# Matrices

A matrix is a two dimensional data structure that is similar to a vector but additionally contains the dimension attribute.

## 1. Creating

When creating a matrix, the last two arguments to matrix tell it the number of rows and columns the matrix should have.

#### 1.1 Matrix Code Example

You can use the byrow=TRUE argument to create a matrix by rows instead of by columns as shown below:

## Example   
# ---  
# Question: Let's create a matrix mymat  
# ---  
#   
mymat <- matrix(1:12,4,3)  
  
# And then print out mymat  
# ---  
# OUR CODE GOES BELOW  
#   
mymat

## [,1] [,2] [,3]  
## [1,] 1 5 9  
## [2,] 2 6 10  
## [3,] 3 7 11  
## [4,] 4 8 12

#### 1.2 Matrix Code Example

## Example   
# ---  
# Question: Let's use the byrow=TRUE argument to create a matrix   
# by rows instead of by columns as shown below  
# ---  
mymat <- matrix(1:12,ncol=3,byrow=TRUE)  
  
# And then print out the mymat variable below  
mymat

## [,1] [,2] [,3]  
## [1,] 1 2 3  
## [2,] 4 5 6  
## [3,] 7 8 9  
## [4,] 10 11 12

## 2. Naming

#### 2.1 Matrix Naming Code Example

## Example   
# ---  
# In order to remember what is stored in a matrix, you can add the names of the columns and rows.   
# This will also help you to read the data as well as select elements from the matrix.  
# ---  
# Question: Lets create the vectors kenya, ethiopia and chad  
# ---  
# OUR CODE GOES BELOW  
#   
kenya <- c(460.998, 314.4)   
ethiopia <- c(290.475, 247.900)   
chad <- c(309.306, 165.8)  
  
# then create a matrix geography\_matrix  
# ---  
#   
geography\_matrix <- matrix(c(kenya, ethiopia, chad), nrow = 3, byrow = TRUE)  
  
# Uncomment the following two vectors; location and countries which will be used for naming  
# ---  
#  
location <- c("Lat", "Long")  
countries <- c("Kenya", "Ethiopia", "Chad")  
  
# Then also uncomment the line below and so as to name the columns with location  
# ---  
#   
colnames(geography\_matrix) <- location  
  
# More on uncomment also the following line so as to name the rows with countries  
# ---  
#  
rownames(geography\_matrix) <- countries  
  
# And lastly print out geography\_matrix  
# ---  
#  
geography\_matrix

## Lat Long  
## Kenya 460.998 314.4  
## Ethiopia 290.475 247.9  
## Chad 309.306 165.8

## Challenge   
# ---  
# Question: Create a matrix family with column names Name, Age, Gender and Occupation.   
# Populate it with 5 your own family members.  
# ---  
# OUR CODE GOES HERE  
#   
Elsie <- c(18, "Female", "Student")  
Joy <- c(45, "Female", "Doctor")  
John <- c(50, "Male", "Accounts")  
Peter <- c(24, "Male", "Engineer")  
  
family <- matrix(c(Elsie, Joy, John, Peter), nrow = 4, byrow = TRUE)  
  
Bio <- c("age", "gender", "Occupation")  
person <- c("Elsie", "Joy", "John", "Peter")  
  
colnames(family) <- Bio  
  
# More on uncomment also the following line so as to name the rows with countries  
# ---  
#  
rownames(family) <- person  
  
# And lastly print out geography\_matrix  
# ---  
#  
family

## age gender Occupation  
## Elsie "18" "Female" "Student"   
## Joy "45" "Female" "Doctor"   
## John "50" "Male" "Accounts"  
## Peter "24" "Male" "Engineer"

## 3. Adding a Column

You can add a row to a matrix using the rbind() function.

