



UNIVERSITÀ  
DEGLI STUDI DELLA  
**TUSCIA**

# INFORMATICA

Linguaggio R  
Variabile e set istruzioni

*Dott. Franco Liberati*  
*franco.liberati@unitus.it*

# LINGUAGGIO R

## Argomenti del corso





# LINGUAGGIO R

## La nozione di Oggetto-R

- ❑ L'entità atomica che il Linguaggio R crea e manipola è definita **oggetto** (*R-object*).
- ❑ Per oggetto si intende un contenitore (una porzione di memoria) nel quale è stipata una variabile, un vettore di numeri, una stringa di caratteri, una funzione o strutture più complesse costruite da tali componenti

# LINGUAGGIO R

## Definizione di un Oggetto-R

- ❑ Ogni oggetto-R è definito da una **etichetta**, ovvero un identificatore cioè una stringa alfanumerica che deve iniziare con un carattere (mai una cifra) e può sfruttare il carattere underscore (`_`) e il punto (`.`); dopo il punto non può esserci un numero

### Oggetto-R variabile

- `x`
- `valore_intero`
- `valore_reale`

### Oggetto-R funzione

- ```
CALCOLO <- function(a,b,c)
{
    return((a*b)+c)
}
```



# LINGUAGGIO R

## I commenti in R

- ❑ Un commento (su una singola riga) si realizza antepo-  
nendo il simbolo #

# Questo è un commento



**Oggetto-R: variabile**



# LINGUAGGIO R

## Variabile: generalità

- ❑ Il più semplice R-object è la **variabile**
- ❑ Una variabile è un vettore di elementi dello stesso tipo
- ❑ Una variabile è rappresentata da un identificatore

# LINGUAGGIO R

## Variabile: vettore atomico



- ❑ Un vettore di un solo elemento è detto **vettore atomico**
- ❑ Un vettore atomico corrisponde ad una **cella di memoria** nel quale è contenuto un solo operando
- ❑ Per definire una vettore atomico è sufficiente esprimere l'identificatore
- ❑ La sintassi è:

### IDENTIFICATORE

non è necessario mettere il ; alla fine

Nelle successive slide con il termine variabile si intende il vettore atomico  
Inoltre sarà sempre usato il ;

### Oggetto-R variabile: esempi

Definizione di una vettore atomico (che preserva un operando di tipo numerico):  
**valore\_reale**

Definizione di una vettore atomico (che preserva un operando di tipo Intero)  
**valore\_intero**

Definizione di una vettore atomico (che preserva un carattere)  
**carattere**

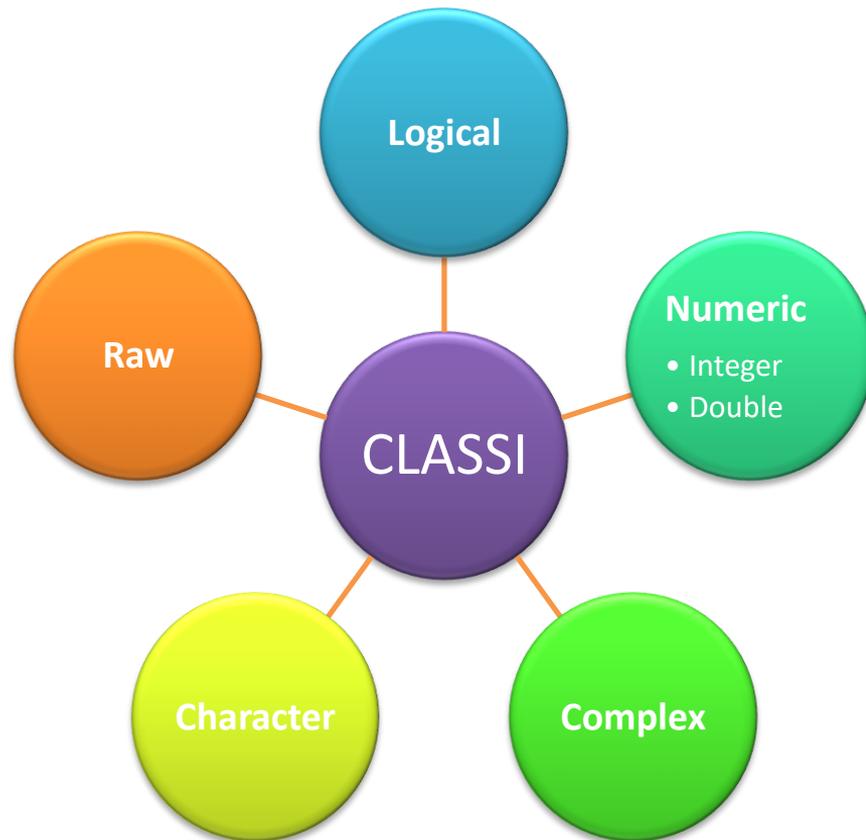
Definizione di una vettore atomico (che preserva un valore logico)  
**booleano**

