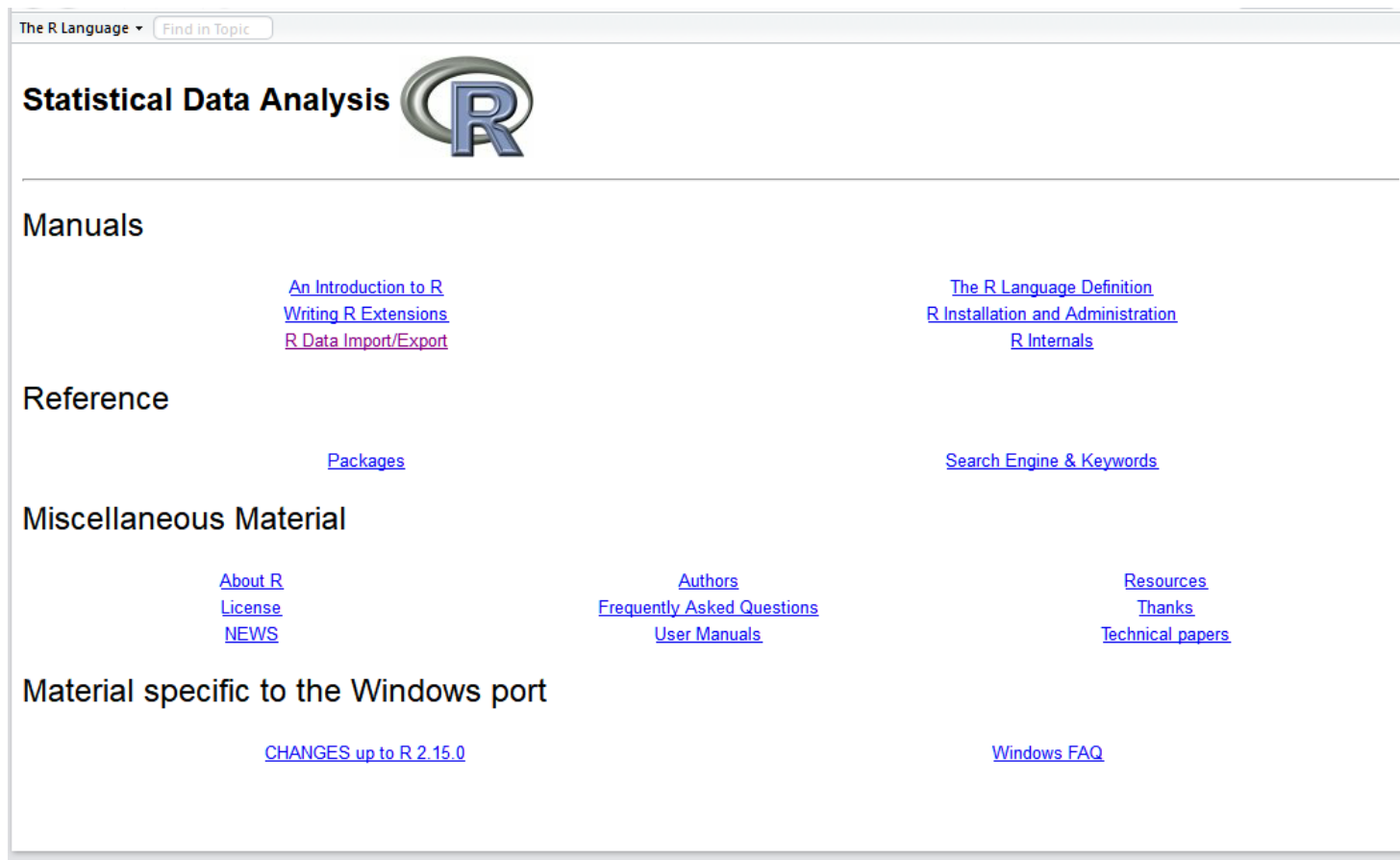
- R является свободно распространяемым программным пакетом;
- достаточно просто устанавливается под Windows, MacOS, Linux;
- базовая комплектация R занимает немного места на жёстком диске и включает в себя все функции, необходимые для статистического анализа;
- дополнительно устанавливаются вспомогательные пакеты с необходимыми функциями;
- разработаны пакеты, применимые практически во всех областях знания, где используется статистика;
- можно работать с большими массивами данных (несколько сотен тысяч наблюдений);
- возможность самостоятельного написания необходимых функций;
- R обеспечивает прямой интерфейс с C, C++, Java и т.д.

Введение в статистический пакет R.

Недостатки

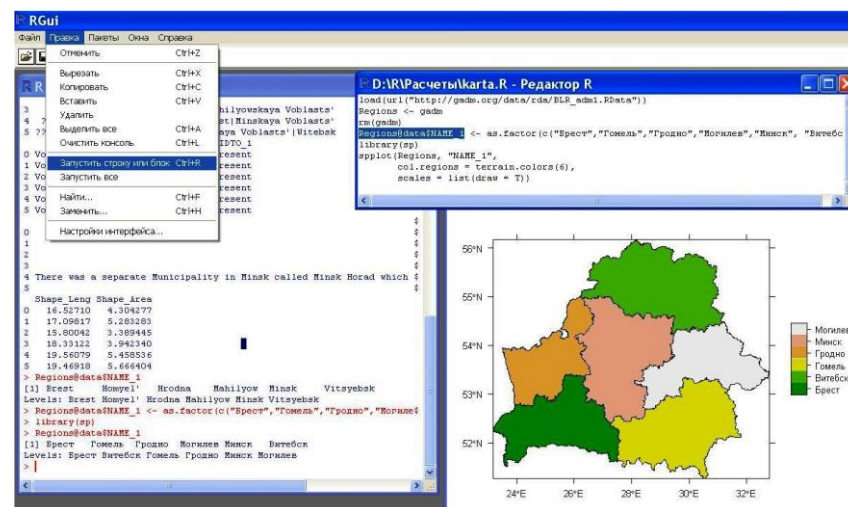
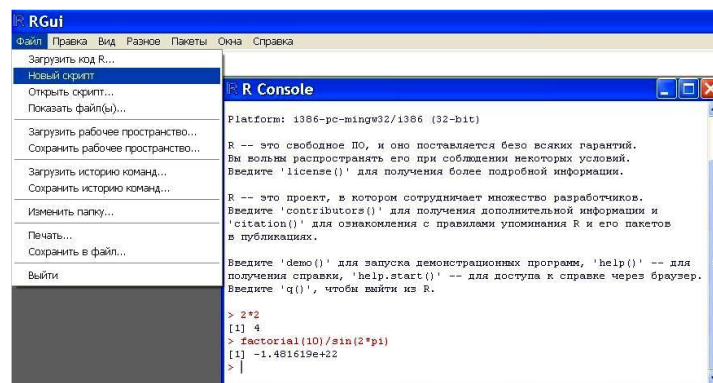
- в отличие от большинства коммерческих программ R имеет не графический интерфейс, а интерфейс командной строки, таким образом нужно знать необходимые для работы функции и синтаксис языка программирования;
- нет коммерческой поддержки (но есть международная система рассылки сообщений об обновлениях);
- довольно мало литературы по R на русском языке.

