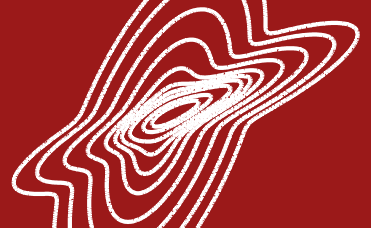


UNIVERSITÀ
DEGLI STUDI
DI PADOVA

METODI STATISTICI PER LA BIOINGEGNERIA

Laboratorio 2

A.A. 2024-2025

Enrico Longato

Grazie alla Prof.ssa Alessandra Bertoldo e al Dott. Marco Castellaro per la versione iniziale delle slide.

Dal lab 1: Account DEI

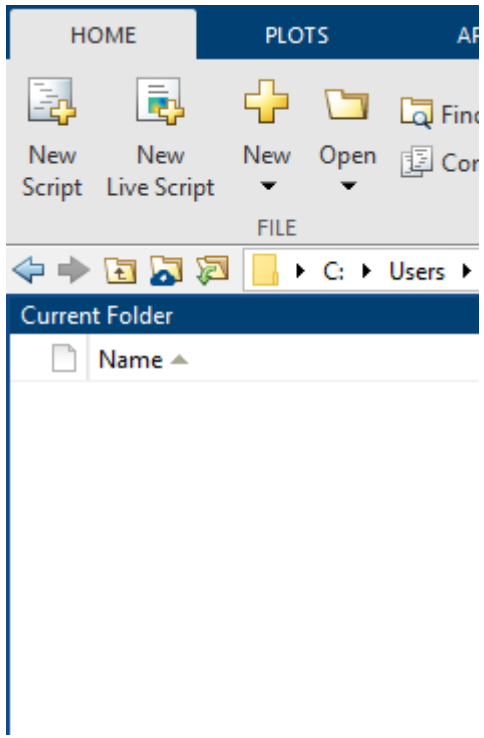
https://www.dei.unipd.it/account

RICHIESTA NUOVO ACCOUNT: STUDENTI (tratto dal link qui sopra)
Per gli STUDENTI: iscrizione ai laboratori informatici/creazione account studenti
Gli studenti dell'Università di Padova **iscritti ad uno dei corsi di laurea del Dipartimento di Ingegneria dell'Informazione** possono utilizzare i sistemi dei laboratori informatici disponendo di una login personale.

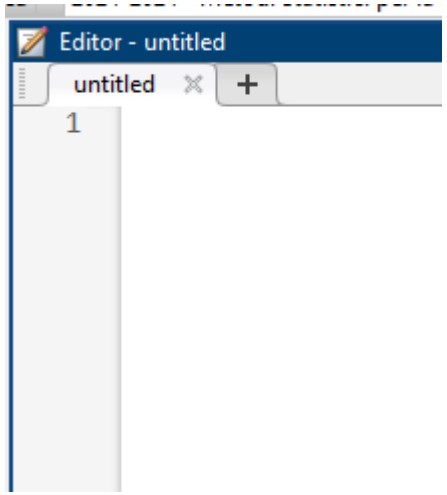
Per attivare l'account al DEI e poi impostare o resettare la password si segue la procedura informatizzata partendo da:
https://www.dei.unipd.it/nuovoaccount

N.B. Pagina accessibile tramite le credenziali SingleSignOn (UNIWEB @studenti.unipd.it)
Gli studenti ERASMUS o di altri corsi di laurea devono richiedere al loro professore Tutor del DEI di inviare una email ai Servizi Informatici.

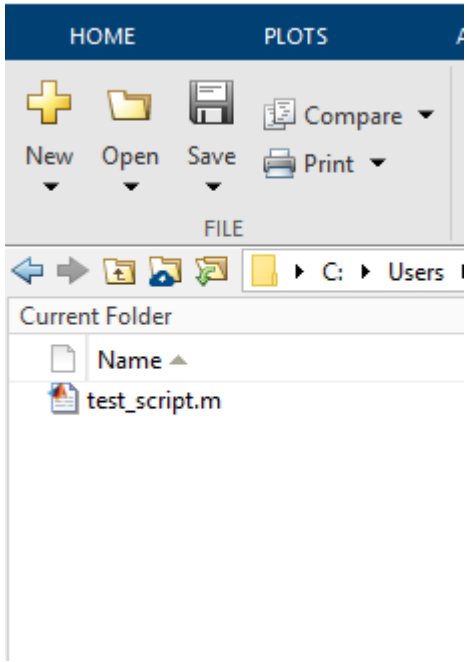
Dal lab 1: Script MATLAB



1) Clicko su "New script"

