Основы анализа больших данных

Лекция 4. Многомерные массивы, списки, факторы и таблицы, таблицы данных в R.

Класс данных: многомерные массивы (array)

Массивы – объекты данных в R, которые могут хранить данные одного типа в более, чем 2 измерениях.

Основной способ создания массивов:

```
array(data = NA, dim = length(data), dimnames = NULL)
```

Пример

```
# Создадим два вектора разной длины.
vector1 <- c(5,9,3)
vector2 <- c(10,11,12,13,14,15)
# Возьмём эти векторы в качестве входных данных для массива.
a <- array(c(vector1,vector2),dim = c(3,3,2))
```

Именование столбцов и строк (параметр dimnames)

Пример

```
vector1 <- c(5,9,3)
vector2 <- c(10,11,12,13,14,15)
column.names <- c("COL1","COL2","COL3")
row.names <- c("ROW1","ROW2","ROW3")
matrix.names <- c("Matrix1","Matrix2")

a <- array(c(vector1,vector2),dim = c(3,3,2), dimnames =
       list(row.names,column.names,matrix.names))
```

Класс данных: многомерные массивы (array) – продолжение 1

Доступ к элементам массива

Пример

```
vector1 <- c(5,9,3)
vector2 <- c(10,11,12,13,14,15)
column.names <- c("COL1","COL2","COL3")
row.names <- c("ROW1","ROW2","ROW3")
matrix.names <- c("Matrix1","Matrix2")
result <- array(c(vector1,vector2),dim = c(3,3,2),
       dimnames = list(row.names, column.names,
       matrix.names))
# выводим третью строку второй матрицы массива:
print(a[3,,2])
# выводим элемент в 1й строке, 3-м столбце первой
матрицы:
print(a[1,3,1])
# выводим вторую матрицу:
print(a[,,2])
```

Операции над элементами массива

Пример

```
vector1 <- c(5,9,3)
vector2 <- c(10,11,12,13,14,15)
array1 <- array(c(vector1,vector2),dim = c(3,3,2))
vector3=c(-4,-9,-3)
vector4=c(-10,-10,-12,-13,-14,-14)
array2 <- array(c(vector3,vector4),dim = c(3,3,2))
# создадим матрицы из этих массивов
matrix1 <- array1[,,2]
matrix2 <- array2[,,2]
# сложим матрицы
result <- matrix1 + matrix2
print(result)
```

Класс данных: многомерные массивы (array) – продолжение 2

Сквозные вычисления (по элементам массива) с помощью apply()

Функция `apply(x, margin, fun)`

Пример (вычислим сумму элементов в строках массива по всем матрицам)

```
vector1 <- c(5,9,3)
vector2 <- c(10,11,12,13,14,15)
new_array <- array(c(vector1,vector2),dim = c(3,3,2))
print(new_array)

result <- apply(new_array, c(1), sum)
print(result)

# что делают следующие команды?
apply(new_array,c(1,2),sum)
apply(new_array,c(1,3),sum)
apply(new_array,c(2,3),sum)
apply(new_array,3,sum)
```

Есть целое семейство функций, которые специализируются на векторизованных вычислениях:
`apply(), by(), lapply(), sapply(), tapply()` и др.

Класс данных: списки (list)

Списки – объекты данных в R, которые содержат элементы разных типов – числа, строки, векторы, матрицы, другие списки.

Создание списков:
`list(…)` или `vector("list", длина)`

Пример
```
# Создадим список, содержащий строки, числа, векторы и логические значения:
list_data <- list("Red", "Green", c(21L,32L,11L), TRUE, 51.23, 119.1)
```

Именование элементов списка

Пример
```
# создадим список, содержащий вектор, матрицу и список:
list_data <- list(c("Jan","Feb","Mar"), matrix(c(3,9,5,1,-2,8), nrow = 2), list("green",12.3))
# Дадим имена элементам списка:
names(list_data) <- c("1st Quarter", "A_Matrix", "A Inner list")
print(list_data)
#Доступ к имени третьего элемента:
names(list_data)[3]
```

Доступ к элементам списка
```
# Доступ к первому элементу списка
list_data[1]   # полученный объект тоже будет списком
list_data[[1]] # полученный объект будет того типа, какого он был бы до объединения в список
list_data$"1st Quarter" # или так: list_data["1st Quarter"]
```

Класс данных: списки (list) - продолжение

Операции над элементами списка

Пример

```
#добавим элемент в конец списка:
list_data[4] <- "New element"
print(list_data[4])

#удалим последний элемент:
list_data[4] <- NULL

#Обновим 3й элемент
list_data[3] <- "updated element"
print(list_data[3])
```

Объединение списков

Пример

```
list1 <- list(1,2,3)
list2 <- list("Sun","Mon","Tue")
merged.list <- c(list1,list2)

#объединение 2 списков с помощью append():
append(list1,list2)
# аргумент after указывает, после какого элемента
#списка сделать вставку:
append(list1,list2,after=2)
```