#### 3.1 Adding a Column Code Example

## Example   
# ---  
# Question: Create the matrix x below  
# ---  
# OUR CODE GOES BELOW  
#   
x <- matrix(1:9, nrow = 3)  
  
  
# Then add a column as shown  
# ---  
#   
cbind(x, c(1, 2, 3))

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

## Challenge   
# ---  
# Question: Add a column residence to your fictional family matrix that you had created earlier  
# ---  
# OUR CODE GOES BELOW  
#   
cbind(family, c("Nai", "Msa", "Nkr", "Nai"))

## age gender Occupation   
## Elsie "18" "Female" "Student" "Nai"  
## Joy "45" "Female" "Doctor" "Msa"  
## John "50" "Male" "Accounts" "Nkr"  
## Peter "24" "Male" "Engineer" "Nai"

## 4. Adding a Row

#### 4.1 Adding a Row Code Example

## Example   
# ---  
# Question: Creating a matrix x  
# ---  
#   
x <- matrix(1:9, nrow = 3)  
  
# You can also add a row using the cbind() function  
# ---  
#   
rbind(x,c(1,2,3))

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

## Challenge   
# ---  
# Question: Add a fictional character to your fictional family matrix  
# ---  
# OUR CODE GOES BELOW  
#   
  
rbind(family ,c(18, "Male", "Banker"))

## age gender Occupation  
## Elsie "18" "Female" "Student"   
## Joy "45" "Female" "Doctor"   
## John "50" "Male" "Accounts"  
## Peter "24" "Male" "Engineer"  
## "18" "Male" "Banker"

## 5. Selecting a Matrix

To select an element of a matrix, one needs to specify both the row and the column as shown:

## Example   
# ---  
# Question: Select the following matrix  
# ---  
# OUR CODE GOES BELOW  
#   
x <- matrix(1:9, nrow = 3)  
  
# Select the elements from the above matrix   
# By uncommenting the following lines.  
# ---  
#   
x[1,3] # select the element at 1nd row, 3rd column

## [1] 7

x[2, ] # the 2nd row

## [1] 2 5 8

x[ ,3] # the 3rd column

## [1] 7 8 9

## Challenge   
# ---  
# Question: Select the last member of your family member   
# ---  
# OUR CODE GOES BELOW  
#   
family[4, ]

## age gender Occupation   
## "24" "Male" "Engineer"

## Challenge   
# ---  
# Question: Select the first member of your family member   
# ---  
# OUR CODE GOES BELOW  
#   
family[1, ]

## age gender Occupation   
## "18" "Female" "Student"

## 6. Operations

### Matrix addition & subtraction

#### 6.1 Matrix Addition & Subtraction Code Example

## Example   
# ---  
# Question: Matrix addition and subtract require the matrices to have the same dimensions.   
# Let's start by creating matrices x and y.  
# ---  
# OUR CODE GOES BELOW  
#   
x <- matrix(c(3, 9, -1, 4, 2, 6), nrow = 2)  
y <- matrix(c(5, 2, 0, 9, 3, 4), nrow = 2)  
  
# Print matrix x below  
# ---  
#   
x

## [,1] [,2] [,3]  
## [1,] 3 -1 2  
## [2,] 9 4 6

# Print the matrix y below  
# ---  
#   
y

## [,1] [,2] [,3]  
## [1,] 5 0 3  
## [2,] 2 9 4

# Add the matrices and print out the result  
# ---  
#  
x+y

## [,1] [,2] [,3]  
## [1,] 8 -1 5  
## [2,] 11 13 10

# Subtract the matrices  
# ---  
#   
x-y

## [,1] [,2] [,3]  
## [1,] -2 -1 -1  
## [2,] 7 -5 2

### Matrix Multiplication & Division

#### 6.2 Matrix Multiplication & Division Code Example

## Challenge  
# ---  
# Question: Let's create two 2 x 3 matrices x and y  
# ---  
# OUR CODE GOES BELOW  
#   
x <- matrix(c(3, 9, -1, 4, 2, 6), nrow = 2)  
y <- matrix(c(5, 2, 0, 9, 3, 4), nrow = 2)  
x

## [,1] [,2] [,3]  
## [1,] 3 -1 2  
## [2,] 9 4 6

y

## [,1] [,2] [,3]  
## [1,] 5 0 3  
## [2,] 2 9 4

# And then multiply these matrices, assign the result to to the variable z  
# ---  
#   
z = x\*y  
  
# Now print out the matrix z  
# ---  
#   
z

## [,1] [,2] [,3]  
## [1,] 15 0 6  
## [2,] 18 36 24

# Lastly divide matrix x by y and assign the result to the variable z  
# ---  
#  
z= x/y  
z

## [,1] [,2] [,3]  
## [1,] 0.6 -Inf 0.6666667  
## [2,] 4.5 0.4444444 1.5000000