



Esercizi con le table

Eserguire gli esercizi in un file di script (.m) oppure di live script (.mlx) e NON NELLA COMMAND WINDOW. Alla fine salvare il file, con un nome a piacere, in modo da confrontare quello che avete fatto con il file di soluzione. Chi sta lavorando nei computer dei laboratori può mandare il file all'indirizzo email che crede.

Creare la variabile `a` contenente il numero 10

Caricare in memoria la table `intrtrade` tramite l'istruzione `load intrtrade`

Sommare il valore all'incrocio della terza riga e seconda colonna della table `intrtrade` con quello della variabile `a` in modo tale che.

1) il risultato in output sia una table di dimensione 1x1 (assegnare a questa variabile il nome `btable`)

2) il risultato in output sia un double (assegnare a questa variabile il nome `bdouble`)

Estrarre le righe in posizione dispari della table `inttrade` ed aggiungere il valore contenuto nella variabile `a` elevato alla terza in modo tale che

1) il risultato in output sia una table (assegnare a questa variabile il nome `dtable`). Mostrare le prime tre righe

2) il risultato in output sia una matrice numerica di doubles (assegnare a questa variabile il nome `ddouble`). Mostare le prime 3 righe

Solo sistema operativo Windows

Creare su C:\ la cartella denominata `MATLABesercizi`

Copiare in questa cartella il file `Firm.xlsx` presente nella sottocartella `capUnivariate` oppure nella cartella `capStruttureDati`

della cartella dove avete clonato gli esercizi di corredo al libro da GitHub; Fate riferimento a pagina 27 del libro per le istruzioni su come clonare l'archivio GitHub degli esercizi.

Posizionarsi nel percorso `C:\MATLABesercizi`

Verificare che il percorso in cui si sta lavorando sia `'C:\MATLABesercizi'` tramite l'istruzione `MATLAB pwd`

Creare un'istruzione `assert` che generi l'errore "Non sono nel percorso corretto" se l'utente non si trova nel percorso `C:\MATLABesercizi`

Osservazione: l'istruzione di seguito viene eseguita solo se il sistema operativo è Windows

Solo sistema operativo MacOS

Supponiamo di impersonare l'utente `marco` (sostituire `marco` con il vostro nome utente di MacOS).

Per conoscere il vostro nome utente, eseguite l'istruzione seguente nella Command Window

```
getenv('USER')
```

Creare sotto la cartella home (`/Users/marco/`) la cartella denominata `MATLABesercizi`

Posizionarsi nella cartella `/Users/marco/MATLABesercizi`

Verificare che il percorso in cui si sta lavorando sia effettivamente `'/Users/marco/MATLABesercizi'` tramite l'istruzione `MATLAB pwd`

Creare un'istruzione `assert` che generi l'errore "Non sono nel percorso corretto" se l'utente non si trova nel percorso `/Users/marco/MATLABesercizi`

Osservazione: l'istruzione di seguito viene eseguita solo se il sistema operativo è MacOS

PARTE DI NUOVO COMUNE AI DUE SISTEMI OPERATIVI

Copiare nella cartella MATLABesercizi il file `Firm.xlsx` presente nella sottocartella `capUnivariate` di `DSconMATLAB/matlabfiles`

della cartella dove avete clonato gli esercizi di corredo al libro da GitHub. Fate riferimento a pagina 27 del libro per le istruzioni su come clonare l'archivio GitHub degli esercizi.

Caricare il contenuto del range `A1:H15` del foglio data nella table denominata `X` con l'opzione di lettura dei nomi delle righe

Mostare il record riferito alla riga `P0213` della table `X`

Estrarre i record `P0213` e `P0219` e le colonne 2, 3, 4 della table `X`

Estrarre tutte le righe dei campi denominati "Surname" e "Wage" della table `X`. Assegnare l'output alla variabile `Uno`. Inserire `Surname` e `Wage` prima dentro un cell array e poi dentro un string array. Selezionare le variabili della table prima utilizzando il cell array e poi lo string array. Controllare tramite `assert` che i due modi di estrazione delle variabili producono lo stesso risultato. Mostrare le ultime 5 righe della table `Uno`.

Estrarre la prima e le ultime due colonne della table `X`. Mostrare le prime 8 righe tramite istruzione `head`

Estrarre le righe 5,7,...,13 della variabile "Wage" in modo tale che l'output sia un array e non una table

Creare una variabile booleana come logical array, che è vera quando

il salario è compreso tra 3000 e 4000 e denominarla boo (v. Esercizio 1.8 del libro per un esercizio simile).

