

- Типы данных
- Импорт данных
- Ввод с клавиатуры
- Ввод баз данных в программную среду
- Просмотр данных
- Пропущенные значения
- Дата и время

Данные в **R** могут быть представлены в виде большого числа типов данных (скаляры, векторы, матрицы, базы данных, списки).

Векторы

```
a <- c(1,2,5.3,6,-2,4) # числовой вектор
b <- c("one","two","three") # строковый вектор
c <- c(TRUE,TRUE,TRUE,FALSE,TRUE,FALSE)
#логический вектор
Выбор отдельных элементов вектора
a[c(2,4)] # 2-ой и 4-ый элементы вектора a
```

Матрицы

Все колонки матрицы должны быть одного типа и одной и той же длины

Общая форма:

```
mymatrix <- matrix(vector, nrow=r, ncol=c,
  byrow=FALSE,dimnames=list(char_vector_rownames
  , char_vector_colnames))
```

byrow=**TRUE** означает, что матрица заполняется по строкам

byrow=**FALSE** означает, что матрица заполняется по столбцам

dimnames – опционально: имена строк и столбцов

Матрицы

```
# создание числовой матрицы 5 x 4
y<-matrix(1:20, nrow=5,ncol=4)
# другой пример
cells <- c(1,26,24,68)
rnames <- c("R1", "R2")
cnames <- c("C1", "C2")
mymatrix <- matrix(cells, nrow=2, ncol=2,
  byrow=TRUE, dimnames=list(rnames, cnames))
#Определение отдельных элементов матрицы
x[,4] # 4-ая колонка матрицы
x[3,] # 3-ая колонка матрицы
x[2:4,1:3] # строки 2,3,4, столбцы 1,2,3
```

Массивы

Массивы имеют структуру, схожую с матрицами, но могут иметь более 2 измерений

help(array)

Таблицы данных

Таблица данных по структуре напоминает расширенную матрицу, однако данные в разных колонках могут быть различного типа.

```
d <- c(1,2,3,4)
e <- c("red", "white", "red", NA)
f <- c(TRUE,TRUE,TRUE,FALSE)
mydata <- data.frame(d,e,f)
names(mydata) <- c("ID", "Color", "Passed")
#имена переменных
```

Таблицы данных

Существует множество способов идентификации элементов таблицы данных

```
myframe[3:5] # колонки 3,4,5
myframe[c("ID", "Age")] # колонки ID и Age
myframe$X1 # переменная x1
```

Списки

Список – нумерованная последовательность объектов (компонентов списка). Список позволяет собрать множество различных объектов (не обязательно одного и того же класса)

```
# Пример списка из 4 компонентов
w <- list(name="Fred", mynumbers=a,
mymatrix=y, age=5.3)
```

```
# Пример списка из двух списков
v <- c(list1,list2)
```

Списки

Компоненты списка определяются с использованием следующих знаков: `[[]]`
`mylist[[2]]` # 2-ой элемент списка

Факторы

Переменную, принимающую конечное число значений, можно преобразовать в фактор. Объект типа «Фактор» хранит возможные значения переменной в виде вектора.

```
# variable gender with 20 "male" entries and
# 30 "female" entries
gender <- c(rep("male",20), rep("female", 30))
gender <- factor(gender)
summary(gender)
```

Полезные функции

```
length(object) # Число элементов в объекте
str(object) # Структура объекта
class(object) # Класс объекта
names(object) # Имена элементов объекта
c(object,object,...) # соединить объекты в вектор
cbind(object, object, ...) # объединить объекты как столбцы
rbind(object, object, ...) # объединить объекты как строки
ls() # Показать список текущих объектов
rm(object) # Удалить объект
```

Ввод данных с клавиатуры

Для создания данных в интерактивном режиме используются следующие функции

```
age <- c(25, 30, 56)
gender <- c("male", "female", "male")
weight <- c(160, 110, 220)
mydata <- data.frame(age,gender,weight)
```

Ввод данных с клавиатуры

```
# Ввод данных в режиме редактора
mydata <- data.frame(age=numeric(0),
gender=character(0), weight=numeric(0))
mydata <- edit(mydata)
```

Просмотр данных

Существуют различные способы просмотра имеющихся данных

- # Просмотр объектов в рабочей среде
ls()
- # Просмотр переменных объекта
names(mydata)
- # Просмотр структуры объекта
str(mydata)
- # Размерность объекта
dim(object)

Просмотр данных

Существуют различные способы просмотра имеющихся данных

- # Просмотр класса объекта
class(object)
- # Вывод на экран объекта
mydata
- # Вывод первые 10 строк объекта
head(mydata, n=10)
- # Вывод первых 10 строк объекта
tail(mydata, n=5)