



UNIVERSITÀ  
DEGLI STUDI DELLA  
**TUSCIA**

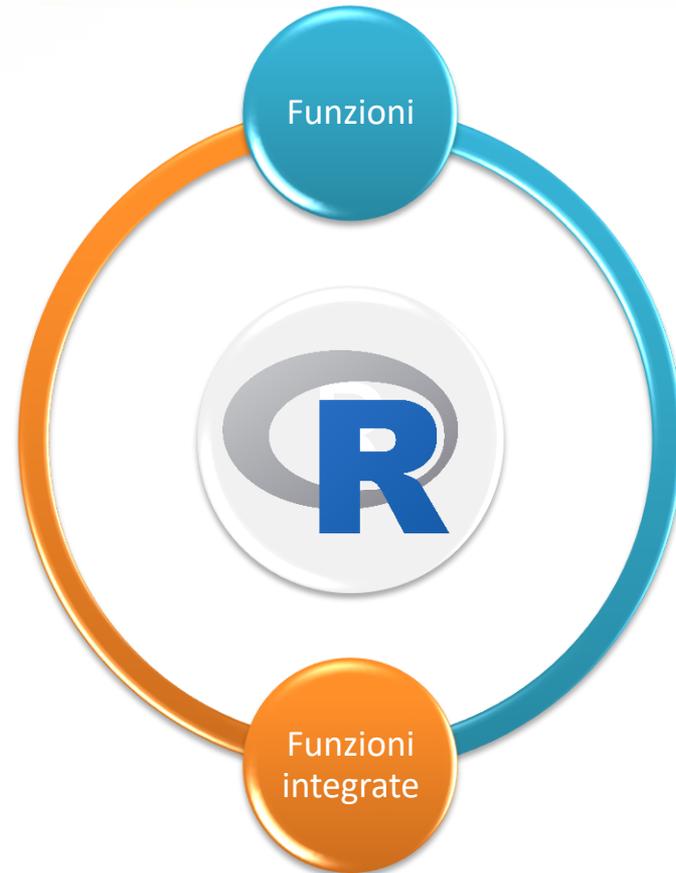
# PROGRAMMAZIONE

Linguaggio R  
Funzione

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

# LINGUAGGIO R

## Argomenti del corso





# LINGUAGGIO R

## Funzione: generalità

- ❑ La maggior parte dei programmi che risolvono problemi reali hanno dimensioni molto grandi. L'esperienza ha dimostrato che il modo migliore per sviluppare e mantenere un programma con molte istruzioni è quello di suddividerlo in parti più piccole, ognuna delle quali risulta più maneggevole e facilmente testabile (tecnica del 'dividi e conquista' - dal latino *divide et impera*)
- ❑ Una strategia per facilitare la progettazione, l'implementazione, l'utilizzo e la manutenzione di programmi di grandi dimensioni mediante suddivisione del problema è quella di sfruttare la tecnica del partizionamento di blocchi di istruzioni riutilizzabili, ovvero la **programmazione organizzata a funzione**