4. Введение в статистический пакет R. `help.start()`

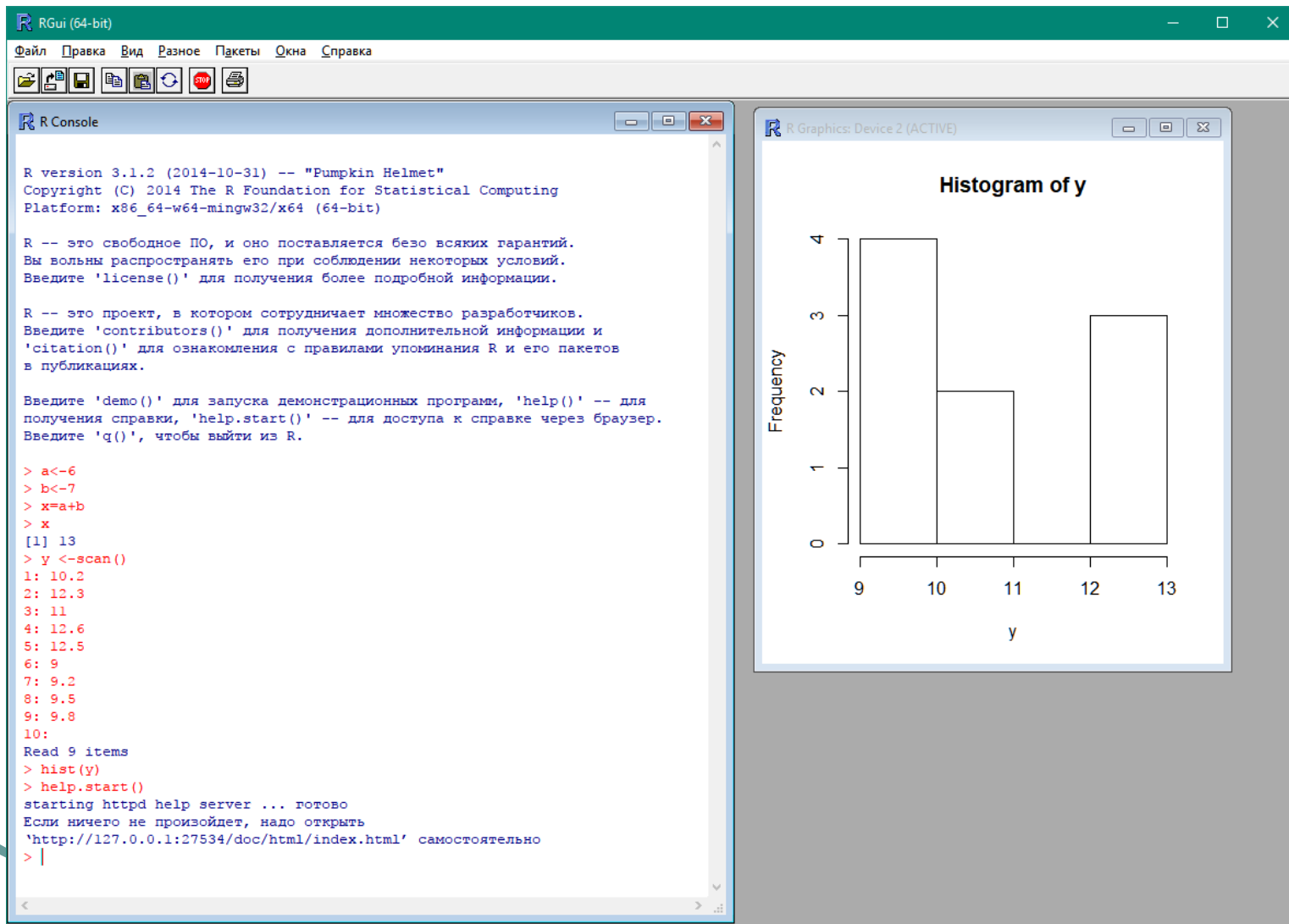


4. Введение в статистический пакет R. Интерпретация команд и скрипты

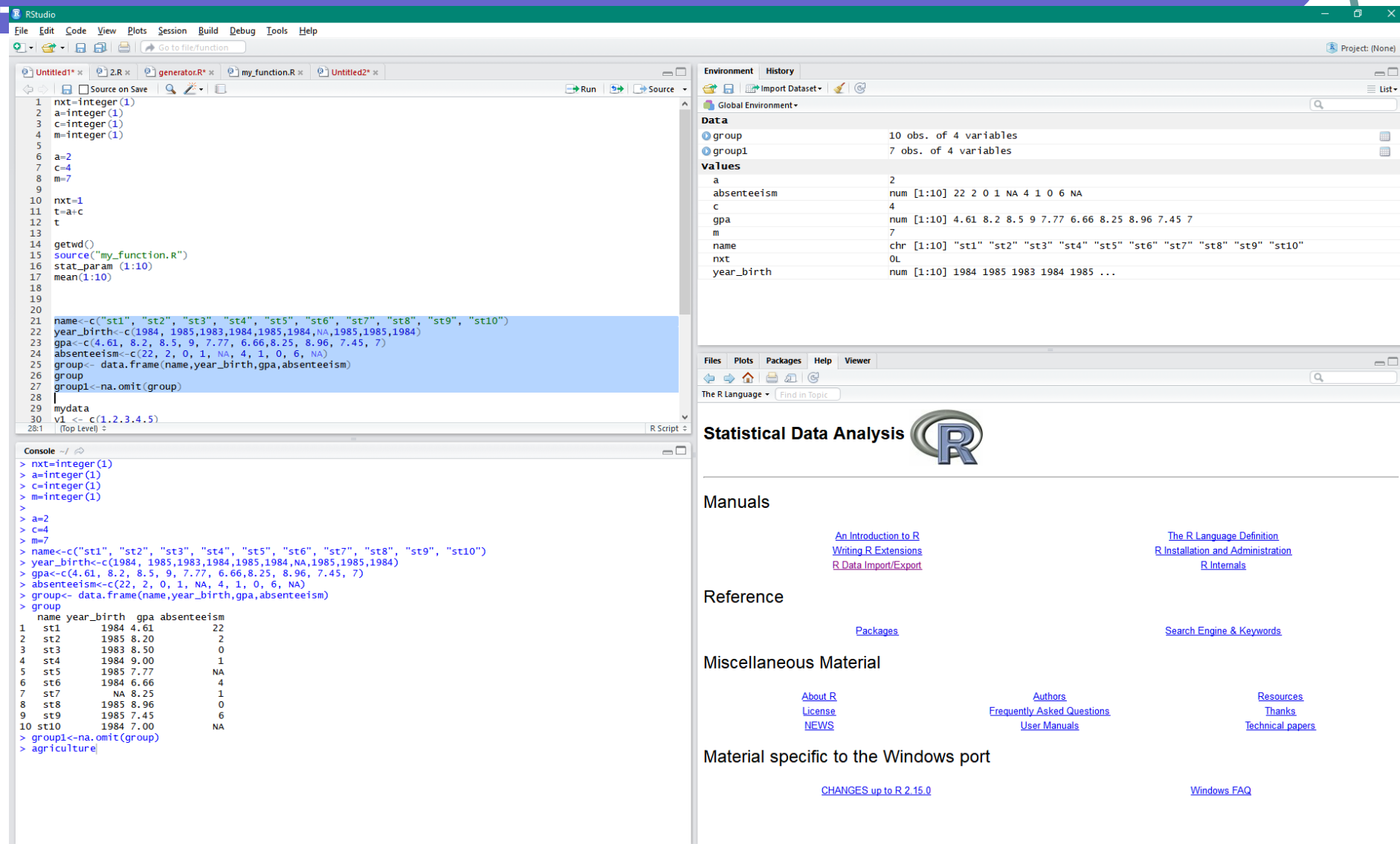
- Командный режим : после знака **>** в командном окне вводится команда, которая немедленно выполняется (интерпретируется)
- Разделитель команд – новая строка или ;
- Скрипты:
 - Создать текстовый файл с расширением *.r
 - Открыть в окне «Редактор R»
 - Запустить скрипт целиком или запустить выделенный фрагмент



Введение в статистический пакет R.



Введение в статистический пакет R.



