



INTRODUCTION TO R

Create and Name Matrices

Matrix

- Vector: 1D array of data elements
- Matrix: 2D array of data elements
- Rows and columns
- One atomic vector type

Create a matrix

matrix()

```
> matrix(1:6, nrow = 2)
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
> matrix(1:6, ncol = 3)
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
> matrix(1:6, nrow = 2, byrow = TRUE)
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
```

Create a matrix: recycling

```
> matrix(1:3, nrow = 2, ncol = 3)
```

```
      [,1] [,2] [,3]  
[1,]     1     3     2  
[2,]     2     1     3
```

```
> matrix(1:4, nrow = 2, ncol = 3)
```

```
      [,1] [,2] [,3]  
[1,]     1     3     1  
[2,]     2     4     2
```

Warning message:

In matrix(1:4, nrow = 2, ncol = 3) :

data length [4] is not a sub-multiple or multiple of the
number of columns [3]

`rbind()`, `cbind()`

```
> cbind(1:3, 1:3)
```

	[,1]	[,2]
[1,]	1	1
[2,]	2	2
[3,]	3	3

```
> rbind(1:3, 1:3)
```

	[,1]	[,2]	[,3]
[1,]	1	2	3
[2,]	1	2	3

`rbind()`, `cbind()`

```
> m <- matrix(1:6, byrow = TRUE, nrow = 2)
```

```
> rbind(m, 7:9)
```

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

```
> cbind(m, c(10, 11))
```

	[,1]	[,2]	[,3]	[,4]
[1,]	1	2	3	10
[2,]	4	5	6	11

Naming a matrix `rownames()`, `colnames()`

```
> m <- matrix(1:6, byrow = TRUE, nrow = 2)
```

```
> rownames(m) <- c("row1", "row2")
```

```
> m
```

	[,1]	[,2]	[,3]
row1	1	2	3
row2	4	5	6

```
> colnames(m) <- c("col1", "col2", "col3")
```

```
> m
```

	col1	col2	col3
row1	1	2	3
row2	4	5	6

Naming a matrix

```
> m <- matrix(1:6, byrow = TRUE, nrow = 2)
```


Naming a matrix

```
> m <- matrix(1:6, byrow = TRUE, nrow = 2,  
              dimnames = list(c("row1", "row2"),  
                              c("col1", "col2", "col3")))
```

```
> m
```

	col1	col2	col3
row1	1	2	3
row2	4	5	6

Coercion

```
> num <- matrix(1:8, ncol = 2)
> num
      [,1] [,2]
[1,]    1    5
[2,]    2    6
[3,]    3    7
[4,]    4    8
> char <- matrix(LETTERS[1:6], nrow = 4, ncol = 3)
> char
      [,1] [,2] [,3]
[1,] "A"  "E"  "C"
[2,] "B"  "F"  "D"
[3,] "C"  "A"  "E"
[4,] "D"  "B"  "F"
```

Coercion

```
> num <- matrix(1:8, ncol = 2)
> char <- matrix(LETTERS[1:6], nrow = 4, ncol = 3)

> cbind(num, char)
      [,1] [,2] [,3] [,4] [,5]
[1,] "1"  "5"  "A"  "E"  "C"
[2,] "2"  "6"  "B"  "F"  "D"
[3,] "3"  "7"  "C"  "A"  "E"
[4,] "4"  "8"  "D"  "B"  "F"
```

Contain different types? `list` or `data.frame`



INTRODUCTION TO R

Subsetting Matrices

Subset element

```
> m <- matrix(sample(1:15, 12), nrow = 3)
```

```
> m
```

	[,1]	[,2]	[,3]	[,4]
[1,]	5	11	15	3
[2,]	12	14	8	9
[3,]	6	1	4	2

```
> m[1,3]
```

```
[1] 15
```

```
> m[3,2]
```

```
[1] 1
```

Subset column or row

```
> m[3,]  
[1] 6 1 4 2  
  
> m[,3]  
[1] 15 8 4  
  
> m[4]  
[1] 11  
  
> m[9]  
[1] 4
```

```
> m  
      [,1] [,2] [,3] [,4]  
[1,]    5   11   15    3  
[2,]   12   14    8    9  
[3,]    6    1    4    2
```