# LINGUAGGIO R

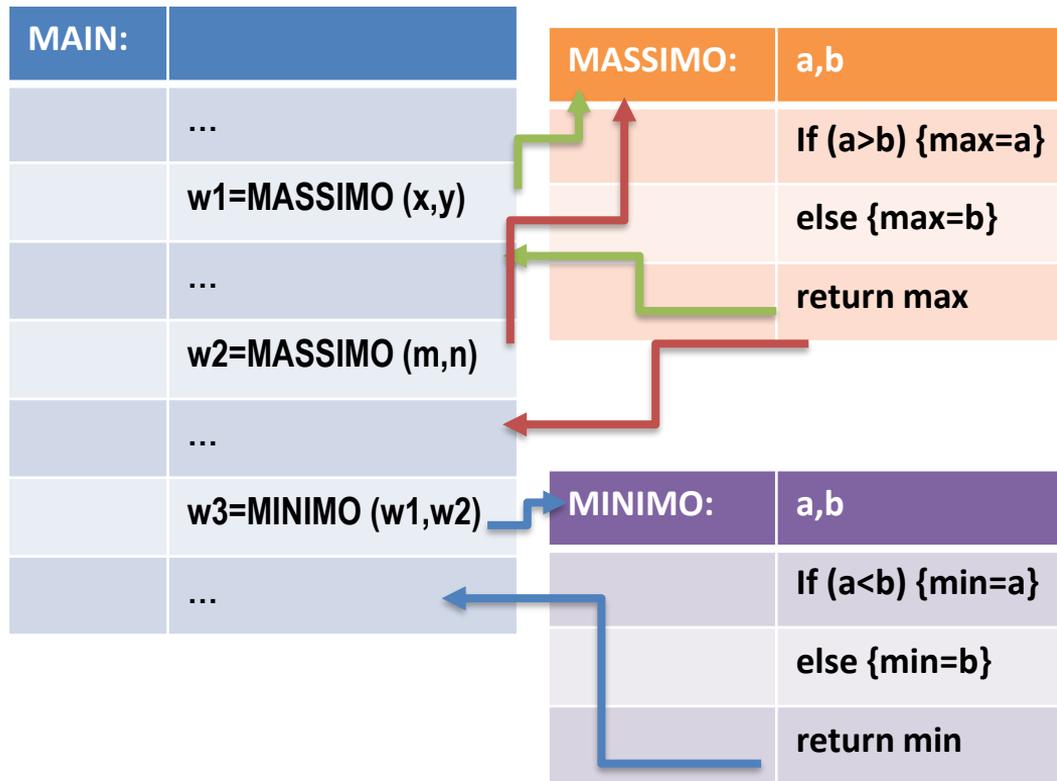
## Funzione: generalità

- ❑ Una **funzione** (sottoprogramma o subroutine) è un blocco di istruzioni che svolgono una determinata operazione, spesso con una valenza universale (es.: il minimo comune multiplo tra due numeri)
- ❑ Vantaggi:
  - ❑ **replicabilità**: si può usare la funzione più volte all'interno del codice
  - ❑ **riuso**: si possono usare funzioni già definite per definirne di nuove
  - ❑ **condivisione**: si possono usare funzioni già definite da altri programmatori.  
In R se una funzione è stata definita e convalidata dal gruppo di lavoro *Comprehensive R Archive Network* si chiama **FUNZIONE INTEGRATA**  
Un insieme di funzioni integrate realizzano un **pacchetto** (*package*)

# LINGUAGGIO R

## Funzione: generalità

- L'impiego di una funzione avviene mediante l'istruzione di **salto a subroutine** (o salto a sottoprogramma o chiamata di funzione) che permette di passare da una parte del **programma principale** (o main) a un **sottoprogramma**, di eseguirlo e di tornare alla istruzione successiva a quella del salto (**si salva il valore del program counter dopo la chiamata del sottoprogramma**)
- L'interprete R, quando incontra l'identificatore del nome della funzione (e gli eventuali argomenti), richiama la subroutine (in realtà la riscrive). La funzione a sua volta svolge il suo compito e restituisce il controllo all'interprete insieme al risultato





# LINGUAGGIO R

## Funzione in R

- ❑ Nel Linguaggio R una funzione è un **oggetto primitivo**, al pari dei vettori
- ❑ Pertanto è possibile assegnare funzioni a variabili; passare come argomento ad una funzione un'altra funzione; e restituire una funzione come risultato
- ❑ Questa libertà fa di **R un linguaggio funzionale**

# LINGUAGGIO R

## Funzione in R: definizione

- ❑ Per definire una funzione in Linguaggio R, nella sua forma canonica, si usa il costrutto **function** seguito da una collezione di argomenti (dei parametri che possono essere passati alla funzione) racchiusi tra parentesi tonde
- ❑ Seguono un insieme di istruzioni, il **corpo della funzione**, all'interno di parentesi graffe
- ❑ L'identificazione della funzione avviene mediante il **nome della funzione**
- ❑ La restituzione di quanto elaborato si ha con l'ultima istruzione prima della graffa di chiusura o usando il costrutto **return**

### SINTASSI

```
nome_funzione <- function(arg_1, arg_2, ...) {  
  Corpo della funzione  
  return(risultato)  
}
```

### ESEMPIO

```
massimo <- function(x,y) {  
  if (x>y){max=x;}  
  else{max=y;}  
  return(max);  
}
```

# LINGUAGGIO R

## Funzione in R: nome e argomenti

- ❑ Il **nome della funzione** è una etichetta, è archiviata nell'ambiente R e ci si riferisce quando va richiamata per usare la subroutine (anche in questo caso è context sensitive)
- ❑ Un **argomento** è uno (o più) vettore atomico o una struttura dati che può essere passata alla funzione. Gli argomenti di una chiamata di funzione sono forniti nella stessa sequenza in accordo alla definizione oppure hanno una sequenza diversa ma, in questo caso, è richiesta l'assegnazione dei nomi degli argomenti

