



```
disp('Data di ultima modifica')
data0dierna=datetime('today','Format','dd/MM/yyyy');
disp(data0dierna)
```

## Esercizi di base al capitolo 1: introduzione all'utilizzo di MATLAB e alla gestione dei dati

Esercizi preliminari per controllare che i file di corredo al libro siano stati scaricati dal sito di GitHub e che il toolbox FSDA ultima versione sia stato caricato in memoria

### Esercizio per testare che FSDA sia stato installato correttamente

Digitare nella Command Window il comando `tuna` (`tuna` sta per *Toolbox Update Notification Alert*). Se il comando `tuna` è andato buon fine dovreste visualizzare un output del tipo (ovviamente nel corso del tempo i numeri della versione cambiano).

```
>> tuna
Installed version of "FSDA": 8.7.9.1
Latest available version: 8.7.9.1
Toolbox FSDA is up-to-date.
```

Test per capire se si riesce a caricare in memoria un file di FSDA.

Eseguire nella Command Window il comando

```
load intrade.mat
```

Questo comando dovrebbe caricare in memoria la table denominata `intrade`

Nella finestra di Workspace vi deve apparire

| Workspace                                                                                |             |  |
|------------------------------------------------------------------------------------------|-------------|--|
| Name^                                                                                    | Value       |  |
|  intrade | 180x2 table |  |

Se il comando precedente è andato a buon fine

nella Command Window l'istruzione

```
intrade(1:3,:)
```

deve produrre

```
ans =  
  
          I  
3x2 table  
  
    Weight    Value  
    _____    _____  
    0.3384    2.1353  
    0.8176    3.392  
    0.3868    1.9854
```

Scaricare i file di corredo al libro (dopo essersi posizionati su un percorso dove si hanno i diritti di scrittura) tramite l'istruzione

```
!git clone https://github.com/UniprJRC/DsconMATLAB.git
```

Se questa istruzione non è andata a buon fine vedere il libro di testo.

### Esercizio per testare che i file di corredo al libro siano stati scaricati correttamente da GitHub

Navigare nella sottocartella `DsconMATLAB/matlabfiles/capUnivariate` della cartella dove avete clonato gli esercizi di corredo al libro da GitHub. Ad esempio se avete clonato i files nella cartella `D:\temp` (sistema

operativo windows) e avete navigato nel percorso DSconMATLAB/matlabfiles/capUnivariate i files nella **Current Folder** che visualizzate sono i seguenti (Sistema operativo Windows)