Преобразование списка в вектор

```
unlist(x, recursive = TRUE, use.names = TRUE)
```

Пример

```
list1 <- list(1:5); list1
list2 <- list(10:14); list2
v1 <- unlist(list1); v1
v2 <- unlist(list2); v2
result <- v1+v2
print(result)
```

Пример (на использование параметров `recursive` и `use.names`)

```
list_data[[3]] <- list(T,F,T)
unlist(list_data)
unlist(list_data, recursive=T, use.names=F)
unlist(list_data, recursive=F, use.names=T)

# Обратно: relist(flesh, skeleton)
```

Класс данных: факторы (factor)

Факторы – объекты данных в R, которые используются для категоризации данных и хранения их в виде уровней (levels)

Создание факторов
Факторы создаются с помощью функции factor() , принимая вектор в качестве входных данных.
Пример

```
data <-c("East","West","East","North","North","East","West","West","West","East","North")
factor_data <- factor(data)
```

Изменение порядка уровней
Пример

```
new_order_data <- factor(factor_data, levels = c("East","West","North"))
```

Еще пример

```
m <- c("L", "S", "XL", "XXL", "S", "M", "L")
m.f <- factor(m)
m.f
m.o <- ordered(m.f, levels=c("S", "M", "L", "XL", "XXL"))
```

Класс данных: таблица данных (data frame)

Таблицы (фрейм) данных - это гибридный тип представления данных, **одномерный список из векторов одинаковой длины**.
Таблица данных — это прямоугольная двумерная таблица с данными или структура, подобная двумерному массиву, в которой каждый столбец содержит значения одной переменной, а каждая строка содержит один набор значений из каждого столбца.

Характеристические особенности таблиц данных:
- Все данные внутри отдельного столбца должны быть одного типа.
- Имена столбцов не должны быть пустыми.
- Имена строк должны быть уникальными.
- Данные, хранящиеся в таблице данных, могут иметь числовой, факторный или символьный тип.
- Каждый столбец должен содержать одинаковое количество элементов данных.

Основной способ создания таблиц данных – с помощью функции **`data.frame()`**:

```
data.frame(..., row.names = NULL, check.rows = FALSE, check.names = TRUE,
       fix.empty.names = TRUE, stringsAsFactors = FALSE)
```

Аргументы функции:

`...` – список данных (может содержать произвольное число элементов)
row.names – определяет имена строк (м.б. одно целое число, или вектор целых, или вектор символов)
check.names – проверяет синтактическую валидность имён переменных и отсутствие повторов
fix.empty.names – обозначает, что «безымянные» столбцы по умолчанию получают автоматически имя
stringsAsFactors – определяет, должны ли векторы символов конвертироваться в факторы

Таблицы данных: пример 1

```
> # Вектор веса
> w <- c(69, 68, 93, 87, 59, 82, 72)
> names(w) <- c("Коля", "Женя", "Петя", "Саша", "Катя", "Вася", "Жора")
> w
Коля Женя Петя Саша Катя Вася Жора
  69   68   93   87   59   82   72
> # Вектор роста
> x <- c(174, 162, 188, 192, 165, 168, 172)
> # Вектор размера одежды
> m <- c("L", "S", "XL", "XXL", "S", "M", "L")
> m.f <- factor(m)
> m.o <- ordered(m.f, levels=c("S", "M", "L", "XL", "XXL"))
> m.o
[1] L   S   XL  XXL S   M   L
Levels: S < M < L < XL < XXL
> # Вектор пола
> sex <- c("male", "female", "male", "male", "female", "male", "male")
> sex.f <- factor(sex)
> # Создание таблицы данных df по четырем переменным (колонкам) w, x, m.o, sex.f:
> df <- data.frame(weight = w, height = x, size = m.o, sex = sex.f)
```

Таблицы данных: пример 1 (продолжение)

```
> df
       weight height size  sex
Коля     69    174    L   male
Женя     68    162    S female
Петя     93    188   XL   male
Саша     87    192  XXL   male
Катя     59    165    S female
Вася     82    168    M   male
Жора     72    172    L   male
> str(df)
'data.frame':  7 obs. of  4 variables:
 $ weight: num  69 68 93 87 59 82 72
 $ height: num  174 162 188 192 165 168 172
 $ size  : Ord.factor w/ 5 levels "S"<"M"<"L"<"XL"<..: 3 1 4 5 1 2 3
 $ sex   : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 2
> summary(df)
 weight          height       size       sex
 Min.   :59.00   Min.   :162.0   S  :2   female:2
 1st Qu.:68.50   1st Qu.:166.5   M  :1   male  :5
 Median :72.00   Median :172.0   L  :2
 Mean   :75.71   Mean   :174.4   XL :1
 3rd Qu.:84.50   3rd Qu.:181.0   XXL:1
 Max.   :93.00   Max.   :192.0
```

Доступ к именам строк и столбцов таблиц данных: ф-ии `rownames()`, `colnames()`, `dimnames()`.