### ESEMPIO

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

```
ris1<-CALCOLO (2,3,4)  
#ris1 ha valore 10
```

```
ris2<-CALCOLO (b=2,c=3,a=4)  
# ris2 ha valore 11
```

# LINGUAGGIO R

## Funzione in R: argomenti

- ❑ Gli argomenti possono essere anche inizializzati Il valore di inizializzazione rimane tale se non è modificato quando si richiama la funzione

- ❑ Gli argomenti sono facoltativi; cioè, una funzione può non contenere argomenti

### ESEMPIO

```
PRODOTTO_SEMPlice<- function(a=5,b=4) {  
  return(a*b)  
}  
ris1<- PRODOTTO_SEMPlice ()  
# il valore di ris1 è 20
```

```
ris2<-PRODOTTO_SEMPlice(b=2, a=4)  
#Il valore di ris2 è 8
```

### ESEMPIO

```
MESSAGGIO_ERRORE <- function() {  
  return("ERRORE NELL'ALGORITMO")  
}  
ris1<-MESSAGGIO_ERRORE();  
print(ris1)  
#Mostra a video "ERRORE NELL'ALGORITMO"
```

# LINGUAGGIO R

## Funzione in R: Corpo e ritorno

- ❑ Il **corpo della funzione** ha una insieme di istruzioni che definiscono le operazioni da svolgere; esso è contenuto tra parentesi graffe. Le variabili usate al suo interno sono da considerarsi come **variabili locali**, cioè **cessano di esistere una volta terminata la funzione** (e quindi, dopo la chiamata a subroutine, non sono visibili né riutilizzabili né stampabili)
- ❑ Il valore elaborato da una funzione può essere ottenuto con il costrutto **return(oggetto)**, si parla di ritorno esplicito

### ESEMPIO

```
CONTROLLO<- function(x) {  
  if (x > 0) {result <- "Positivo"}  
  else if (x < 0) {result <- "Negativo"}  
  else {result <- "Zero"}  
  return(result)  
}
```

```
#Lettura da tastiera di un valore e sua conversione in numero  
valore<-readline(prompt = "\nInserire un numero: ")  
valore<-as.integer(valore)
```

```
#Richiamo della funzione e copia del risultato nell'oggetto ris  
ris<- CONTROLLO (valore)
```

```
#stampa del risultato ottenuta dalla funzione  
print(ris)
```



# LINGUAGGIO R

## Funzione in R: ritorno

- ❑ Il costrutto **return**(oggetto) può restituire solo un singolo oggetto.
- ❑ Se c'è la necessità di restituire più valori si possono sfruttare strutture dati come il vettore, la lista o altri oggetti

### ESEMPIO

```
RITORNI_MULTIPLI <- function(x) {  
  quadrato<-x*x  
  cubo<-x*x*x  
  risultati<-c(quadrato,cubo)  
  return(risultati)  
}
```

```
#Lettura da tastiera di un valore e sua conversione in numero  
valore<-readline(prompt = "\nInserire un numero: ")  
valore<-as.integer(valore)
```

```
#Richiamo della funzione e copia del risultato nell'oggetto ris  
ris<- RITORNI_MULTIPLI(valore)
```

```
#Stampa del risultato ottenuta dalla funzione (con input 3)  
print(ris)  
[1] 9 27
```



# LINGUAGGIO R

## Funzione in R: esempio

Il riuso di una subroutine si può mostrare in un programma che risolve il massimo di tre numeri (Val1,Val2,Val3)

```
MIO_MAX<-function(a,b){  
if (a>b) {max=a;}  
else{max=b;}  
return(max);  
}  
val1<-readline(prompt="\nInserire il primo valore: ")  
val2<-readline(prompt="\nInserire il secondo valore: ")  
val3<-readline(prompt="\nInserire il terzo valore: ")  
val1<-as.integer(val1)  
val2<-as.integer(val2)  
val3<-as.integer(val3)  
massimo<-MIO_MAX(MIO_MAX(val1,val2),val3)  
print(massimo)
```



**Principali funzioni integrate**



# LINGUAGGIO R

## Funzioni integrate

- ❑ Le subroutine che sono già state create o definite nel framework di programmazione R sono note come **funzioni integrate** (*built-in function*) e appartengono al pacchetto base
- ❑ Il Linguaggio R è dotato di un ricco insieme di funzioni integrate. Queste subroutine sono suddivise in tre categorie differenziate in relazione alla loro funzionalità e operano non solo sul singolo operando ma anche su strutture dati più complesse come vettori, liste e altri oggetti: MATEMATICA-STRINGHE-PROBABILITÀ E STATISTICA

