METODI STATISTICI PER LA BIOINGEGNERIA

Laboratorio 3

A.A. 2026-2027

Enrico Longato

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

Dal lab 2: Prediligere le operazioni vettoriali

Per abitudine o per non uscire dalla propria comfort zone, potreste essere tentate o tentati di implementare le condizioni logiche con un ciclo for, invece che con l'operazione booleana vettoriale corrispondente.

PER FAVORE NON FATELO!

Oggi, in laboratorio, non c'è problema; domani, all'esame, in preda all'ansia, è molto facile sbagliare l'indicizzazione.

```
i_gss_high = find(subset_data(:, 1) > 90);
```



Dal lab 2: Libertà nei plot

Per i plot, anche all'esame, usate le funzioni che volete!

Basta che riusciate a rappresentare e/o a dedurre quanto richiesto dalla consegna.

Indifferente!

hist

(Not recommended) Histogram plot

`hist` is not recommended. Use `histogram` instead

For more information, including suggestions on updating code, see Repla

Syntax

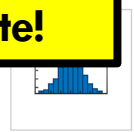
```
hist(x)
hist(x,nbins)
hist(x,xbins)

hist(ax, __ )

counts = hist( __ )
[counts,centers] = hist( __ )
```

histogram

Histogram plot



Description

Histograms are a type of bar plot that group data into bins. After you create a Histogra

Creation

Syntax

```
histogram(X)
histogram(X,nbins)
histogram(X,edges)
histogram('BinEdges',edges,'BinCounts',counts)

histogram(C)
histogram(C,Categories)
histogram('Categories',Categories,'BinCounts',counts)

histogram( __ ,Name,Value)
histogram(ax, __ )
h = histogram( __ )
```

Dal lab 2: Soggetti e variabili

age	bmi	waist	heart rate	ferritin	c reactive protein	haemoglobin	fibrinogen	glucose
57	32.427	113.3	49.5	156	2.6	16.9	3.4	NaN
68	31.201	106.7	71	168	3.4	15	3.6	82.8
79	30.033	118.8	63	28	2	17.2	4.1	84.6
77	29.017	103.7	66.5	147	5.7	15.5	3	99
73	32.198	106.2	60.5	NaN	NaN	NaN	NaN	NaN
61	30.409	111.7	41.5	490	1.2	15.6	3.1	162
73	29.006	98.8	45.5	108	0.8	16.4	3.3	NaN
63	23.238	76.7	51	134	0.2	13.2	2.8	NaN
67	28.107	105.3	55.5	450	0.6	15.9	3.5	NaN
76	28.103	89.6	52.5	NaN	NaN	NaN	NaN	NaN
60	28.236	100.4	72	NaN	NaN	NaN	NaN	NaN
62	26.341	105.8	51	287	1	14.7	2.9	79.2
78	28.05	98.6	66	72	1.6	12.6	2.9	86.4
:	:	:	:	:	:	:	:	:
62	25.184	96.8	46	NaN	NaN	NaN	NaN	NaN
78	NaN	104.5	83.5	69	1.5	16.5	2.7	88.2
55	31.141	102.15	23	160	8.9	12.1	3.8	NaN
85	30.115	108.8	93.5	206	1.6	14.9	3.1	NaN
75	28.995	112.4	63.5	126	2.6	16.2	2.9	90
65	24.452	89.6	49	NaN	NaN	NaN	NaN	NaN
76	24.547	93.1	54.5	14	0.4	12.7	3.1	NaN
67	21.265	92.65	79	344	0.3	NaN	3.2	73.8
62	24.031	83	45	191	1.5	13	2.8	79.2
55	25.766	83.45	50.5	NaN	NaN	NaN	NaN	NaN
86	28.017	94.75	51.5	164	2.2	11.9	3.7	NaN
71	25.179	82.5	77.5	115	0.7	14.4	3.4	81
54	23.706	82.75	81.5	27	1.1	14	3.6	97.2

Colonne = variabili

NB: noi avremo sempre le etichette in un cell array e i dati in una matrice; la struttura qui accanto è una table (non in programma).

