

Exercise1__wbstyle

March 24, 2025

0.0.1 Checking Astropy Package Information

In this step, we will display detailed information about the installed version of the **astropy** Python package, including its version, dependencies, installation location, and metadata.

```
!pip show astropy
```

0.0.2 Importing Astropy and Checking Its Version

Here, we will import the **astropy** package and print its version to confirm that it is installed correctly and ready for use.

```
import astropy
print(astropy.__version__)
```

0.0.3 Loading FITS Data and Plotting All Stars (Basic Visualization)

In this section, we will:

- Import essential libraries for data analysis, WCS handling, and plotting.
- Load data from a FITS file (`hst_results_nd.fits`).
- Extract key information such as header details and data arrays.
- Visualize all stars from the table using default markers.

```
import numpy as np
import matplotlib.pyplot as plt
from astropy.io import fits
from astropy.wcs import WCS
from astropy.table import Table
from astropy.visualization import make_lupton_rgb
from astropy import units as u
plt.ion()
import os
```

```
hdul = fits.open('hst_results_nd.fits')
```

```
hdr = hdul[0].header
wcs = WCS(hdr)
data = hdul[0].data
```

```
t = Table.read('hst_results_nd.fits')
```

```

ra = t[0]['RA']
dec = t[0]['DEC']
age = t["logA_p50"]
mass = t["M_ini_p50"]
f475W = t["F475W_VEGA"]
f814W = t["F814W_VEGA"]
columns = t[0].colnames
print(columns)

plt.figure()
plt.plot(t['RA'], t['DEC'], 'o')
plt.xlabel('RA')
plt.ylabel('DEC')
plt.title('All stars in the table')
plt.show()