Таблицы данных: доступ к элементам (индексация)

Доступ к элементам таблицы данных осуществляется 2 способами:

1. Как к элементам матрицы
2. Как к элементам списка

Вернемся к примеру 1:

```
# Выведем столбец данных weight первым способом
> df[,1]
[1] 69 68 93 87 59 82 72
> df[,"weight"]
[1] 69 68 93 87 59 82 72
# Выведем столбец данных weight вторым способом
> df[[1]]
[1] 69 68 93 87 59 82 72
> df$weight
[1] 69 68 93 87 59 82 72
```

Выбор нескольких конкретных колонок.

```
# хотим вывести все столбцы, кроме weight
> df[,2:4]
# или так:
> df[,-1] # отрицательная индексация
> df[c("weight","size")] # только столбцы weight и size
```

Чтобы избежать «схлопывания» размерности:

```
> df[,1, drop=FALSE] #таблица данных, состоящая из одного столбца
```

Таблицы данных: доступ к элементам (продолжение)

Доступ к элементам таблицы данных с помощью логического условия (логическая индексация):

```
> df[df$sex == "female",]
       weight height size    sex
Женя     68    162    S       female
Катя     59    165    S       female

> df$sex=="female"
[1] FALSE TRUE FALSE FALSE TRUE FALSE FALSE

# Аналогично, в «матричном» стиле:
> df[,"sex"] == "female"
[1] FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE

> df[df[,"sex"] == "female",]
       weight height size    sex
Женя     68    162    S       female
Катя     59    165    S       female

# Фильтрация по условию с помощью функции subset():
> subset(df, sex=='female')
```

Также можно редактировать таблицы данных во встроенном редакторе (функция `fix()`)

Таблицы данных: сортировка

Применение функции **order()**.

```
# Сортировка всех данных по весу
> df[order(df$weight),]
     weight height size    sex
Катя     59    165    S female
Женя     68    162    S female
Коля     69    174    L   male
Жора     72    172    L   male
Вася     82    168    M   male
Саша     87    192  XXL   male
Петя     93    188   XL   male
# Сортировка данных сначала по полу, а потом по росту
> df[order(df$sex, df$height), ]
     weight height size    sex
Женя     68    162    S female
Катя     59    165    S female
Вася     82    168    M   male
Жора     72    172    L   male
Коля     69    174    L   male
Петя     93    188   XL   male
Саша     87    192  XXL   male
> df[ ,order(colnames(df))]
> df[order(rownames(df)), ]
```

Таблицы данных: расширение

Таблицу данных можно расширить, добавив столбцы и/или строки.

Добавление столбца:

```
> birthday = as.Date(c("2002-01-01", "2003-09-23", "2004-11-15", "2004-05-11", "2005-03-27",
"1999-04-05", "1985-02-10"))
> df$birthday <- birthday
> df
       weight height size    sex     birthday
Коля       69    174    L       male    2002-01-01
Женя       68    162    S       female  2003-09-23
Петя       93    188   XL       male    2004-11-15
Саша       87    192  XXL       male    2004-05-11
Катя       59    165    S       female  2005-03-27
Вася       82    168    M       male    1999-04-05
Жора       72    172    L       male    1985-02-10
```

Или с помощью функции cbind():

```
> cbind(df, as.data.frame(birthday))
```

Таблицы данных: расширение (продолжение)

Добавление строки:

```
df_new <- data.frame(weight=c(75,85), height=c(175,185), size=c("L","XL"),
       sex=c("female","male"), birthday=as.Date(c("2006-01-01", "2000-10-10")),
       stringsAsFactors = F, row.names = c("Маша", "Ваня"))
> df_new
     weight height size    sex   birthday
Маша     75    175    L female 2006-01-01
Ваня     85    185   XL   male 2000-10-10
> df_final <- rbind(df,df_new)
> df_final
     weight height size    sex   birthday
Коля     69    174    L   male 2002-01-01
Женя     68    162    S female 2003-09-23
Петя     93    188   XL   male 2004-11-15
Саша     87    192  XXL   male 2004-05-11
Катя     59    165    S female 2005-03-27
Вася     82    168    M   male 1999-04-05
Жора     72    172    L   male 1985-02-10
Маша     75    175    L female 2006-01-01
Ваня     85    185   XL   male 2000-10-10
```

Таблицы данных: комбинирование по ключу

Комбинирование по ключу с помощью функции merge():

```
> df_tshirts <- data.frame(size = c("S","L","M"), color = c("red", "green", "blue"))
> df_tshirts
  size color
1    S   red
2    L green
3    M  blue

> merge(df, df_tshirts, by="size")
 size weight height    sex   birthday color
1    L     69    174   male 2002-01-01 green
2    L     72    172   male 1985-02-10 green
3    M     82    168   male 1999-04-05  blue
4    S     68    162 female 2003-09-23   red
5    S     59    165 female 2005-03-27   red
```