FILE

NAVIGATE

CODE

ANALYZE

SEC

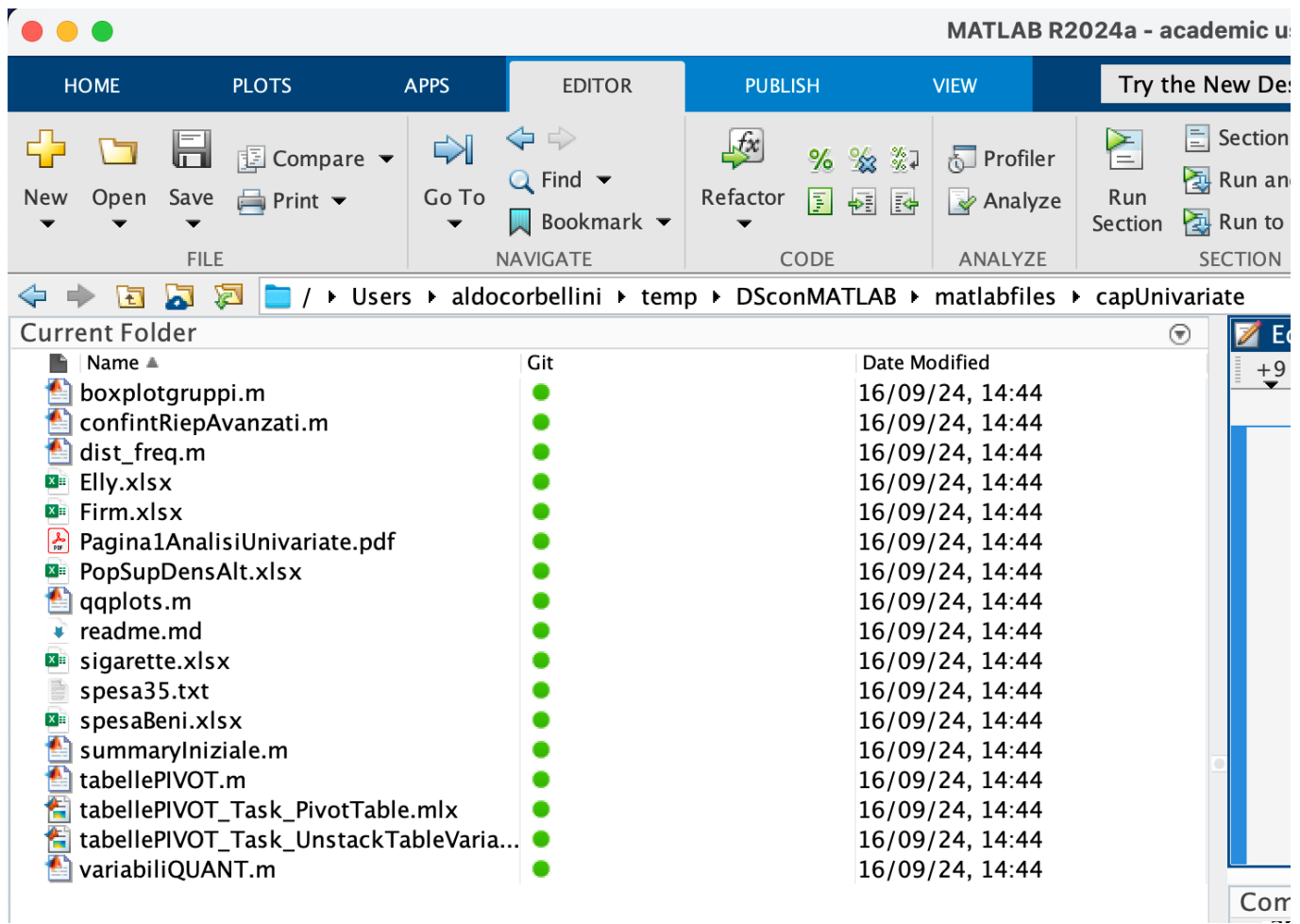
D: ▶ temp ▶ DSconMATLAB ▶ matlabfiles ▶ capUnivariate