The screenshot displays the RStudio environment. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Tools, and Help. The toolbar contains icons for file operations and running code. The editor pane shows a script with the following code:

```
1 n1=integer(1)
2 a=integer(1)
3 c=integer(1)
4 m=integer(1)
5
6 a=2
7 c=4
8 m=7
9
10 n1=1
11 t=a+c
12 t
13
14 getwd()
15 source("my_function.R")
16 stat_param (1:10)
17 mean(1:10)
18
19
20
21 name<-c("st1", "st2", "st3", "st4", "st5", "st6", "st7", "st8", "st9", "st10")
22 year_birth<-c(1984, 1985,1983,1984,1985,1984,NA,1985,1985,1984)
23 gpa<-c(4.61, 8.2, 8.5, 9, 7.77, 6.66,8.25, 8.96, 7.45, 7)
24 absenteeism<-c(22, 2, 0, 1, NA, 4, 1, 0, 6, NA)
25 group<- data.frame(name,year_birth,gpa,absenteeism)
26 group
27 group1<-na.omit(group)
28
29 mydata
30 v1 <- c(1.2,3.4,5)
31
```

The console shows the execution of the code, resulting in the following output:

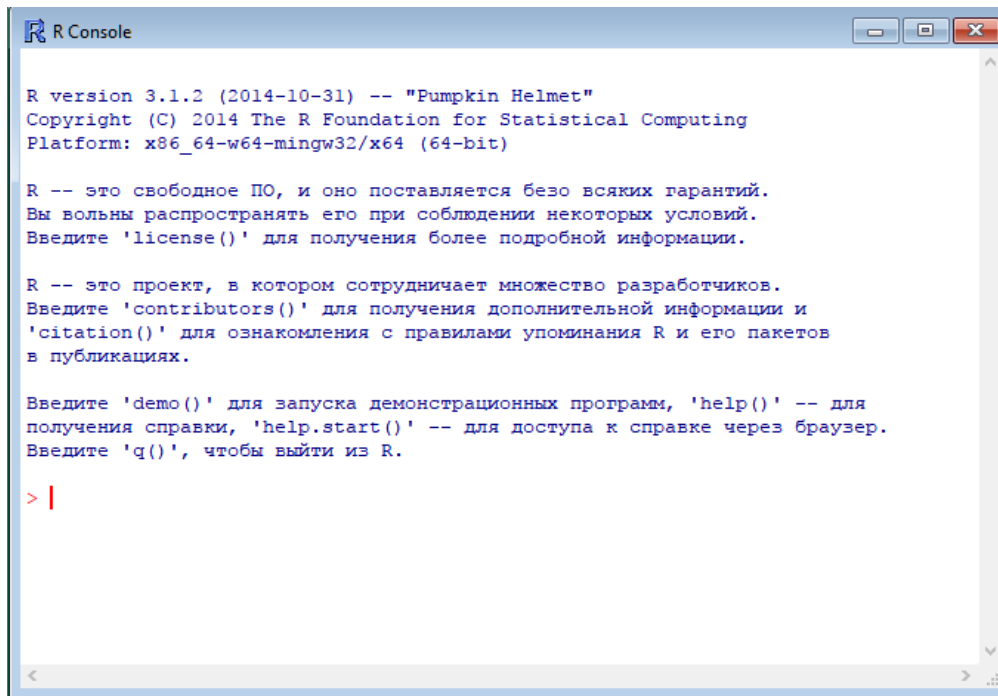
```
> n1=integer(1)
> a=integer(1)
> c=integer(1)
> m=integer(1)
>
> a=2
> c=4
> m=7
> name<-c("st1", "st2", "st3", "st4", "st5", "st6", "st7", "st8", "st9", "st10")
> year_birth<-c(1984, 1985,1983,1984,1985,1984,NA,1985,1985,1984)
> gpa<-c(4.61, 8.2, 8.5, 9, 7.77, 6.66,8.25, 8.96, 7.45, 7)
> absenteeism<-c(22, 2, 0, 1, NA, 4, 1, 0, 6, NA)
> group<- data.frame(name,year_birth,gpa,absenteeism)
> group
  name year_birth gpa absenteeism
1 st1      1984 4.61          22
2 st2      1985 8.20           2
3 st3      1983 8.50           0
4 st4      1984 9.00           1
5 st5      1985 7.77          NA
6 st6      1984 6.66           4
7 st7        NA 8.25           1
8 st8      1985 8.96           0
9 st9      1985 7.45           6
10 st10     1984 7.00          NA
> group1<-na.omit(group)
> agriculture
```

The Environment pane shows the Global Environment with the following data:

| Object | Class | Attributes |
|-------------|------------|---|
| group | data.frame | 10 obs. of 4 variables |
| group1 | data.frame | 7 obs. of 4 variables |
| a | numeric | 2 |
| absenteeism | numeric | [1:10] 22 2 0 1 NA 4 1 0 6 NA |
| c | numeric | 4 |
| gpa | numeric | [1:10] 4.61 8.2 8.5 9 7.77 6.66 8.25 8.96 7.45 7 |
| m | numeric | 7 |
| name | character | [1:10] "st1" "st2" "st3" "st4" "st5" "st6" "st7" "st8" "st9" "st10" |
| n1 | numeric | 0L |
| year_birth | numeric | [1:10] 1984 1985 1983 1984 1985 ... |

The bottom pane displays the RStudio website, titled "Statistical Data Analysis", with links to Manuals, Reference, and Miscellaneous Material.

Введение в статистический пакет R.



```
R Console

R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-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
'scitation()' 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.

> |
```

- `setwd('~/.')` – установить директорию
- `getwd('~/.')` – уточнить текущую директорию

Введение в статистический пакет R.

- CTRL+Enter – запускает на выполнение строку (выделенные строки) и переходит на следующую
- ALT+Enter – запускает на выполнение строку (выделенные строки) и остается на текущей
- ALT+ - (минус) - вводит оператор присваивания <-
- CTRL+SHIFT+C – комментирует выделенный блок
- CTRL+L – очистка консоли

4. Введение в статистический пакет R. Алфавит языка. Синтаксис инструкций. Комментарии

- Алфавит: буквы, цифры, знаки (. , _ = > < - и др.)
- Строчные и заглавные буквы различаются. Имена переменных (идентификаторы) состоят из букв, цифр и знаков точки (.) и подчёркивания (_). Имя объекта не может начинаться с цифры, и если первый символ — это точка, то цифра не может быть вторым символом.
- Операторы:
 - Присваивания: =, <-, ->
 - Арифметические: +, -, *, /, ^, %/%, %%
 - Сравнения: ==, !=, <, >, <=, >=, &, |, !
- # - начало комментария;
- на все функции (встроенные) есть описание
 - help("названиеф-ции")
 - ?названиеф-ции

4. Введение в статистический пакет R. Основные вычисления, присваивание переменных

```
1+1
```

```
[1] 2
```

```
2*2
```

```
[1] 4
```

```
3^2
```

```
[1] 9
```

```
Variable <-2
```

```
print (Variable+10)
```

```
[1] 12
```

```
Variable+10
```

```
[1] 12
```

```
v1 <- 1
```

```
2 -> v1
```

```
v1 = 3
```

```
assing("v1", 4)
```


4. Введение в статистический пакет R.

Объекты

- Объекты для хранения данных (переменные, векторы, матрицы и массивы, списки, таблицы данных)
- Функции – поименованные программы, предназначенные для создания новых объектов или выполнения действий над ними
- Объекты среды R комплектуются в пакеты
- Пакеты инсталлируются в определенных директориях
- Для установки пакета вводят команду вида:

```
install.packages(c("vegan", "xlsReadWrite", "car"))
```

- Для инициализации пакета перед его использованием вводят команду вида:

```
library (<имя_пакета>).
```

- Модель ООП в R: метод – функция, вызываемая другой универсальной (generic) функцией, например:

```
predict.smooth.spline()
```

4. Введение в статистический пакет R.

Типы данных

- **numeric** - объекты, к которым относятся целочисленные (integer) и действительные (double или real) числа;
- **logical** - логические объекты, принимающие два значения FALSE (F) и TRUE (T);
- **character** - символьные объекты (значения переменных задаются либо в двойных, либо в одинарных кавычках);
- **complex** – объекты комплексного типа (представление комплексных чисел);

4. Введение в статистический пакет R.

Типы данных

```
as.numeric(TRUE)
```

```
[1] 1
```

```
as.character(4.9)
```

```
[1] "4.9"
```

```
as.numeric(4.9)
```

```
[1] 4.9
```

```
as.integer(4.9)
```

```
[1] 4
```

```
as.numeric("Hello")
```

```
[1] NA
```

Warning message: NAs
introduced by coercion

4. Введение в статистический пакет R.

Типы данных

- **Концепция типа данных основывается на следующих положениях:**
 1. Любой тип данных определяет множество значений, к которому принадлежит константа, которые может принимать переменная или выражение или вырабатывать операция или функция.
 2. Каждая операция или функция требует аргументов фиксированного типа и выдает результат фиксированного типа. Если операция допускает аргументы нескольких типов, то тип результата можно определить по специальным правилам языка.
- **Статическая типизация (Fortran, Pascal, C, C++):**
 3. Тип значения, задаваемого константой, переменной или выражением, можно определить по их виду или описанию и остается неизменным для переменных.
- **Динамическая типизация (R, Python, Ruby, Perl):**
 3. Тип значения, задаваемого константой, переменной или выражением определяется присвоенным или выработанным им значением в момент присваивания (выработки), может быть определен по их значению и для переменных изменен в процессе выполнения программы .