Righe = soggetti

(o altra unità di osservazione, es. visite, ...)

[Eccezione (non in programma): per convenzione, i dati -omici (genomica, proteomica, transcrittomica, ecc.) solitamente hanno i soggetti sulle colonne e i geni (o simili) sulle righe]

Dal lab 2: Estensione dinamica di un vettore

```
%% Utilizzo tipico del ciclo for: vettori "autoestendenti"
vettore_da_estendere = [];
for i=1:10
    elemento_da_inserire = i^2; % Qualunque cosa
    vettore_da_estendere = [vettore_da_estendere elemento_da_inserire];
end

% Altro modo: MATLAB capisce da solo che deve estendere il vettore
vettore_da_estendere_2 = [];
for i=1:10
    elemento_da_inserire = i^2; % Qualunque cosa
    vettore_da_estendere_2(i) = elemento_da_inserire;
end
risultato_atteso = (1:10).^2; % Operazione vettoriale equivalente
```

Concatenare un vettore con se stesso esegue l'operazione di "append" (come in python: mylist.append(myvalue))

In ogni caso, per essere esteso dinamicamente, il vettore deve esistere (qui inizializzato vuoto)

Tipico di MATLAB: assegnare a un indice "out of bounds" non dà errore (**ATTENZIONE!!**), bensì, prima, **estende il vettore** quanto necessario e, **poi, inserisce il valore** al posto giusto. (viceversa, accedere a un elemento "out of bounds" dà errore come ci si poteva aspettare)

Dal lab 2: Concatenazione di stringhe

```
>> ['prima stringa', ' <-- spazio --> ', 'seconda stringa']

ans =

    'prima stringa <-- spazio --> seconda stringa'

>> ['prima stringa' ' <-- spazio --> ' 'seconda stringa']

ans =

    'prima stringa <-- spazio --> seconda stringa'

>> ['prima stringa' ' <-- spazio --> ' num2str(pi)]

ans =

    'prima stringa <-- spazio --> 3.1416'
```

Il problema è lo spazio!

```
>> ['attenzione che l''apice è la trasposizione' num2str(pi)'errore']
 ['attenzione che l''apice è la trasposizione' num2str(pi)'errore']
                                                              ↑

>> ['attenzione che l''apice è la trasposizione' num2str(pi) 'errore']

ans =

    'attenzione che l'apice è la trasposizione3.1416errore'
```

```
>> stringa_con_apice = 'attenzione che l''apice va escapato con il doppio apice'

stringa_con_apice =

    'attenzione che l'apice va escapato con il doppio apice'
```

L'escape dei caratteri speciali è l'operazione che tipicamente si fa con il backslash