Current Folder

| Name                                                      | Size   | Git         | Date Modified    | Type             |
|-----------------------------------------------------------|--------|-------------|------------------|------------------|
| <div><div></div>boxplotgruppi.m</div>                     | 2 KB   | <div></div> | 14/09/2024 16:25 | Script           |
| <div><div></div>confintRiepAvanzati.m</div>               | 7 KB   | <div></div> | 14/09/2024 16:25 | Script           |
| <div><div></div>dist_freq.m</div>                         | 5 KB   | <div></div> | 14/09/2024 16:25 | Script           |
| <div><div></div>Elly.xlsx</div>                           | 14 KB  | <div></div> | 14/09/2024 16:25 | Foglio di lav... |
| <div><div></div>Firm.xlsx</div>                           | 19 KB  | <div></div> | 14/09/2024 16:25 | Foglio di lav... |
| <div><div></div>Pagina1AnalisiUnivariate.pdf</div>        | 204 KB | <div></div> | 14/09/2024 16:25 | Documento...     |
| <div><div></div>PopSupDensAlt.xlsx</div>                  | 15 KB  | <div></div> | 14/09/2024 16:25 | Foglio di lav... |
| <div><div></div>qqplots.m</div>                           | 1 KB   | <div></div> | 14/09/2024 16:25 | Script           |
| <div><div></div>readme.md</div>                           | 1 KB   | <div></div> | 14/09/2024 16:25 | File di origi... |
| <div><div></div>sigarette.xlsx</div>                      | 14 KB  | <div></div> | 14/09/2024 16:25 | Foglio di lav... |
| <div><div></div>spesa35.txt</div>                         | 1 KB   | <div></div> | 14/09/2024 16:25 | Documento...     |
| <div><div></div>spesaBeni.xlsx</div>                      | 11 KB  | <div></div> | 14/09/2024 16:25 | Foglio di lav... |
| <div><div></div>summaryIniziale.m</div>                   | 3 KB   | <div></div> | 14/09/2024 16:25 | Script           |
| <div><div></div>tabellePIVOT.m</div>                      | 5 KB   | <div></div> | 14/09/2024 16:25 | Script           |
| <div><div></div>tabellePIVOT_Task_PivotTable.mlx</div>    | 46 KB  | <div></div> | 14/09/2024 16:25 | Live Script      |
| <div><div></div>tabellePIVOT_Task_UnstackTableVaria</div> | 5 KB   | <div></div> | 14/09/2024 16:25 | Live Script      |
| <div><div></div>variabiliQUANT.m</div>                    | 4 KB   | <div></div> | 14/09/2024 16:25 | Script           |

Se invece avete il Mac ed il vostro nome utente è "aldocorbellini" ed avete scaricato i file dentro /Users/aldocorbellini/temp

dovete navigare nel percorso /Users/aldocorbellini/temp/DSconMATLAB/matlabfiles/capUnivariate