# LINGUAGGIO R

## Variabile: generalità

☐ Sulla variabile si possono usare sei tipologie di dati, anche dette **classi del vettore atomico**

- ☐ **Logical** è il tipo booleano che accetta due soli valori: true e false
- ☐ **Numeric**
  - ☐ **Double** è il tipo reale (si riporta il numero con il punto tra la parte intera e frazionata)
  - ☐ **Integer** è il tipo intero (si riporta il numero seguito dalla lettera **L**)
- ☐ **Complex** è il tipo impiegato per definire un numero complesso formato da una parte reale e una immaginaria es:  $3+5i$
- ☐ **Character** è il tipo per rappresentare un carattere o una stringa (il carattere è racchiuso tra singoli apici `' '` la stringa tra doppi apici `" "`)
- ☐ **Raw** è il tipo stringa archiviata come interi ai quali ogni numero corrisponde il valore in ASCII



# LINGUAGGIO R

## Variabile: vettore atomico - inizializzazione

- ❑ Per inizializzare una variabile si usa la seguente sintassi:

**IDENTIFICATORE**<-*valore*  
(inizializzazione a sinistra)

*oppure*

**IDENTIFICATORE**=*valore*

*oppure*

*valore* ->**IDENTIFICATORE**  
(inizializzazione a destra)

- ❑ Si può usare l'inizializzazione a destra e a sinistra applicandola a due variabili

### Oggetto-R variabile

Definizione di una variabile che preserva operandi di tipo numerico:

```
valore_reale1=12.74
valore_reale2<-22.11
56.99 ->valore_reale3
valore_4<-66.6->valore_5
```

| INDIRIZZO | VALORE |
|-----------|--------|
| ...       |        |
| 2000000   | 12.74  |
| 2000001   | 22.11  |
| 2000002   | 56.99  |
| 2000003   | 66.6   |
| 2000004   | 66.6   |
| ...       |        |

MEMORIA CENTRALE

# LINGUAGGIO R

## Variabile: vettore atomico – inizializzazione (esempio)

### Oggetto-R variabile

Definizione di una variabile che preserva un operando di tipo numerico reale:

```
valore_reale1=12.74;
```

Definizione di una variabile che preserva un operando di tipo numerico intero:

```
valore_intero=23L;
```

Definizione di una variabile che preserva un operando di tipo logical:

```
valore_booleano=TRUE;
```

Definizione di una variabile che preserva un operando di tipo complesso:

```
valore_complesso=5+3i;
```

Definizione di una variabile che preserva una stringa:

```
stringa="Benson";
```

| INDIRIZZO | VALORE |
|-----------|--------|
| ...       |        |
| 2000000   | 12.74  |
| 2000001   | 23     |
| 2000002   | TRUE   |
| 2000003   | 5      |
| 2000004   | +3i    |
| 2000005   | B      |
| 2000006   | e      |
| 2000007   | n      |
| 2000008   | s      |
| 2000009   | o      |
| 2000010   | n      |
| 2000011   | \0     |
| ...       |        |

MEMORIA CENTRALE

# LINGUAGGIO R

## Variabile: valore corrente e tipo

❑ Durante l'esecuzione di un programma ciascuna variabile ha un **valore corrente** che può mutare il tipo del vettore atomico

