Laboratori tecnico scientifici

per l'economia, la statistica e la matematica applicata

Advanced level

Corso Matlab :

Seconda lezione (Esercitazione, 27/09/13)

Samuela Persia, Ing. PhD.

Sommario

- Presentazione Programma delle Esercitazioni
- Ogni Esercitazione prevede:
  - Richiami Teoria e Sintassi – Cenni
  - Esercizi

Programma Esercitazioni

- Seconda lezione corso: 27/09/2013
  - Elementi base di Matlab: Array e Matrici
- Terza lezione corso: 04/10/2013
  - File script, Grafici, Acquisizione dati esterni
- Quarta lezione corso: 18/10/2013
  - Come usare i toolbox di matlab
  - Esempi di applicazioni di alcune funzioni selezionate nel toolbox di finanza: «*finance toolbox*»
- Quinta lezione corso: 25/10/2013
  - Esercizio finale in cui vengono utilizzate tutte le nozioni spiegate durante il corso, ed in cui viene affrontato un problema di finanza (**Caso studio**).

Richiami Teoria

- Per lavorare in una sessione di lavoro occorre:
  - Aprire Matlab
  - Scrivere nella sezione “Command Window”
- Tutte le variabili create saranno visibili nella sezione “WorkSpace”
- Una volta chiuso Matlab tutti i dati creati in questo modo **verrano persi (variabili locali)**

Richiami Teoria e Sintassi

- **Logaritmo**
  - **in base naturale:** log()
  - **in base 2:** log2()
  - **in base 10:** log10()

- **Funzioni trigonometriche**
  - **Seno:** sin()
  - **Coseno:** cos()
  - **Tangente:** tan()

Richiami Teoria e Sintassi

- **Array**
  - **Si definisce nei seguenti modi:**
    - **con una coppia di parentesi quadre:** []
    - **linespace(i,f,n):** genera un array di **n** elementi da **i** a **f** con punti equidistanti
    - **logspace(i,f,n):** genera un array di n elementi da 10^i a 10^f con punti logaritmicamente equidistanti
    - **operatore ":"** per selezionare un sottoinsieme di valori

Richiami Teoria e sintassi

- **Matrici**
  - Si definisce con una coppia di parentesi quadre con il ; che separa le righe dalle colonne: [;]

  – **Matrici speciali**

    - **eye(n):** crea una matrice identità nxn
    - **ones(n):** crea una matrice nxn i cui elementi sono uguali a 1
    - **zeros(n):** crea una matrice nxn i cui elementi sono uguali a 0
    - **ones(m,n):** crea una matrice mxn i cui elementi sono uguali a 1
    - **zeros(m,n):** crea una matrice mxn i cui elementi sono uguali a 0

Richiami Teoria e Sintassi

- **Semplici operazioni con Array (1/2)**
  - **max(x)** restituisce il valore più grande di x se x è un vettore; se x è una matrice restituisce un vettore riga i cui elementi corrispondono ai valori più grandi delle corrispondenti colonne di x
  - **min(x)** equivalente a max(x) con la differenza che restituisce i valori più bassi
  - **ndims(x)** restituisce il numero di dimensioni di un array
  - **mean(x)** fornisce la media dei valori dell'array o un vettore riga con le medie calcolate per colonne se x è una matrice
  - **std(x)** fornisce la deviazione standard dei valori dell'array o un vettore riga con le deviazioni standard calcolate per colonne se x è una matrice

Richiami Teoria e Sintassi

- **Semplici operazioni con Array (2/2)**
  - **size(x)** restituisce un vettore riga con due valori [i j] uguali al numero di righe e colonne di x
  - **length(x)** restituisce il numero degli elementi di x o di colonne in caso di matrice
  - **sum(x)** restituisce la somma dei valori di x se x è un vettore; se x è una matrice restituisce un vettore riga i cui elementi corrispondono alla somma degli elementi di ciascuna colonna di x
  - **sort(x)** restituisce un array della stessa dimensione di x; se x è un vettore, ordina gli elementi in senso crescente; se x è una matrice, ordina ogni singola colonna