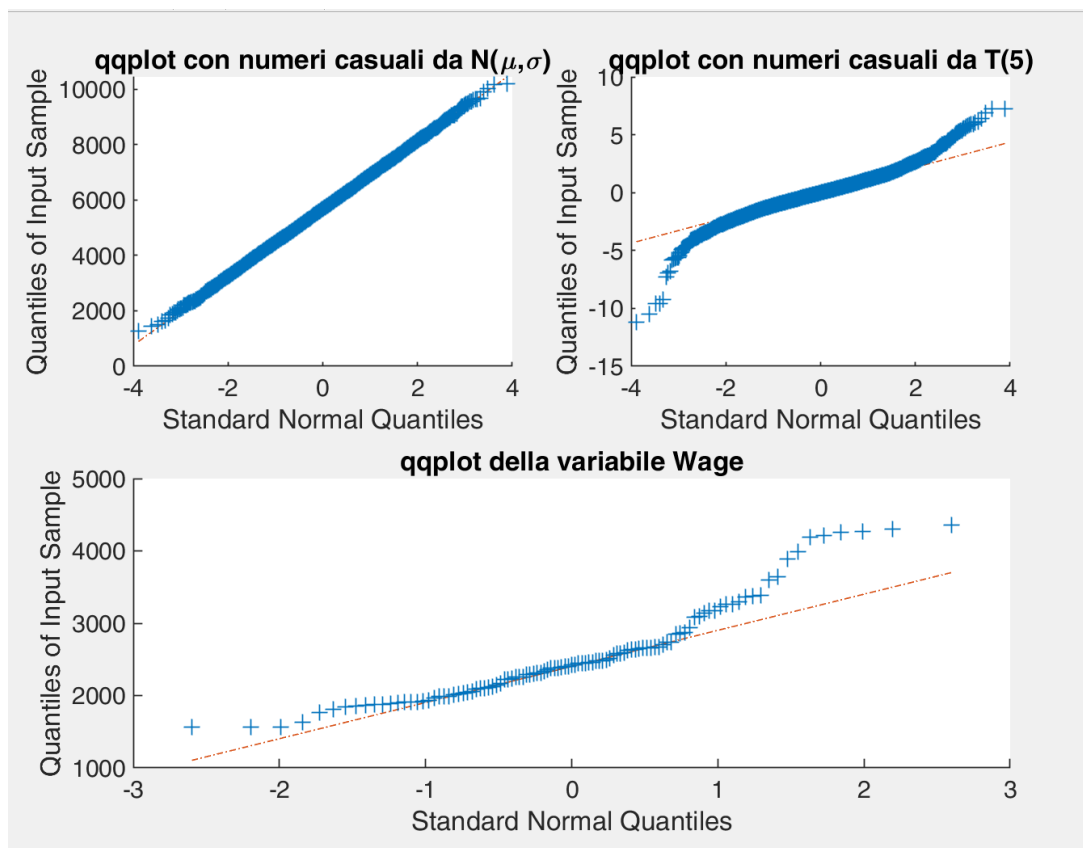


Aprire il file `qqplots.m`. Eseguire il run del file `qqplots.m`, vedi p. 149 del libro.

Se siete riusciti a fare il run del file `qqplots.m` nella Command Window vi dovrebbe apparire il seguente output

```
>> qqplots
Welcome to the third edition of the book
CONGRATULATIONS AND ENJOY!
```

Il Run del file `qqplots.m` deve aver prodotto un grafico molto simile a quello di seguito.



*Osservazione:* si parla di molto simile e non di uguale in quanto i primi due pannelli in alto sono generati da numeri casuali che cambiano ad ogni esecuzione successiva. Ad ogni run, quindi, i due pannelli in alto cambiano leggermente. Al contrario il grafico del pannello in basso (qqplot della variabile Wage del file Firm.xlsx) non cambia nei diversi run.

Al termine dell'esecuzione del file, nella finestra Workspace dovreste avere le seguenti variabili

| Workspace                                                                                     |                |
|-----------------------------------------------------------------------------------------------|----------------|
| Name ^                                                                                        | Value          |
|  miofile   | "Firm.xlsx"    |
|  n         | 10000          |
|  nu        | 5              |
|  X         | 107x9 table    |
|  xnorm     | 10000x1 double |
|  xstudentt | 10000x1 double |

Il contenuto dei vettori xnorm e xstudentt cambia ad ogni run.

Al contrario il contenuto della table X rimane sempre lo stesso ad ogni run

Ad esempio la quinta riga della table X è

```
>> X(5,:)
```

```
ans =
```

```
1×9 table
```

|       | Surname      | Name           | Gender  | BirthDate   | Education | Wage | C |
|-------|--------------|----------------|---------|-------------|-----------|------|---|
| P0215 | { 'PRENNA' } | { 'FEDERICA' } | { 'F' } | 22-Jan-1966 | { 'C' }   | 3893 |   |

Chiudere dalla finestra di editor il file `qqplots.m`

Eseguire il `clear` della memoria

Eseguire il run del file `qqplots`, senza aprirlo tramite il comando

`run qqplots`

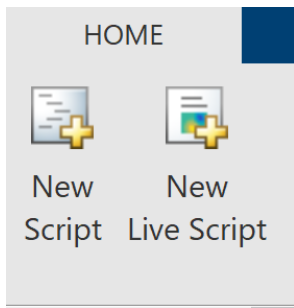
nella Command Window.

```
% Se non riuscite ad eseguire l'istruzione precedente e ricevete il  
% messaggio di errore (scritto in rosso) di seguito  
%  
% Error using run (line 74)  
% qqplots.m not found.  
%  
% significa che la vostra Current Folder (ossia la cartella in cui vi  
% trovate) non contiene il file qqplots.m  
% Ad esempio se avete il sistema operativo Windows e avete scaricato i file  
% di GitHub nella cartella C:\PIPP0 dovete trovarvi nella cartella  
% C:\PIPP0\DSconMATLAB\matlabfiles\capUnivariate  
% Se invece avete il sistema operativo MacOS, il vostro nome utente è  
% pippo e avete scaricato i file di GitHub nella cartella  
% /Users/pippo/temp/  
% dovete trovarvi nella cartella  
% /Users/pippo/temp/DSconMATLAB/matlabfiles/capUnivariate
```

*Osservazione:* chi non sapesse come navigare nelle sottocartelle è necessario che colmi questo gap, altrimenti al momento dell'esame non saprà nemmeno dove si trova il file che deve consegnare.