❑ Per **conoscere il tipo** si può usare la funzione **typeof** che ha sintassi:

**typeof**(variabile)

### ESEMPIO

```
real1=1.22;  
int1=23L;  
real2=int1*valore_real1;
```

| Tempo        | real1 | int1 | real2 |
|--------------|-------|------|-------|
| $\Delta t_0$ | 1.22  | -    | -     |
| $\Delta t_1$ | 1.22  | 23   |       |
| $\Delta t_2$ | 1.22  | 23   | 28.06 |

### ESEMPIO

```
real1=1.22;  
print(typeof(real1)); // stampa double  
int1=23L;  
print(typeof(int1)); // stampa integer  
real2=valore_int*valore_real1;  
print(typeof(real2)); // stampa double
```

# LINGUAGGIO R

## Variabile conversione di tipo

- ❑ Se si vuole **modificare** il tipo di un **operando intero in uno reale** è necessario sfruttare la funzione

**`as.double(int_var)`**

- ❑ Se si vuole **modificare** il tipo di un **operando reale in intero** (con un troncamento del valore reale in intero) è necessario sfruttare la funzione:

**`as.integer(dbl_car)`**

### ESEMPIO

```
#Definizione di una variabile intera
int_var <- 6L;
#Funzione di conversione da intero a double. Si può
#ottenere lo stesso risultato con as.numeric(int_var)
d_var<-as.double(int_var);
print(d_var)
[1] 6
```

| Tempo        | int_var | d_var |
|--------------|---------|-------|
| $\Delta t_0$ | 6       | -     |
| $\Delta t_1$ | 6       | 6.0   |

### ESEMPIO

```
#Definizione di una variabile double
dbl_var <- 4.9
#Funzione di conversione da double a intero (con troncamento)
i_var <-as.integer(dbl_var)
print(i_var)
[1] 4
```



**Set Istruzioni**

# LINGUAGGIO R

## Set istruzioni: interazione con dispositivi di I/O

- ❑ Per **inserire o visualizzare dati sfruttando i dispositivi** di ingresso (tastiera) e riproduzione (terminale) si può ricorrere alle funzioni di interazione

| Operatore                   | Descrizione                                                                                                                                                                                 | Esempio                                                                                                                           | Esecuzione                                                  |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| <code>print(object)</code>  | Stampa l'oggetto specificato (con un indice per evidenziare i vari elementi)                                                                                                                | <pre>variabile&lt;-9<br/>print(variabile)</pre>                                                                                   | <pre>[1] 9</pre>                                            |
| <code>cat("",object)</code> | Stampa di stringhe e di un oggetto (non si riporta l'indice)                                                                                                                                | <pre>cat("Inserire un numero:")<br/>input3&lt;-readline()<br/>int_input3=as.integer(input3)<br/>cat("Num. ins:",int_input3)</pre> | <pre>Inserisci un numero:<br/>5<br/><br/>Num. ins.: 5</pre> |
| <code>scan()</code>         | Lettura di una serie di NUMERIC e immissione in una variabile, un vettore o in una lista (la terminazione avviene con la pressione del tasto <i>Enter</i> senza l'immissione di alcun dato) | <pre>input1&lt;-scan()<br/>print(input1)</pre>                                                                                    | <pre>4<br/>↵<br/>[1] 4</pre>                                |
| <code>readline()</code>     | Lettura di dati da tastiera (intervallati da uno spazio) e immissione in un vettore. Tutti i dati sono convertiti in un'unica stringa                                                       | <pre>input2&lt;-readline()<br/>print(input2)</pre>                                                                                | <pre>4 pippo 7.8 5-3i<br/>[1] "4 pippo 7.8 5-3i"</pre>      |