Dal lab 2: Accesso a cell array

```
>> cell_array = {'prima stringa', 'seconda stringa', 'terza stringa'}

cell_array =

  1×3 cell array

    {'prima stringa'}    {'seconda stringa'}    {'terza stringa'}
```

```
>> cell_array(2)

ans =

  1×1 cell array

    {'seconda stringa'}
```

```
>> cell_array{2}

ans =

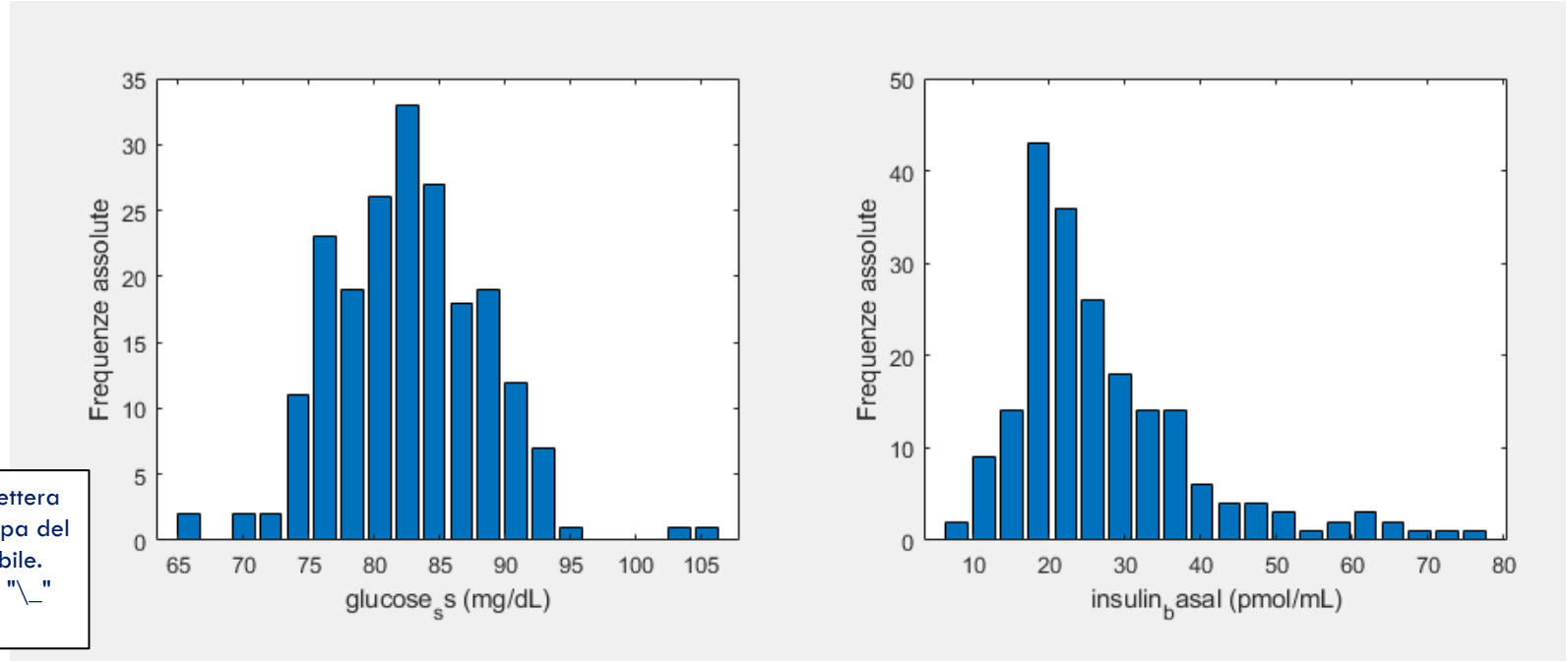
    'seconda stringa'
```

Parentesi tonde: restituisce un sottoarray del cell array (poco utile ai nostri scopi)

Parentesi graffe: restituisce il contenuto della cella (quello che tipicamente serve a noi)

Cosa fare se non mi ricordo all'esame?
In questo caso e "per la vita", provare con un caso semplice nella command window!

Dal lab 2: Gaussianità "a occhio"



Non preoccupatevi della lettera che è finita al pedice: è colpa del "_" nel nome della variabile. Potevamo trasformarlo in "_" per risolvere.

La variabile di sinistra (glucosio allo stato stazionario) potrebbe essere gaussiana: è simmetrica e non sembra troppo acuta o spanciata.

La variabile di destra (insulina basale) è molto difficile che sia gaussiana: già "ad occhio" vediamo che è troppo asimmetrica.

- ⇒ Ulteriori verifiche possibili con skewness e curtosi
- ⇒ Esistono test di statistica inferenziale per escludere la gaussianità con alto livello di confidenza (<u>non</u> ne esistono per essere certi della gaussianità, ma vedremo questa sottigliezza a lezione + al prossimo laboratorio)

Laboratorio 3: Contenuti e obiettivi

1. Esercitazione "alla lavagna"
   - Caricamento ed esplorazione dati
   - Identificazione di dati mancanti, infiniti e implausibili sull'intera tabella
   - Identificazione di dati mancanti, infiniti e implausibili per colonna/variabile
   - Creazione di sottomatrici
2. Esercizi da svolgere in autonomia (per superare la "paura del file bianco")
   - Identificazione di dati mancanti, infiniti e implausibili per soggetto/riga
   - Sostituzione di valori in posizione specifiche della matrice
   - Confronto tramite media/varianza/mediana e istogrammi

IL PRE-PROCESSING DEI DATI

Nella vita reale e nella professione, c'è sempre un divario tra la forma in cui i dati vengono acquisiti o ci vengono forniti e quella necessaria per la loro analisi.