Operazioni Matematiche

- **Esercizio1:** Dopo aver avviato una sessione di lavoro, definire le variabili x=10 e y=3, ed effettuare i seguenti calcoli:

1. u = x + y [13]
2. v = xy [30]
3. w = x/y [3.3333]
4. z = sin x [-0.5440]
5. r = 8 sin x [-4.3522]
6. s = 5 sin (2y) [-1.3971]
7. $p = x^{y}$ [1000]
8. $k = x^{y/x}$ [1.9953]
9. $f = \sqrt{\sin(x + y)}$ [0.6482]
10. $g = \frac{x}{5\sqrt{y}}$ [1.1547]

Operazioni Matematiche

- **Svolgimento Esercizio1**

  1. Aprire Matlab
  2. Scrivere nel Command Window
  3. x = 10;
  4. y = 3;

- **Svolgimento 1-1: u = x + y**

  u = x + y

  u =

  13

Nota: senza ";" si visualizza il valore

Operazioni Matematiche

- **Svolgimento 2-1:** **v = xy**
  - v = x * y
- **Svolgimento 3-1:** **w = x/y**
  - w = x / y
- **Svolgimento 4-1:** **z = sinx**
  - z = sin(x)
- **Svolgimento 5-1:** **r = 8 sin x**
  - r = 8* sin(x)

Operazioni Matematiche

- **Svolgimento 6-1:** **$s = 5 \sin(2y)$**
  - s = 5*sin(2*y)
- **Svolgimento 7-1:** **$p = x^y$**
  - p = x^y
- **Svolgimento 8-1:** **$k = x^{y/x}$**
  - k = x^(y/x)

Operazioni Matematiche

- **Svolgimento 9-1:** $f = \sqrt{\sin(x + y)}$
  - f = sqrt(sin(x+y))

- **Svolgimento 10-1:** $g = \frac{x}{5\sqrt{y}}$
  - g = x/(5*sqrt(y))

  – Nota: nelle espressioni si usano solo le parentesi tonde, quindi la precendenza delle operazione si ottiene con l'annidamento delle parentesi

Operazioni Matematiche

- **Esercizio2:** calcolare i seguenti logaritmi:

1. Logaritmo naturale di 24 [3.1781]
2. Logaritmo in base 2 di 73 [6.1898]

Array (uso di [] e linespace)

- **Esercizio3:** Creare il vettore i cui elementi sono linearmente intervallati con incrementi di 0.4 tra 2 e 14. Utilizzare due metodi diversi.
- **Esercizio4:** Creare il vettore che ha 100 elementi linearmente intervallati tra 5 e 28. Utilizzare due metodi diversi.

Array (uso di [] e linspace)

- **Svolgimento 3-1: Array utlizzando []**
  - A=[2:0.4:14]
  - E' un array di 31 elementi
- **Svolgimento 3-2: Array utlizzando linespace**
  - n1=(14-2)/0.4
  - A=linspace(2,14, n1+1)

Nota: devo aggiungere 1 altrimenti otterrei un vettore di 30 elementi che non arriverebbe a 14

- **Svolgimento 4-1: Array utlizzando []**
  - n2=(28-5)/(100-1);
  - B=[5:n2:28]

Nota: devo togliere 1 altrimenti otterrei un vettore 101 elementi

- **Svolgimento 4-2: Array utlizzando linespace**
  - B=linspace(5,28,100)

Array (uso di [] e linespace)

- **Conclusione:**
  - Quando il passo è noto è meglio usare **la coppia di parentesi** quadre per generare l'array
  - Quando il passo non è noto ma è noto il numero max di elementi è meglio usare la funzione **linspace()** per generare l'array. Il passo verrà calcolato dalla funzione linspace().

Array (uso di logspace)

- **Esercizio 5:** Creare il vettore che ha 20 elementi logaritmicamente intervallati tra 100 e 10000.

