



ARRAYS & MATRICES | МАСИВИ І МАТРИЦІ

Table of Contents | Зміст

This lesson will cover some basics concepts about arrays and matrices in R:

Цей тренінг стосується базових концепцій щодо масивів та матриць в R:

- [What is an Array?](#) | [Що таке масив?](#)
- [Creating an array from a vector](#) | [Створення масиву із вектору.](#)
- [What is a parameter?](#) | [Що таке параметр?](#)
- [Array indexing](#) | [Індексування масиву.](#)
- [What is a Matrix?](#) | [Що таке матриця?](#)
- [Concatenation Function](#) | [Функція зчеплення](#)

About the Dataset | Про базу даних

Imagine you got many movie recommendations from your friends and compiled all of the recommendations in a table, with specific info about each movie.

The table has one row for each movie and several columns

Ви отримали багато рекомендацій про фільми від своїх друзів і внесли усі рекомендації до таблиці зі специфічною інформацією про кожен фільм.

У цій таблиці одному рядкові відповідає один фільм а також кілька колонок з його параметрами.

- **name** - The name of the movie | Назва фільму
- **year** - The year the movie was released | Рік релізу фільму
- **length_min** - The lenght of the movie in minutes | Тривалість фільму (хвилини)
- **genre** - The genre of the movie | Жанр фільму
- **average_rating** - Average rating on Imdb | Середній рейтинг на сайті IMDB
- **cost_millions** - The movie's production cost in millions | Вартість фільму (млн доларів США)
- **sequences** - The amount of sequences | Кількість частин

- **foreign** - Indicative of whether the movie is foreign (1) or domestic (0) | Фільм є зарубіжним (1) чи вітчизняним (0)?
- **age_restriction** - The age restriction for the movie | Вікові обмеження для перегляду фільму

You can see part of the dataset below

Ви можете побачити частину бази даних нижче

name	year	length_min	genre	average_rating	cost_millions	foreign	age_restriction
Toy Story	1995	81	Animation	8.3	30	0	0
Akira	1998	125	Animation	8.1	10.4	1	14
The Breakfast Club	1985	97	Drama	7.9	1	0	14
The Artist	2011	100	Romance	8	15	1	12
Modern Times	1936	87	Comedy	8.6	1.5	0	10
Fight Club	1999	139	Drama	8.9	63	0	18
City of God	2002	130	Crime	8.7	3.3	1	18
The Untouchables	1987	119	Drama	7.9	25	0	14
Star Wars	1977	121	Action	8.7	11	0	10
American Beauty	1999	122	Drama	8.4	15	0	14
Room	2015	118	Drama	8.3	13	1	14
Dr. Strangelove	1964	94	Comedy	8.5	1.8	1	10
The Ring	1998	95	Horror	7.3	1.2	1	18
Monty Python and the Holy Grail	1975	91	Comedy	8.3	0.4	1	18
High School Musical	2006	98	Comedy	5.2	4.2	0	0
Shaun of the Dead	2004	99	Horror	8	6.1	1	18
Taxi Driver	1976	113	Crime	8.3	1.3	1	14
The Shawshank Redemption	1994	142	Crime	9.3	25	0	16
Interstellar	2014	169	Adventure	8.6	165	0	10
Casino	1995	178	Biography	8.2	50	0	18
The Goodfellas	1990	145	Biography	8.7	25	0	14
Blue is the Warmest Colour	2013	179	Romance	7.8	4.5	1	18
Black Swan	2010	108	Thriller	8	13	0	16
Back to the Future	1985	116	Sci-fi	8.5	19	0	0
The Wave	2008	107	Thriller	7.6	5.5	1	16
Whiplash	2014	106	Drama	8.5	3.3	1	12
The Grand Hotel Budapest	2014	100	Crime	8.1	25.5	0	14
Jumanji	1995	104	Fantasy	6.9	65	0	12
The Eternal Sunshine of the Spotless Mind	2004	108	Drama	8.3	20	0	14
Chicago	2002	113	Comedy	7.2	45	0	12

What is an Array? | Що таке масив?