4. Введение в статистический пакет R.

- Inf – положительная или отрицательная бесконечность (обычно результат деления вещественного числа на 0);
- NA – "отсутствующее значение" (Not Available);
- NaN – "не число" (Not a Number).

4. Введение в статистический пакет R. Структуры данных (объекты). Векторы

- **Вектор** представляет собой поименованный одномерный объект, содержащий набор однотипных элементов (числовые, логические либо текстовые значения – сочетания не допускаются).
- Для создания векторов используются функции
 - **c()** – функция конкатенации, в качестве аргументов указываются объединяемые в вектор значения; `y<- c(2, 4, 6, 8)`
 - **scan()** – функция считывает последовательно либо вводимые с клавиатуры значения, либо из файла; `y<- scan("scan.txt", what = double(50))`
 - **seq()** – функция создает векторы, содержащие последовательную совокупность чисел; `y<- seq(1,7)`, `y<- seq(from = 1, to = 4, by = 0.5)`
 - **rep()** - функция создает векторы, содержащие одинаковые значения; `rep("test",7)`
 - **integer(n), logical(n)** - создают векторы длины n соответствующего типа, инициализированные значениями по умолчанию.
- Всем компонентам вектора присваиваются индексные номера, начиная с 1.
- `y[7]`, `y[5:8]` – обращения к элементам вектора `y`.

4. Введение в статистический пакет R. Структуры данных (объекты). Векторы

- **which()**

```
x=scan()
```

```
1: 1
```

```
2: 2
```

```
3: 5
```

```
4: 6
```

```
5: 7
```

```
6: 8
```

```
7: 4
```

```
8: 2
```

```
9: 0
```

```
10: 45
```

```
11: 5
```

```
12: Read 11 items
```

```
which(x>10)
```

```
[1] 10
```

```
which(x==2)
```

```
[1] 2 8
```

4. Введение в статистический пакет R. Структуры данных (объекты). Матрицы

- Для создания матриц используется функция **matrix()**
- по умолчанию заполнение матрицы происходит по столбцам, порядок меняется аргументом `byrow = TRUE`
- Доступ к элементам матрицы происходит по индексу. `M[i, j]` ссылается на элемент *i*-й строки и *j*-го столбца матрицы `M`
- `dim()` – определяет количество строк и столбцов матрицы;
- `t()` – транспонирует матрицу

4. Введение в статистический пакет R. Структуры данных (объекты). Матрицы

```
MA <- matrix (seq(1,16), nrow = 4, ncol = 4, byrow = TRUE)
```

```
MA
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
[4,]   13   14   15   16
```

```
MA <- matrix (seq(1,16), nrow = 4, ncol = 4)
```

```
MA
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
```

```
MA <- matrix (c(1,2,3), nrow = 2, ncol = 3)
```

```
MA
     [,1] [,2] [,3]
[1,]    1    3    2
[2,]    2    1    3
```

- **m1+m2** #сложение
- **m1*2** #умножение на число
- **m1%*%m2** #умножение матриц
- **solve** #нахождение обратной
- **colSums()** #суммы по столбцам
- **rowSums()** #суммы по строкам

4. Введение в статистический пакет R. Структуры данных (объекты). Матрицы

```
m1 <- matrix (1, nrow = 2, ncol = 3)
```

```
m1  
      [,1] [,2] [,3]  
[1,]    1    1    1  
[2,]    1    1    1
```

```
m1[,2] #печатает 2-ой столбец
```

```
m1[2,] <- c(3, 4, 5) #меняет данные во второй строке
```

```
m1  
      [,1] [,2] [,3]  
[1,]    1    1    1  
[2,]    3    4    5
```

```
m1[2,3]
```

```
5
```

```
m2 <- matrix (0, nrow = 3, ncol = 3)
```

```
diag(m2) <- 1
```

```
m2  
      [,1] [,2] [,3]  
[1,]    1    0    0  
[2,]    0    1    0  
[3,]    0    0    1
```

4. Введение в статистический пакет R. Структуры данных (объекты). Матрицы

```
m1
      [,1] [,2] [,3]
[1,]    1    1    1
[2,]    3    4    5
```

```
m2
      [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```

```
rbind(m1,m2)
      [,1] [,2] [,3]
[1,]    1    1    1
[2,]    3    4    5
[3,]    1    0    0
[4,]    0    1    0
[5,]    0    0    1
```

```
cbind(m1,m2) Error in cbind(m1, m2) : number of rows of matrices must match (see arg 2)
```

```
m3 <- matrix (1, nrow = 3, ncol = 2)
```

```
cbind(m2,m3)
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    1    1
[2,]    0    1    0    1    1
[3,]    0    0    1    1    1
```

4. Введение в статистический пакет R. Структуры данных (объекты). Списки

- Списки могут включать в себя сочетание любых типов данных.
- Для создания списков используется функция **list()**
- **Пример:**
 - `> v1 <- c("A", "Б", "B")`
 - `> v2 <- seq(1, 3, 0.5)`
 - `> v3 <- c(FALSE, TRUE)`
 - `> spisok <- list(Name=v1, Numb=v2, L=v3)`
 - `> spisok`
 - `$Name`
 - `[1] "A" "Б" "B"`
 - `$Numb`
 - `[1] 1.0 1.5 2.0 2.5 3.0`
 - `$L`
 - `[1] FALSE TRUE`
- Обращение к элементам списка: `spisok$Name[2]`, `spisok$Numb[2:4]`, `spisok[[1]]`, `spisok[[2]][2:4]`

4. Введение в статистический пакет R.

Структуры данных (объекты). Фреймы(таблицы)

- Фреймы (таблицы) - основной класс объектов R, используемый для хранения данных.
- Создание фреймов - функция **data.frame()**; аргументы - произвольное количество элементов (столбцов) фрейма. Элементы фрейма данных - векторы, факторы, матрицы, списки или другие фреймы. Все векторы должны иметь одинаковую длину, а матрицы и фреймы - одинаковое число строк.
- Загрузка наборов данных из файла в фрейм - функция **read.table**(file, header = FALSE, sep = " ", ...), на вход подается путь к текстовому файлу с данными (значения в каждой строке разделяются символом sep). Параметр header позволяет указать, следует ли интерпретировать первую строку файла как имена столбцов таблицы.

4. Введение в статистический пакет R. Структуры данных (объекты). Фреймы(таблицы)

- **str ()** – функция, отображающая структуру объекта;
- **name ()** – отображает имена переменных, входящих в таблицу данных;
- **head ()** – отображает несколько первых значений каждой переменной, входящей в состав таблицы;
- **tail ()** - отображает несколько последних значений каждой переменной, входящей в состав таблицы;
- Для внесения изменений в таблицу можно воспользоваться встроенным редактором данных. Запускается из меню *Правка – Редактор данных* , либо командой **fix ()**;

4. Введение в статистический пакет R. Структуры данных (объекты). Фреймы(таблицы)

```
v1 <- c(1,2,3,4,5)
v2 <- c("name1", "name2", "name3", "name4", "name5")
v3 <- c(TRUE,TRUE,TRUE,FALSE, TRUE)
v4 <- c(7.2, 8, 5.6, 9.1, 8.35)
mydata <- data.frame(v1,v2,v3,v4)
names(mydata) <- c("ID", "Name", "Logexp", "Average")
```

```
  ID Name Logexp Average
1  1 name1  TRUE   7.20
2  2 name2  TRUE   8.00
3  3 name3  TRUE   5.60
4  4 name4 FALSE   9.10
5  5 name5  TRUE   8.35
```

```
mydata[[2]]
[1] name1 name2 name3 name4 name5
mydata$Average[4]
[1] 9.1
```

4. Введение в статистический пакет R. Функции. Встроенные функции и библиотеки

```
имя_функции <- function(arg1, arg2,...)  
{  
  группа_выражений  
  return(object)  
}
```

Оператор `return()` нужен в случаях, когда группа выражений не возвращает целевого результата.

Перед своим первым выполнением функция должна быть определена в текущем скрипте, либо загружена с помощью команды `source()` из скриптового файла, где она была предварительно подготовлена.

4. Введение в статистический пакет R. Функции. Пример 1