# LINGUAGGIO R

## Variabile: operazioni matematiche

- ❑ Le istruzioni aritmetiche operano sui singoli operandi nel caso di vettori atomici

| Simbolo | Significato            | Esempio                                                                  |
|---------|------------------------|--------------------------------------------------------------------------|
| +       | <b>Addizione</b>       | <pre>var1 &lt;- 3; var2 &lt;- 5; var3 = var1+var2 #Il valore è 8</pre>   |
| -       | <b>Sottrazione</b>     | <pre>var1 &lt;- 3; var2 &lt;- 5; var3 = var1-var2 #Il valore è -2</pre>  |
| *       | <b>Moltiplicazione</b> | <pre>var1 &lt;- 3; var2 &lt;- 5; var3 = var1*var2 #Il valore è 15</pre>  |
| /       | <b>Divisione</b>       | <pre>var1 &lt;- 3; var2 &lt;- 5; var3 = var1/var2 #Il valore è 0.6</pre> |

# LINGUAGGIO R

## Variabile: operazioni matematiche

- ❑ Le **istruzioni aritmetiche** operano sui singoli operandi nel caso di vettori atomici

| Simbolo   | Significato                 | Esempio                                                                  |
|-----------|-----------------------------|--------------------------------------------------------------------------|
| <b>%%</b> | <b>Quoziente</b>            | <pre>var1 &lt;- 3; var2 &lt;- 5; var3 = var1%%var2 #Il valore è 0</pre>  |
| <b>%%</b> | <b>Resto</b>                | <pre>var1 &lt;- 3; var2 &lt;- 5; var3 = var1%%var2 #Il valore è 3</pre>  |
| <b>^</b>  | <b>Elevamento a potenza</b> | <pre>var1 &lt;- 3; var2 &lt;- 5; var3 = var1^var2 #Il valore è 243</pre> |

# LINGUAGGIO R

## Variabile: operazioni logiche

- Le **istruzioni logiche** sono applicabili solo a variabili di tipo logico, numerico o complesso. Tutti i numeri diversi da 0 sono considerati come valore logico TRUE

| Simbolo               | Significato | Esempio                                                                 |                                                                        |
|-----------------------|-------------|-------------------------------------------------------------------------|------------------------------------------------------------------------|
| <b>&amp;</b>          | <b>AND</b>  | var1 <- 3;<br>var2 <- 5;<br>var3 = var1&var2<br>#Il valore è TRUE       | var1 <- 0;<br>var2 <- 5;<br>var3 = var1&var2<br>#Il valore è FALSE     |
| <b> </b>              | <b>OR</b>   | var1 <- 3;<br>var2 <- 5;<br>var3 = var1 var2<br>#Il valore è TRUE       | var1 <- 0;<br>var2 <- 5;<br>var3 = var1 var2<br>#Il valore è TRUE      |
| <b>!</b>              | <b>NOT</b>  | var1 <- 3;<br>var3 = !var1<br>#Il valore è FALSE                        | var1 <- 0;<br>var3 = !var1<br>#Il valore è TRUE                        |
| <b>xor(var1,var2)</b> | <b>XOR</b>  | var1 <- 3;<br>var2 <- 5;<br>var3 = xor(var1,var2)<br>#Il valore è FALSE | var1 <- 0;<br>var2 <- 5;<br>var3 = xor(var1,var2)<br>#Il valore è TRUE |

# LINGUAGGIO R

## Variabile: istruzioni di confronto

- ❑ Le **istruzioni di confronto** operano analizzando gli operandi delle variabili
- ❑ Il confronto **non è ammesso tra numeri complessi**

| Simbolo      | Significato              | Esempio                                                                       |
|--------------|--------------------------|-------------------------------------------------------------------------------|
| <b>==</b>    | <b>Uguale</b>            | <pre>var1 &lt;- 3; var2 &lt;- 5; print(var1==var2) #Il valore è FALSE</pre>   |
| <b>!=</b>    | <b>Diverso</b>           | <pre>var1 &lt;- 3; var2 &lt;- 5; print (var1!=var2) #Il valore è TRUE</pre>   |
| <b>&gt;</b>  | <b>Maggiore</b>          | <pre>var1 &lt;- 3; var2 &lt;- 5; print(var1&gt;var2) #Il valore è FALSE</pre> |
| <b>&gt;=</b> | <b>Maggiore o uguale</b> | <pre>var1 &lt;- 5; var2 &lt;- 5; print(var1&gt;=var2) #Il valore è TRUE</pre> |