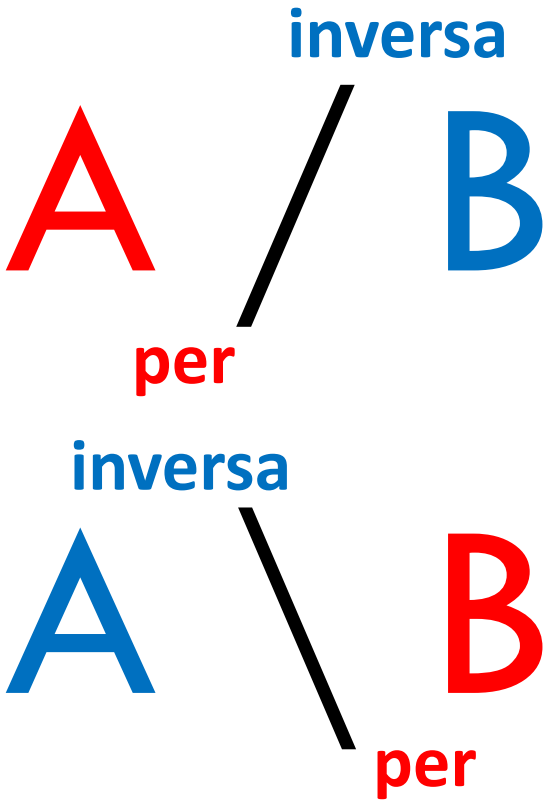


2) Si apre nell'editor uno script vuoto di nome "untitled", senza titolo



3) Clicko su "Save" e salvo lo script nella cartella corrente: apparirà dove al punto 1 non avevamo nulla

Dal lab 1: Operatori / e \



Per ricordarmi se il "per" va sopra o sotto, posso ricordarmi che la divisione tra scalari vuol dire "moltiplicare il primo numero per il reciproco (cioè l'inverso) del secondo". Così, desumo che A/B vuol dire "A per l'inverso di B" e, quindi, A\B è l'altro operatore.

Dal lab 1: "A per inversa di A"

```
>> A*inv(A) - A/A

ans =

   1.0e-14 *

   -0.0222   -0.0056   -0.0444   -0.0444
   -0.1332         0         0   -0.2220
    0.0222    0.0167         0    0.1332
   -0.0083   -0.0056    0.0333    0.0888
```

MATLAB segnala di preferire A/A perché, internamente, lo implementa con maggior precisione. Altrimenti, il calcolo viene impreciso di una quantità molto molto piccola (nell'ordine di 10^{-14}), a causa della **propagazione degli errori di approssimazione numerica del calcolatore** attraverso le moltiplicazioni e divisioni dell'inversione di matrice e del prodotto.

Dal lab 1: A(:) = …

```
A =

    -8     2     7     3
    -8    -9   -10     5
     9    -6   -10     3
    10    -3    -7    -1
```

Non vi stupisce che, indicizzando A per righe e colonne, si possa modificare A.

```
>> A(2, 2) = 999

A =

    -8     2     7     3
    -8   999   -10     5
     9    -6   -10     3
    10    -3    -7    -1
```

L'utilizzo di A(:) è concettualmente simile, soltanto che agisce su tutti gli elementi della matrice nell'ordine del find lineare.

Si può interpretare come "Assegna [=] ad A, srotolandola e riarrotolandola dopo l'assegnazione [A(:)], i valori della sequenza 1:16"

```
>> A(:) = 1:16

A =

     1     5     9    13
     2     6    10    14
     3     7    11    15
     4     8    12    16
```

Dal lab 1: Attenzione agli autogoal

```
%% Trovare minimo e massimo della matrice A e i loro indici
[min_A, i_min_A] = min(A(:)); % Sono il min e l'argmin
[r_min_A, c_min_A] = ind2sub(size(A), i_min_A); % Solita conversione

[max_A, i_max_A] = max(A(:)); % Sono il max e l'argmax
[r_max_A, c_max_A] = ind2sub(size(A), i_max_A); % Solita conversione
```

```
%% Ripetere l'operazione senza vettorizzare
[max_A_nonvec, i_max_A_nonvec] = max(A); % Notare che A non è "srotolata"

r_max_A_nonvec = i_max_A_nonvec; % Indice riga
c_max_A_nonvec = 1:size(A, 2); % Le colonne sono semplicemente 4

% Di default, le operazioni vettoriali sono svolte per colonne; in questo
% caso si poteva usare il parametro "dim", cioè il terzo, per ovviare e
% trovare il massimo di ciascuna riga.
[max_A_nonvec_by_rows, i_max_A_nonvec_by_rows] = max(A, [], 2);

r_max_A_nonvec_by_rows = 1:size(A, 1); % Le righe sono  4
c_max_A_nonvec_by_rows = i_max_A_nonvec_by_rows; % Indice colonna
```

