

# Data manipulation



**Feng Li**

**feng.li@cufe.edu.cn**

**School of Statistics and Mathematics  
Central University of Finance and Economics**

# Today we are going to learn...

- 1 Generate a sequence
- 2 Vectors and matrices
- 3 Vectorization
- 4 Data Structures

# Sequences

- Generate a sequence: `seq()`
- Repeat a vector: `rep()`

# Vectors

- Numerical vectors
- Logical vectors
- Characters
- Length of a vector
- Vector calculations

# Mathematical functions

- `sqrt()`, `log()`
- `sin()`, `cos()`, `tan()`

# Matrices

- Create a matrix: `matrix()`
- Dimension of a matrix: `dim()`
- How many elements in a matrix: `length()`
- Extract elements from a matrix.
- Replace elements with new entries.
- Create special matrices: diagonal matrix, identity matrix, zero matrix...
- Matrix multiplications: `%*%`
- Matrix inverse: `solve()`
- Transpose of a matrix: `t()`
- Element-wise operation with a matrix.
- Combine two or more matrices: `rbind()`, `cbind()`

# The concept of vectorization

- Vectorization: same operation applies to each element of an object.
- It is not a loop.
- If you want to do four additions

$$c_1 = a_1 + b_1$$

$$c_2 = a_2 + b_2$$

$$c_3 = a_3 + b_3$$

$$c_4 = a_4 + b_4$$

in C you have to write it as

```
for (i=0; i<4; i++)  
    c[i] = a[i] + b[i];
```

- There is a vectorizing compiler in C that transform such a loop into a sequence of vector operations, that perform additions on length-four (in our example) blocks of elements from the arrays a, b and c when you compile you C code.
- In R, you can just do it as  
 $c = a + b$   
and R will do the rest for you.

# Vectorization with vectors and matrices

- Vectorize your code as much as possible.
- Vectorization works with vectors, matrices and arrays.
- Special case
  - When matrix multiplies (\*) a scalar or a vector, it will first repeat the vector to be the same length of the matrix. Then do the element-wise multiplication.
  - Same rule applies to other types of arithmetic.



# Array

- An array is a high dimensional matrix.
- A matrix is a special case of an array when the dimension is two.
- A vector is a special array when there is no dimension (In R the dimension is usually dropped in this situation)

# List

- Special data structure that matrix could not handle.
  - Data length are not the same.
  - Data type are not the same.
  - Nested data structure within a list.
- Create a list: `list()`
- Extract elements of a list: `[[ ]]` or `$`
- Delete an element within a list: set `NULL` to that element.

# Data frame

- `data.frame()`: tightly coupled collections of variables which share many of the properties of matrices and of lists, used as the fundamental data structure by most of R's modeling software.
- In most cases, the operation with a data frame is similar to matrix operation.

# Discussion

What type of data structure would you choose when you meet the following situations.

- Data are of the same length but different types.
- Data are not of the same length.
- Hierarchical structure of the data.

## Suggested reading

- R-intro (2015): **Chapter 2, 3, 5, 6**