*Osservazione:* i comandi per risolvere la parte di seguito devono essere inseriti in un nuovo file (con denominazione a piacere) nella finestra di Editor oppure di Live Editor.

Di conseguenza, create un nuovo file di script (ossia un nuovo file con estensione `.m`) oppure un nuovo file di live script (ossia un nuovo file con estensione `.mlx`) tramite il pulsante "New Script" oppure "New Live Script" della scheda HOME



e nel nuovo file scrivete le istruzioni (con i commenti riga per riga). Ogni volta che aggiungete una nuova istruzione, per essere sicuri che la nuova istruzione sia corretta, fate il Run completo del file oppure solo dell'ultima istruzione tramite dx del mouse "Evaluate Selection in Command Window".

% dominio della v.a.

x = -4:0.1:4;

Per la differenza tra file .m e file .mlx v. libro di testo

## Esercizi di base

Fare il `clear` della memoria (ossia cancellare tutte le variabili presenti) e chiudere tutti i grafici aperti.

```
clear
close all
```

Creare una variabile denominata `a` che contiene il numero 12.

Creare un'altra variabile denominata `b` che contiene il numero 3

Creare un'altra variabile denominata `c` che contiene il numero -8

Calcolare `d` come `a` elevato al quadrato più tre volte la variabile `b`. Mostrare il contenuto di `d` nella Command Window (oppure nella finestra di output del file `mlx`).

*Osservazione:* la dizione abbreviata "mostrare il contenuto della variabile nella Command Window" significa mostrare la variabile nella Command Window oppure nella finestra di output del file `mlx`.

Calcolare a+d ed assegnare il risultato alla variabile pippo

Mostrare la variabile pippo nella Command Window

Supponiamo che le variabili a, b e c siano i coefficienti dell'equazione di secondo grado

$$ax^2 + bx + c = 0$$

Calcolare le soluzioni x1 e x2 dell'equazione di secondo grado tramite implementazione manuale della formula di soluzione

$$\begin{cases} x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \\ x_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \end{cases}$$

Lista degli errori tipici

1) La formula

$$x1 = (-b - \text{sqdelta}) / 2 * a;$$

sarebbe stata scorretta in quanto il numeratore viene diviso per 2 ed il risultato viene moltiplicato per a;

2) La formula per il numeratore di x1 scritta come segue

$$-b - (b^2 - 4 * a * c)^{0.5}$$

è scorretta in quanto con questa scrittura il Delta viene elevato alla 1 ed il risultato viene diviso per 2.

3) Nelle formule in programmazione esistono solo le parentesi tonde e non le parentesi quadre oppure le parentesi graffe. I simboli in MATLAB [ ] servono per racchiudere un insieme di numeri in un vettore (matrice) (v. libro). Al contrario, le parentesi graffe in MATLAB servono per creare un cell array (v. libro) oppure per accedere al contenuto degli elementi di un cell array (v. libro) oppure di una table. Di conseguenza, se avete bisogno di un'altra parentesi dovete aggiungere una nuova parentesi tonda. Questa avvertenza vale per tutti i linguaggi di programmazione, Excel ecc.

