

Relazione

Introduzione

Il programma realizzato su MATLAB è un'implementazione del processo di calcolo dei quartili, dei quantili e di alcuni indici di dispersione.

Il concetto di quartile deriva da quello di mediana: data una popolazione, la mediana è il valore della popolazione stessa che la divide in due parti di uguale numerosità e viene denotata con m . Le mediane delle due sottopopolazioni ottenute dividendo la popolazione complessiva tramite la mediana centrale, insieme a quest'ultima, costituiscono i quartili.

Si definisce quantile di ordine $\alpha \in [0,1]$, e lo si indica con q_α , un valore per cui a sinistra compare almeno il $100\alpha\%$ delle osservazioni e alla sua destra il $100(1-\alpha)\%$. Inoltre, si usa il termine percentile, al posto di quantile, se α è espresso in percentuale.

Invece, si dicono primo quartile, secondo quartile e terzo quartile i quantili rispettivamente di ordine 0.25, 0.5 e 0.75, con il secondo quartile che rappresenta la mediana della popolazione considerata, il primo quartile che è la mediana della sottopopolazione di sinistra e il terzo quartile che coincide con la mediana della sottopopolazione di destra.

Effettuando misure di tipo numerico su una determinata popolazione, si possono notare quattro dati fondamentali:

1. I valori assunti dalla popolazione, che possono essere espressi come intervalli o cifre numeriche e che vengono indicati con X_1, X_2, \dots, X_n ;
2. Le frequenze assolute associate ai valori della popolazione, indicate con f_1, f_2, \dots, f_n ;
3. Le frequenze relative, dette anche probabilità empiriche, indicate con p_1, p_2, \dots, p_n ;
4. Le frequenze cumulate, che sono funzioni di ripartizione empiriche costruibili una volta note le frequenze relative e che si indicano con F_1, F_2, \dots, F_n .

Esiste una tabella che contiene i dati appena elencati nel medesimo ordine con cui sono stati presentati, conosciuta con il nome di tabella di raccolta dei dati.

Quando il carattere esaminato è di tipo continuo, ossia assume valori reali, o discreto con un numero elevato di modalità, conviene raggruppare i dati in classi, che, di solito, hanno tutte la stessa ampiezza.

Per comprendere in quante classi bisogna suddividere la popolazione, esistono due semplici regole:

1. Scegliere un numero di classi $k \approx \lceil \sqrt{n} \rceil + 1$, dove n è il numero dei dati e $\lceil \sqrt{n} \rceil$ coincide con la parte intera di \sqrt{n} ;
2. Scegliere l'ampiezza delle classi come pari ad $a = R/k$, dove R è il campo di variazione, cioè la differenza tra il valore massimo e quello minimo assunto dai dati.

Gli indici di dispersione scelti sono: momento centratò empirico di ordine 4, varianza empirica, deviazione standard empirica, indice di Kurtosi e differenza interquartile.

Raggruppamento per classi di valori

Sia $\alpha \in [0,1]$, si vuole determinare il quantile di ordine α . Se α coincide con uno dei valori della frequenza cumulata, è facile da determinare $F(q_\alpha) = \alpha$. Altrimenti bisogna individuare q_α mediante un'approssimazione lineare. q_α è l'ascissa del punto di intersezione tra l'approssimazione lineare e la retta di equazione $y=\alpha$:

1. Si individua l'intervallo in corrispondenza del quale cade il quantile $q_\alpha: [x_{j-1}, x_j]$, ossia $F_{j-1} \leq \alpha \leq F_j$;

2. Si scrive l'equazione della retta passante per i punti (x_{j-1}, F_{j-1}) e (x_j, F_j) , che è:

$$\frac{F(x) - F_{j-1}}{F_j - F_{j-1}} = \frac{x - x_{j-1}}{x_j - x_{j-1}} \rightarrow F(x) - F_{j-1} = \frac{F_j - F_{j-1}}{x_j - x_{j-1}} * (x - x_{j-1}) \rightarrow F(x) = F_{j-1} + \frac{F_j - F_{j-1}}{x_j - x_{j-1}} * (x - x_{j-1});$$

