

Verifica python

Dal file:

https://github.com/MilenaValentini/TRM_Dati/blob/main/Nemo_6670.dat

leggere le seguenti colonne:

`M_ass`

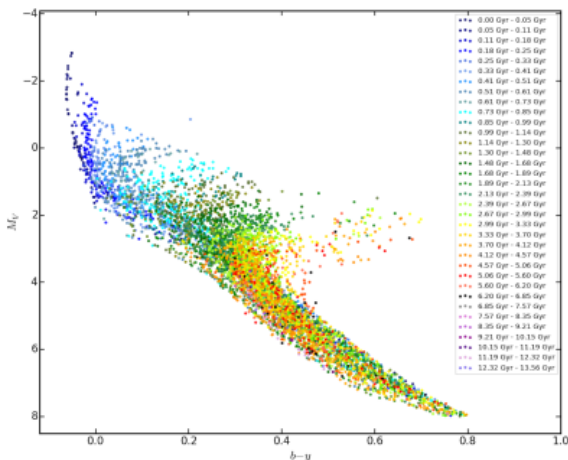
`b-y`

`age_parent [Gyr]`

e produrre un grafico simile a quello riportato a destra.

In questo grafico, come spiegato a lezione, si grafica `M_ass` in funzione di `b-y` (diagramma colore-magnitudine) per le 6670 stelle riportate nel file.

Il colore codifica l'età (`age_parent`) di stelle in un determinato bin di età e deve essere spiegato nella legenda.



Leggere poi anche la prima colonna del file (`MsuH`, cioè la metallicità delle stelle).

Dividere il campione di 6670 stelle in tre sotto-popolazioni a seconda della loro età (ad esempio per `age_parent` < 1 Gyr, compresa tra 1 e 5 Gyr, e > 5 Gyr) e studiare la distribuzione delle metallicità stellari per le stelle nei tre sotto-campioni (tramite tre istogrammi sullo stesso plot).

Identificare gli istogrammi con diversi colori e riportare una legenda sul grafico.

Calcolare anche media e mediana delle tre distribuzioni e riportarne con delle linee verticali i valori sul plot.

Leggere poi la seconda colonna (`m_ini`, cioè la massa iniziale) del file.

Graficare per le tre popolazioni individuate al punto precedente la metallicità in funzione della massa.

Riportare le tre distribuzioni sullo stesso grafico.

Opzionale: capire come ottimizzare la visualizzazione del contenuto del plot (ad esempio valutando diverse trasparenze dei simboli; o provando ad utilizzare contorni di densità per una o più distribuzioni ad esempio con l'istruzione `matplotlib.pyplot.contour` o `pylab.contour`; oppure provando a produrre un istogramma 2D ad esempio con `numpy.histogram2d`).

Verifica bash

Preparare lo script bash di esecuzione della soluzione dell'esercizio assegnato che deve:

- scaricare il file
https://github.com/MilenaValentini/TRM_Dati/blob/main/Nemo_6670.dat
Suggerimento: usare il link
https://raw.githubusercontent.com/MilenaValentini/TRM_Dati/main/Nemo_6670.dat
- lanciare lo script che risolve il problema assegnato passandogli come parametro il nome del file.

Lo script python deve essere scritto in modo da ricevere il parametro 'nome_del_file' in input ed usarlo opportunamente.

Preparare lo script di installazione della piccola applicazione appena codificata che:

- crea una directory dove copiare l'applicazione,
- copia lo start script bash precedentemente preparato ed il/i file python in un path opportuno
- attribuisce opportunamente i permessi di esecuzione
- modifica il PYTHONPATH ed il PATH di sistema in modo che l'applicazione nel suo complesso sia eseguibile con un singolo comando

Consegna dell'elaborato

L'elaborato deve essere consegnato come repository github (la consegna consiste nel fornire alle docenti il link al repository). Nel repository deve essere presente il file README.md contenente una breve descrizione dell'applicazione risultante dalla soluzione degli esercizi proposti e le istruzioni su come utilizzarla: installazione ed esecuzione.

La verifica dell'elaborato consisterà nel seguire le istruzioni per installare ed eseguire l'applicazione secondo le istruzioni fornite.