

---

## Elementi minimi di MATLAB

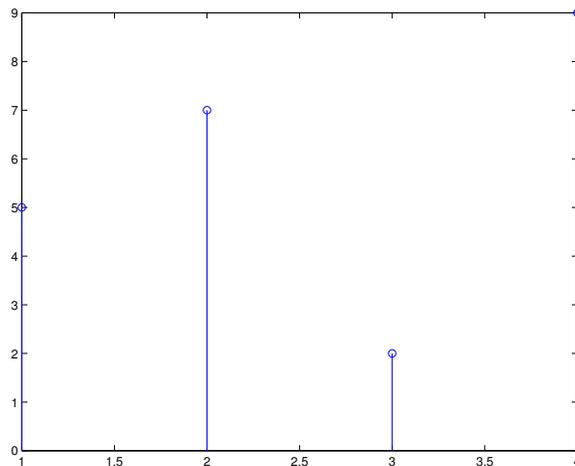
- » `help`  
invoca l'help in linea di MATLAB e restituisce la lista di argomenti sui quali è possibile avere informazioni; è una buona idea usarlo almeno una volta.
- » `help <topic/command>`  
invoca l'help su un particolare argomento o comando, e suggerisce una lista di argomenti o comandi correlati con quello richiesto.
- » `help help`  
spiega come usare l'help e indica altri utili comandi, es `lookfor`.
- » `cd`  
mostra la directory corrente.
- » `dir`  
mostra il contenuto della directory corrente.
- » `cd <nomedir>`  
porta nella nuova directory di lavoro `nomedir`.
- » `x=3;`  
definisce la variabile `x` e vi assegna il valore 3; se si omette il punto-e-virgola, il valore della variabile viene anche visualizzato.
- » `x`  
(senza punto-e-virgola) mostra il valore corrente della variabile `x`.
- » `whos;`  
mostra l'insieme delle variabili definite, il loro tipo, e la loro occupazione di memoria. Da notare che il tipo assegnato per default alle variabile è `double`.
- » `clear x;`  
cancella la variabile `x` dalla memoria; `clear all` cancella tutte le variabili definite.
- » `x=[5, 2, 3];`  
definisce un vettore riga di tre elementi con valori 5,2 e 3; se al posto delle virgole si lasciano degli spazi si ottiene lo stesso risultato.
- » `x=[5; 2; 3];`  
definisce un vettore colonna di tre elementi con valori 5,2 e 3.
- » `z=[x, y];`  
definisce la matrice `z` affiancando le due matrici `x` e `y`, che devono avere lo stesso numero di righe.
- » `z=[x; y];`  
definisce la matrice `z` sovrapponendo le due matrici `x` e `y`, che devono avere lo stesso numero di colonne.

- 
- » `x=zeros(N,M);`  
definisce una matrice di N righe ed M colonne con elementi tutti nulli.
  - » `x=ones(N,M);`  
definisce una matrice di N righe ed M colonne con elementi tutti unitari.
  - » `x=eye(N);`  
definisce una matrice identità quadrata di N righe ed N colonne.
  - » `y=x.'`;  
definisce y come trasposto di x.
  - » `y=x'`;  
definisce y come hermitiano (trasposto e coniugato) di x. Per i vettori reali coincide con il trasposto.
  - » `[M,N]=size(x);`  
restituisce in M ed N il numero di righe e di colonne della matrice x; con `size(x)` tali dimensioni vengono mostrate su schermo.
  - » `L=length(x);`  
restituisce in L la lunghezza del vettore x.
  - » `y=flipr(x);`  
inverte l'ordine delle righe della matrice x; `flipud(x)` opera in modo analogo sulle colonne.
  - » `min(x)`  
se x è un vettore restituisce il minimo di x; se è una matrice calcola i minimi lungo ogni colonna e li restituisce in un vettore riga. Per ottenere il minimo della matrice usare `min(min(x))`.
  - » `max(x)`  
restituisce il massimo, funziona come `min(x)`.
  - » `mean(x)`  
restituisce il valor medio, funziona come `min(x)`.
  - » `std(x)`  
restituisce la deviazione standard, funziona come `min(x)`; si noti però che se x è una matrice `std(std(x))` restituisce la deviazione standard delle deviazioni standard calcolate lungo ogni colonna, che è diverso dalla deviazione standard di tutti gli elementi della matrice.
  - » `y=abs(x);`  
definisce y come la matrice dei valori assoluti di x
  - » `sum(x)`  
calcola la somma degli elementi di x, funziona come `min(x)`;
  - » `y=cumsum(x)`  
restituisce in y la somma cumulativa degli elementi di x, cioè se x è un vettore, allora y(n) è la somma di  $x(1)+\dots+x(n)$ ; di conseguenza x e y hanno le stesse dimensioni; se x è una matrice, la somma cumulativa opera lungo le colonne.