Mostrare la variabile matrice che contiene nella prima colonna il contenuto della variabile Wage e nella seconda colonna il vettore booleano boo

Mostrare i record con salario compreso tra 3000 e 4000 per le variabili

Surname e Name, in formato table

Mostrare i record con salario compreso tra 3000 e 4000 per le variabili

Wage e CommutingTime, in formato table

Mostrare i record con salario compreso tra 3000 e 4000 per le variabili

Wage e CommutingTime, in formato array

Mostrare i nomi delle righe in posizione 10:13 nella Command Window

Assegnare alla quinta riga il nome "nomeNONSO" ed alla penultima riga il nome 'P0999'

Assegnare alle righe 1,4,7... della variabile Wage il valore 5000.

Rinominare la variabile "Name" "Nome"

Mostrare la table modificata con i nuovi nomi riga e i nuovi valori di Wage

Copiare il contenuto della table X nella table denominata X1

Utilizzando la funzione `randperm` generare una permutazione casuale dei 7 numeri associati alle 7 colonne della table X1. Inserire la permutazione casuale dei 7 numeri in un vettore denominato `permCasuale`;

Assegnare alle variabili presenti dentro X1 l'ordine definito dal vettore `permCasuale`. Ad esempio, se il vettore `permCasuale` è uguale a

5 6 7 1 2 3 4

le colonne 5:7 devono diventare le prime 3 colonne e le prime 4 colonne devono andare in posizione 4:7.

Ad ogni run il vettore `permCasuale` cambia di conseguenza cambia l'ordine delle colonne in X1.

Scrivere un codice che rinomini la variabile `Nome` con `NomeProprio` senza sapere la posizione della variabile `Nome` dentro X1.

Modo 1 di risolvere l'esercizio

Modo 2 di risolvere l'esercizio

Controllo l'uguaglianza delle due implementazioni.

Mostrare i nuovi nomi delle variabili X1 nella Command Window

IF e CICLI FOR

Fissare il seed a 123459

Generare un numero casuale da $N(0,1)$

in una variabile denominata `x`. Scrivere un programma che mostri nella Command Window la stringa "numero inferiore a - 0.5" se `x < -0.5`

oppure "numero tra - 0.5 e 0.5" se `x` appartiene all'intervallo

chiuso [-0.5 0.5], oppure "numero superiore a 0.5" se $x > 0.5$

Esempio di CICLO FOR

Creare la sequenza che segue

$$x_t = 0.3x_{t-1} - 0.5x_{t-2} - 2x_{t-3}, \quad t=4, 5, \dots, 20, \text{ con } x_1=4, x_2=5, x_3=2$$

Calcolare i valori della sequenza per $t=4, 5, 6, \dots, 20$

Fare il plot della sequenza al variare di t

Aggiungere all'asse x la label t

Aggiungere come titolo del grafico il testo riportato nell'immagine precedente.

Osservazione: MATLAB supporta il linguaggio Latex. Per capire come inserire gli apici e pedici in Latex. In questo linguaggio gli apici si fanno con il simbolo $\wedge$ e i pedici con il simbolo $_$. Nel caso in cui siano presenti più di un carattere nell'apice (pedice) questi devono essere inseriti dentro le parentesi graffe.

Risolvere il seguente sistema di equazioni lineari di 4 equazioni e 4 incognite

$$\begin{cases} 3x_1 + 4x_2 + 17x_3 + 18x_4 = 2 \\ x_1 + 2x_4 = 100 \\ x_1 - x_3 + 3x_4 = 2 \\ x_1 + 3x_2 - 2x_4 = 8 \end{cases}$$

Trovare il vettore $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$

Verificare la soluzione trovata tramite `assert`.

Calcolare la media potenziata di ordine s dei seguenti numeri 21 23 22 28 22 18 24 30 21. Aggiungere una slider bar per la variabile s con valore minimo -3, valore massimo 3, passo 1 e valore di default 2. Inserire come label del controllo "Media potenziata di ordine s ".

Osservazione: la slider bar può essere inserita se e solo se sono nel file in formato .mlx.

Mostrare il dettaglio dei calcoli manuali da effettuare per calcolare la media potenziata tramite la chiamata a `GUIpowermean`.