  - **Svolgimento 5:**
    - D = logspace(2, 4, 20);

Nota: in logspace si inserisce la potenza di 10 per ottenere l'estremo dell'array desiderato

Array

- **Esercizio 6:** Definito il vettore x=[-3, 4, 2, 1, 0, 2, 3, 5, 10] calcolare:
  1. Numero di elementi del vettore (length())
  2. La dimensione in termini di righe e colonne (size())
  3. La media di x (mean())
  4. Assegnare agli elementi con indici pari il valore 7, digitando un unico comando (suggerimento: usare l'operatore ":")

Array

- **Svolgimento 6-3:**
  - mean(x)

- **Svolgimento 6-4:**
  - x(2:2:8) = 7 -OPPURE- x(2:2:end) = 7
  - x =

    -3 7 2 7 0 7 3 7 10
  - Nota: L'operatore due punti (:) permette di selezionare sottoinsiemi di valori all'interno di un array.

Array e Matrici

- **Esercizio7:** Definito il vettore x=[2, 3, 4, 5, 6, 7, 12], costruire a partire da esso il vettore y

y =

2, 3, 4, 5, 6, 7, 12, 12, 7, 6, 5, 4, 3, 2.

Array e Matrici

- **Svolgimento 7 – (continua):**
  - x=[2, 3, 4, 5, 6, 7, 12],
  - Il vettore y è caratterizzato da:
    1. **concatenazione** di 2 “x” essendo “x” composto da 7 elementi e y da 14
    2. Gli elementi da 8 al 14 sono gli elementi x ordinati dal più grande al più piccolo
  - Quindi
    1. y=[x x]

Array e Matrici

- **Svolgimento 7:**

```
y=
Columns 1 through 13
2  3  4  5  6  7 12  2  3  4  5  6  7
Columns 14 through 26
12
```

2. Utilizzare l'help per la funzione sort()
3. y1=sort(x,2,'descend')
4. y(8:end)=y1

```
y=
Columns 1 through 13
2  3  4  5  6  7 12 12 7  6 5  4  3
Columns 14 through 26
2
```

Nota: devo specificare l'indice di righe o colonne e il tipo di ordinamento

Array e Matrici

- **Svolgimento 7-2:**

```
y = [x sort(x,'descend')]

y=
    Columns 1 through 13
    2  3  4  5  6  7  12  12  7  6  5  4  3
    Columns 14 through 26
    2
```

Matrici

- **Esercizio8:** Data la seguente matrice:

$$A=\begin{pmatrix} 1 & 0 & 6 & -3 \\ -1 & 2 & 0 & 2 \\ 0 & 3 & -1 & -2 \\ -6 & 0 & 4 & 1 \end{pmatrix}$$

- Determinare
  1. il vettore «a» composto dalla seconda colonna
  2. la media della seconda colonna
  3. la matrice X composta dagli elementi appartenenti alla prima e terza riga ed alla seconda e quarta colonna

Matrici

- **Svolgimento 8:**
  - Matrice A:

    A=[1 0 6 -3; 1 2 0 2; 0 3 -1 -2; 0 3 -1 -2; -6 0 4 1]

    1. a=A(:,2)

       Rows 1 through 4

       0

       2

       3

       0
    2. mean(a)
    3. X=A([1 3],[2 4])

Richiami di Teoria e Sintassi

- **Prodotto tra Matrici ed Array**
- Si definiscono tre tipi di prodotti (e analogamente divisioni):

  - **Prodotto Elemento per Elemento:** La moltiplicazione tra l'array A e l'array B, di stessa dimensione, elemento-per-elemento (.*) restituisce un nuovo array C i cui elementi sono il prodotto dei corrispondenti elementi di A e di B. Sintassi **A.*B**

    C=A.*B

    A[n1, m1] x B[n2, m2] = C[n, m]

    dove **n = n1 = n2**

    m = m1 = m2

Richiami di Teoria e Sintassi

- **Prodotto scalare:** il prodotto scalare è un'operazione tra un vettore riga a=[a1, …, an] ed un vettore colonna b=[b1, …, bn] aventi entrambi n elementi. E' definito come il valore scalare ottenuto dalla somma dei prodotti degli elementi corrispondenti dei due vettori: a1 b1 + …+ an bn. Sintassi **a*b**

C=A*B

A[1, m1] x B[n2,1] = C[1,1]

dove m1=n2

Richiami di Teoria

- **Prodotto matrice-vettore:** È una generalizzazione del prodotto scalare. Richiede che il numero di colonne della matrice coincida con il numero di righe della seconda matrice.