Consultando l'help di MATLAB, si può venire a sapere che i due argomenti in uscita da max sono il valore massimo e l'indice a cui si trova il massimo.
Questo sempre, a prescindere dalla forma della matrice in ingresso A.
Tuttavia

- Se A è srotolata, la funzione max vede un vettore e, quindi, restituisce un unico massimo e un unico indice lineare.
- Se A è una matrice 2x2, la funzione max, di default, restituisce il massimo di ciascuna colonna e l'indice a cui quel massimo si trova all'interno della colonna.

Possibile autogoal: se non consulto correttamente l'help di MATLAB, potrei essere tentato di dire, per similitudine con il find, che, se A non è srotolata, i due argomenti sono l'indice riga e l'indice colonna a cui si trova il massimo.

- Questo si manifesta, solitamente, nel fatto che, a fronte della convinzione sbagliata di cui sopra, potrei assegnare un nome ingenuamente sbagliato agli argomenti di uscita (cioè chiamare indice riga quello che è il massimo e indice colonna quello che è l'indice riga, solo perché sono due argomenti)

Laboratorio 2: Contenuti e obiettivi

1. Esercitazione "alla lavagna"
   - Caricamento ed esplorazione dati
   - Rappresentazione dati in figura tramite boxplot e scatterplot
   - Struct di MATLAB
   - Comparazione tra stringhe
   - Statistica descrittiva
2. Esercizi da svolgere in autonomia (per superare la "paura del file bianco")
   - Selezione sottoporzioni di dati
   - Istogrammi, frequenze relative e assolute

Operazioni preliminari

I dati sono caricati sulla pagina elearning del corso, nella stessa cartella del testo dell'esercitazione.

Prima di svolgere gli esercizi

1. Scaricare eventuali file di dati (.mat)
2. Scaricare eventuali file sorgente aggiuntivi forniti dal docente (.m)
3. Mettere tutti questi file nella stessa cartella degli script che creerete durante l'esercitazione *

* Se non fate così, sta a voi caricarli utilizzando il percorso all'interno del filesystem (facile per i dati, più complesso per gli script).

CONTESTO BIOLOGICO DELL'ESERCITAZIONE

L'assorbimento del glucosio all'interno delle cellule è necessario perché esse producano energia ed è mediato da alcuni ormoni, tra cui l'insulina.

- Banalizzando all'estremo: meno insulina = meno glucosio nelle cellule = più glucosio nel sangue.

Il diabete è una malattia metabolica caratterizzata da una concentrazione elevata di glucosio nel sangue.

- Esistono due tipi principali di diabete:
  - Tipo 1, in cui il pancreas non produce insulina
  - Tipo 2 in cui, per diverse ragioni e con diversi meccanismi, l'effetto dell'insulina è inferiore rispetto a quello fisiologicamente necessario.