» `stem(x, y)`  
 rappresenta graficamente i valori del vettore `y` come impulsi centrati in corrispondenza dei valori del vettore `x`.

► **Esempio 1 (comando `stem`)** Le seguenti linee di codice producono l'output rappresentato in Fig.1:

```
» x = [1, 2, 3, 4];
» y = [5, 7, 2, 9];
» stem(x, y)
```



**Fig. 1.** `stem(x,y)`

» `plot(x, y)`  
 restituisce il grafico del vettore `y` in corrispondenza del vettore `x`. Se `x` (o `y`) una matrice, allora le righe o le colonne di `x` (o `y`) vengono diagrammate in funzione di `y` (o `x`). È possibile scegliere il tipo di linea, i simboli ed il colore del grafico aggiungendo delle opzioni nell'argomento del comando.

► **Esempio 2 (comando `plot`)** Le seguenti linee di codice producono l'output rappresentato in Fig.2:

```
» x = [1, 2, 3, 4];
» y = [5, 7, 2, 9];
» plot(x, y, '-o m')
```

» `bar(x, y)`  
 restituisce il diagramma a barre del vettore `y` in funzione di `x`. Se `y` una matrice  $M \times N$ , vengono diagrammate le colonne di `y` come `M` gruppi di `N` barre verticali.

► **Esempio 3 (comando `bar`)** Le seguenti linee di codice producono l'output rappresentato in Fig.3:

```
» x = [1, 2, 3, 4];
» y = [[2, 5, 8]; [2, 4, 6]; [1, 9, 7]; [2, 2, 2]];
» bar(x, y)
```