Garbage in, garbage out
Più educatamente: "Se i dati in ingresso sono di scarsa qualità, il risultato di un'analisi (anche se condotta perfettamente dal punto di vista matematico/operativo) non potrà che essere fallace."

Per evitare questo fenomeno, si effettua il cosiddetto pre-processing dei dati.

- Il pre-processing è tipicamente la componente maggioritaria, a livello di tempo, di qualunque analisi.
- Tra le attività di pre-processing includiamo
  - Esplorazione dei dati (numerosità, tipo, statistiche descrittive, outlier, …)
  - Identificazione di valori inattendibili (es., perché non fisiologici o perché chiari errori di misura)
  - Studio dei dati mancanti (numerosità, esclusione di soggetti o variabili, ipotetiche strategie per ridurre l'impatto della "missingness")
  - (molto altro)

CONTESTO DELL'ESERCITAZIONE

Caso di studio reale: il database ELSA

ELSA (English Longitudinal Study of Ageing) è uno studio epidemiologico multidisciplinare su un campione rappresentativo di persone residenti in Inghilterra con età maggiore di 50 anni.

Il suo scopo è capire le complesse dinamiche dell'invecchiamento che intercorrono tra le relazioni familiari, affettive, economiche, sociali, biologiche e come queste si legano alla salute e al benessere delle persone.

ELSA English Longitudinal Study of Ageing

Measures

Demographic data
Including Ethnicity, Marital status, and Education

Income and assets
Including Earnings, Pensions, and Housing wealth

Employment
Including Employment status, Job details, and Reasons for retirement

Psychosocial factors
Including Control, Demand, Effort-reward balance

Consumption
Including Transfers, Housing, and Expenditures

Expectations
Including on Mortality, Employment, and Finance

Social and civic participation
Including Transport, Social support, and Loneliness

Physical Health
Including Mobility, Pain, and Disability

Behavioural health
Including Smoking, Physical Activity, and Sleep

Mental health
Including Psychiatric problems, General Health Questionnaire, and CES-D depression scale

Psychological and social well-being
Including Quality of life (CASP-19) and Personality

Cognitive Function
Including Memory, Literacy, and Fluid intelligence

Psychosocial factors
Including Control, and Effort-reward balance

Physical examination and performance
Including Blood pressure, Grip strength, and Balance

https://www.elsa-project.ac.uk/

Un articolo scientifico del nostro gruppo sul database ELSA: Roversi, C., Tavazzi, E., Vettoretti, M. *et al.* A dynamic probabilistic model of the onset and interaction of cardio-metabolic comorbidities on an ageing adult population. *Sci Rep* **14,** 11514 (2024). https://doi.org/10.1038/s41598-024-61135-x

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)
- isnan, isinf, mean, var, sum

Parte 2 (svolto)
- any

Parte 3 (proposto)
- --

Parte 4 (proposto)
median, nanmedian

Parte 5 (proposto)
- --

PARTE 1: CARICAMENTO E ANALISI ESPLORATIVA NAIVE (svolto)

- Caricare il file `ELSA.mat`
  - `elsa` = matrice di dimensione 1000x16 ("mille soggetti per sedici variabili")
  - `elsa_labels` = cell array di etichette con i nomi delle 16 variabili
  - `elsa_units` = cell array con le unità di misura corrispondenti

- Ispezionare visivamente i dati: per ciascuna variabile (= colonna) stabilire se
  - I valori ivi memorizzati siano (apparentemente) discreti o continui
  - Eventuali valori negativi siano plausibili

- Effettuare un'analisi naïve sull'intera tabella, individuando e contando (istruzioni `find`, `length` e `sum`)
  - I valori mancanti (`NaN`), con la funzione `isnan`
  - I valori infiniti (`Inf`), con la funzione `isinf`
  - I valori negativi, con la consueta indicizzazione logica

- Provare a calcolare la media e la varianza di ciascuna colonna. Cosa si nota "di strano"? Perché succede?