An array is a structure that holds values grouped together, like a 2 x 2 table of 2 rows and 2 columns. Arrays can also be **multidimensional**, such as a 2 x 2 x 2 array.

Масив - це структура, яка містить певні згруповані між собою значення, наприклад таблиця 2 x 2 з 2 рядками та 2 стовпцями. Масиви також можуть бути **багатовимірними**, наприклад масивом типу 2 x 2 x 2.

What is the difference between an array and a vector? | Яка різниця між масивом та вектором?

Vectors are always one dimensional like a single row of data. On the other hand, an array can be multidimensional (stored as rows and columns). The "dimension" indicates how many rows of data there are.

Вектори завжди одномірні, наче єдиний рядок даних. Тоді як масив може бути багатовимірним (зберігатись у вигляді рядків та колонок). "Вимір" вказує, скільки існує рядків даних в масиві.

Let's create a 4 x 3 array (4 rows, 3 columns) | Давайте створимо масив 4 x 3 (4 рядки, 3 колонки)

The example below is a vector of 9 movie names, hence the data type is the same for all the elements.

Нижче наведений приклад - це вектор з 9 назв фільмів, тому тип даних у ньому однаковий для всіх елементів.

In []:

```
#lets first create a vector of nine movies
movie_vector <- c("Akira", "Toy Story", "Room", "The Wave", "Whiplash",
                  "Star Wars", "The Ring", "The Artist", "Jumanji")
movie_vector
```

Creating an array from a vector | Створення масиву із вектору

To create an array, we can use the **array()** function.

Щоб створити масив, ми можемо скористатися функцією **array()**

In []:

```
movie_array <- array(movie_vector, dim = c(4,3))
movie_array
```

Note that **arrays are created column-wise**. Did you also notice that there were only 9 movie names, but the array was 4 x 3? The original **vector doesn't have enough elements** to fill the entire array (that should have 3 x 4 = 12 elements). So R simply fills rest of the empty values by going back to the beginning of the vector and starting again ("Akira", "Toy story", "Room" in this case).

Зверніть увагу, що **масиви заповнюються по колонках**. Чи ви помітили, що у нас було лише 9 назв фільмів, тоді як вимір масиву був 4 x 3? Первинний **вектор не має достатньо елементів** для заповнення всього масиву (який повинен мати 3 x 4 = 12 елементів). Таким чином, R просто заповнює залишки порожніх значень, повертаючись до початку вектора і починаючи знов ("Акіра", "Історія іграшок", "Кімната" в даному випадку).

We also needed to provide `c(4,3)` as a second *argument* to specify the number of rows (4) and columns (3) that we wanted.

Ми також повинні вказати `c(4,3)` у якості другого *аргументу функції*, щоб зазначити ту кількість рядків (4) та колонок (3), яку ми хочемо створити.

In []:

```
?array
```

[Tip] What is an "argument"? How are "arguments" different from "parameters"?

Arguments and parameters are terms you will hear constantly when talking about **functions**.

- The **parameters** are the input variables used in a function, like **dim** in the function **array()**.
- The **arguments** refer to the *values* for those parameters that a function takes as inputs, like **c(4,3)**

We actually don't need to write out the name of the parameter (dim) each time, as in:

```
array(movie_vector, c(4,3))
```

As long as we write the arguments out in the correct order, R can interpret the code.

Arguments in a function may sometimes need to be of a **specific type**. For more information on each function, you can open up the help file by running the function name with a `?` beforehand, as in:

```
?array
```

[Лайфхак] Що таке "аргумент"? Як "аргументи" відрізняються від "параметрів"?

