Course: R Language in Computational Probability and Statistics

Lecture 3: Factors. Arrays and matrices.

Lecturer: Nataliia Kruhlova

Будь-який об'єкт даних в **R** має дві головні властивості (**attributes)**: тип (**mode**) і довжину (**length**). Кожний об'єкт може мати так само інші додаткові властивості: клас (**class)** і розмірність (**dimension** або **dim**). Об'єкти і їх можливі властивості представлені в наступній таблиці.

Таблиця 1

Об'єкт	**Тип (mode)**	**Застосування різних типів в одному обєкті**	**Класс (class)**
1.Вектор (*vector)*	numeric, character, complex, logical	ні	numeric, character, complex, logical
2. Фактор(factor)	numeric, character	ні	factor
3.Рангова змінна (factor ordered)	numeric, character	ні	factor ordered
4. Масив(array)	numeric, character, complex, logical	ні	NULL
5.Матриця(matrix)	numeric, character, complex, logical	ні	matrix
6.Фрейм(data frame)	List	так	data. frame
7. Список(list)	List	так	Залежить від способу формування

Командами:
attributes() - виводяться властивості об'єкту**,**
mode() – тип об'єкту**,**
class() – визначається клас об'єкту.

Матриці

Матриці в **R** обов'язково складаються з елементів одного типу (наприклад, тільки з чисел або тільки з логічних значень). Є багато різних способів створити матрицю, наприклад, декілька рядків об'єднуються у

матрицю за допомогою функції **rbind**, а за допомогою функції **cbind** «склеюються» вектори-стовпці.

Приклади.

```
> x1<-c(0,1,1,1)
> x2<-c(F,F,T,F)
> m1<-rbind(x1,x2)
> m1
   [,1] [,2] [,3] [,4]
x1    0    1    1    1
x2    0    0    1    0
> x3<-c(-1,2,3,6)
> m2<-cbind(x1,x3)
> m2
     x1 x3
[1,]  0 -1
[2,]  1  2
[3,]  1  3
[4,]  1  6
```

У створених матрицях імена векторів, що об’єднувались, перетворились на імена відповідних рядків чи стовпчиків матриці.

Щоб звернутись до відповідного елемента матриці, потрібно у квадратних дужках вказати першим індексом номер рядка елемента і, через кому, другим індексом – номер стовпчика. Наприклад, **m2[4,1]** - це елемент четвертого рядка і першого стовпчика матриці **m2.** Правила індексації дуже гнучкі.

Приклад.

```
> m2[3,]
x1 x3
 1  3
```

У цьому прикладі ми виводимо всі елементи третього рядка матриці m2.

Весь другий стовпчик цієї матриці можна вивести наступною командою.

Приклад.

```
> m2[,2]
[1] -1  2  3  6
```

Можна звертатись до потрібних рядків чи стовпчиків за їх іменами та вив одити частину елементів рядка або стовпчика.

Приклад.

```
> m1["x1",]
[1] 0 1 1 1
> m2[c(1,3),"x3"]
[1] -1  3
```

Якщо ми з матриці вибираємо рядок чи стовпчик, то результатом буде вектор-рядок. Іноді, для коректної роботи деяких операцій, нам потрібно одержати в результаті матрицю-рядок чи матрицю-стовпчик. Для цього є опція **drop=F**.

Приклади.

```
> m2[c(1,3),"x3", drop=F]
     x3
[1,] -1
[2,]  3
> m2[1,,drop=F]
     x1 x3
[1,]  0 -1
```

Для виділення частини об'єкта можна використовувати не індексацію, а функцію **subset().**

Параметри**:**

x — об'єкт, з якого вибирається певна частина,

subset — умова на рядки, що виділяються,

select — умова або номери стовпчиків, які вибираються**.**

Приклади.

```
> m3<-cbind(x1,x2,x3)
> m3
     x1 x2 x3
[1,]  0  0 -1
[2,]  1  0  2
[3,]  1  1  3
[4,]  1  0  6
> subset(m3,select=3,c(T,F,F,T))
     x3
[1,] -1
[2,]  6
```

Також матрицю можна створити за допомогою функції **matrix,** застосувавши її до вектора. В якості першого параметру функції виступає вектор, з якого формується матриця, кількість рядків і стовпчиків задають за допомогою параметрів **nrow** і **ncol**.