# LINGUAGGIO R

## Variabile: istruzioni di confronto

- ❑ Le **istruzioni di confronto** operano analizzando gli operandi delle variabili
- ❑ Il confronto **non è ammesso tra numeri complessi**

| Simbolo | Significato          | Esempio                                                                        |
|---------|----------------------|--------------------------------------------------------------------------------|
| <       | <b>Minore</b>        | <pre>var1 &lt;- 3; var2 &lt;- 5; print(var1&lt;var2) #Il valore è TRUE</pre>   |
| <=      | <b>Minore uguale</b> | <pre>var1 &lt;- 3; var2 &lt;- 5; print (var1&lt;=var2) #Il valore è TRUE</pre> |



# LINGUAGGIO R

## Esempio: problema da risolvere

Calcolare il minimo comune multiplo(MCM) di due numeri interi positivi. Dati due numeri interi  $N1$  e  $N2$ , il minimo comune multiplo è il più piccolo numero  $M$  che è divisibile (con resto pari a zero) sia per  $N1$  che per  $N2$ .

Suggerimento. Si considerino due numeri interi  $N1$  e  $N2$ . Sia  $N1$  più grande di  $N2$ . Il MCM è il primo multiplo di  $N1$  che è divisibile per  $N2$ .

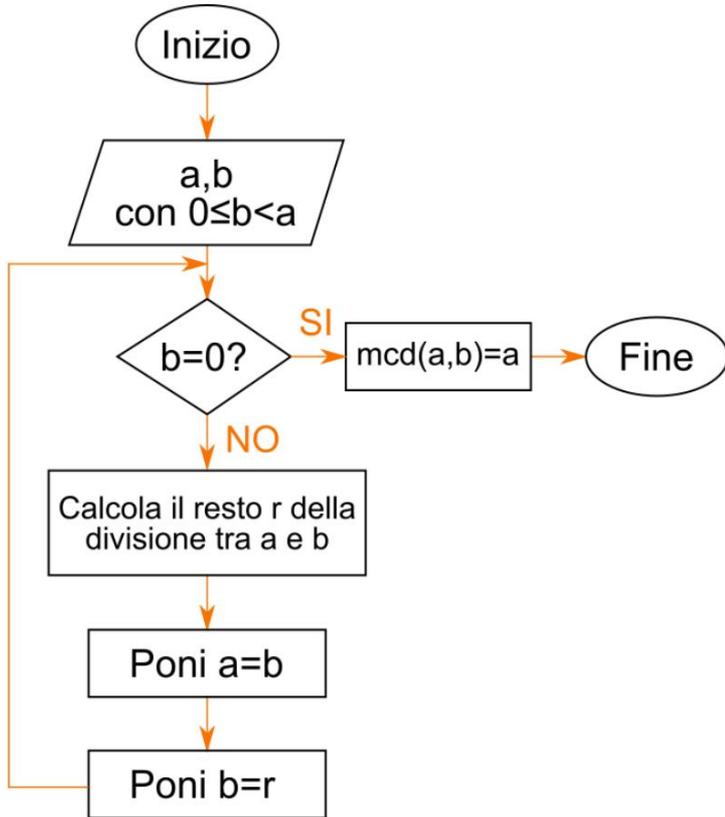
Suggerimento 2. usare il teorema di euclide per il calcolo del MCD



# LINGUAGGIO R

## Esempio: algoritmo

CALCOLO MASSIMO COMUNE DIVISORE (ALGORITMO DI EUCLIDE)



$$\text{mcm}(a,b) = a * b / \text{MCD}(a,b)$$



# LINGUAGGIO R

## Esempio: soluzione

```
#Calcolo del minimo comune multiplo
val1<-as.integer(readline(prompt="Inserisci il primo valore: "));
val2<-as.integer(readline(prompt="Inserisci il secondo valore: "));
a<-val1;
b<-val2;
if (a<b) {
    t<-a;
    a<-b;
    b<-t;
}
while(b!=0)
{
    r<-a%%b;
    a<-b;
    b<-r;
}
MCD<-a;
mcm<-(val1*val2)%/%MCD;
cat("\nIl minimo comune multiplo tra ",val1," e ",val2," è: ",mcm)
```



Fine