

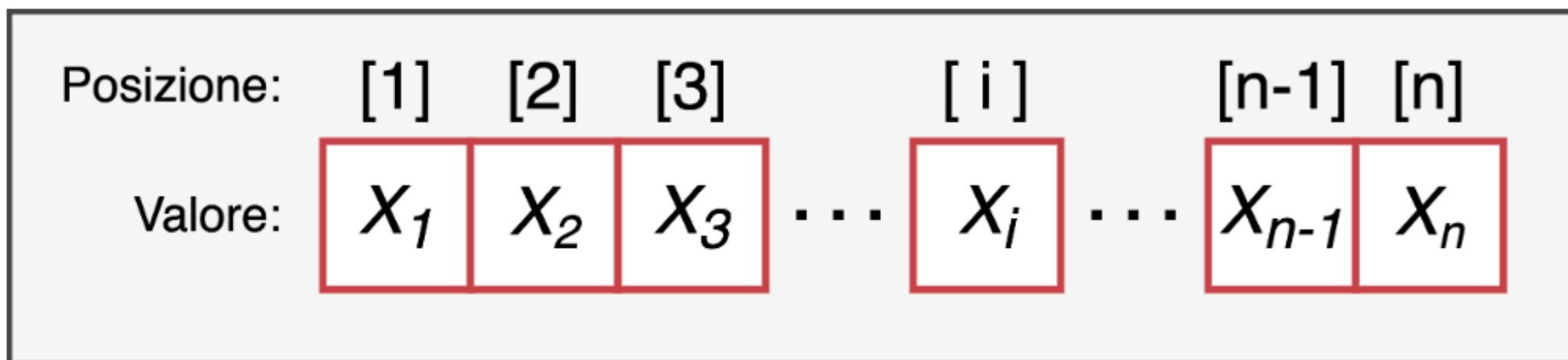
2.1_vettori

Strutture Dati

- *vettori*
- fattori
- liste
- matrici
- array
- dataframe

Vettori

I vettori sono una struttura dati unidimensionale e sono la più semplice presente in R.

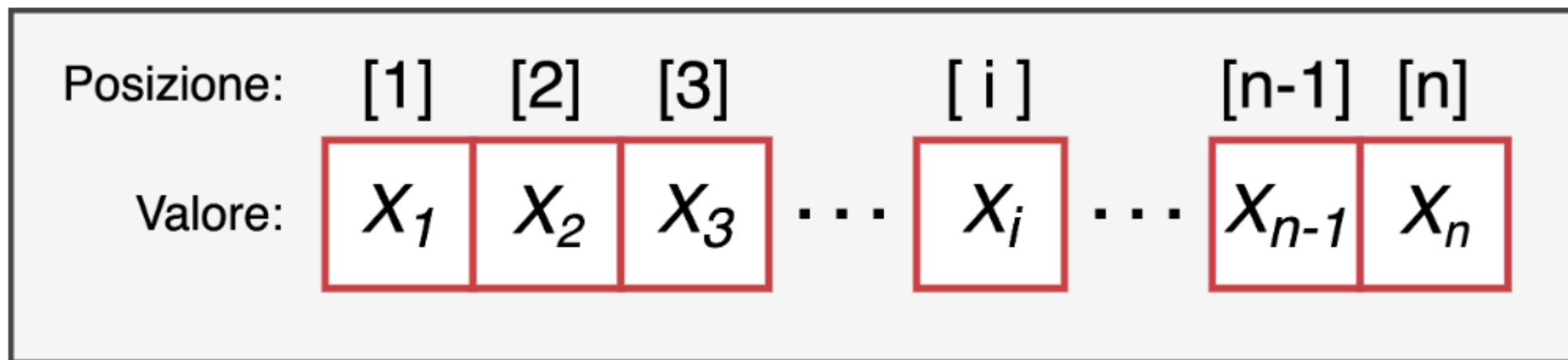


Caratteristiche di un vettore

- **la lunghezza:** il numero di elementi da cui è formato il vettore
- **la tipologia:** la tipologia di dati da cui è formato il vettore. Un vettore infatti deve essere formato da **elementi tutti dello stesso tipo!**

Caratteristiche degli elementi di un vettore

- **un valore:** il valore dell'elemento che può essere di qualsiasi tipo ad esempio un numero o una serie di caratteri
- **un indice di posizione:** un numero intero positivo che identifica la sua posizione all'interno del vettore.