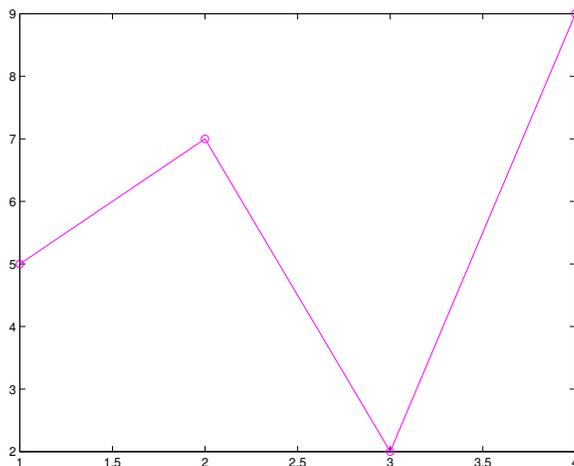


Fig. 2. `plot(x,y,'-o m')`

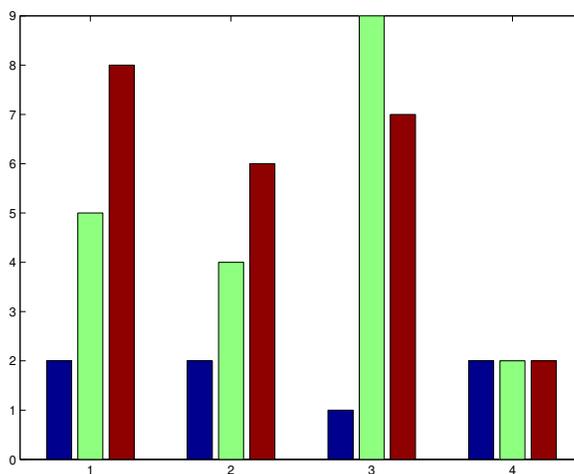


Fig. 3. `bar(x,y)`

» `hist(x)`  
 produce l'istogramma di x.

► **Esempio 4 (comando `hist`)** Le seguenti linee di codice producono l'output rappresentato in Fig.4:

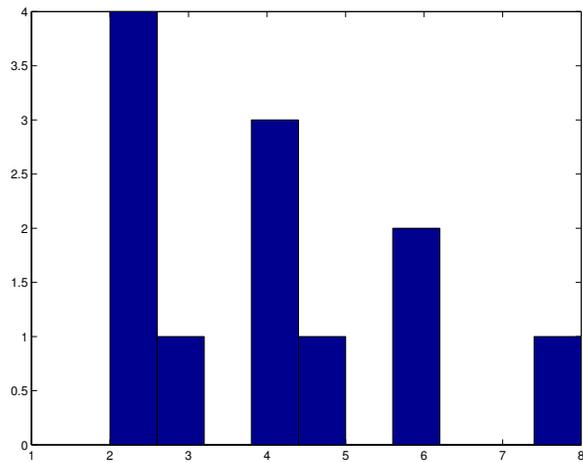
```
» x = [2, 2, 5, 6, 3, 4, 4, 4, 2, 2, 8, 6];
» hist(x)
```



» `subplot`  
 consente di visualizzare pi grafici all'interno della stessa finestra. Il comando `subplot(m,n,p)` suddivide la finestra in  $m \times n$  quadranti e posiziona il grafico corrente nella posizione p-esima.

► **Esempio 5 (comando `subplot`)** Le seguenti linee di codice producono l'output rappresentato in Fig.5:

```
» subplot(2, 2, 1)
» stem(x, y)
```

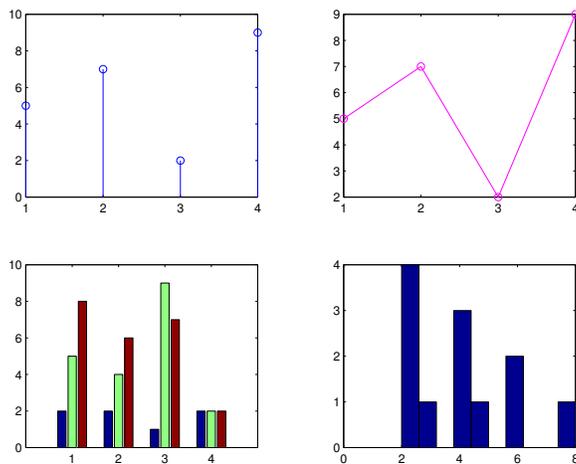


**Fig. 4.** hist(x)

```

» subplot(2,2,2)
» plot(x,y,'-o m')
» subplot(2,2,3)
» bar(x1,y1)
» subplot(2,2,4)
» hist([2,2,5,6,3,4,4,4,2,2,8,6])

```



**Fig. 5.** subplot

- » `axis([xmin xmax ymin ymax])`  
consente di settare la scala degli assi di un grafico tra i valori  $x_{\min}$  e  $x_{\max}$  per l'asse delle ascisse e tra  $y_{\min}$  e  $y_{\max}$  per l'asse delle ordinate.
- » `figure(H)`  
crea una figura alla quale assegnato l'indice H.

» `close`

chiude la figura. Il comando `close(H)` chiude la H-esima figura, mentre `close(all)` chiude tutte le figure aperte.