3. Si determina l'intersezione della retta ottenuta al passo precedente con la retta di equazione $F(x)=\alpha$:

$$\begin{cases} F(x) = \alpha \\ F(x) = F_{j-1} + \frac{F_j - F_{j-1}}{x_j - x_{j-1}} * (x - x_{j-1}) \end{cases} \rightarrow \begin{cases} F(x) = \alpha \\ F_{j-1} + \frac{F_j - F_{j-1}}{x_j - x_{j-1}} * (x - x_{j-1}) = \alpha \end{cases} \rightarrow$$

$$\begin{cases} F(x) = \alpha \\ \frac{F_j - F_{j-1}}{x_j - x_{j-1}} * (x - x_{j-1}) = \alpha - F_{j-1} \end{cases} \rightarrow \begin{cases} F(x) = \alpha \\ x - x_{j-1} = (\alpha - F_{j-1}) * \frac{x_j - x_{j-1}}{F_j - F_{j-1}} \end{cases} \rightarrow$$

$$\begin{cases} F(x) = \alpha \\ x = x_{j-1} + (\alpha - F_{j-1}) * \frac{x_j - x_{j-1}}{F_j - F_{j-1}} \end{cases}$$

Primo esempio da considerare

Vengono effettuate delle misure circa la concentrazione di polveri sottili nell'aria.

Classi	f_i	p_i	F_i
[0, 0.5[10	0.37	0.37
[0.5, 1[8	0.3	0.67
[1, 1.5[6	0.2	0.87
[1.5, 2[3	0.1	≈ 1

Si calcolano i quartili:

- 1° quartile: $q_{0.25} = x_{j-1} + \frac{\alpha - F_{j-1}}{F_j - F_{j-1}} * (x_j - x_{j-1}) = 0 + \frac{0.25 - 0}{0.37 - 0} * (0.5 - 0) = 0.34$;
- 2° quartile: $q_{0.5} = x_{j-1} + \frac{\alpha - F_{j-1}}{F_j - F_{j-1}} * (x_j - x_{j-1}) = 0.5 + \frac{0.5 - 0.37}{0.67 - 0.37} * (1 - 0.5) = 0.72$;
- 3° quartile: $q_{0.75} = x_{j-1} + \frac{\alpha - F_{j-1}}{F_j - F_{j-1}} * (x_j - x_{j-1}) = 1 + \frac{0.75 - 0.67}{0.87 - 0.67} * (1.5 - 1) = 1.2$.

Secondo esempio da considerare

Questo esempio appartiene al caso in cui si hanno dati puntuali ma con frequenze assolute maggiori di 1: Il regolamento di ammissione di un certo corso di laurea prevede un bonus di 10 punti agli studenti che hanno riportato un voto non inferiore al novantesimo

percentile e bonus di 5 punti agli studenti che hanno riportato un voto non inferiore all'ottantesimo percentile e inferiore al novantesimo percentile.

Voto	5	6	7.5	8.5	9	10
fi	1	5	6	3	4	1
pi	0.05	0.25	0.3	0.15	0.2	0.05
Fi	0.05	0.3	0.6	0.75	0.95	1

Si definiscono le classi:

x	5	6	7.5	8.5	9	10
F(x)	0.05	0.3	0.6	0.75	0.95	1

- 1° classe: [0, 0.05[;
- 2° classe: [0.05, 0.3[;
- 3° classe: [0.3, 0.6[;
- 4° classe: [0.6, 0.75[;
- 5° classe: [0.75, 0.95[;
- 6° classe: [0.95, 1[.