Приклади.

```
> y<-c(-2:9)
> matrix(y,ncol=3)
     [,1] [,2] [,3]
[1,]   -2    2    6
[2,]   -1    3    7
[3,]    0    4    8
[4,]    1    5    9
> matrix(y,nrow=3)
     [,1] [,2] [,3] [,4]
[1,]   -2    1    4    7
[2,]   -1    2    5    8
[3,]    0    3    6    9
> matrix(y,nrow=3,ncol=3)
     [,1] [,2] [,3]
[1,]   -2    1    4
[2,]   -1    2    5
[3,]    0    3    6
```

Порядок заповнення матриці можна вказати параметром **byrow=T**. Тоді заповнення відбувається по рядкам.

Приклад.

```
> matrix(y,nrow=3,ncol=3,byrow=T)
     [,1] [,2] [,3]
[1,]   -2   -1    0
[2,]    1    2    3
[3,]    4    5    6
```

Щоб задати імена стовпчикам і рядкам матриці, використовують функцію **dimnames.**

Приклад.

```
> M<-matrix(y,ncol=3)
> dimnames(M)<-list(LETTERS[1:4],letters[1:3])
> M
  a b c
A -2 2 6
B -1 3 7
C  0 4 8
D  1 5 9
```

В цьому прикладі за допомогою функції **list** було створено список з двох векторів. Функція **letters** виводить в алфавітному порядку маленькі літери латинського алфавіту, а **LETTERS -** відповідні великі літери.

Щоб задати або змінити імена тільки рядків або стовпчиків використовуються функції **rownames** і **colnames** відповідно**.** Ці самі функції використовуються, щоб вивести імена рядків чи стовпчиків деякої матриці.

Приклади.

```
> colnames(M)
[1] "a" "b" "c"
> colnames(M)<-c(1:3)
> M
  1 2 3
A -2 2 6
B -1 3 7
C  0 4 8
D  1 5 9
```

В останньому прикладі ми змінили назви рядочків.

Щоб із вектору створити діагональну матрицю, на головній діагоналі якої розміщуються елементи вектору, використовують функцію **diag.**

Приклад.

```
> x<-c(F,T,T,T)
> diag(x)
      [,1]  [,2]  [,3]  [,4]
[1,] FALSE FALSE FALSE FALSE
[2,] FALSE  TRUE FALSE FALSE
[3,] FALSE FALSE  TRUE FALSE
[4,] FALSE FALSE FALSE  TRUE
```

Якщо в якості параметру цієї функції подати матрицю, то виводиться вектор, що містить елементи головної діагоналі матриці.

Приклад.

```
> M<-matrix(seq(2,18,by=2),nrow=3)
> diag(M)
[1]  2 10 18
```

Щоб змінити головну діагональ матриці, потрібно функцію **diag** записати зліва від символу привласнення.

Приклад.

```
> diag(M)<-mean(M)
> M
     [,1] [,2] [,3]
[1,]   10    8   14
[2,]    4   10   16
[3,]    6   12   10
```

Арифметичні операції та логічні дії виконуються для матриць поелементно.

%*% - оператор матричного множення.

t() – функція для знаходження транспонованої матриці.

solve(A,b) розв'язує систему лінійних алгебраїчних рівнянь виду **Ax = b.**

solve(A) обчислює обернену матрицю A^{-1}.

det() рахує визначник матриці.

eigen() обчислює власні числа і власні вектори матриці. Вертає список, що містить і власні числа, і власні вектори, що їм відповідають.

chol() виконує розклад Холецького симетричної додано визначеної матриці.

ncol() визначає кількість стовпчиків матриці.

nrow() виводить кількість рядків матриці.