  C=A*B

  A[n1, m1] x B[n2,m2] = C[n1,m2]

  dove m1=n2

Array e Matrici

- **Esercizio9:** Definita x nell'intervallo [1,5], con passo 0.1, tabellare le funzioni e graficarle
  1. y1= log(x)
  2. $y2 = e^{3x}\sin(5\pi x)$

- Nota1: Con il termine "tabellare" si intende creare una tabella, tale che la prima colonna rappresenti la variabile x e la seconda la variabile y.
- Nota 2: Il π in Matlab si indica con "pi"
- Nota 3: il grafico si visualizza con la funzione ("plot")

Array e Matrici

- **Svolgimento 9-1:**
  1. x= [1:0.1:5];
  2. y1= log(x)
  3. A = [x;y1]
  4. Tab=A';
  5. Plot(x,y1)

- Nota: Con l'operatore " ' " si ottiene la matrice trasposta

Array e Matrici

- **Svolgimento 9-2:**

1. y2= exp(3*x).*sin(5*pi*x)
2. A 1= [x;y2]
3. Tab1=A1';
4. Plot(x,y2)

- Nota: Con l'operatore " .* " si ottiene il prodotto elemento per elemento

Array e Matrici

- **Esercizio 10:** Date le seguenti matrici

$$A = \begin{pmatrix} -7 & 16 \\ 4 & 9 \end{pmatrix} \qquad B = \begin{pmatrix} 6 & -5 \\ 12 & -2 \end{pmatrix}$$

$$C = \begin{pmatrix} -3 & -9 \\ 6 & 2 \end{pmatrix}$$

1. Calcolare il prodotto elemento per elemento di A e B
2. Elevare al quadrato gli elementi di B
3. Elevare alla 3 gli elementi di C

Array e Matrici

- **Svolgimento 10-1:**
- La moltiplicazione tra A e B elemento-per-elemento (.*) restituisce una nuova matrice D i cui elementi sono il prodotto dei corrispondenti elementi di A e di B.
- La sintassi è ".*"
- D=A.*B

Array e Matrici

- **Svolgimento 10-2:**
- Il quadrato di B si ottiene con la moltiplicazione tra B e B elemento-per-elemento (.*)
- B2=B.*B
- **Svolgimento 10-3:**
- Il cubo di C si ottiene
- C2=C.*C
- C3=C2.*C

Matrici

- **Esercizio11:** La seguente tabella riporta il consumo medio, i km percorsi e l'indice di affidabilità di 5 diverse automobili.

	1	2	3	4	5
Consumo km/l	11.2	8.8	9.1	12	16.4
Km percorsi	240	550	302	667	980
Affifabilità	0.921	0.454	0.980	0.777	0.602

1. Calcolare i litri di benzina consumati da ogni automobile
2. Calcolare il consumo medio delle 5 automobili prese in esame
3. Calcolare la percorrenza media delle 5 automobili prese in esame
4. Usare le funzioni corrette per determinare l'auto più affidabile e quella che consuma di meno

Array e Matrici

- **Svolgimento 11:**
- Creo i vettori
  - Vetture=[1 2 3 4 5]
  - Consumo =[11.2 8.8 9.1 12 16.4]
  - km_percorsi=[240 550 302 667 980];
  - Affidab = [0.921 0.454 0.980 0.777 0.602]
  - Litri= km percorsi / consumo quindi per calcolare i litri devo fare una divisone elemento per elemento (./)
  - litri =km_percorsi./Consumo;
  - mean(Consumo)
  - mean(km_percorsi)
  - auto più affidabile: max(Affidab)
  - consuma di meno: min(Consumo)

Matrici

- **Esercizio 12:** Data la matrice

$$G = \begin{pmatrix} 3 & 58 & -4 & 3 \\ -5 & 42 & 80 & 2 \\ 6 & -19 & 9 & 55 \\ -17 & 5 & 44 & 1 \end{pmatrix}$$

1. Ordinare gli elementi di ogni colonna in ordine crescente e assegnare il risultato all'array H.
2. Ordinare gli elementi di ogni riga in ordine crescente e assegnare il risultato all'array I.
3. Sommare gli elementi di ogni colonna e assegnare il risultato all'array L.
4. Sommare gli elementi di ogni riga e assegnare il risultato all'array M.