PARTE 2: PULIZIA DEI DATI A LIVELLO DI VARIABILE/COLONNA (svolto)

Poiché i valori infiniti e negativi non sono plausibili per nessuna delle variabili, per semplificare le analisi, possiamo equipararli al valore `NaN`, che, convenzionalmente, indentifica i dati mancanti.

<u>Ragionamento</u>: se un valore è implausibile è come se non fosse stato registrato.

- Sostituire tutti i valori infiniti e negativi della matrice `elsa` con `NaN`
- Calcolare la percentuale di valori mancanti per ogni variabile (= per ciascuna colonna)

- Identificare le colonne corrispondenti a variabili con più del 20% di dati mancanti e memorizzare una nuova matrice senza quelle colonne in una variabile `elsa_reduced`. Soluzione: dovrebbero rimanere 6 colonne.

PARTE 3: PULIZIA DEI DATI A LIVELLO DI SOGGETTO/RIGA (proposto)

- A partire dalla matrice `elsa_reduced` creata al punto precedente, identificare i soggetti (= righe) che hanno valori mancanti in almeno una delle variabili. **Suggerimento**: dopo aver applicato `isnan`, usare la funzione `sum` oppure `any` con il secondo argomento = 2 per modificare il comportamento di default e farle agire sulle righe; per fare allenamento, provare a consultare **autonomamente** l'help di MATLAB)

- Memorizzare in una nuova matrice `elsa_reduced_filtered` la sottomatrice di `elsa_reduced` che corrisponde ai soggetti senza dati mancanti. Soluzione: le dimensioni finali dovrebbero essere 826x6.

NB: l'esercizio è appositamente lasciato per lo svolgimento autonomo in aula. L'obiettivo è prendere confidenza con due sensazioni tipiche dell'esame al calcolatore e capire come porvi rimedio:

1. Trovarsi di fronte a una leggera variante di un esercizio svolto in aula (la parte 3 non è altro che la parte 2 fatta "per righe" invece che "per colonne"), che, però, è concettualmente diversa (togliere soggetti contro togliere variabili).
   - Rimedio cercando di rifarmi quanto più possibile a quello che so e implementando solo le minime variazioni richieste.
2. La dimenticanza (momentanea, all'esame) della sintassi di MATLAB per eseguire operazioni semplici, ma "non di default" (in questo caso, l'utilizzo di sum o any per righe e non per colonne, specificando la dimensione lungo cui operare).
   - Rimedio ricordandomi che esiste l'help di MATLAB e abituandomi a consultarlo in autonomia.

PARTE 4: UNA PRIMA IMPUTAZIONE DATI (proposto)

Imputare i dati significa trovare un valore sensato da sostituire ai dati mancanti (tipicamente, le analisi successive al pre-processing richiedono che non ci siano dati mancanti).
L'imputazione è un ambito di ricerca piuttosto complesso che esula dagli obiettivi del corso. Qui, una semplice imputazione ci è utile per fare esercizio sulla statistica descrittiva e l'indicizzazione delle matrici.

- Sostituire ai valori mancanti il valore mediano dei dati non mancanti della colonna a cui appartengono. **Suggerimento:** in un ciclo for che scandisce le colonne della matrice elsa, usare la funzione `nanmedian` oppure calcolare la mediana sui soli valori non `NaN`.

PARTE 5: VISUALIZZAZIONE E CONFRONTO (proposto)

- Confrontare numerosità, media, varianza, mediana e istogrammi dei dati originariamente non mancanti di ciascuna colonna e di quelli dopo l'imputazione con la mediana. NB: ricordarsi le unità di misura nel cell array `elsa_units`.