Приклади.

```
> M<-matrix(seq(2,18,by=2),nrow=3)
> D<-matrix(1,nrow=3,ncol=3)
> M+D
     [,1] [,2] [,3]
[1,]    3    9   15
[2,]    5   11   17
[3,]    7   13    1
> M*D
     [,1] [,2] [,3]
[1,]    2    8   14
[2,]    4   10   16
[3,]    6   12   18
> M%*%D
     [,1] [,2] [,3]
[1,]   24   24   24
[2,]   30   30   30
[3,]   36   36   36
```

Матрицю також можна ввести з клавіатури шляхом резервування пам'яті і вказання розмірності матриці. У наступному прикладі ми створюємо нульовий вектор з 18 елементів, а потім задаємо розмірність майбутньої матриці, після чого з клавіатури вводимо елементи матриці.

Приклад.

```
> vec <- numeric(18)

> dim(vec)<-c(3,6)

> A <- edit(vec)
```

Масиви.

Узагальненням матриць є масиви. Вони можуть мати довільну кількість розмірностей. Створити їх можна за допомогою функції **array,** вказавши вектор розмірностей.

Приклад.

```
> A<- array (1:18 , c(3, 3, 2))
> A
, , 1

     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

, , 2

     [,1] [,2] [,3]
[1,]   10   13   16
[2,]   11   14   17
[3,]   12   15   18
```

Або за допомогою функції **dim**, вказавши вектор розмірностей.

Приклад.

```
> x<-c(1:18)
> dim(x)<-c(3,3,2)
> x
, , 1
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
, , 2
     [,1] [,2] [,3]
[1,]   10   13   16
[2,]   11   14   17
[3,]   12   15   18
 > x[1,2,2]

[1] 13
```

Операції виконуються аналогічно, як для матриць.

Фактори

Ще один тип векторних даних, що заслуговує уваги, - це фактори. Елементи факторної змінної можуть приймати значення лише з певного фіксованого набору. Цей тип даних часто зустрічається в статистичних дослідженнях, коли об'єкти розбиваються на кілька груп за певною ознакою, наприклад, люди за національністю, статтю, відношенням до військової служби, юридичні особи за формою власності, слова за частинами мови (іменник, прикметник, дієслово і т.д.). Значення, які може приймати фактор, називають рівнями. Різні рівні, зазвичай, позначають їх назвами, наприклад, національність - українець, німець, американець.

Наприклад, створимо факторну змінну, що містить дані про стать людей, які відвідали лікаря Пілюлькіна в його перший робочий день.

Приклад.

```
> x<-c("чоловік", "жінка", "чоловік", "чоловік", "жінка", "жінка", "чоловік",
"жінка")
> x
[1] "чоловік" "жінка"   "чоловік"   "чоловік" "жінка"   "жінка" "чоловік"
[8] "жінка"
> str(x)
 chr [1:8] "чоловік" "жінка" "чоловік" "чоловік" "жінка" "жінка" "чоловік"
"жінка"...
```

В цьому прикладі вектор **x** має символьний тип. Це ми перевірили за допомогою функції **str().**

Щоб перетворити вектор на фактор, потрібно використати функцію **factor().**

Приклад.

```
> xf<-factor(x)
> xf
 [1:8] чоловік жінка   чоловік   чоловік жінка   жінка чоловік жінка
Levels: жінка чоловік
```

Хоча на екрані рівні фактора показуються їхніми назвами, в комп'ютерному представленні вони кодуються цілими числами. Список різних рівнів виводиться у рядку **"Levels"** у порядку зростання кодів. Якщо вам потрібен лише цей список у вигляді символьного рядка, ви можете скористатися функцією **levels().**

Приклад.

```
> levels(xf)
[1] «жінка»   «чоловік»
```

Застосувавши функцію **unclass,** можна вивести на екран ці коди.

