

# Vectors (part 3)

## R Data Objects

Gaston Sanchez

CC BY-NC-SA 4.0

R Coding Compendium

[Donate](#)

# About

To make the best of the R language, you'll need a strong understanding of the basic **data types** and **data structures** and how to operate on them.

# Vector Manipulation

## Subsetting: Bracket Notation

opening bracket                      closing bracket

⋮    ⋮

**vec** [*index*]

⋮    ⋮

atomic vector                      index vector

# Bracket Notation System

- ▶ To extract values from R objects use brackets: `[ ]`
- ▶ Inside the brackets specify a vector of indices
- ▶ Vector of indices can be of type numeric (integer or double), logical, and sometimes character

# Indexing

Consider the following vector `x`

```
# some vector  
x <- c(2, 4, 6, 8)  
  
# adding names  
names(x) <- letters[1:4]  
  
x
```

```
## a b c d
```

```
## 2 4 6 8
```

# Numeric Indexing

```
# first element
```

```
x[1]
```

```
## a
```

```
## 2
```

```
# second element
```

```
x[2]
```

```
## b
```

```
## 4
```

# Numeric Indexing

```
# last element
```

```
x[length(x)]
```

```
## d
```

```
## 8
```

```
# recall that first position is 1
```

```
x[0]
```

```
## named numeric(0)
```



# Numeric Indexing

```
# first 3 elements
```

```
x[1:3]
```

```
## a b c
```

```
## 2 4 6
```

```
# non-consecutive elements
```

```
x[c(1, 3)]
```

```
## a c
```

```
## 2 6
```

# Numeric Indexing

```
# different order
```

```
x[c(3, 2, 4, 1)]
```

```
## c b d a
```

```
## 6 4 8 2
```

```
# different order (and repetition)
```

```
x[c(3, 2, 4, 1, 1, 1)]
```

```
## c b d a a a
```

```
## 6 4 8 2 2 2
```

# Numeric Indexing

Negative numbers indicate exclusion

```
# exclude 2nd element
```

```
x[-2]
```

```
## a c d
```

```
## 2 6 8
```

```
# exclude non-consecutive elements
```

```
x[-c(1, 3)]
```

```
## b d
```

```
## 4 8
```

# Logical Indexing (aka logical subsetting)

You can also use logical index vectors

```
# first element  
x[c(TRUE, FALSE, FALSE, FALSE)]
```

```
## a
```

```
## 2
```

```
# exclude first element  
x[c(FALSE, TRUE, TRUE, TRUE)]
```

```
## b c d
```

```
## 4 6 8
```

Note: you won't typically be typing logical vectors like in these examples; instead you'll be using logical vectors that result from logical operators or from comparison operators

# Logical Indexing (aka logical subsetting)

Logical subsetting is extremely powerful

```
# elements equal to 2  
x[x == 2]
```

```
## a
```

```
## 2
```

```
# elements different to 2  
x[x != 2]
```

```
## b c d
```

```
## 4 6 8
```

# Logical Indexing (aka logical subsetting)

```
# elements greater than 1
```

```
x[x > 1]
```

```
## a b c d
```

```
## 2 4 6 8
```

```
# greater than 1 and less than or equal to 3
```

```
x[x > 1 & x <= 3]
```

```
## a
```

```
## 2
```

## Logical Indexing (aka logical subsetting)

```
# try this
```

```
x[TRUE]
```

```
# and this
```

```
x[FALSE]
```

```
# what about this?
```

```
x[as.logical(c(0, 1, pi, -10))]
```

# Character Indexing (aka logical subsetting)

You can use a character index vector as long as `x` has named elements:

```
# element named "a"  
x["a"]
```

```
## a
```

```
## 2
```

```
# elements named "b" and "d"  
x[c("b", "d")]
```

```
## b d
```

```
## 4 8
```



## Character Indexing (aka logical subsetting)

You can use a character index vector as long as `x` has named elements:

```
# repeated elements  
x[rep("a", 5)]
```

```
## a a a a a
```

```
## 2 2 2 2 2
```

# More Indexing

This is less common but possible:

```
x[-1][2:3]
```

```
## c d
```

```
## 6 8
```

```
x[-length(x)][-2][2]
```

```
## c
```

```
## 6
```

# Double Brackets

# Double Bracket Notation

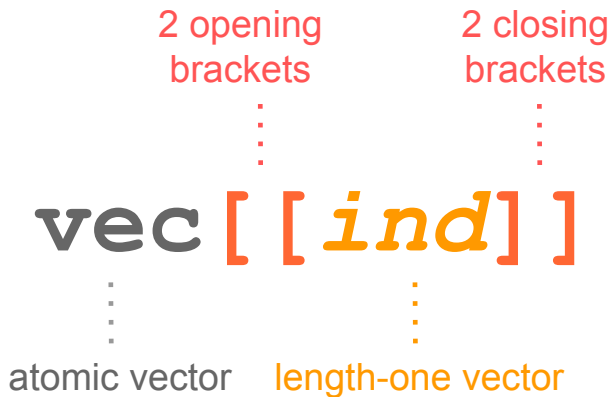
2 opening  
brackets

2 closing  
brackets

`vec` `[ [ind] ]`

atomic vector

length-one vector



The diagram illustrates the components of the R expression `vec[[ind]]`. The word `vec` is in dark grey, with a vertical ellipsis below it labeled 'atomic vector'. The first opening square bracket `[` is red, with '2 opening brackets' written above it and a vertical ellipsis below it. The word `ind` is in orange italics, with a vertical ellipsis below it labeled 'length-one vector'. The closing square bracket `]` is red, with '2 closing brackets' written above it and a vertical ellipsis below it. The second opening square bracket `[` is also red, with a vertical ellipsis below it.

# Double Brackets

You can also use double brackets `[[ ]]` but in this case only to extract a **single element**

```
x <- c(a = 2, b = 4, c = 6, d = 8)
x
```

```
## a b c d
## 2 4 6 8
```

```
x[[2]]
```

```
## [1] 4
```

```
x[["a"]]
```

```
## [1] 2
```

# Double Brackets

With double brackets, you **cannot** pass an index vector of length greater than one

```
x[[1:2]]
```

```
Error in x[[1:2]] :
```

```
  attempt to select more than one element in vectorIndex
```

## Donation

*If you find any value and usefulness in this set of slides, please consider making a one-time donation in any amount (via paypal). Your support really matters.*

Donate

[https://www.paypal.com/donate?business=ZF6U7K5MW25W2&currency\\_code=USD](https://www.paypal.com/donate?business=ZF6U7K5MW25W2&currency_code=USD)