*Osservazione:* ovviamente la soluzione

$$x1 = (-b - \text{sqrt}(b^2 - 4 * a * c)) / (2 * a);$$

$$x2 = (-b + \text{sqrt}(b^2 - 4 * a * c)) / (2 * a);$$

è corretta, ma inefficiente dal punto di vista computazione, in quanto le espressioni  $\sqrt{b^2-4*a*c}$  e  $(2*a)$  vengono calcolate due volte. Al contrario, nella soluzione proposta, le parti comuni alle due soluzioni vengono inserite in due variabili (`sqdelta` e `den`), in modo da calcolare queste due quantità una sola volta.

Creare il vettore colonna `sol` che contiene come primo elemento `x1` e come secondo elemento `x2`

Mostrare nella Command Window le soluzioni `x1` e `x2` come vettore riga

Creare il vettore `p` come vettore riga di lunghezza 3 contenente le variabili `a`, `b` e `c`

Calcolare le soluzioni tramite la funzione di MATLAB `roots` (v. capitolo 2 del libro) ed assegnare il risultato al vettore denominato `solchk`

Controllare tramite `assert()` (vedi capitolo 1 del libro) che il vettore `sol` sia uguale al vettore `solchk`. Se il risultato non fosse uguale scrivere il messaggio di errore "Implementazione manuale errata".

*Osservazione:* il codice

```
assert(sol==solchk,"Implementazione manuale errata")
```

non è corretto in quanto `sol==solchk` produce in output un vettore di due valori logici.

Se si vuole utilizzare `sol==solchk` allora bisogna premettere la funzione `all`, ossia andiamo a controllare che l'uguaglianza ci sia in corrispondenza di ciascuno dei due elementi

Effettuare il `clear` della memoria

Creare il vettore riga che segue

1 3 5 7 9 11 13 15

e denominarlo x

Creare il vettore colonna che segue e denominarlo y

24

21

18

15

...

-126

Calcolare la somma tra il quarto elemento del vettore y ed il terzo elemento del vettore x. Assegnare il risultato ad una variabile denominata z

Mostrare la variabile z nella Command Window

Creare la matrice magica (tramite la funzione `magic()`), e denominarla MA

17 24 1 8 15

23 5 7 14 16

4 6 13 20 22

10 12 19 21 3

11 18 25 2 9

Calcolare la somma degli elementi della seconda colonna ed assegnare il risultato ad una variabile denominata mio

Effettuare la moltiplicazione matriciale (vedi capitolo 2) tra la riga 2 e la colonna 3 della matrice MA ed assegnare il risultato ad una variabile denominata z

Mostrare la variabile z nella Command Window

Mostrare la size di z nella Command Window

Effettuare la moltiplicazione matriciale tra la colonna 3 e la riga 2 della matrice MA ed assegnare il risultato ad una variabile denominata Z

Mostrare la variabile Z nella Command Window

Mostrare la size della variabile Z nella Command Window

Estrarre le colonne 2 e 5 della matrice MA e mostrarle nella Command Window

Estrarre l'ultima colonna della matrice MA e mostrarla nella Command Window

Estrarre gli elementi all'incrocio delle righe 2 5 1 e delle colonne 1 e 4 della matrice MA in una variabile denominata T. Mostrare la matrice T

Elevare al quadrato gli elementi della colonna 3

Calcolare la somma dei quadrati degli elementi delle colonne 2:4 (vedi pagina 81 per approfondimenti). Il risultato deve essere un vettore riga di lunghezza 3 con il primo elemento uguale alla somma dei quadrati degli elementi della colonna 2, il secondo elemento uguale alla somma dei quadrati degli elementi della colonna 3, ...

Calcolare la somma dei quadrati degli elementi delle righe 3 e 1. Il risultato deve essere un vettore colonna di lunghezza 2 con il primo elemento uguale alla somma dei quadrati degli elementi della riga 3 ed il secondo elemento uguale alla somma dei quadrati degli elementi della riga 1.

Creare una variabile denominata `MAvec` che applica l'operatore `vec` alla matrice `MA`

Estrarre gli elementi in posizione dispari del vettore `MAvec` e mostrarli nella Command Window

```
)
```

Unmatched ')'. Check for missing '('.