Si calcolano i quantili:

1. Quantile relativo al ottantesimo percentile: $q_{0.8} = x_{j-1} + \frac{\alpha - F_{j-1}}{F_j - F_{j-1}} * (x_j - x_{j-1}) = 8.5 + \frac{0.8 - 0.75}{0.95 - 0.75} * (9 - 8.5) = 8.62;$
2. Quantile relativo al novantesimo percentile: $q_{0.9} = x_{j-1} + \frac{\alpha - F_{j-1}}{F_j - F_{j-1}} * (x_j - x_{j-1}) = 8.5 + \frac{0.9 - 0.75}{0.95 - 0.75} * (9 - 8.5) = 8.875;$

Implementazione su MATLAB

Per l'implementazione processo di calcolo dei quartili e dei quantili su MATLAB, sono stati realizzati 12 file di estensione ".m" e 5 dataset di estensione ".tsv": i file ".m" comprendono un main, che coincide con il file da mandare in esecuzione per far partire il programma, e 11 funzioni di assistenza al main stesso, che contengono operazioni di controllo dei valori inseriti in input, di riempimento delle tabelle di raccolta dati per il calcolo dei quartili e dei quantili e di esecuzione vera e propria di tutti i passaggi precedentemente illustrati; i file ".tsv" sono raggruppamenti di informazioni inerenti sport, prodotti di elettronica e argomenti di carattere più generale.

Il nome del file corrispondente al main è "main_quartili_e_quantili.m". Al suo interno, è possibile trovare nell'ordine:

- I passaggi di rimozione degli elementi dal workspace, di chiusura delle figure e dei grafici aperti e di pulizia della finestra di comando, rispettivamente attraverso "clear all", "close all" e "clc";
- L'inizializzazione del numero di dataset proposti, del numero massimo di classi e valori considerabili e del limite minimo da cui partire per l'inserimento di tali valori;

- Il messaggio di esplicitazione del processo implementato, con annesso comando per interrompere in anticipo l'esecuzione del programma;
- Il richiamo della funzione “dati_input”;
- Il richiamo delle funzioni contenenti procedure che dipendono dai valori ottenuti come output della funzione “dati_input”. In particolare, le funzioni richiamate sono:
 1. “procedura_intervalli_con_e_senza_dataset”, nel caso in cui si decida di esprimere le classi come intervalli, indipendentemente dall'utilizzo o meno di uno dei dataset proposti;
 2. “procedura_valori_con_e_senza_dataset”, nel caso in cui si voglia esprimere le classi come valori, anche in questo caso in maniera indipendente dall'utilizzo o meno dei dataset disponibili.
- La rimozione dal workspace di variabili di cui non è necessaria la visualizzazione ai fini del processo implementato.

Le 11 funzioni realizzate, invece, hanno per nome:

1. “calcolo_indici_di_asimmetria”;
2. “calcolo_quantili”;
3. “calcolo_quartili”;
4. “costruzione_tabella_con_intervalli”;
5. “costruzione_tabella_con_valori”;
6. “dati_input”;
7. “procedura_intervalli_con_e_senza_dataset”;
8. “procedura_valori_con_e_senza_dataset”;
9. “scelta_dataset”;
10. “suddivisione_con_dataset”;
11. “suddivisione_senza_dataset”.

Ognuna è presente nel relativo file di estensione “.m”.

La funzione “calcolo_indici_di_asimmetria”:

- Riceve in input 1 parametro:
 1. “w_o_intervalli_o_Valori”, che è il vettore contenente i dati da cui ricavare gli indici di asimmetria.
- Fornisce in output 1 valore:
 1. “indici_di_asimmetria”, che rappresenta il vettore con gli indici di asimmetria calcolati.
- Svolge le seguenti operazioni:
 1. Inizializza due variabili;
 2. Calcola i cinque indici di asimmetria usando funzioni predefinite di MATLAB;
 3. Inizializza due vettori colonna inserendo in uno i nomi degli indici e nell'altro i valori dei medesimi indici;
 4. Visualizza i risultati ottenuti su schermo;
 5. Inserisce i risultati trovati nel vettore degli indici di asimmetria.