Приклад.

```
> unclass(xf)
[1] 2 1 1 2 1 2 2 1
attr(,"levels")
[1] "жінка"   "чоловік"
```

Використання факторів замість символьних рядків економить пам'ять комп'ютера, особливо коли довжина вектора велика, а кількість рівнів помірна. Крім того, вказування переліку рівнів дозволяє виявити непотрібні назви, які можуть виникнути через помилки. У статистиці існує багато алгоритмів обробки даних, які спеціалізуються на категоріальних даних (наприклад, у дисперсійному та регресійному аналізі, а також у аналізі таблиць спряженості). Це одна з причин, чому фактори виділяються у власний тип. Варто зазначити, що у факторній змінній можуть бути відсутні всі можливі рівні, але інформація про їх можливу появу зберігається в атрибуті **levels.**

Наприклад, нам потрібно створити фактор оцінок студента під час останньої сесії, але ми хочемо вказати, які саме оцінки були присутні у відомості (часто потрібно зводити дані у спільну таблицю для аналізу).

Приклад.

```
> x<-factor(c("D", "B", "C", "C", "A"),levels=c("A","B","C","D","E","F","Fx"))
> x
[1] D B C C A
Levels: A B C D E F Fx
```

Уявімо, що Олег Іванович сказав в деканаті надати оцінки певного студента тільки з математичних дисциплін. Екзамени по цим предметам здавались другими і четвертими по черзі.

Приклад.

```
> mat<-x[c(2,4)]
> mat
[1] B C
Levels: A B C D E F Fx
```

Якщо, виділяючи частину факторної змінної, ми хочемо видалити рівні, що не зустрічаються у вибраній частині, то це можна зробити, підключивши параметр **drop.**

Приклад.

```
> mat2<-x[c(2,4),drop=T]
> mat2
[1] B C
Levels: B C
```

Виконуючи статистичні дослідження, іноді виникає потреба в перетворенні кількісної змінної на категоріальну. Наприклад, нам потрібно не саме значення змінної, а номер інтервалу, в який вона потрапляє. Чи, наприклад, вибірка містить дані про зріст людей, а нам потрібно їх поділити на групи: низький, середній і високий зріст.

Це можна зробити за допомогою функції **cut.**

Приклад. Нехай ми знаємо зріст (у см) 20 чоловіків. Нам потрібно їх поділити на групи по зросту: низький (до 165), середній (165-180), високий (вище 180).

```
> x<-c(156,181,201,166,150,175,174,169,189,196,188,159,174,186,167,205,179,18
3,178,199 )
> xf<-cut(x,breaks=c(0,165,180,Inf),labels=c('low','mid','high'))
> xf
 [1] low  high high mid  low  mid  mid  mid  high high high low  mid  high mid
[16] high mid  high mid  high
Levels: low mid high
```

В цьому прикладі ми спочатку створили числовий вектор **x,** а потім поділили вибірку на три групи в залежності від значень **x.**

Параметр **breaks** визначає межі інтервалів, на які ми ділимо вибірку: до першої категорії відносяться об’єкти, для яких $\mathbf{x} \in (-\infty, \mathbf{165}]$**,** до другої —$\mathbf{x} \in (\mathbf{165}, \mathbf{180}]$**,** до третьої —$\mathbf{x} \in (\mathbf{180}, \infty]$**.** В параметрі **labels** ми перераховуємо рівні факторної змінної.

Можна задати порядок для рівнів фактора, наприклад: low<mid<high. Для деяких факторів не існує такого впорядкування, наприклад для частин мови. Але можна самостійно створити впорядкований фактор **(ordered)**. Іноді таке перетворення потрібно зробити, оскільки деякі функції **R** мають певні відмінності в роботі з впорядкованими і невпорядкованими факторами.

Приклад.

```
> xfo<-ordered(xf)
> xfo
 [1] low  high high mid  low  mid  mid  mid  high high high low  mid  high mid
[16] high mid  high mid  high
Levels: low < mid < high
```