Subset multiple elements

```
> m[c(2, 3)]  
[1] 14 8
```

```
> m[c(1, 2), c(2, 3)]  
      [,1] [,2]  
[1,]  11  15  
[2,]  14   8
```

```
> m[c(1, 3), c(1, 3, 4)]  
      [,1] [,2] [,3]  
[1,]   5  15   3  
[2,]   6   4   2
```

```
> m  
      [,1] [,2] [,3] [,4]  
[1,]    5  11  15   3  
[2,]   12  14   8   9  
[3,]    6   1   4   2
```

Subset by name

```
> rownames(m) <- c("r1", "r2", "r3")
> colnames(m) <- c("a", "b", "c", "d")
> m
```

	a	b	c	d
r1	5	11	15	3
r2	12	14	8	9
r3	6	1	4	2

```
> m[2,3]
[1] 8
```

```
> m["r2", "c"]
[1] 8
```

```
> m[2, "c"]
[1] 8
```

```
> m[3, c("c", "d")]
c d
4 2
```




INTRODUCTION TO R

Matrix Arithmetic

Matrix Arithmetic

- `colSums()`, `rowSums()`
- Standard arithmetic possible
- Element-wise computation

lotr_matrix

```
> the_fellowship <- c(316, 556)
> two_towers <- c(343, 584)
> return_king <- c(378, 742)

> lotr_matrix <- rbind(the_fellowship, two_towers, return_king)
> colnames(lotr_matrix) <- c("US", "non-US")
> rownames(lotr_matrix) <- c("Fellowship", "Two Towers",
                             "Return King")

> lotr_matrix
```

	US	non-US
Fellowship	316	556
Two Towers	343	584
Return King	378	742

Matrix - Scalar

```
> lotr_matrix / 1.12
```

	US	non-US
Fellowship	282.1429	496.4286
Two Towers	306.2500	521.4286
Return King	337.5000	662.5000

```
> lotr_matrix - 50
```

	US	non-US
Fellowship	266	506
Two Towers	293	534
Return King	328	692

```
> lotr_matrix
```

	US	non-US
Fellowship	316	556
Two Towers	343	584
Return King	378	742

Matrix - Matrix

```
> # Definition of theater_cut omitted
```

```
> theater_cut
```

```
      [,1] [,2]
```

```
[1,]    50    50
```

```
[2,]    80    80
```

```
[3,]   100   100
```

```
> lotr_matrix - theater_cut
```

```
      US non-US
```

```
Fellowship  266    506
```

```
Two Towers  263    504
```

```
Return King 278    642
```

```
> lotr_matrix
```

```
      US non-US
```

```
Fellowship  316    556
```

```
Two Towers  343    584
```

```
Return King 378    742
```

Recycling

```
> lotr_matrix - c(50, 80, 100)
      US non-US
Fellowship 266    506
Two Towers 263    504
Return King 278    642

> matrix(c(50, 80, 100), nrow = 3, ncol = 2)
      [,1] [,2]
[1,]   50   50
[2,]   80   80
[3,]  100  100
```

```
> lotr_matrix
      US non-US
Fellowship 316    556
Two Towers 343    584
Return King 378    742
```

Matrix Multiplication

```
> # Definition of rates omitted  
> rates
```

```
      [,1] [,2]  
[1,] 1.11 1.11  
[2,] 0.99 0.99  
[3,] 0.82 0.82
```

```
> lotr_matrix * rates
```

```
      US non-US  
Fellowship 350.76 617.16  
Two Towers 339.57 578.16  
Return King 309.96 608.44
```

```
> lotr_matrix
```

	US	non-US
Fellowship	316	556
Two Towers	343	584
Return King	378	742

Matrices and Vectors

- Very similar
- Vector = 1D, matrix = 2D
- Coercion if necessary
- Recycling if necessary
- Element-wise calculations

Subset with logical vector

```
> m[c(FALSE, FALSE, TRUE),  
     c(FALSE, FALSE, TRUE, TRUE)]
```

```
c d  
4 2
```

```
> m[c(FALSE, FALSE, TRUE),  
     c(FALSE, TRUE)]
```

```
b d  
1 2
```

```
> m[c(FALSE, FALSE, TRUE),  
     c(FALSE, TRUE, FALSE, TRUE)]
```

```
b d  
1 2
```

```
> m  
      a  b  c d  
r1    5 11 15 3  
r2   12 14  8 9  
r3    6  1  4 2
```