**What's a matrix?**

In R, a matrix is a collection of elements of the same data type (numeric, character, or logical) arranged into a fixed number of rows and columns. Since you are only working with rows and columns, a matrix is called two-dimensional.

You can construct a matrix in R with the [**matrix()**](http://www.rdocumentation.org/packages/base/functions/matrix) function. Consider the following example:

matrix(1:9, byrow = TRUE, nrow = 3)

In the [**matrix()**](http://www.rdocumentation.org/packages/base/functions/matrix) function:

* The first argument is the collection of elements that R will arrange into the rows and columns of the matrix. Here, we use 1:9 which is a shortcut for c(1, 2, 3, 4, 5, 6, 7, 8, 9).
* The argument byrow indicates that the matrix is filled by the rows. If we want the matrix to be filled by the columns, we just place byrow = FALSE.
* The third argument nrow indicates that the matrix should have three rows.

##Instruction

Construct a matrix with 3 rows containing the numbers 1 up to 9, filled row-wise.

# Analyze matrices, you shall

It is now time to get your hands dirty. In the following exercises you will analyze the box office numbers of the Star Wars franchise. May the force be with you!

In the editor, three vectors are defined. Each one represents the box office numbers from the first three Star Wars movies. The first element of each vector indicates the US box office revenue, the second element refers to the Non-US box office (source: Wikipedia).

In this exercise, you'll combine all these figures into a single vector. Next, you'll build a matrix from this vector.

##Instruction

* Use c(new\_hope, empire\_strikes, return\_jedi) to combine the three vectors into one vector. Call this vector box\_office.
* Construct a matrix with 3 rows, where each row represents a movie. Use the matrix() function to do this. The first argument is the vector box\_office, containing all box office figures. Next, you'll have to specify nrow = 3 and byrow = TRUE. Name the resulting matrix star\_wars\_matrix.

# Naming a matrix

To help you remember what is stored in star\_wars\_matrix, you would like to add the names of the movies for the rows. Not only does this help you to read the data, but it is also useful to select certain elements from the matrix.

Similar to vectors, you can add names for the rows and the columns of a matrix

rownames(my\_matrix) <- row\_names\_vector

colnames(my\_matrix) <- col\_names\_vector

We went ahead and prepared two vectors for you: region, and titles. You will need these vectors to name the columns and rows of star\_wars\_matrix, respectively.

##Instructions

* Use colnames() to name the columns of star\_wars\_matrix with the region vector.
* Use rownames() to name the rows of star\_wars\_matrix with the titles vector.
* Print out star\_wars\_matrix to see the result of your work.

# Calculating the worldwide box office

The single most important thing for a movie in order to become an instant legend in Tinseltown is its worldwide box office figures.

To calculate the total box office revenue for the three Star Wars movies, you have to take the sum of the US revenue column and the non-US revenue column.

In R, the function [**rowSums()**](http://www.rdocumentation.org/packages/base/functions/colSums) conveniently calculates the totals for each row of a matrix. This function creates a new vector:

rowSums(my\_matrix)

Calculate the worldwide box office figures for the three movies and put these in the vector named worldwide\_vector.

# Adding a column for the Worldwide box office

In the previous exercise you calculated the vector that contained the worldwide box office receipt for each of the three Star Wars movies. However, this vector is not yet part of star\_wars\_matrix.

You can add a column or multiple columns to a matrix with the [**cbind()**](http://www.rdocumentation.org/packages/base/functions/cbind) function, which merges matrices and/or vectors together by column. For example:

big\_matrix <- cbind(matrix1, matrix2, vector1 ...)

#Instructions

Add worldwide\_vector as a new column to the star\_wars\_matrix and assign the result to all\_wars\_matrix. Use the [**cbind()**](http://www.rdocumentation.org/packages/base/functions/cbind) function.

**Adding a row**

Just like every action has a reaction, every [**cbind()**](http://www.rdocumentation.org/packages/base/functions/cbind) has an [**rbind()**](http://www.rdocumentation.org/packages/base/functions/cbind). (We admit, we are pretty bad with metaphors.)

Your R workspace, where all variables you defined 'live' has already been initialized and contains two matrices:

* star\_wars\_matrix that we have used all along, with data on the original trilogy,
* star\_wars\_matrix2, with similar data for the prequels trilogy.

Type the name of these matrices in the console and hit Enter if you want to have a closer look. If you want to check out the contents of the workspace, you can type ls() in the console.

Use rbind() to paste together star\_wars\_matrix and star\_wars\_matrix2, in this order. Assign the resulting matrix to all\_wars\_matrix.

#Instructions

Use rbind() to paste together star\_wars\_matrix and star\_wars\_matrix2, in this order. Assign the resulting matrix to all\_wars\_matrix.

# The total box office revenue for the entire saga

Just like [**cbind()**](http://www.rdocumentation.org/packages/base/functions/cbind) has [**rbind()**](http://www.rdocumentation.org/packages/base/functions/cbind), [**colSums()**](http://www.rdocumentation.org/packages/base/functions/colSums) has [**rowSums()**](http://www.rdocumentation.org/packages/base/functions/colSums). Your R workspace already contains the all\_wars\_matrix that you constructed in the previous exercise; type all\_wars\_matrix to have another look. Let's now calculate the total box office revenue for the entire saga.

##Instructions

* Calculate the total revenue for the US and the non-US region and assign total\_revenue\_vector. You can use the [**colSums()**](http://www.rdocumentation.org/packages/base/functions/colSums) function.
* Print out total\_revenue\_vector to have a look at the results.

**Selection of matrix elements**

Similar to vectors, you can use the square brackets [ ] to select one or multiple elements from a matrix. Whereas vectors have one dimension, matrices have two dimensions. You should therefore use a comma to separate the rows you want to select from the columns. For example:

* my\_matrix[1,2] selects the element at the first row and second column.
* my\_matrix[1:3,2:4] results in a matrix with the data on the rows 1, 2, 3 and columns 2, 3, 4.

If you want to select all elements of a row or a column, no number is needed before or after the comma, respectively:

* my\_matrix[,1] selects all elements of the first column.
* my\_matrix[1,] selects all elements of the first row.

Back to Star Wars with this newly acquired knowledge! As in the previous exercise, all\_wars\_matrix is already available in your workspace.

##Instruction

* Select the non-US revenue for all movies (the entire second column of all\_wars\_matrix), store the result as non\_us\_all.
* Use mean() on non\_us\_all to calculate the average non-US revenue for all movies. Simply print out the result.
* This time, select the non-US revenue for the first two movies in all\_wars\_matrix. Store the result as non\_us\_some.
* Use mean() again to print out the average of the values in non\_us\_some.

# A little arithmetic with matrices

Similar to what you have learned with vectors, the standard operators like +, -, /, \*, etc. work in an element-wise way on matrices in R.

For example, 2 \* my\_matrix multiplies each element of my\_matrix by two.

As a newly-hired data analyst for Lucasfilm, it is your job to find out how many visitors went to each movie for each geographical area. You already have the total revenue figures in all\_wars\_matrix. Assume that the price of a ticket was 5 dollars. Simply dividing the box office numbers by this ticket price gives you the number of visitors.

#Instructions

* Divide all\_wars\_matrix by 5, giving you the number of visitors in millions. Assign the resulting matrix to visitors.
* Print out visitors so you can have a look.