```
stat_param <- function(x)
{
  a<-mean(x); b<-sd(x);
  x<-c(MEAN=a, SD=b)
  x # Вывод на экран
}
```

```
source("my_function.R")
stat_param (1:10)
```

```
MEAN  SD
5.5    3.02765
```

4. Введение в статистический пакет R. Функции. Пример 2

```
stat_param <- function(x)
{
  a<-mean(x); b<-sd(x);
  return (c(MEAN=a, SD=b)) # Возвращаемое значение
}
```

```
source("my_function.R")
z<-stat_param (1:10)
z
```

```
MEAN  SD
5.5    3.02765
```

4. Введение в статистический пакет R. Функции. Пример 3

```
stat_param <- function(x=1:20) # значение аргумента по умолчанию
{
  a<-mean(x); b<-sd(x);
  x<-c(MEAN=a, SD=b)
  return (x)
}
```

```
source("my_function.R")
z<-stat_param (1:10)
```

```
z
  MEAN  SD
5.5    3.02765
```

```
z<-stat_param()
```

```
z
  MEAN  SD
10.5   5.91608
```

4. Введение в статистический пакет R. Функции. Пример 4

```
stat_param <- function(x, y=1:10)
{
  a<-mean(x); b<-sd(x);
  c<-mean(y); d<-sd(y);
  x<-c(MEANx=a, SDx=b, MEANy=c, SDy = d)
  return (x)
}
```

```
source("my_function.R")
```

```
z<-stat_param (y=1:5, x=1:10) # сопоставление параметров по имени
```

```
z
```

| MEANx | SDx | MEANy | SDy |
|-------|---------|-------|----------|
| 5.5 | 3.02765 | 3.0 | 1.581139 |

```
z<-stat_param (1:4) # сопоставление параметра по порядку следования
```

```
z
```

| MEANx | SDx | MEANy | SDy |
|-------|----------|-------|---------|
| 2.5 | 1.290994 | 5.5 | 3.02765 |

4. Введение в статистический пакет R. Математические и статистические функции

- `sin(x)`, `cos(x)`, `tan(x)`,
- `exp(x)`, `log(x)`, `log10(x)`, `log(x, base)`,
- `max(x)`, `min(x)`,
- `sum(x)`, `prod(x)`,
- `mean(x)`,
- `median(x)`,
- `quantile(x, probs = seq(0, 1, 0.25))`,
- `var(x)`, `sd(x)`.

4. Введение в статистический пакет R.

Ввод и вывод

- `getwd()` – текущая директория;
- `setwd(dir)` – изменить текущую директорию на `dir`;
- `library(package)` – подключить пакет `package`;
- `read.table()` – считывает таблицу данных и создаёт по ним `data.frame`;
- `write.table()` – печатает объект, конвертируя его в `data.frame`;
- `read.csv()` – считывает csv-файл;

4. Введение в статистический пакет R.

Циклы

- **for** (index in for_object)
 выражение
- **for** (index in for_object)
 {
 группа_выражений
 }
- for_object – выражение, результатом которого является вектор (числовая последовательность)

```
s <- 0  
for (i in 1:10)  
  s <- s+i  
s
```

4. Введение в статистический пакет R.

Циклы

- **while** (логическое_выражение)

```
{  
  группа_выражений  
}
```

- **repeat**

```
{  
  группа_выражений  
  if (условие_выхода) break  
}
```

```
z <- 10  
repeat  
{  
  z <- z-1  
  if (z<0) break  
}  
z  
  
-1
```


4. Введение в статистический пакет R.

Ветвления

- `if(логическое_выражение)`
 `выражение_1`
`else`
 `выражение_2`
- `if(логическое_выражение)`
 {
 `группа_выражений_1`
 }
`else`
 {
 `группа_выражений_2`
 }
- `ifelse (условие, выражение_1, выражение_2)`

4. Введение в статистический пакет R. Ветвления. Пример

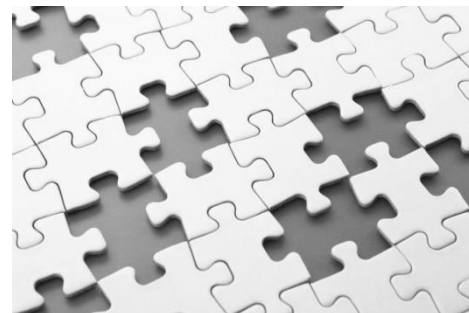
```
compare <- function(x, y)
{
  z=length(x) - length(y)
  if(z>0)
    cat ("Первый вектор имеет на ",z," элементов больше")
  else
    if (z<0)
      cat ("Второй вектор имеет на ", -z ," элементов больше")
    else
      cat("Количество элементов одинаково ",n1)
}
```

```
x <- c(1:4); y <- c(1:9)
compare (x, y)
```

Первый вектор имеет на 5
элементов больше

4. Введение в статистический пакет R. Пропущенные данные.

NA – отсутствующие данные;



- Идентификация недостающих данных.
- Исследование закономерностей появления отсутствующих значений.
- Формирование наборов данных, не содержащих пропуски (в результате удаления или замены соответствующих фрагментов).
- **is.na** () – позволяет проверить данные на наличие пропущенных значений; возвращает объект такой же размерности, где элементы заменены TRUE, если значение было пропущено и FALSE – в противном случае;

4. Введение в статистический пакет R. Пропущенные данные.

- **na.rm=TRUE** – параметр, который удаляет пропущенные значения перед вычислениями и позволяет применить функцию к оставшимся элементам;
 - `x<-c(2,4, NA, 3)`
 - `y<-sum(x)`
 - `y`
 - `[1] NA`

 - `x<-c(2,4, NA, 3)`
 - `y<-sum(x, na.rm=TRUE)`
 - `y`
 - `[1] 9`
- **na.omit ()** – функция, удаляющая все строки, в которых есть хотя бы одно пропущенное значение
- **complete.cases()** - возвращает логический вектор, указывающий какие данные являются полными

4. Введение в статистический пакет R. Пропущенные данные.

```
name<-c("st1", "st2", "st3", "st4", "st5", "st6", "st7", "st8", "st9", "st10")
year_birth<-c(1984, 1985,1983,1984,1985,1984,NA,1985,1985,1984)
gpa<-c(4.61, 8.2, 8.5, 9, 7.77, 6.66,8.25, 8.96, 7.45, 7)
absenteeism<-c(22, 2, 0, 1, NA, 4, 1, 0, 6, NA)
group<- data.frame(name,year_birth,gpa,absenteeism)
group
```

| | name | year_birth | gpa | absenteeism |
|----|------|------------|------|-------------|
| 1 | st1 | 1984 | 4.61 | 22 |
| 2 | st2 | 1985 | 8.20 | 2 |
| 3 | st3 | 1983 | 8.50 | 0 |
| 4 | st4 | 1984 | 9.00 | 1 |
| 5 | st5 | 1985 | 7.77 | NA |
| 6 | st6 | 1984 | 6.66 | 4 |
| 7 | st7 | NA | 8.25 | 1 |
| 8 | st8 | 1985 | 8.96 | 0 |
| 9 | st9 | 1985 | 7.45 | 6 |
| 10 | st10 | 1984 | 7.00 | NA |

```
na.omit(group)
```

| | name | year_birth | gpa | absenteeism |
|---|------|------------|------|-------------|
| 1 | st1 | 1984 | 4.61 | 22 |
| 2 | st2 | 1985 | 8.20 | 2 |
| 3 | st3 | 1983 | 8.50 | 0 |
| 4 | st4 | 1984 | 9.00 | 1 |
| 6 | st6 | 1984 | 6.66 | 4 |
| 8 | st8 | 1985 | 8.96 | 0 |
| 9 | st9 | 1985 | 7.45 | 6 |

```
complete.cases(group)
```

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

4. Введение в статистический пакет R. Пропущенные данные.

Замена пропущенных значений средним, медианой, модой (в случае, когда вариация данных невелика);

- `library(Hmisc)`
- `impute(имя столбца, mean) # заменить средним`
- `impute(имя столбца, median) # медианой`
- `impute(имя столбца, 20) # заменить заданным числом`

```
impute(group$year_birth, mean)
```

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------|----------|----------|----------|----------|----------|-----------|----------|----------|----------|
| 1984.000 | 1985.000 | 1983.000 | 1984.000 | 1985.000 | 1984.000 | 1984.333* | 1985.000 | 1985.000 | 1984.000 |