Creare un vettore

I vettori si possono creare attraverso il comando `c()`, indicando tra le parentesi i valori degli elementi nella successione desiderata e separati da una virgola.

```
1 num_vect = c(1,2,3,4)
2 # è possibile anche utilizzare la funzione seq()
3 num_vect_seq = seq(from = 1,to = 4, by = 1)
4 num_vect;num_vect_seq
```

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

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

```
1 char_vect = c("R","R","R","ok")
2 # è possibile anche utilizzare la funzione rep()
3 char_vect_rep = c(rep("R", 3), "ok")
4 char_vect;char_vect_rep
```

```
[1] "R"  "R"  "R"  "ok"
```

```
[1] "R"  "R"  "R"  "ok"
```

Tiplogia di vettore

La tipologia di dati da cui è formato il vettore.

```
1 class(num_vect)
```

```
[1] "numeric"
```

```
1 class(char_vect)
```

```
[1] "character"
```

Tiplogia di vettore

Un vettore deve essere formato da **elementi tutti dello stesso tipo!**

```
1 wrong = c(1,2,3,"non so", 4)
2 class(wrong)
```

```
[1] "character"
```

```
1 wrong
```

```
[1] "1"      "2"      "3"      "non so" "4"
```

Altrimenti si “rischia” che tutto venga trasformato a carattere.

```
1 correct = c(1,2,3,NA, 4)
2 class(correct)
```

```
[1] "numeric"
```

```
1 correct
```

```
[1] 1 2 3 NA 4
```


is.* & as.*

Possiamo testare o convertire (quando possibile) la tipologia del vettore attraverso queste funzioni **is.** & **as.**

```
1 is.character(char_vect)
```

```
[1] TRUE
```

```
1 as.numeric(char_vect) #!!
```

```
[1] NA NA NA NA
```

```
1 is.numeric(num_vect)
```

```
[1] TRUE
```

```
1 as.character(num_vect) #!!
```

```
[1] "1" "2" "3" "4"
```

```
1 logi_vect = c(TRUE,FALSE,TRUE)
2 is.logical(logi_vect)
```

```
[1] TRUE
```

```
1 as.numeric(logi_vect)
```

```
[1] 1 0 1
```

Attributi del vettore

Ogni elemento del vettore può essere associato ad un nome.

```
1 names(num_vect) #nessun nome associato
```

NULL

```
1 names(num_vect) = letters[1:4]  
2  
3 num_vect
```

a b c d

1 2 3 4

Ogni vettore è caratterizzato da una dimensione (**dim()**), in realtà essendo il vettore unidimensionale usiamo il comando **length()** per ottenere la lunghezza del vettore

```
1 dim(num_vect)
```

```
NULL
```

```
1 length(num_vect)
```

```
[1] 4
```

Indicizzazione

Possiamo selezionare, eliminare, estrarre elementi semplicemente usando l'indice di posizione tramite le parentesi quadre `vettore[pos]`

```
1 # Creo un vettore formato da 20 numeri casuali pescati da 1 a 100
2 my_vect = round(runif(20,1,100))
3 my_vect
```

```
[1]  3 74 44 12 35 32 90 40 79 78 93 97 25 56 49 98 47 55 59 76
```

```
1 my_vect[1] # estraggo il primo elemento
```

```
[1] 3
```

```
1 my_vect[1:5] # estraggo i primi 5 elementi
```

```
[1]  3 74 44 12 35
```

```
1 my_vect[c(1,4,2,9)] # estraggo elementi a scelta
```

```
[1]  3 12 74 79
```

```
1 my_vect[length(my_vect)] #ultimo elemento (perchè?)
```

```
[1] 76
```

Inidicizzazione negativa

Allo stesso modo possiamo decidere di estrarre tutti gli elementi del vettore eccetto alcuni

```
1 my_vect[-c(1)] #tutti tranne il primo elemento
```

```
[1] 74 44 12 35 32 90 40 79 78 93 97 25 56 49 98 47 55 59 76
```

```
1 my_vect[-c(1:10)] #tutti tranne i primi 10
```

```
[1] 93 97 25 56 49 98 47 55 59 76
```

Se assegniamo nomi agli elementi del vettore, possiamo indicizzare questi “chiamandoli per nome” (meno comune)

```
1 names(my_vect) = letters[1:length(my_vect)]  
2  
3 names(my_vect)
```

```
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r"  
"s"  
[20] "t"
```

```
1 my_vect
```

```
a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t  
3 74 44 12 35 32 90 40 79 78 93 97 25 56 49 98 47 55 59 76
```

```
1 my_vect["a"]
```

```
a  
3
```

```
1 my_vect[1]
```

```
a  
3
```

Indicizzazione Logica

Indicizzare con la posizione è l'aspetto più semplice e intuitivo. E' possibile anche selezionare tramite valori **TRUE** e **FALSE**: possiamo estrarre elementi dal vettore basandoci su specifiche condizioni logiche

```
1 numeri = 1:7  
2 numeri
```

```
[1] 1 2 3 4 5 6 7
```

```
1 numeri[numeri<2]
```

```
[1] 1
```