La funzione “calcolo_quantili”:

- Riceve in input 3 parametri:
 1. “Fi”, che rappresenta il vettore colonna con le frequenze cumulate;
 2. “intervalli”, che coincide con il vettore riga contenente gli estremi delle classi;
 3. “tipo_classi”, che rappresenta il numero che esprime se le classi sono intervalli o valori.
- Fornisce in output 1 valore:
 1. “quantili”, che corrisponde ad una matrice n*n avente nella prima colonna il nome del quantile calcolato e del relativo percentile e nella seconda colonna il rispettivo valore.
- Svolge le seguenti operazioni:
 1. Inizializza variabili e vettori;
 2. Avvia un ciclo che viene concluso solo quando l’utente decide di non calcolare altri quantili;
 3. All’interno di un ciclo while, richiede di inserire il percentile di cui calcolare il relativo quantile con un valore da 1 a 100, escludendo 25, 50 e 75, oltre a eventuali percentili già scelti, controllando eventuali errori e ripetendo la precedente richiesta in caso di operazione non andata a buon fine;
 4. Aggiunge il percentile scelto nel vettore riga dei percentili già scelti, che viene successivamente sistemato, e ricava l’ordine del quantile dividendo il percentile per 100;
 5. Controlla se l’ordine del quantile è compreso tra 0 e il primo valore di Fi ed eventualmente assegna 0 e 1 agli indici per il calcolo del relativo quantile;
 6. Nel caso in cui gli indici precedenti non sono stati inizializzati, controlla se l’ordine del quantile è compreso tra valori di Fi successivi al primo ed eventualmente assegna i e i+1 ai rispettivi indici per il calcolo del quantile associato;
 7. Calcola il quantile in questione considerando se il primo dei due indici di sopra è nullo o meno e tenendo presente come sono espresse le classi;
 8. Visualizza i risultati ottenuti su schermo;
 9. Salva i nomi dei quantili e i relativi risultati in due vettori colonna separati;
 10. All’interno di un ciclo while, richiede di inserire un valore che sia 1 o 2, per comprendere se desidera continuare a calcolare altri quantili o meno, controllando eventuali errori e ripetendo la precedente richiesta in caso di operazione non andata a buon fine;
 11. Nel caso in cui desidera continuare, trasferisce il contenuto dei vettori colonna nella matrice n*n e termina il ciclo avviato al punto 2;
 12. Nel caso in cui, invece, vuole terminare, aggiorna il valore di alcune variabili e continua ad eseguire il ciclo al punto 2.

La funzione “calcolo_quartili”:

- Riceve in input 2 parametri:
 1. “Fi”;
 2. “intervalli”.

- Fornisce in output 1 valore:
 1. “quartili”, che corrisponde ad una matrice 3×2 avente nella prima colonna il nome del quartile calcolato e del relativo percentile e nella seconda colonna il rispettivo valore.
- Svolge le seguenti operazioni:
 1. Controlla se i quantili di ordine 0.25, 0.5 e 0.75 sono compresi tra 0 e il primo valore di F_i ed eventualmente assegna 0 e 1 agli indici per il calcolo del primo, secondo e terzo quartile;
 2. Nel caso in cui non tutti gli indici precedenti siano stati inizializzati, controlla se i quantili di ordine 0.25, 0.5 e 0.75 sono compresi tra valori di F_i successivi al primo ed eventualmente assegna i e $i+1$ ai rispettivi indici per il calcolo del primo, secondo e terzo quartile;
 3. Se almeno uno dei tre indici di sopra è pari a 0, calcola il primo, secondo e terzo quartile considerando se il relativo indice è nullo o meno;
 4. Se, invece, nessuno dei tre indici in questione è nullo, ricava il primo, secondo e terzo quartile con la formula generale;
 5. Salva i nomi dei quartili e i relativi risultati nella matrice 3×2 ;
 6. Visualizza i risultati ottenuti su schermo.

