Enter the Matrix

Lesson:
A matrix is a collection of data elements arranged in a two-dimensional rectangular layout. Here is a visual representation of a matrix with **2 rows** and **3 columns**.

$$\begin{bmatrix} 1 & 5 & 4 \\ 3 & 7 & 2 \end{bmatrix}$$

You can in making a matrix in R is you will use the **matrix()** function. Here are a few ways to use this:

- matrix(c(1,3,5,7,9,11))

  Output:

  1
  3
  5
  7
  9
  11

- matrix(1:5)

  Output:

  1
  2
  3
  4
  5

- matrix(1:6, nrow = 3)

  Output:

  1 4
  2 5
  3 6

- matrix(1:6, byrow = TRUE, nrow = 3)

  Output:

  1 2
  3 4
  5 6

- Using the **c()** function will let you insert your choice of data.
- While using **1:5** will give you values raiding from 1 to 5.
- **nrow** indicates the number of rows the matrix should have.
- The argument **byrow** will indicate if the matrix will be filled by rows or not and this is indicated by using **TRUE** or **FALSE.**

To do:

1. Create a matrix of **9** numbers that has **3 rows** and **3 columns**.
2. Create a matrix that will produce this output:

   1 3 5
   2 4 6

Analyzing Matrices

Lesson:
Since one of R's use is statistical computing, what better way to ____ than with statistics! Here's an example activity you can run.

To do:

1. Copy the following lines to your editor. Understand and run it.

```
# Fruit Stand Sales (in kgs)
apples <- c(17, 22, 14, 7 ,36 )
oranges <- c(32, 42, 22,2, 12)
bananas <- c(20, 15, 36, 11, 14)
total_sales <- c(apples,oranges, bananas)
total_sales_matrix <- matrix(total_sales, byrow = TRUE, nrow = 3)
total_sales_matrix
```

Labels are Important

In order to make sense of what is store in **total_sales_matrix**, let's add names in order for the data to be coherent. Using **rownames()** and **colnames()** we can add vectors containing the titles to be used as names for columns and rows.

To do:

1. Using the 3 existing fruit vectors in a **c()**, place them inside to a new **total_sales_matrix** vector but with **matrix()** and having **nrow = 3** and **byrow = TRUE.**
   total_sales_matrix <- matrix(c(apples, oranges, bananas), nrow = 3, byrow = TRUE
2. Create a vector **days** with the values **"Monday", "Tuesday", "Wednesday", "Thursday", "Friday"**
3. Create a vector **fruits** with the values **"Apples", "Oranges", "Bananas"**
4. Name the columns with this line
   colnames(total_sales_matrix) <- days
5. Do the same thing above but with **rownames()** and **fruits**
6. Display **total_sales_matrix**

Hacking the Matrix

Lesson:
We are now able to create and label matrices and now it's time we do some manipulations. Take note of the following functions.

- **rowSums()** – Allows you to get the sum of values per row.
- **colSums()** – Provides you the sum of values per column.
- **cbind()** – Allows you to add a column or multiple columns to an existing matrix.
- **rbind()** – Merges matrices with similar columns
- **mean()** – Calculates the average of values.

To do (use the existing fruit vectors):

1. Get the sum of fruits sold **per day** and save it to a variable named **total_sales_day**.
2. Get the sum of fruits sold **per type** and save it to a variable named **total_sales_fruit.**
3. Get the average of fruits sold **per day.**
4. Get the average of fruits sold **per type**.
5. Add the following vectors and add them to a matrix named **total_sales_matrix2.**
   lemons <- c(49, 39, 17, 33, 9)
   kiwis <- c(36, 19, 47, 27, 35)
   mangoes <-c(10, 31, 16, 4, 24)
6. Create a vector **fruits2** with values **"Lemons", "Kiwis", "Mangoes"** and use **fruits2** to label **total_sales_matrix2.**
7. Create a vector **all_sales_matrix** and merge **total_sales_matrix** and **total_sales_matrix2** inside it**.**
8. Display **all_sales_matrix.**
   Expected Output:

	Monday	Tuesday	Wednesday	Thursday	Friday
Apples	17	22	14	7	36
Oranges	32	42	22	2	12
Bananas	20	15	36	11	14
Lemons	49	39	17	33	9
Kiwis	36	19	47	27	35
Mangoes	10	31	16	4	24

End!
Now you're just about done with the basics of R. You can now head on to the Data Camp course for a quick review and a more advanced approach to R.