L'idea è che se abbiamo un vettore di lunghezza **n** e un'altro vettore logico di lunghezza **n**, tutti gli elementi **TRUE** saranno selezionati:

```
1 numeri<2 # vettore logico
```

```
[1] TRUE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
1 # verrà quindi estratto il primo elemento  
2 # perchè è l'unico che risulta TRUE  
3 numeri[numeri<2]
```

```
[1] 1
```

```
1 # non vale solo per elementi del vettore numeri ma per qualsiasi vettore  
2 lettere = letters[1:6]  
3 lettere
```

```
[1] "a" "b" "c" "d" "e" "f"
```

```
1 lettere[numeri<2]
```

```
[1] "a"
```


Indicizzazione Interna

Attraverso la funzione `which()` possiamo ottenere la posizione associata ad una selezione logica:

```
1 lettere
```

```
[1] "a" "b" "c" "d" "e" "f"
```

```
1 which(lettere == "a")
```

```
[1] 1
```

```
1 my_vect
```

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
3	74	44	12	35	32	90	40	79	78	93	97	25	56	49	98	47	55	59	76

```
1 which(my_vect > 20)
```

b	c	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
2	3	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

```
1 my_vect[which(my_vect > 20)] # Cosa cambia tra questi 2 comandi?
```

b	c	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
74	44	35	32	90	40	79	78	93	97	25	56	49	98	47	55	59	76

Indicizzazione e assegnazione

Indicizzare può essere utile anche per assegnare un valore ad un determinato elemento del vettore

```
1 lettere
```

```
[1] "a" "b" "c" "d" "e" "f"
```

```
1 lettere[1] = "z"
```

```
2 lettere
```

```
[1] "z" "b" "c" "d" "e" "f"
```

```
1 lettere[9] = "?"
```

```
2 lettere
```

```
[1] "z" "b" "c" "d" "e" "f" NA NA  "?"
```

```
1 # Possiamo anche modificare l'elemento
```

```
2 lettere[2] = paste(lettere[2], sep = "", "o")
```

```
3 lettere
```

```
[1] "z"  "bo" "c"  "d"  "e"  "f"  NA   NA   "?"
```

Operazioni sui vettori

Possiamo eseguire operazioni sui vettori, ed applicare la stessa operazione a tutti gli elementi del vettore (element-wise)

```
1 new_vect = rep(2:4, each = 2)
2 new_vect
```

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

```
1 # potete svolgere qualsiasi operazione
2 new_vect/2
```

```
[1] 1.0 1.0 1.5 1.5 2.0 2.0
```

```
1 log(new_vect)
```

```
[1] 0.6931472 0.6931472 1.0986123 1.0986123 1.3862944 1.3862944
```

```
1 exp(new_vect)
```

```
[1] 7.389056 7.389056 20.085537 20.085537 54.598150 54.598150
```

Summary Statistics

Le operazioni più comuni sono per esempio la media `mean()`, la deviazione standard `sd()`, la mediana `median()`, il valore massimo `max()` e minimo `min()`.

```
1 # creo un vettore campionando 1000 da una distribuzione normale
2 new_vect = rnorm(1000, mean = 1, sd = 4)
3
4 mean(new_vect)
```

```
[1] 1.041632
```

```
1 sd(new_vect)
```

```
[1] 3.714709
```

```
1 median(new_vect)
```

```
[1] 1.208813
```

```
1 summary(new_vect)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-12.139	-1.469	1.209	1.042	3.500	11.596

Valori NA

A volte (molto spesso) può capitare di avere dati mancanti

```
1 new_vect[1] = NA # Assegnamo NA al primo elemento del vettore
2
3 mean(new_vect) # è impossibile calcolare la media!
```

```
[1] NA
```

```
1 mean(new_vect, na.rm = T) # soluzione!
```

```
[1] 1.050141
```

Frequenze

Un'altra operazione comune che viene svolta sui vettori (e sulle altre strutture dati), è il conteggio delle frequenze

```
1 # riprendiamo il vettore creato all'inizio della lezione
2 char_vect
```

```
[1] "R"  "R"  "R"  "ok"
```

```
1 # attraverso la funzione table possiamo calcolare la frequenza
2 table(char_vect)
```

```
char_vect
```

```
ok  R
```

```
1  3
```

```
1 # ricordatevi che R è case-sensitive!
2 char_vect[1] = "r";char_vect
```

```
[1] "r"  "R"  "R"  "ok"
```

```
1 table(char_vect)
```

```
char_vect
```

```
ok  r  R
```

```
1  1  2
```

Facciamo un po' di pratica!

Aprirete e tenete aperto questo link:

<https://etherpad.wikimedia.org/p/arca-corsoR>