La funzione “costruzione_tabella_con_intervalli”:

- Riceve in input 4 parametri:
 1. “creazione”, che specifica se creare la tabella di raccolta dati con o senza dataset;
 2. “ f_i ”, che coincide con il vettore colonna contenente le frequenze assolute per ogni classe;
 3. “intervalli”;
 4. “n”, che rappresenta il numero di valori da inserire nelle classi, nonché la somma di tutte le frequenze assolute contate dal programma.
- Fornisce in output 2 valori:
 1. “T”, che è la tabella in cui si visualizzano le classi e ogni propria frequenza;
 2. “ F_i ”.
- Svolge le seguenti operazioni:
 1. Inizializza i valori dei vettori colonna “Classi”, “ p_i ” e “ F_i ”;
 2. Riempie tali vettori incolonnando rispettivamente gli intervalli che formano le classi e le frequenze assolute, relative e cumulate di ogni classe;
 3. Nel caso di utilizzo di un dataset, trasforma “ f_i ” da vettore riga a vettore colonna;
 4. Crea e mostra su schermo la tabella di raccolta dati ottenuta.

La funzione “costruzione_tabella_con_valori”:

- Riceve in input 4 parametri:
 1. “creazione”;
 2. “ f_i ”;
 3. “n”;

- 4. “Valori”, che è il vettore riga contenente i valori inseriti dall’utente.
- Fornisce in output 4 valori:
 1. “T1”, che è la tabella in cui si visualizzano i valori e ogni propria frequenza;
 2. “T2”, che è la tabella in cui si visualizzano le classi e ogni propria frequenza;
 3. “Fi”;
 4. “intervalli”.
- Svolge le seguenti operazioni:
 1. Inizializza i valori dei vettori colonna “Classi”, “pi” e “Fi”;
 2. Riempie tali vettori incolonnando rispettivamente i valori che formano le classi e le frequenze assolute, relative e cumulate di ogni classe;
 3. Nel caso di utilizzo di un dataset, trasforma “fi” da vettore riga a vettore colonna;
 4. Trasforma “Valori” da vettore riga a vettore colonna;
 5. Crea e mostra su schermo prima la tabella di raccolta dati con i valori inseriti dall’utente e poi la medesima tabella con le classi a posto dei valori.

La funzione “dati_input”:

- Non riceve in input alcun parametro:
- Fornisce in output 3 valori:
 1. “creazione”;
 2. “tipo_classi”;
 3. “tipo_quantili”, che coincide con il valore che specifica se si vogliono calcolare solo i quartili o solo i quantili o entrambi.
- Svolge le seguenti operazioni:
 1. Inizializza i valori di “creazione”, “tipo_classi” e “tipo_quantili” a 0, al fine di permettere l’attraversamento dei successivi cicli while proposti;
 2. All’interno di tre cicli while successivi, richiede di inserire per le variabili “creazione” e “tipo_classi” un valore numerico che sia 1 o 2 e per la variabile “tipo_quantili” un valore numerico compreso tra 1 e 3, controllando eventuali errori e ripetendo ogni precedente richiesta in caso di operazione non andata a buon fine.

La funzione “procedura_intervalli_con_e_senza_dataset”:

- Riceve in input 5 parametri:
 1. “creazione”;
 2. “dataset_o_limite”, che è il numero di dataset pronti e utilizzabili, nel caso di uso di dataset, o la cifra minima da inserire immettendo l’estremo inferiore del primo intervallo, nel caso di mancato uso di dataset;
 3. “k_o_kmax”, che rappresenta il numero di classi previste usando i dataset o il numero massimo di classi previste senza i dataset;
 4. “tipo_classi”;
 5. “tipo_quantili”.
- Fornisce in output 4 valori:
 1. “T”;

2. "quantili";
3. "quartili";
4. "indici_di_asimmetria".

- Svolge le seguenti operazioni:

