

Esercitazioni 5 (Matlab) 29/11/2013:

Comandi avanzati Matlab

Alberto Castellini

Dalla esercitazione precedente:

1. Correzione esercizio 5 della esercitazione precedente.
2. IMPORTAZIONE DATI DA FILE CSV: importare il file `cellConcentrations.csv` (allegato) in modo da ottenere una matrice *data* contenente i soli valori numerici presenti nel file. Il file contiene nella prima colonna le età in giorni di 80 pazienti e nelle successive colonne le concentrazioni di 5 sostanze trovate nel sangue dei pazienti stessi. Eseguire l'esercizio prima tramite l'interfaccia grafica di Matlab (File/Import Data) e poi da riga di comando tramite l'opportuna funzione Matlab.
3. GENERAZIONE DI GRAFICI (1): generare 5 scatter plot, ciascuno avente nell'asse *x* l'età in giorni dei pazienti e nell'asse *y* i valori di concentrazione di una sostanza
4. IMPLEMENTAZIONE DI FUNZIONI (1): Scrivere una funzione Matlab chiamata "*firstStat*" che prenda in input un array di numeri e che restituisca in output:
 - a. la media dei valori contenuti nell'array,
 - b. la mediana dei valori contenuti nell'array,
 - c. la deviazione standard dei valori contenuti nell'array,
 - d. la varianza dei valori contenuti nell'array,
 - e. gli outliers dei valori contenuti nell'array e le loro posizioni, dove un punto è detto outlier se la sua distanza dalla media dei valori dell'array è maggiore di 3 volte la deviazione standard dell'array stesso.

Utilizzare la funzione per calcolare le varie statistiche su ciascuna colonna della matrice *data* definita al punto 6.

Nuovi esercizi:

5. CICLO FOR e PREALLOCAZIONE: Scrivere una funzione Matlab chiamata "*fibonacci*" che prenda in input un numero *n* e restituisca in output i primi *n* valori della successione di Fibonacci (i primi due elementi della successione di Fibonacci sono $F(0)=0$ e $F(1)=1$, mentre i successivi elementi si calcolano a partire dai precedenti due tramite la seguente formula $F(n)=F(n-1)+F(n-2)$. Vedi http://it.wikipedia.org/wiki/Successione_di_Fibonacci).
6. CICLO FOR: Scrivere una funzione Matlab chiamata "*fattoriale*" che prenda in input un numero *n* e restituisca in output il suo fattoriale (es: $fattoriale(6)=6!=6*5*4*3*2*1$). L'esercizio deve essere svolto utilizzando un ciclo for.

7. CICLO WHILE: Scrivere una funzione Matlab chiamata “*fattorialeW*” che calcola il fattoriale (vedi sopra) utilizzando un ciclo while.
8. CONTINUE, BREAK e RETURN: Scrivere una funzione Matlab contenente il ciclo for riportato alla slide 53 del corso e testarne il funzionamento. Cosa accade se l’istruzione *continue* viene sostituita con l’istruzione *break*? E se viene sostituita con l’istruzione *return*?
9. CICLO FOR E VETTORIALIZZAZIONE: scrivere una funzione Matlab che prenda in input un numero n e restituisca un vettore di tutti i valori del logaritmo in base 10 dei numeri compresi tra 1 ed n . Eseguire l’esercizio prima con un ciclo for e poi con la tecnica della settorializzazione (vedi slide 60 del corso), Quale tecnica possiede le migliori performance?
10. ARRAY MULTIDIMENSIONALI: scrivere una funzione Matlab che prenda in input due numeri n ed m e restituisca una matrice $3 \times 3 \times 3$ in cui:
 - gli elementi nelle posizioni $(:, :, 1)$ sono numeri random uniformemente distribuiti tra n ed m ,
 - gli elementi nelle posizioni $(:, :, 2)$ sono numeri random uniformemente distribuiti tra $n+1$ ed $m+1$,
 - gli elementi nelle posizioni $(:, :, 3)$ sono numeri random uniformemente distribuiti tra $n+2$ ed $m+2$.
11. CELL ARRAY: ripetere l’esercizio precedente facendo in modo che l’output sia un cell array contenente 3 matrici di dimensione 3×3 , 4×4 e 5×5 .