Creare una nuova variabile denominata `MAvecScambiato` che contenga come primo elemento l'ultimo del vettore `MAvec`, come secondo elemento il penultimo del vettore `MAvec`, ..., come ultimo elemento il primo del vettore `MAvec`. Risolvere l'esercizio prima utilizzando la funzione `flip` e poi un ciclo `for` e poi tramite sequenza

Creare una matrice denominata `M` che contiene come prima colonna il vettore `MAvec` e come seconda colonna il vettore `MAvecScambiato`.

Calcolare la somma di ogni colonna della matrice `M` e mostrare il risultato nella Command Window

Calcolare la somma di ogni riga della matrice `M`

Calcolare la somma di tutti gli elementi della matrice `M`

Calcolare la somma dei quadrati di tutti gli elementi della matrice `M`

Calcolare il seno di tutti gli elementi della matrice `M` ed assegnare il risultato ad una matrice denominata `MAs`, mostrando l'output

"Plottare" tramite linee le due colonne delle matrici MAs.

Aggiungere al grafico la legenda "Prima colonna" e "Seconda colonna"

Inserire la legenda utilizzando la *best location*

Cancellare dalla memoria la matrice MAs.

Estrarre le prime 5 righe della matrice M in una nuova matrice denominata A (vedi pagina 51 del libro).

Mostrare la matrice A nella Command Window

Creare una matrice logica delle stesse dimensioni di A (denominata L) che contenga il valore vero (1) in corrispondenza degli elementi di A maggiori di 10. Per un'applicazione della logica booleana vedi esercizio 6.1 del libro.

Mostrare la matrice L.

Mostrare la classe della matrice L

Partendo dalla matrice A creare una table denominata Atable con i seguenti nomi riga

A, B, C, D e E ed i seguenti nomi colonna "Spesa" "Conteggio" (vedi pagine 43-44 del libro per un esercizio simile)

Assegnare alla table il nome Atable

Assegnare alla table la seguente descrizione "Questa è la mia prima table".

Mostrare la descrizione nella Command Window

Salvare tutto il Workspace nel file `MI0primo.mat` (vedi sezione 1.6.1 per ulteriori opzioni di salvataggio)

Tramite l'istruzione `who` mostrare tutte le variabili attualmente presenti nel Workspace

Cancellare tutte le variabili in memoria tranne `Atable`

Mostrare le variabili presenti nel workspace

Cancellare la variabile `Atable`.

Mostrare le variabili presenti in memoria (tutte le variabili sono state cancellate dalla memoria)

Caricare in memoria tutte le variabili presenti nel file `MIOpri.mo.mat`

## Operazioni con characters e stringhe

Fare il clear della memoria

Creare il character vector 'Oggi vado al mare' e denominarlo `ch`

Mostrare le dimensioni di `ch`

Estrarre gli elementi in posizione 6:9 e mostrarli

Mostrare i codici ascii di 'Oggi'.

Trasformare ch in una stringa e denominarlo ST

Concatenare ST con "BCDEFGHIJKLMNOPQRSTUVWXYZ[\]^\_`abcdefghijklmnopqrstuvwxyz";

Creare lo string array di seguito e denominarlo Mio

"20" "17" "14" "11" "8" "5" "2" ... "-145" "-148"

Partendo dallo string array precedente creare il char array di dimensione 6x2 denominato d riportato di seguito  
6x2 char array

'20'

'17'

'14'

'11'

'8 '

'5 '

Estrarre gli elementi della prima colonna del char array precedente e mostrarli nella Command Window

Partendo dai due string arrays di seguito

nomi = ["data" "report" "slides"]

estensioni = [".xlsx" ".docx" ".pptx"]

Creare lo string array di seguito

1x3 **string** array

"data.xlsx" "report.docx" "slides.pptx"