1. Nel caso di utilizzo di dataset, inizializza una variabile, usando il parametro di ingresso "dataset_o_limite", e richiama le funzioni "scelta_dataset", "suddivisione_con_dataset" e "calcolo_indici_di_asimmetria";
2. Nel caso di mancato utilizzo di dataset, inizializza una variabile utilizzando il parametro di ingresso "dataset_o_limite" e inizializza un'altra variabile a 0. Dopo, all'interno di un ciclo while, richiede di inserire per la variabile "k" un valore numerico compreso tra 1 e il numero massimo di classi previste, ossia 20 in questo caso, controllando eventuali errori e ripetendo la precedente richiesta in caso di operazione non andata a buon fine. Infine, richiama le funzioni "suddivisione_senza_dataset" e "calcolo_indici_di_asimmetria";
3. Richiama la funzione "costruzione_tabella_con_intervalli";
4. Se si vuole calcolare solo i quartili, richiama la funzione "calcolo_quartili" e assegna alla matrice n*n "quantili" una matrice vuota;
5. Se si vuole calcolare solo i quantili, richiama la funzione "calcolo_quantili" e assegna alla matrice 3x2 "quartili" una matrice vuota;
6. Se si vuole calcolare sia i quartili sia i quantili, richiama prima la funzione "calcolo_quartili" e poi la funzione "calcolo_quantili".

La funzione "procedura_valori_con_e_senza_dataset":

- Riceve in input 5 parametri:
 1. "creazione";
 2. "dataset_o_limite";
 3. "k_o_kmax";
 4. "tipo_classi";
 5. "tipo_quantili".
- Fornisce in output 5 valori:
 1. "T1";
 2. "T2";
 3. "quantili";
 4. "quartili";
 5. "indici_di_asimmetria".
- Svolge le seguenti operazioni:
 1. Nel caso di utilizzo di dataset, inizializza una variabile, usando il parametro di ingresso "dataset_o_limite", richiama le funzioni "scelta_dataset", "suddivisione_con_dataset" e "calcolo_indici_di_asimmetria" e ricava il numero totale di frequenze assolute;
 2. Nel caso di mancato utilizzo di dataset, inizializza una variabile utilizzando il parametro di ingresso "dataset_o_limite" e inizializza un'altra variabile a 0. Dopo, all'interno di un ciclo while, richiede di inserire per la variabile "numero_valori" un numero tra 1 e il numero massimo di valori previsti, ossia

- 20 in questo caso, controllando eventuali errori e ripetendo la precedente richiesta in caso di operazione non andata a buon fine. Infine, richiama le funzioni “suddivisione_senza_dataset” e “calcolo_indici_di_asimmetria”;
3. Richiama la funzione “costruzione_tabella_con_valori”;
 4. Se si vuole calcolare solo i quartili, richiama la funzione “calcolo_quartili” e assegna alla matrice $n \times n$ “quantili” una matrice vuota;
 5. Se si vuole calcolare solo i quantili, richiama la funzione “calcolo_quantili” e assegna alla matrice 3×2 “quartili” una matrice vuota;
 6. Se si vuole calcolare sia i quartili sia i quantili, richiama prima la funzione “calcolo_quartili” e poi la funzione “calcolo_quantili”.

La funzione “scelta_dataset”:

- Riceve in input 2 parametri:
 1. “dataset_proposti”, che rappresenta il numero di dataset pronti e utilizzabili;
 2. “tipo_classi”.
- Fornisce in output 4 valori:
 1. “intervalli”;
 2. “k”, che indica il numero di classi previste usando i dataset;
 3. “n”;
 4. “w”, che è il vettore colonna di elementi della colonna estratta dal dataset.
- Svolge la seguente operazione:
 1. Inizializza il valore di “dataset” a 0, al fine di permettere l’attraversamento del successivo ciclo while proposto, e del vettore riga “intervalli”;
 2. All’interno di un ciclo while, richiede di inserire per la variabile “dataset” un valore numerico compreso tra 1 e il numero di dataset proposti, ossia 5 nel caso in questione, controllando eventuali errori e ripetendo ogni precedente richiesta in caso di operazione non andata a buon fine;
 3. Scrive il percorso in cui si trova il dataset scelto;
 4. Crea un vettore riga con i nomi dei dataset in formato “.tsv”;
 5. Legge il dataset selezionato;
 6. Seleziona la colonna da estrarre in base al numero del dataset scelto;
 7. Crea il vettore colonna con i valori estratti;
 8. Ricava il numero di elementi del vettore colonna, il numero di classi richiesto, il campo di variazione e il minimo dei dati e l’ampiezza delle classi;
 9. Realizza degli intervalli aventi la stessa ampiezza per costituire le classi;
 10. Mostra su schermo alcune o tutte le informazioni inerenti ai valori ricavati.