```
impute(group$year_birth, median)
```

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|------|------|------|------|------|------|-------|------|------|------|
| 1984 | 1985 | 1983 | 1984 | 1985 | 1984 | 1984* | 1985 | 1985 | 1984 |

4. Введение в статистический пакет R. Графическое представление данных.

- `plot()` – график;
- `hist()` – гистограмма;
- `boxplot()` – ящик с усами;
- `scatterplot()` – диаграмма рассеяния.

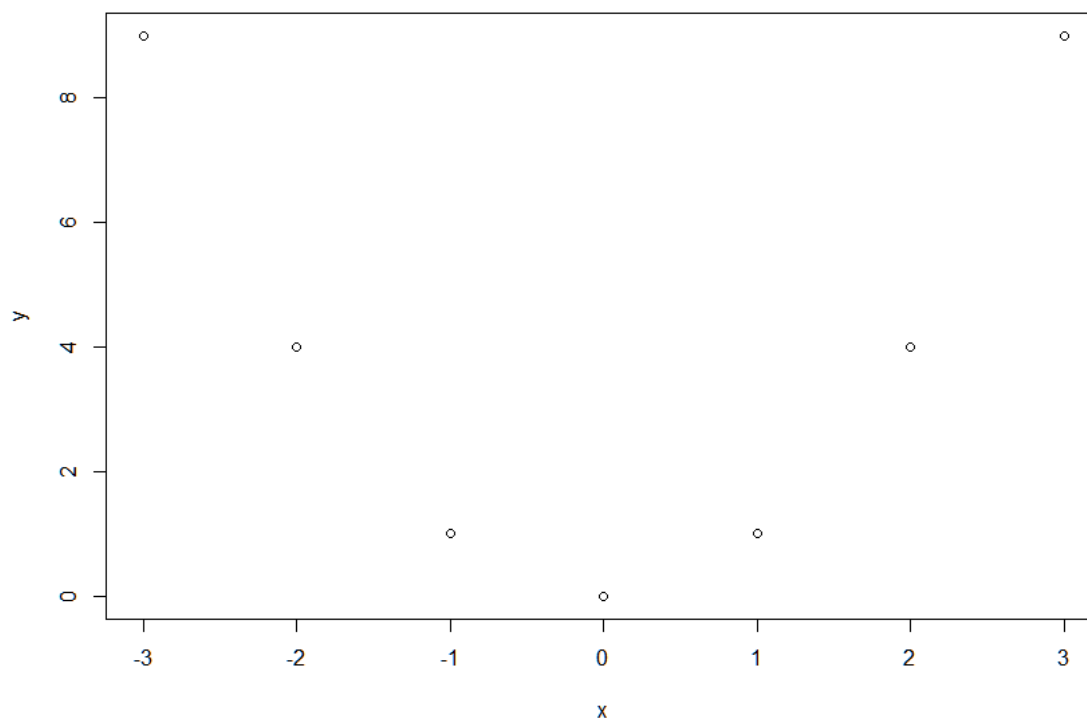
дополнительные элементы:

- `lines()`,
- `points()`,
- `text()`,
- `axis()`.

4. Введение в статистический пакет R. Графическое представление данных.

● plot()

```
x=c(-3,-2,-1,0,1,2,3)
> y=x^2
> y
[1] 9 4 1 0 1 4 9
> plot(x,y)
```



4. Введение в статистический пакет R. Графическое представление данных.

xlab, ylab – подписи осей;

type – вид точек на графике;

"p" – точки (*points*; используется по умолчанию)

"l" – линии (*lines*)

"b" – изображаются и точки, и линии (*both points and lines*)

"o" – точки изображаются поверх линий (*points over lines*)

"h" – гистограмма (*histogram*)

"s" – ступенчатая кривая (*steps*)

"n" – данные не отображаются (*no points*)

xlim, ylim – предельные значения на осях;

axes, ann – отображение осей и названий;

main – заголовок графика;

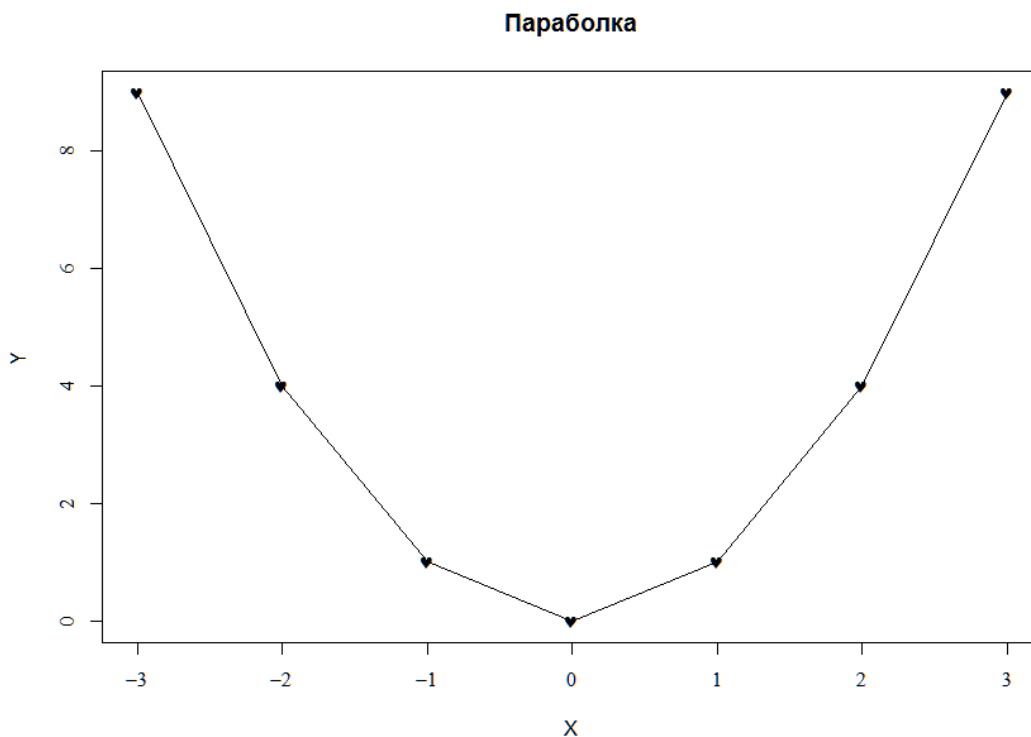
Размеры и цвета маркеров, осей, заголовков, типы линий, рамки и т.п.

4. Введение в статистический пакет R. Графическое представление данных.

```
x=c(-3,-2,-1,0,1,2,3)
```

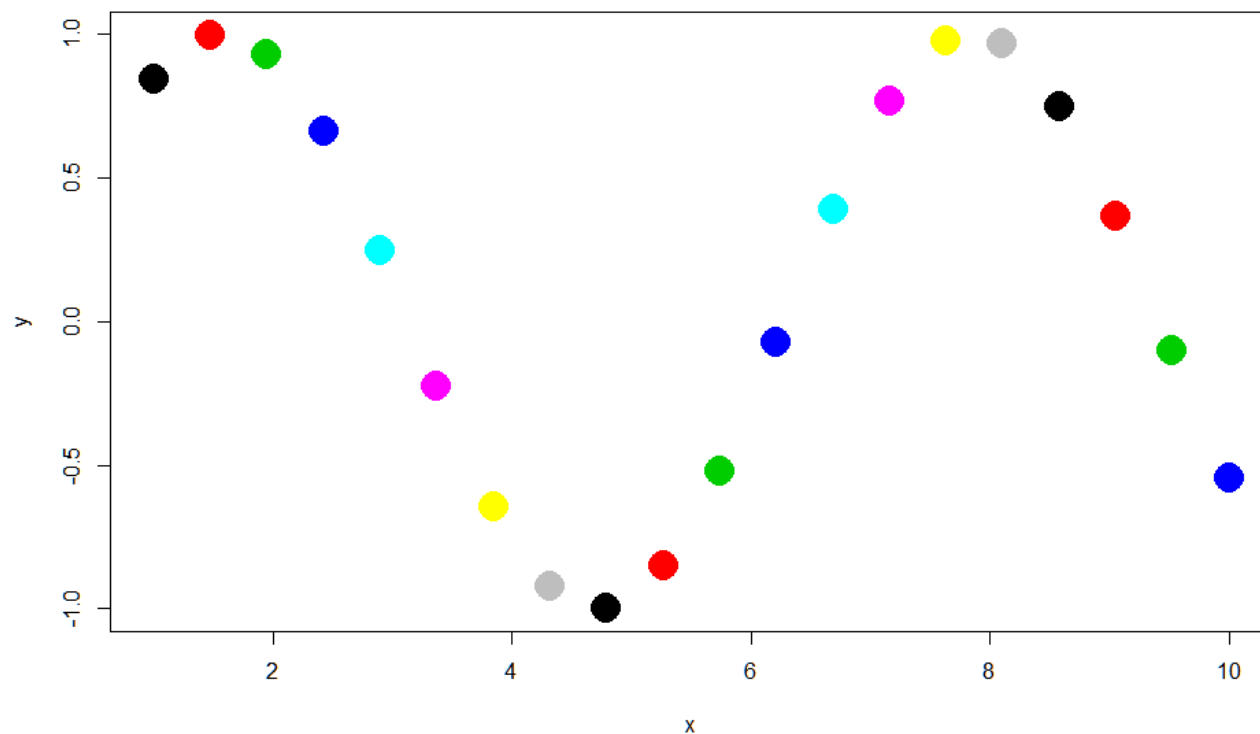
```
y=x^2
```