# LINGUAGGIO R

## Funzione integrate: MATEMATICHE

| Funzione                  | Significato                                                                                                        | Esempio                                                      |
|---------------------------|--------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| <b>abs(x)</b>             | Valore assoluto degli operandi dell'oggetto x                                                                      | <pre>x&lt;- -4 print(abs(x)) [1] 4</pre>                     |
| <b>sqrt(x)</b>            | Radice quadrata degli operandi dell'oggetto x                                                                      | <pre>x&lt;- 25 print(sqrt(x)) [1] 5</pre>                    |
| <b>ceiling(x)</b>         | Restituisce il più piccolo intero che sia uguale o superiore agli operandi nell'oggetto x (parte intera inferiore) | <pre>x&lt;- 7.5 print(ceiling(x)) [1] 8</pre>                |
| <b>floor(x)</b>           | Restituisce il più grande intero che sia uguale o inferiore agli operandi dell'oggetto x (parte intera inferiore)  | <pre>x&lt;- 2.5 print(floor(x)) [1] 2</pre>                  |
| <b>trunc(x)</b>           | Elimina la parte decimale degli operandi dell'oggetto x                                                            | <pre>x&lt;- c(1.8,3.5,8.9) print(trunc(x)) [1] 1 3 8</pre>   |
| <b>round(x, digits=n)</b> | Arrotonda il valore degli operandi dell'oggetto x a n cifre dopo la virgola                                        | <pre>x&lt;- 7.3456570283 print(round(x,5)) [1] 7.34566</pre> |
| <b>sin(x)</b>             | Calcolo del seno degli operandi dell'oggetto x espresso in radianti                                                | <pre>x&lt;- pi/2.0 print(sin(x)) [1] 1</pre>                 |
| <b>cos()</b>              | Calcolo del coseno degli operandi dell'oggetto x espresso in radianti                                              | <pre>x&lt;- pi/2.0 print(cos(x)) [1] 0</pre>                 |
| <b>tan(x)</b>             | Calcolo della tangente degli operandi dell'oggetto x espresso in radianti                                          | <pre>x&lt;- pi/4.0 print(tan(x)) [1] 1</pre>                 |

# LINGUAGGIO R

## Funzione integrate: MATEMATICHE

| Funzione        | Significato                                                                                        | Esempio                                |
|-----------------|----------------------------------------------------------------------------------------------------|----------------------------------------|
| <b>log(x)</b>   | Calcolo del logaritmo naturale degli operandi dell'oggetto x                                       | x<- 4<br>print(log(x))<br>[1] 1.386294 |
| <b>log2(x)</b>  | Calcolo del logaritmo in base 2 degli operandi dell'oggetto x                                      | x<-16<br>print(log2(x))<br>[1] 4       |
| <b>log10(x)</b> | Calcolo del logaritmo in base 10 degli operandi dell'oggetto x                                     | x<-1000.0<br>print(log10(x))<br>[1] 3  |
| <b>exp(x)</b>   | Calcolo dell'esponente in base e degli operandi dell'oggetto x (in altre parole si calcola $e^x$ ) | x<-5<br>print(exp(x))<br>[1] 148.4132  |

# LINGUAGGIO R

## Funzione integrate: STRINGHE

| Funzione                          | Significato                                                                                                                         | Esempio                                                                                                                                                                                                            |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>substr(x,first,last)</code> | Estrae la sottostringa di <code>x</code> a partire dalla posizione <i>first</i> fino a <i>last</i>                                  | <pre>a &lt;- "987654321" print(substr(a, 3, 6)) 7654</pre>                                                                                                                                                         |
| <code>nchar (x)</code>            | Conta il numero di caratteri (spazi inclusi) negli elementi di un oggetto<br>Se <code>x</code> è un vettore lo fa per ogni elemento | <pre>x&lt;-"Ciao a tutti voi" print(nchar(x)) [1] 16</pre>                                                                                                                                                         |
| <code>strsplit(x, split)</code>   | Suddivide gli elementi di un vettore ( <code>x</code> ) ad ogni carattere specificato in <i>split</i>                               | <pre>a&lt;-"Suddivide ogni parola quando split ha il carattere vuoto" print(strsplit(a, " ")) [1] "Suddivide" [2] "ogni" [3] "parola" [4] "quando" [5] "split" [6] "ha" [7] "il" [8] "carattere" [9] "vuoto"</pre> |
| <code>tolower(x)</code>           | Converte tutti i caratteri in minuscolo                                                                                             | <pre>st1&lt;- "Franco Liberati" print(tolower(st1)) [1] franco liberati</pre>                                                                                                                                      |
| <code>toupper(x)</code>           | Converte tutti i caratteri in maiuscolo                                                                                             | <pre>print(toupper(st1)) [1] FRANCO LIBERATI</pre>                                                                                                                                                                 |