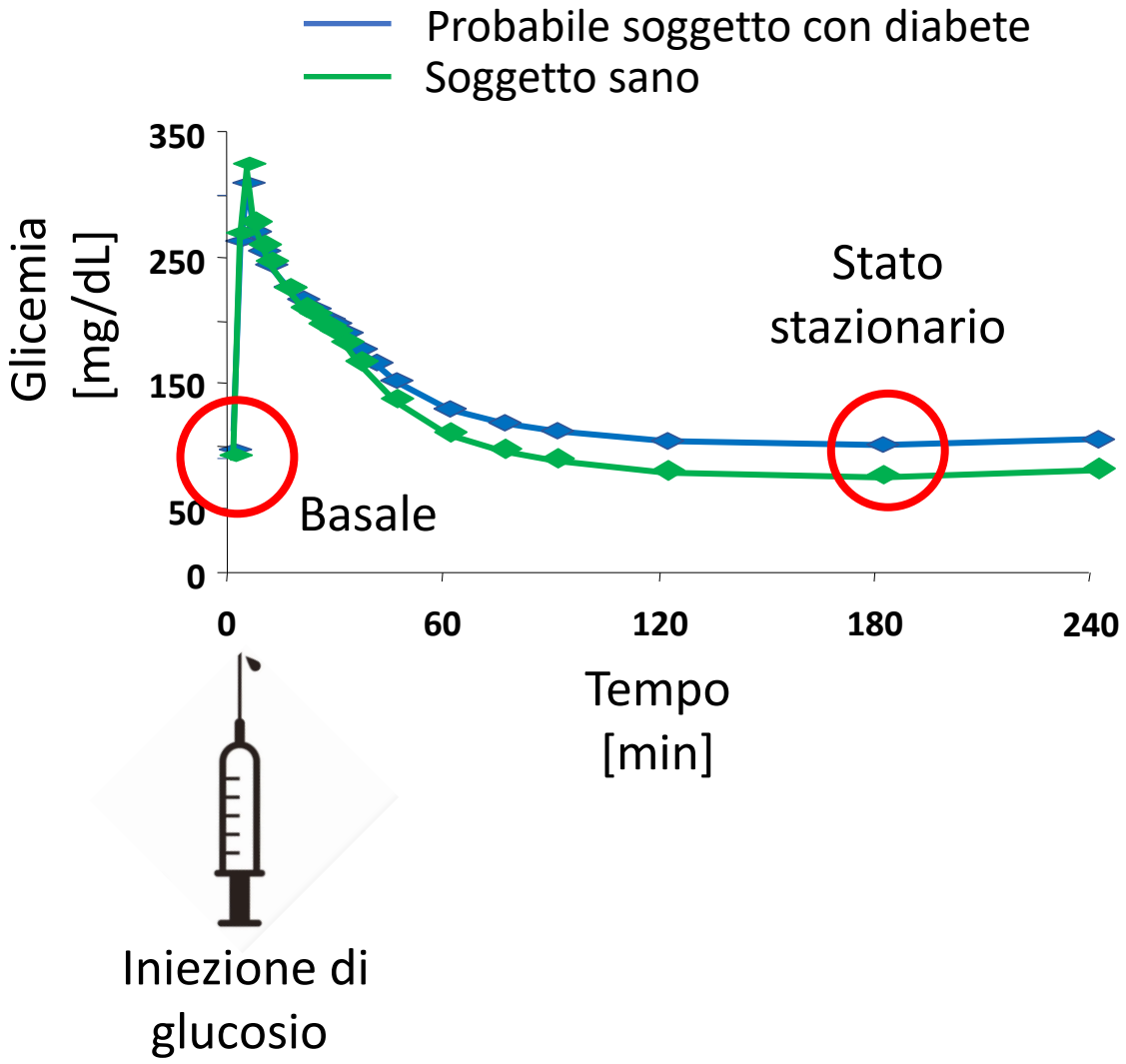
CONTESTO OPERATIVO DELL'ESERCITAZIONE

I dati a disposizione per il laboratorio provengono da un centro clinico americano e sono relativi ad un test per la determinazione della sensibilità insulinica, noto come Intravenous Glucose Tollerance Test (IVGTT)

Schema semplificato dell'IVGTT

1. Dopo un periodo di digiuno di almeno 12 ore, viene effettuata un'iniezione di glucosio.
2. A seguito di questa iniezione di glucosio, naturalmente, la concentrazione del glucosio in circolo (= la glicemia) aumenta.
3. L'organismo, quindi, cerca di riportare la glicemia nel range euglicemico (<100 mg/dL per soggetti sani)
4. Per campionare questo fenomeno, vengono effettuati N prelievi di sangue.
5. Sulla base della capacità del soggetto di far rientrare il glucosio entro livelli fisiologici, si determina se questi è o non è probabilmente affetto da diabete.

DATI A DISPOSIZIONE



Abbiamo a disposizione i dati di concentrazione di glucosio (glicemia) corrispondente a due prelievi:

1) **Basale (cioè a digiuno, prima dell'esperimento)**
2) **Allo stato stazionario (cioè quando l'organismo dovrebbe aver ripristinato il livello basale)**

I dati sono contenuti nel file **data_IVGTT.mat**

- **raw_data:** matrice double 2D (una variabile = quantità misurata per colonna)
- **labels:** cell array con le etichette delle colonne (= nomi delle variabili)

NB: come accade nella vita reale e nella professione, nel file sono presenti anche quantità misurate (insulina, C-peptide, …) che non sono di interesse per il laboratorio; impareremo subito a gestire questa situazione.

Prima di svolgere l'esercitazione (oppure al bisogno), utilizzare il comando help di MATLAB seguito dal nome delle seguenti function, utili allo svolgimento degli esercizi.