Array e Matrici

- **Svolgimento 12-1:**
- La funzione sort per una matrice A restituisce una matrice i cui elementi di ogni colonna sono ordinati in modo crescente
- Costruiamo G
- H=sort(G)

-17	-19	-4	1
-5	5	9	2
3	42	44	3
6	58	80	55

Array e Matrici

- **Svolgimento 12-2:**
- Per ordinare ogni riga bisogna prima considerare la trasposta di G
  - G1=G'
  - G1=

    3 -5 6 -17

    58 42 -19 5

    -4 80 9 44

    3 2 55 1
  - I1=sort(G1) e poi fare la trasposta di I1
  - I1=

    -4 -5 -19 -17

    3 2 6 1

    3 42 9 5

    58 80 55 44

    I=I1'

    I=

    -4 3 3 58

    -5 2 42 80

    -19 6 9 55

    -17 1 5 44

Array e Matrici

- **Svolgimento 12-3:**
  - Per sommare ogni colonna si usa la funzione sum
  - L=sum(G)
  - L=

    -13 86 129 61
- **Svolgimento 12-4**
  - Per sommare ogni riga si usa la funzione sum della matrice inversa di G e poi fare la trasposta di nuovo
  - M1=sum(G1)
  - M1 =60 119 51 33
  - M=M1’

Array e Matrici

- **Domanda Svolgimento esercizio 12:**
  - C'è un modo più semplice di trovare l'ordinamento per righe e la somma per righe?
  - Consultare l'help delle funzioni
    - Ordinamento per righe di G
    - I=sort(G,2)
    - Somma per righe di G
    - I=sum(G,2)

Matrici

- **Esercizio 13:** Creare un array tridimensionale D le cui tre pagine siano le seguenti matrici

$$A = \begin{pmatrix} 12 & 34 & 4 \\ 77 & 21 & 1 \\ 3 & 5 & 9 \end{pmatrix} \qquad B = \begin{pmatrix} 6 & -5 & -7 \\ 16 & 24 & -2 \\ 23 & 19 & 27 \end{pmatrix}$$

$$C = \begin{pmatrix} 11 & 41 & 14 \\ 7 & -1 & -12 \\ 31 & 15 & 94 \end{pmatrix}$$

1. Trovare l'elemento massimo in ogni strato di D e l'elemento massimo di D.

Matrici

- **Svolgimento 13:**
  - L'array tridimensionale sarà composto da tre indici.
  - Ogni matrice costituirà una "pagina" dell'array
  - D = zeros(3,3,3) (per creare una matrice 3 D)
  - Per visualizzare:

D(:,:,1) =

0 0 0

0 0 0

0 0 0

D(:,:,2)=

…

Matrici

- **Svolgimento 13:**
  - D(:,:,1)=A; D(:,:,2)=B; D(:,:,3)=C;
  - D1=max(A); D2=max(B); D3=max(C)
  - Dmax=[D1 D2 D3]
  - max(Dmax)

Matrici

- **Esercizio 14:** Creare un matrice E che abbia
  - Prima colonna uguale a vettore x costituito da 100 elementi da -10 a 10
  - Seconda colonna uguale a vettore y1=Asin(4πx +c)
  - Terza colonna y2=y1*Asin(4π(x+d) +c)
- Con A=2, e c=0.7 d=3π
- Graficare le due funzioni in un unico grafico

Matrici

- **Svolgimento 14:**
  - x=linspace(-10,10,100)
  - A=2; c=0.7; d=3*pi
  - y1=A*sin(4*pi*x +c)
  - y2_=A*sin(4*pi*(X+d) +c);
  - y2=y1.*y2_;
  - E=zeros(100,3);
  - E(:,1)=x; E(:,2)=y1; E(:,3)=y2;
  - plot(x,y1,x,y2) oppure
  - plot(E(:,1), E(:,2), E(:,1), E(:,3))

Domande?

Samuela Persia

spersia@fub.it