

## Oggetti ed attributi in R

Giorgio Valentini  
e-mail: [valentini@dsi.unimi.it](mailto:valentini@dsi.unimi.it)

DSI – Dipartimento di Scienze dell' Informazione  
Università degli Studi di Milano

1

## Oggetti e attributi

- Le entità su cui R opera sono informalmente definite come **oggetti**.
- La nozione oggetto ha un significato tecnico preciso nei linguaggi *object-based* ed *object-oriented*. In queste slide, seguendo l'approccio informale dei corsi e manuali introduttivi su R, il termine “oggetto” si riferisce invece a qualsiasi entità definita in R, caratterizzata da proprietà definite tramite **attributi**.
- Da questo punto di vista sono oggetti sia i vettori (a valori numerici, a caratteri o logici), sia le liste, gli array ed anche le funzioni.
- Gli attributi degli oggetti possono essere modificati ed esistono funzioni (*attributes(object)* e *attr(object, names)*) che permettono di accedervi.
- In questa sezione vedremo solo qualche esempio di attributi utilizzabili per i vettori. Altre tipologie di attributi veranno introdotte nelle prossime lezioni

2

## Attributi

Ogni oggetto è caratterizzato da un insieme di attributi.

Es: i vettori sono costituiti da elementi base di un medesimo **modo**:

```
> x <- c(1,2,3,4)
> mode(x)
[1] "numeric"
> y <- x>3
> y
[1] FALSE FALSE FALSE TRUE
> mode(y)
[1] "logical"
> x <- c("ciao", "topo")
> mode(x)
[1] "character"
```

I vettori sono caratterizzati da una determinata **lunghezza**:

```
> x <- c(1,2,3,4)
> length(x)
[1] 4
```

3

## Cambio di modo

In R è spesso possibile è spesso possibile forzare *un cambio di modo degli oggetti*.

Ad es: con i vettori è possibile forzare un vettore di modo numerico ad un modo carattere:

```
> x <- 0:9
> x
[1] 0 1 2 3 4 5 6 7 8 9
> car <- as.character(x)
> car
[1] "0" "1" "2" "3" "4" "5" "6" "7" "8" "9"
```

E viceversa:

```
> as.integer(car)
[1] 0 1 2 3 4 5 6 7 8 9
```

In R esistono diverse funzioni del tipo "*as.something()*" per forzare la conversione di modo o per permettere ad un oggetto di acquisire nuovi attributi

4

## Cambiare la lunghezza di un oggetto

In R si possono costruire oggetti di lunghezza nulla:

```
> vuoto <- numeric()
> length(vuoto)
[1] 0
> mode(vuoto)
[1] "numeric"
```

E si può variare la lunghezza di un oggetto:

```
> v <- numeric()
> v[4] <- 1 # costruzione implicita di un vettore di
           # lunghezza 4
> v
[1] NA NA NA 1
> v <- v[2*1:2] # "accorciamento" implicito tramite
               # assegnamento
> v
[1] NA 1
```

5

## L'attributo names

L'attributo **names**, quando presente, etichetta gli elementi di un vettore o di una lista con una stringa di caratteri.

Es:

```
> x <- numeric(5)
> names(x) <- c("eta", "altezza", "press.", "glicemia", "temp.")
> x
      eta  altezza  press.  glicemia  temp.
      0      0      0      0      0
> x[1] <- 45
> x[2] <- 175
```

L'attributo names può anche essere usato come indice:

```
> x["altezza"]
altezza
      175
> x["altezza"] <- 182
> x[2]
altezza
      182
```

6

## Visualizzare e settare gli attributi

La funzione `attributes(oggetto)` dà una lista di tutti gli attributi non intrinseci (attributi diversi da `mode` e `length`) attualmente definiti per un determinato oggetto:

```
> x <- 1:5
> attributes(x)
NULL
> names(x) <- c(paste("c", 1:5, sep=""))
> attributes(x)
$names
[1] "c1" "c2" "c3" "c4" "c5"
```

La funzione `attr(oggetto,nome)`, può essere utilizzata per selezionare o modificare uno specifico attributo:

```
> attr(x,"names")
[1] "c1" "c2" "c3" "c4" "c5"
> attr(x,"names")[3]
[1] "c3"
> attr(x,"names")[3] <- "componente3" # modifica di un
# attributo
> attr(x,"names")
[1] "c1" "c2" "componente3" "c4" "c5"
```

7

## Esercizi

1. Costruire un vettore numerico  $x$  di 4 elementi etichettati con p,s,t,q. Utilizzare la funzione `names` per visualizzare le etichette. Assegnare alla seconda componente il valore 5 utilizzando l'attributo `names` come indice. Cambiare le etichette di  $x$  in primo,secondo,terzo,quarto.
2. Generare un vettore casuale  $z$  di 10 elementi. Allungare il vettore in modo che abbia 20 elementi. Tramite un assegnamento ridurre  $z$  a lunghezza 7. Verificare tramite la funzione `length` che la lunghezza sia effettivamente 7.
3. Costruire un vettore numerico i cui elementi siano i numeri da 5 a 9. Forzarne la conversione a vettore di caratteri. E' possibile forzare la conversione a vettore logico? Cosa si ottiene in tal caso? Se si forza la conversione di un vettore di stringhe a vettore di interi cosa si ottiene?

8