Аргументи та параметри - це умови, які ви будете чути постійно, говорячи про **функції**. -

Параметри - це *задані змінні*, що використовуються в функції, наприклад **dim** у функції створення масиву **array()**. - **Аргументи** - це *значення* для тих параметрів, які функція вважає заданими, наприклад **c(4,3)**

Нам насправді не потрібно вписувати назву параметра (dim) кожен раз, як у цьому прикладі:

```
array(movie_vector, c(4,3))
```

Якщо ми тільки запишемо аргументи у правильному порядку, R зможе автоматично інтерпретувати наш код.

Аргументи в функції іноді мають бути **специфічного типу**. Для отримання додаткової інформації про кожну функцію, ви можете відкрити файл довідки, запустивши назву функції зі знаком питання (?) попереду, наприклад так:

```
?array
```

Array Indexing | Індексуння масиву

Let's look at our array again:

Поглянемо на наш масив знову:

In []:

```
movie_array
```

To access an element of an array, we should pass in **[row, column]** as the row and column number of that element.

For example, here we retrieve **Whiplash** from row 1 and column 2:

Щоб отримати доступ до елемента масиву, ми повинні зазначити у квадратних дужках **[row, column]** номер рядка та колонки, в яких знаходиться цей елемент. Наприклад, так ми витягуємо з масиву назву **Whiplash** із рядка 1 та колонки 2:

In []:

```
movie_array[1,2] #[row, column]
```

To display all the elements of the first row, we should put 1 in the row and nothing in the column part. Be sure to keep in the comma after the 1.

Щоб відобразити всі елементи першого рядка, ми повинні зазначити "1" на місці рядка і нічого в частині колонки. Не забувайте ставити кому після "1"!

In []:

```
movie_array[1,]
```

Likewise, you can get the elements by column as below.

Аналогічним чином, ви можете отримати усі елементи колонки, як показано нижче.

In []:

```
movie_array[:,2]
```

To get the dimension of the array, **dim()** should be used.

Щоб отримати розмірність масиву, слід використовувати **dim()**.

In []:

```
dim(movie_array)
```

We can also do math on arrays. Let's create an array of the lengths of each of the nine movies used earlier.

Ми також можемо робити обчислення з масивами. Давайте створимо масив тривалості кожного з тих дев'яти фільмів, які ми брали раніше.

In []:

```
length_vector <- c(125, 81, 118, 81, 106, 121, 95, 100, 104)
length_array <- array(length_vector, dim = c(3,3))
length_array
```

Let's add 5 to the array, to account for a 5-min bathroom break:

Додамо 5 до масиву, щоб врахувати 5-хвилинну перерву:

In []:

```
length_array + 5
```

Tip: Performing operations on objects, like adding 5 to an array, does not change the object. **To change the object, we would need to assign the new result to itself.**

Лайфхак: Виконання операцій над об'єктами, наприклад додавання числа 5 до масиву, не змінює самого об'єкту. **Щоб змінити об'єкт, нам потрібно буде присвоїти йому його стару змінну.**

In []:

```
# масив лишився незмінним
length_array
```

In []:

```
# перейменуємо новий масив, зробившу заміну старого масиву
length_array <- length_array + 5
length_array
```

In []:

```
# вернемось до попередніх значень масиву
length_array <- length_array - 5
length_array
```

Using Logical Conditions to Subset Arrays

Використання логічних умов для розподілу масивів

Which movies can I finish watching in two hours? Using a logical condition, we can check which movies are less than 2 hours long.

Які фільми я можу переглянути за дві години? Використовуючи логічну умову, ми можемо перевірити, які фільми тривають менше двох годин.

In []:

```
mask_array <- length_array > 120  
mask_array
```

Using this array of TRUEs and FALSEs, we can subset the array of movie names:

Використовуючи цей масив з елементами TRUE та FALSE, ми можемо розподілити сві старий масив з назвами фільмів:

In []:

```
x_vector <- c("Akira", "Toy Story", "Room", "The Wave", "Whiplash",  
             "Star Wars", "The Ring", "The Artist", "Jumanji")  
x_array <- array(x_vector, dim = c(3,3))  
x_array  
  
x_array[mask_array]
```

What is a Matrix? | Що таке матриця?