La funzione “suddivisione_con_dataset”:

- Riceve in input 4 parametri:
 1. “intervalli”;
 2. “k”;
 3. “tipo_classi”;
 4. “w”.
- Fornisce in output 2 valori:

- 1. “fi”;
- 2. “Valori”.
- Svolge la seguente operazione:
 1. Inizializza il vettore riga “Valori”;
 2. Nel caso di classi espresse come intervalli, inizializza “fi” inserendo “k” valori 0 e assegna ogni elemento di “w” a una classe, in base all’intervallo di cui fa parte, considerando che, per ogni intervallo, va escluso l’estremo superiore;
 3. Nel caso di classi espresse come valori, inizializza “fi” ad un vettore vuoto, crea un vettore colonna di elementi univoci di “w”, ovvero un nuovo vettore avente tutti gli elementi di “w” senza ripetizioni, ricerca il numero di occorrenze di ogni elemento univoco in “w” e, se quel numero non è già presente nel vettore con tutte le occorrenze trovate, lo inserisce al suo interno, immettendo il relativo elemento univoco nel vettore “Valori”.

La funzione “suddivisione_senza_dataset”:

- Riceve in input 3 parametri:
 1. “k_o_valori”, che rappresenta il numero di classi o valori scelto dall’utente;
 2. “limite_minimo”, che è la cifra minima da inserire immettendo l’estremo inferiore del primo intervallo;
 3. “tipo_classi”.
- Fornisce in output 4 valori:
 1. “fi”;
 2. “intervalli”;
 3. “n”;
 4. “Valori”.
- Svolge le seguenti operazioni:
 1. Inizializza alcune variabili a 0 e alcuni vettori a vettori vuoti;
 2. Nel caso di classi espresse come intervalli, inizializza la variabile “inf” e, all’interno di tre cicli while consecutivi inseriti, a loro volta, dentro un ciclo for, richiede, alla prima iterazione di quest’ultimo ciclo, di inserire per “inf” un valore numerico che sia almeno uguale a -1000. In seguito, aggiunge “inf” al vettore “intervalli” e pone “sup” pari a “inf” sottratto 1. Nel successivo ciclo while, richiede di inserire per la variabile “sup” un valore numerico maggiore di “inf”, nel caso della prima iterazione del ciclo for, e maggiore dell’ultimo “sup” inserito, nel caso delle successive iterazioni del medesimo ciclo for. Nell’ultimo ciclo while, richiede di inserire per la variabile “f” un valore numerico superiore a 0. In tutti e tre i cicli while, si procede controllando eventuali errori e ripetendo ogni precedente richiesta in caso di operazione non andata a buon fine. Infine, dentro il solo ciclo for, si ha l’inserimento di “sup” nel vettore “intervalli” e di “f” nel vettore “fi”, oltre all’inizializzazione di “inf” e di “f” e all’incremento del numero totale di frequenze assolute;
 3. Nel caso di classi espresse come valori, inizializza la variabile “valore” e, all’interno di tre cicli while consecutivi inseriti, a loro volta, dentro un ciclo for, richiede, alla prima iterazione di quest’ultimo ciclo, di inserire per “valore” un

numero che sia almeno uguale a -1000. Alle successive iterazioni, invece, inizializza nuovamente la variabile “valore” e passa al successivo ciclo while, che richiede di inserire per la medesima variabile un valore numerico maggiore di “valore_precedente”, ossia dell’ultimo inserito. Nell’ultimo ciclo while, richiede di inserire per la variabile “f” un valore numerico superiore a 0. In tutti e tre i cicli while, si procede controllando eventuali errori e ripetendo ogni precedente richiesta in caso di operazione non andata a buon fine. Infine, dentro il solo ciclo for, si ha l’inserimento di “valore” nel vettore “Valori” e di “f” nel vettore “fi”, oltre all’inizializzazione di “valore_precedente” e di “f” e all’incremento del numero totale di frequenze assolute.