Parte 1 (svolto)

- load, find, figure, subplot, xlabel, ylabel, scatter, boxplot, strcmp

Non possiamo fare s1 == s2 o cose simili; allora, usiamo strcmp

Find Text in Cell Array

Find the word 'upon' in a cell array of character vectors.

```
s1 = 'upon';
s2 = {'Once','upon', 'a','time'};
tf = strcmp(s1,s2)
```

```
tf = 1×4 logical array
     0   1   0   0
```

Parte 2 (svolto)

- struct, mean, mode, median

Parte 3 (proposto)

- --

Parte 4 (proposto)

- hist, bar

PARTE 1: CARICAMENTO E VISUALIZZAZIONE (svolto)

- Caricare i dati (nel file ***data_IVGTT.mat***)

- Estrarre le sole variabili e le relative etichette di interesse (**`Gss_IVGTT`** = glicemia allo stato stazionario in mg/dL e **`Ib_IVGTT`** = insulina basale in pmol/mL) e memorizzare, rispettivamente, le variabili nella matrice **`subset_data`** e le etichette nel cell array **`subset_label`**
- Aggiungere alle etichette così estratte le unità di misura.

- Visualizzare tramite **`scatter`** le due variabili (una sull'asse x e l'altra sull'asse y).

- Visualizzare con dei **`boxplot`**, in figure separate, le variabili con le relative etichette (si realizzi ***un ciclo for***)
- Visualizzare con dei **`boxplot`**, nella stessa figura, ma in due **`subplot`** diversi, le variabili con le relative etichette

PARTE 2: STATISTICA DESCRITTIVA (svolto)

- Calcolare, per ogni variabile selezionata, la media (`mean`), la moda (`mode`) e la mediana (`median`) e inserire i risultati in una `struct` `stats_data` con campi (ciascuno un vettore di tanti elementi quanti sono le variabili selezionate) `stats_data.mean, stats_data.mode e stats_data.median`

Struttura attesa:

```
stats_data =

  struct with fields:

      mean: [82.6740 27.3431]
      mode: [84 19]
    median: [82.8000 24]
```

PARTE 3: RICERCHE NEI DATI TRAMITE FIND E SUE ALTERNATIVE (proposto)

- Cercare tutti i soggetti i cui campioni di glucosio allo stato stazionario (**`Gss_IVGTT`**) sono maggiori di 90 mg/dL (**`find`**)

- Mostrare con un **`boxplot`** solo i valori delle variabili dei soggetti individuati al punto precedente

- Provare a non utilizzare **`find`**, ma direttamente l'array booleano che inseriamo come condizione logica nel **`find`** per effettuare la stessa operazione.

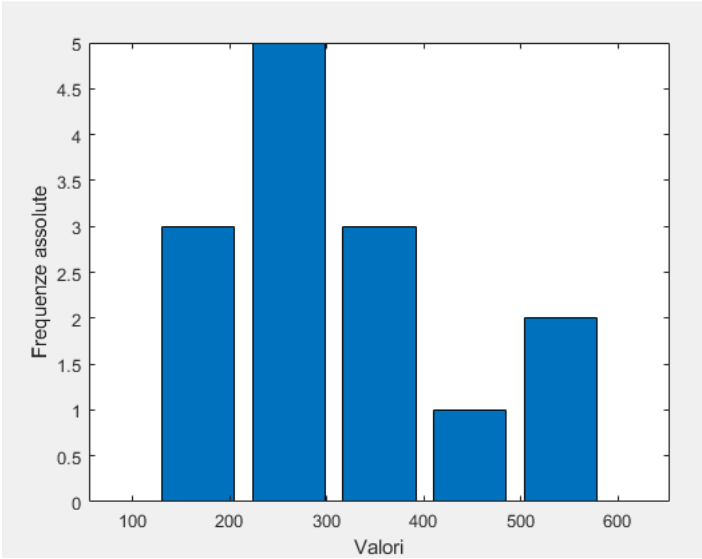
PARTE 4: ISTOGRAMMI, FREQUENZE ASSOLUTE E RELATIVE (proposto)

- Usando il solito ciclo for che scandisce entrambe le variabili selezionate:
  - Creare l'istogramma (**`hist`**) delle frequenze assolute usando 20 bin
  - Creare il corrispondente istogramma delle frequenze relative
  - Disegnare tutti gli istogrammi nella stessa figura (2x2 **`subplot`**) con l'istruzione **`bar`**

```
vettore = [1.2 2.54 3.2 4 2.1 4.35 2.6 1.8 4.9 2.2 3.3 1 2 1] * 120;
numero_di_bin = 5;

[freq_assolute, centri_delle_barre] = hist(vettore, numero_di_bin);

figure
bar(centri_delle_barre, freq_assolute)
xlabel('Valori')
ylabel('Frequenze assolute')
```



- Le due variabili possono essere descritte tramite una funzione di densità di probabilità gaussiana? Per ora, rispondere "a occhio"