```
plot(x, y, xlab = "X", ylab = "Y", main = "Параболка", type = "o", pch = 169, font = 5)
```



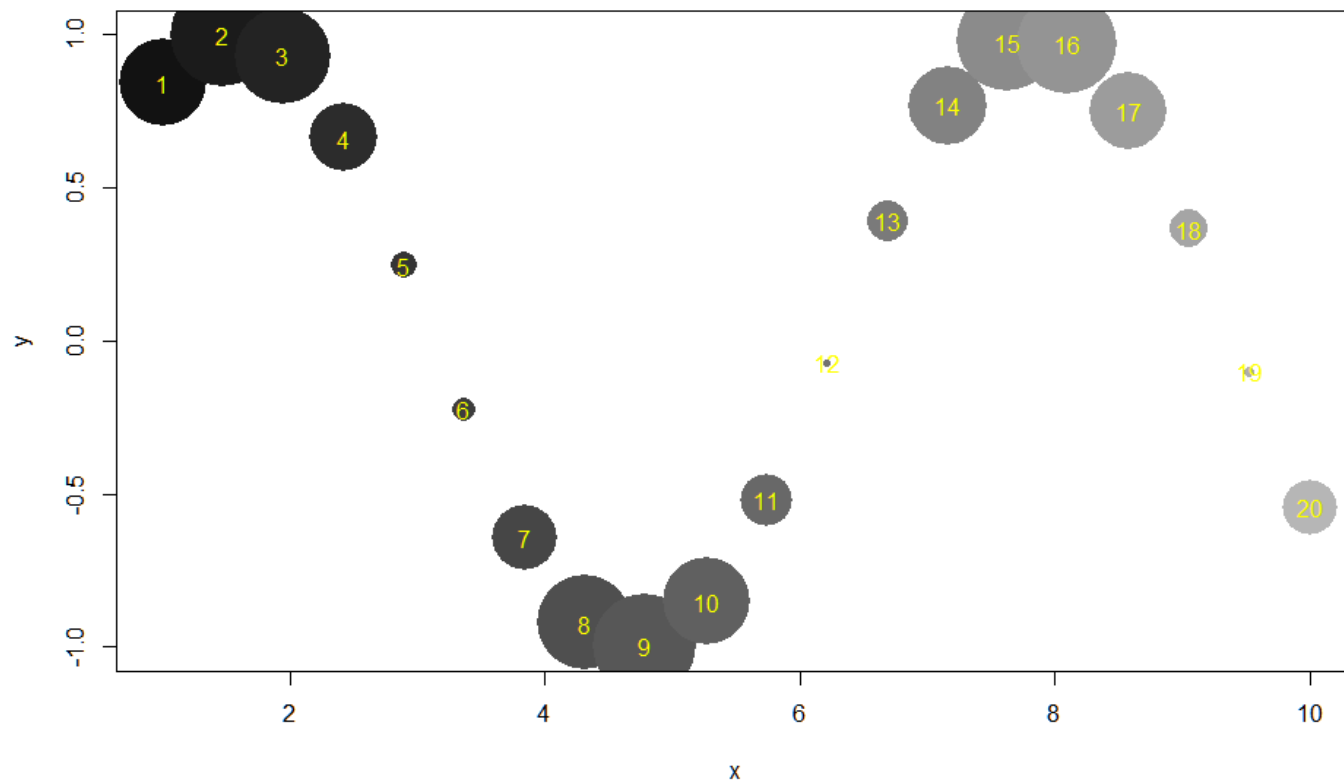
4. Введение в статистический пакет R. Графическое представление данных.

```
x <- seq(from=1, to=10, length=20)  
y <- sin(x)  
plot(x, y, pch=16, cex=3, col=1:20)
```



4. Введение в статистический пакет R. Графическое представление данных.

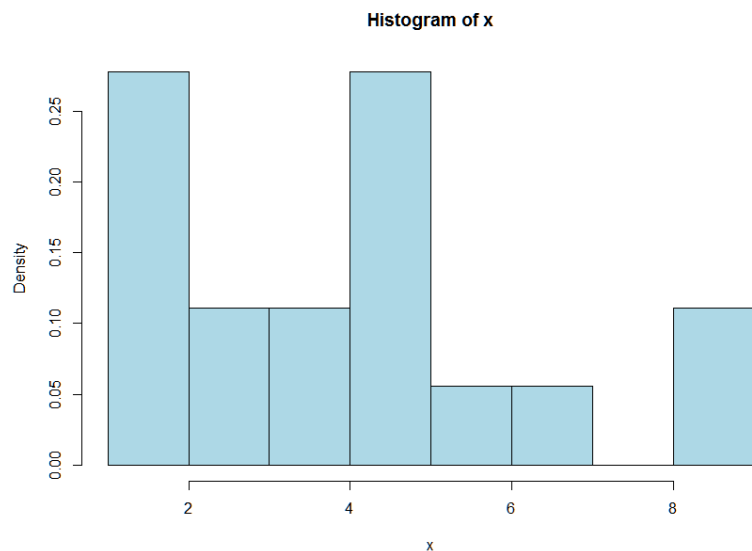
```
x <- seq(from=1, to=10, length=20)
y <- sin(x)
plot(x, y, pch=16, cex=10*abs(y), col=grey(x/14))
text(x, y, 1:20, col="yellow")
```



4. Введение в статистический пакет R. Графическое представление данных.

```
x=c(2.3, 5, 1.2, 3.5, 2, 5, 4.3, 6, 3.9, 8.7, 2.5, 7, 1, 4.1, 4.8, 2, 1.13, 8.12)  
hist(x, freq = FALSE, col = "lightblue")
```

1 1,13 1,2 2 2 2,3 2,5 3,5 3,9 4,1 4,3 4,8 5 5 6 7 8,7 8,12



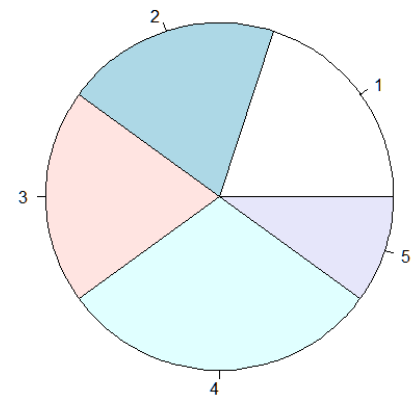
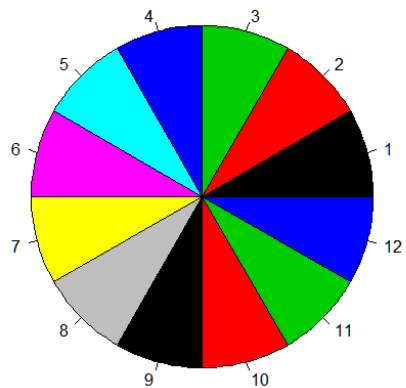
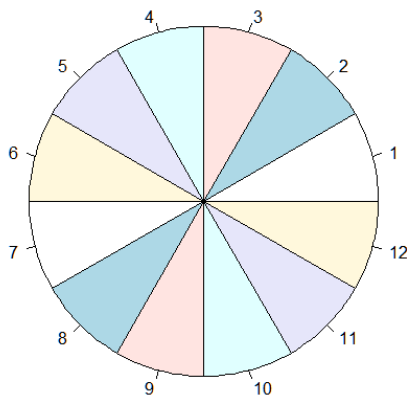
breaks – количество разбиений

4. Введение в статистический пакет R. Графическое представление данных.