Matrices are a subtype of arrays. A matrix **must** have 2 dimensions, whereas arrays are more flexible and can have, 1, 2 or more dimensions.

To create a matrix out of a vector, you can use **matrix()**, which takes in an argument for the vector, an argument for the number of rows and another for the number of columns.

Матриці є підвидом масивів. Матриця **повинна** мати 2 розмірності, тоді як масиви - більш гнучкі і можуть бути 1-, 2- або n-мірними.

Щоб створити матрицю з вектора, ви можете використовувати функцію **matrix ()**, яка конкретизується аргументами для вектора, аргументом для кількості рядків та іншим для кількості колонок.

In []:

```
movie_matrix <- matrix(movie_vector, nrow = 3, ncol = 3)
```

In []:

```
movie_matrix
```

Accessing elements of a matrix | Доступ до елементів матриці

As with arrays, you can use **[row, column]** to access elements of a matrix. To retrieve "Akira", you should use [1,1] as it lies in the first row and first column.

Як і в масивах, ви можете використовувати **[рядок, колонка]** для доступу до елементів матриці. Щоб дістати назву "Акіра", ви повинні використовувати команду [1,1], оскільки фільм знаходиться в першому рядку та в першому стовпчику.

In []:

```
movie_matrix[1,1]
```

To get data from a certain range, the following code can help. This takes the elements from rows 2 to 3, and from columns 1 to 2.

Щоб отримати дані з певного діапазону, може допомогти наступний код. Цей витягує елементи з рядків від 2 до 3, і з колонок від 1 до 2.

In []:

```
movie_matrix[2:3, 1:2]
```

Concatenation function | Функція "зчеплення"

A concatenation function is used to combine two vectors into one vector. It combines values of both vectors. Lets create a new vector for the upcoming movies as `upcoming_movie` and add them to the `movie_vector` to create a `new_vector` of movies.

Функція зчеплення використовується для об'єднання двох векторів у один вектор. Отриманий масив поєднує в собі значення обох векторів.

Давайте створимо новий вектор для майбутніх фільмів під назвою `upcoming_movie` та додамо їх до `movie_vector`, щоб створити `new_vector`.

In []:

```
upcoming_movie <- c("Fast and Furious 8", "xXx: Return of Xander Cage", "Suicide Squad")
```

In []:

```
new_vector <- c(movie_vector, upcoming_movie)
```

In []:

```
new_vector
```

About the Author: | Про автора:

Hi! It's [Helly Patel \(https://ca.linkedin.com/in/helly-patel-90344750\)](https://ca.linkedin.com/in/helly-patel-90344750), the author of this notebook. I hope you found R easy to learn! There's lots more to learn about R but you're well on your way. Feel free to connect with me if you have any questions.

Привіт! Це [Helly Patel \(https://ca.linkedin.com/in/helly-patel-90344750\)](https://ca.linkedin.com/in/helly-patel-90344750), автор цього зошиту. Я сподіваюся, що ви побачили наскільки легко вивчати R! Є ще багато чого дізнатись про R, але ви вже рухаєтесь у вірному напрямку. Якщо у вас виникли запитання, можете без проблем зв'язатися зі мною.

About the Translator: | Про перекладача:

Hello! It's Roman Kornyliuk (<https://www.linkedin.com/in/kornelli>), the translator of this notebook in Ukrainian. Feel free to connect with me if you have any questions too.

Привіт! Мене звуть Roman Kornyliuk (<https://www.linkedin.com/in/kornelli>), перекладач цього зошиту на українську мову. Без вагань пишіть мені також, якщо у вас виникли запитання з приводу R.

Copyright © 2018 Roman Kornyliuk (<https://www.linkedin.com/in/kornelli>).

Copyright © IBM Cognitive Class (<https://cognitiveclass.ai>). This notebook and its source code are released under the terms of the MIT License (<https://cognitiveclass.ai/mit-license/>).