# LINGUAGGIO R

## Funzione integrate: STRINGHE

| Funzione                                                                                        | Significato                                                                                 |
|-------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| <b>grep(pattern,x,ignore.case=FALSE,fixed=FALSE, replacement)</b>                               |                                                                                             |
| <b>Rileva la presenza di una stringa (pattern) o una espressione regolare in un oggetto (x)</b> |                                                                                             |
| <b>pattern</b>                                                                                  | È una stringa o una espressione regolare che deve essere ricercata                          |
| <b>x</b>                                                                                        | È una variabile di tipo carattere (o più in generale un oggetto) che deve essere analizzato |
| <b>ignore.case</b>                                                                              | specifica se la ricerca avviene considerando il case sensitive (FALSE) oppure no (TRUE)     |
| <b>fixed</b>                                                                                    | Se è TRUE il pattern è una stringa                                                          |
| <b>replacement</b>                                                                              | indica un oggetto che sostituisce quanto trovato (non è usato nelle nuove versioni)         |

### ESEMPIO

```
valore<- "bigfoot";  
i <- grep("foo", valore);  
print(i)  
#stampa 1
```

# LINGUAGGIO R

## Funzione integrate: STRINGHE

| Funzione                                                                                                                                                                                                                      | Significato                                                                             |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| <b>sub(pattern, replacement, x, ignore.case=FALSE, fixed=FALSE)</b>                                                                                                                                                           |                                                                                         |
| È una forma semplificata di <b>grep</b> e consente la sostituzione della prima sottostringa ( <b>pattern</b> ) o della prima espressione regolare presente in un oggetto ( <b>x</b> ) con un rimpiazzo ( <b>replacement</b> ) |                                                                                         |
| <b>pattern</b>                                                                                                                                                                                                                | È una stringa o una espressione regolare che deve essere ricercata;                     |
| <b>x</b>                                                                                                                                                                                                                      | È una variabile di tipo carattere (o più in generale un oggetto)                        |
| <b>ignore.case</b>                                                                                                                                                                                                            | specifica se la ricerca avviene considerando il case sensitive (FALSE) oppure no (TRUE) |
| <b>fixed</b>                                                                                                                                                                                                                  | Se è TRUE il pattern è una stringa                                                      |
| <b>replacement</b>                                                                                                                                                                                                            | indica un oggetto che sostituisce quanto trovato (non è usato nelle nuove versioni)     |

### ESEMPIO

```
st1<-"Sopra la panca la capra campa, sotto la panca la capra crepa."  
st2<-sub("la capra","il bue", st1)  
print(st2)  
#Sopra la panca il bue campa, sotto la panca la capra crepa."
```

# LINGUAGGIO R

## ESERCIZIO

Nel 2024 alcuni studenti dell'Università di Viterbo, scoprono un giacimento d'oro. Inizia la febbre dell'oro in tutta Italia.

Generare una coordinata geografica secondo lo standard gg°mm'ss" in maniera casuale su una mappa dell'Italia (quindi con limite: latitudine=[35°29'24"-47°5'3"] e longitudine=[6°37'34"-18°31'10"]) indicante la cava aurea

Indicare la posizione di due cercatori d'oro in coordinate geografiche secondo lo standard gg°mm'ss"

**Determinare quale dei due cercatori è più vicino alla cava aurea.**

Si ricorda che note le coordinate cartografiche di due punti (nello stesso fuso orario) segue il calcolo della distanza reale definita come  $D_r = M_d D$  dove  $M_d$  è il modulo di deformazione lineare pari a 0.9996 e  $D$  è la distanza dei due punti, cioè:

$$D = \sqrt{(lat2 - lat1)^2 + (long2 - long1)^2}$$

### OSS:

Per convertire dalla misura gradi minuti secondi (gg°mm'ss") a gradi decimali bisogna:

- 1) dividere i secondi per 60
- 2) sommare il risultato ai minuti e quindi dividere per 60
- 3) Sommare i gradi



Fine