```
pie(rep(1,12))
```

```
pie(rep(1,12), col=1:12)
```

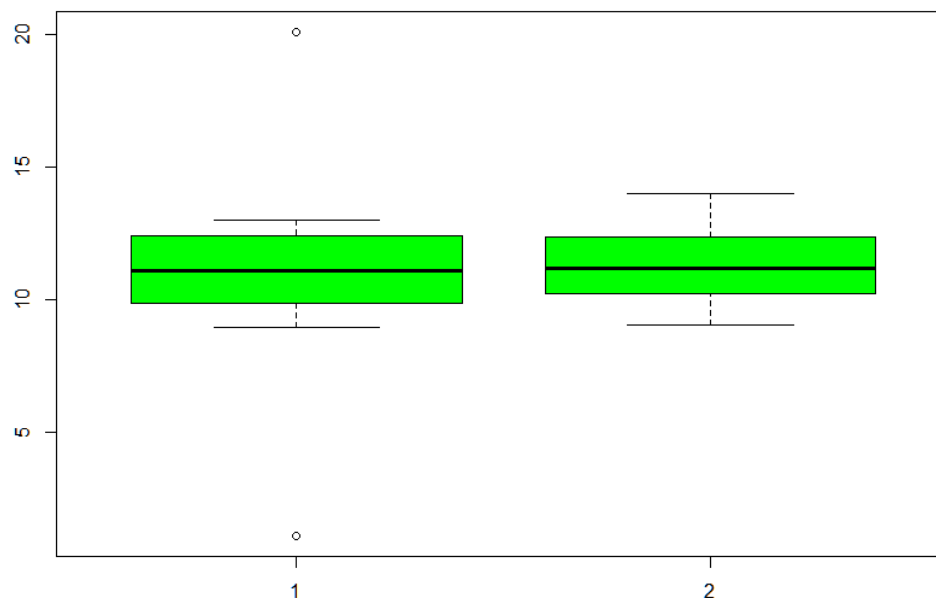
```
x<-c(20, 20, 20, 30, 10)  
pie(x)
```



4. Введение в статистический пакет R. Графическое представление данных.

```
r1=c(12.5, 9.32, 12.26, 20.08, 10.74, 8.96, 1.08, 13, 12, 12.56, 9.69, 10.23, 11.56, 11.08, 9.99)  
r2=c(12.58, 9.4, 12.34, 12.36, 10.82, 9.04, 11.12, 14, 12.08, 12.64, 9.77, 10.31, 11.64, 11.16, 10.07)  
boxplot(r1, r2, col = "green")
```

| ряд 1 | ряд 2 |
|--------------|-------|
| 12,5 | 12,58 |
| 9,32 | 9,4 |
| 12,26 | 12,34 |
| 20,08 | 12,36 |
| 10,74 | 10,82 |
| 8,96 | 9,04 |
| 1,08 | 11,12 |
| 13 | 14 |
| 12 | 12,08 |
| 12,56 | 12,64 |
| 9,69 | 9,77 |
| 10,23 | 10,31 |
| 11,56 | 11,64 |
| 11,08 | 11,16 |
| 9,99 | 10,07 |



4. Введение в статистический пакет R. Описательная статистика на примере

```
library(reshape2)
```

```
tips
```

| | total_bill | tip | sex | smoker | day | time | size |
|----|------------|------|--------|--------|-----|--------|------|
| 1 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 |
| 2 | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 |
| 3 | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 |
| 4 | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 |
| 5 | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 |
| 6 | 25.29 | 4.71 | Male | No | Sun | Dinner | 4 |
| 7 | 8.77 | 2.00 | Male | No | Sun | Dinner | 2 |
| 8 | 26.88 | 3.12 | Male | No | Sun | Dinner | 4 |
| 9 | 15.04 | 1.96 | Male | No | Sun | Dinner | 2 |
| 10 | 14.78 | 3.23 | Male | No | Sun | Dinner | 2 |

```
str(tips)
```

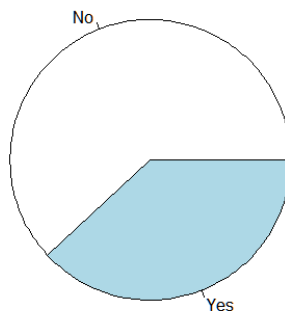
```
'data.frame':      244 obs. of  7 variables:
 $ total_bill: num  17 10.3 21 23.7 24.6 ...
 $ tip       : num  1.01 1.66 3.5 3.31 3.61 4.71 2 3.12 1.96 3.23 ...
 $ sex       : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 2 2 2 2 2 ...
 $ smoker    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ day       : Factor w/ 4 levels "Fri","Sat","Sun",...: 3 3 3 3 3 3 3 3 3 3 ...
 $ time      : Factor w/ 2 levels "Dinner","Lunch": 1 1 1 1 1 1 1 1 1 1 ...
 $ size      : int  2 3 3 2 4 4 2 4 2 2 ...
```


4. Введение в статистический пакет R. Описательная статистика на примере

```
Mytips <- tips  
tab <- table(Mytips$smoker)  
tab
```

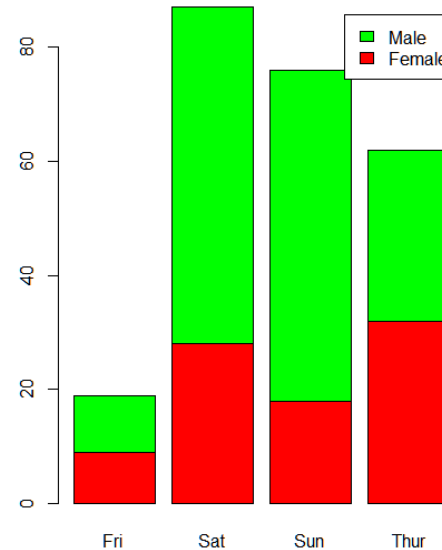
```
  No  Yes  
151  93
```

```
pie(tab)
```



```
tab1 <- table(Mytips$sex, Mytips$day)  
tab1
```

| | Fri | Sat | Sun | Thur |
|--------|-----|-----|-----|------|
| Female | 9 | 28 | 18 | 32 |
| Male | 10 | 59 | 58 | 30 |



```
barplot(tab1, legend.text=rownames(tab1), col=c("red", "green"))
```

4. Введение в статистический пакет R. Описательная статистика на примере

```
c(minTip=min(Mytips$tip), maxTip=max(Mytips$tip))
```

```
minTip maxTip  
1      10
```

```
meanTip=mean(Mytips$tip)
```

```
[1] 2.998279
```

```
sdTip=sd(Mytips$tip)
```

```
[1] 1.383638
```

```
summary(tips)
```

| total_bill | tip | sex | smoker | day | time | size |
|---------------|----------------|------------|---------|---------|------------|--------------|
| Min. : 3.07 | Min. : 1.000 | Female: 87 | No :151 | Fri :19 | Dinner:176 | Min. :1.00 |
| 1st Qu.:13.35 | 1st Qu.: 2.000 | Male :157 | Yes: 93 | Sat :87 | Lunch : 68 | 1st Qu.:2.00 |
| Median :17.80 | Median : 2.900 | | | Sun :76 | | Median :2.00 |
| Mean :19.79 | Mean : 2.998 | | | Thur:62 | | Mean :2.57 |
| 3rd Qu.:24.13 | 3rd Qu.: 3.562 | | | | | 3rd Qu.:3.00 |
| Max. :50.81 | Max. :10.000 | | | | | Max. :6.00 |

4. Введение в статистический пакет R. Описательная статистика на примере

```
boxplot(tip~sex, data=Mytips, horizontal = TRUE)
```

```
boxplot(tip~day, data=Mytips, verticale = TRUE)
```