```

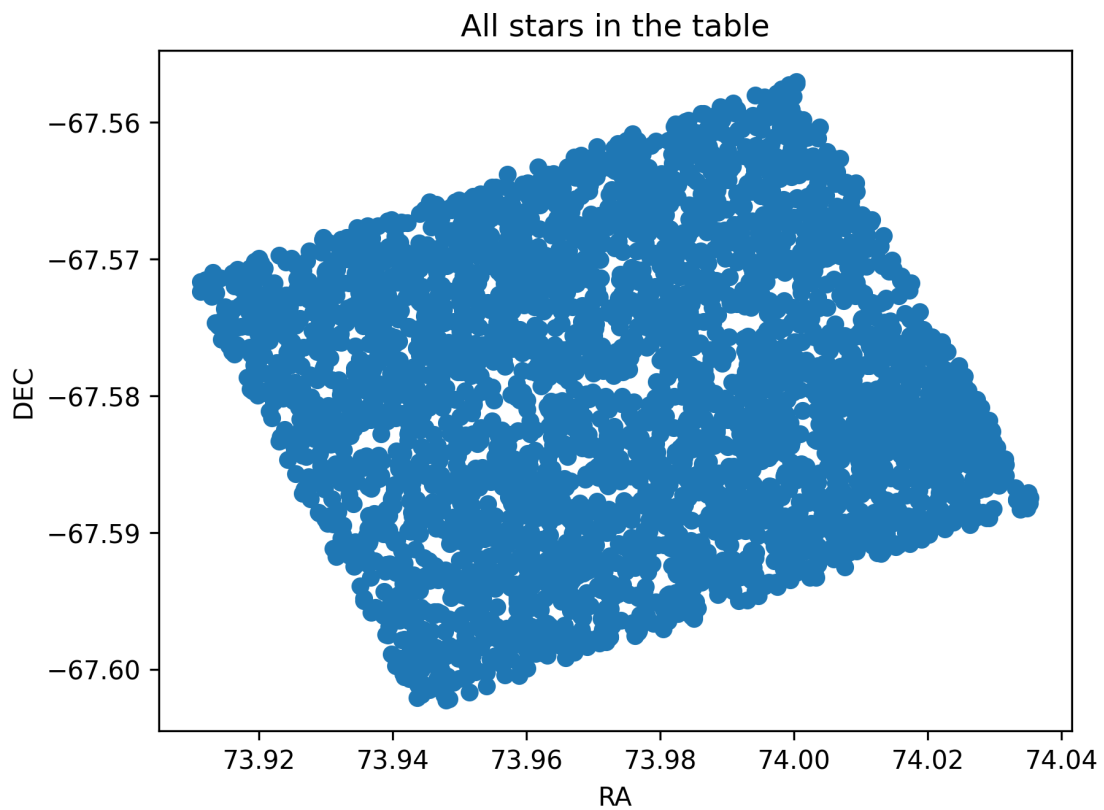


Figure: Spatial distribution of all stars using default circular markers. The plot shows RA vs. DEC coordinates, providing a general view of the stellar field.

0.0.4 Plotting All Stars with Smaller Red Markers

Now, we will re-plot the stars using smaller red markers. This adjustment helps distinguish stars in crowded regions more effectively.

```
plt.figure()
plt.plot(t['RA'], t['DEC'], 'ro', ls='None', markersize=1)
plt.xlabel('RA')
plt.ylabel('DEC')
plt.title('All stars in the table (red markers)')
plt.show()
```

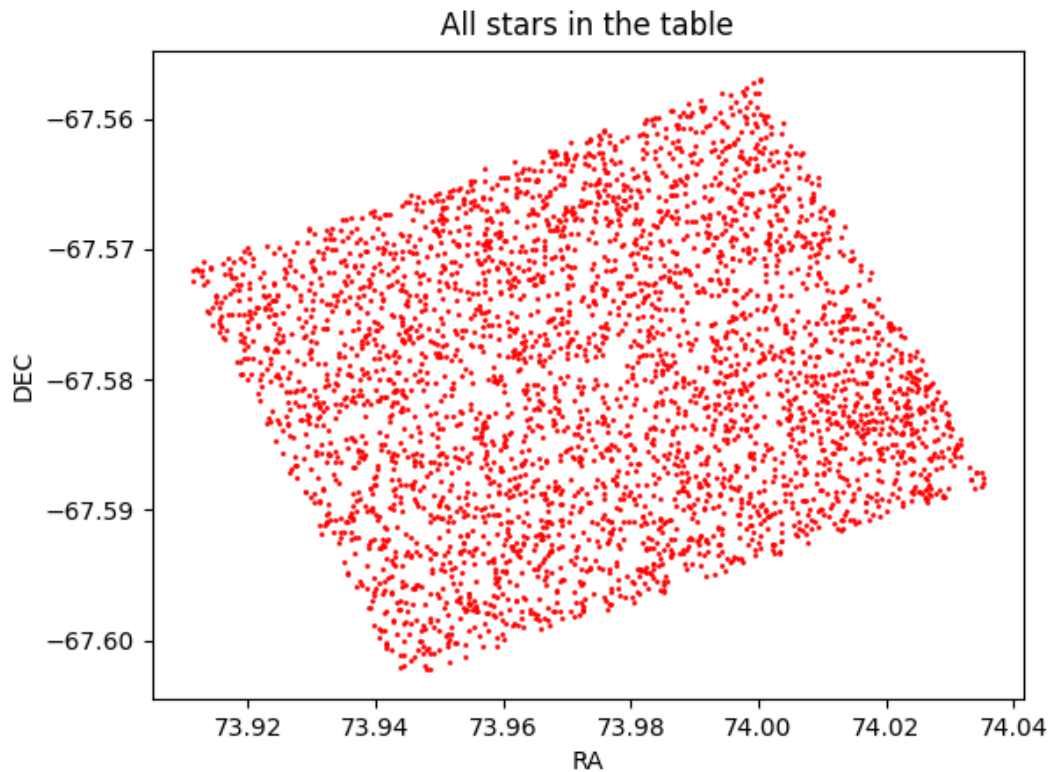


Figure: Spatial distribution of all stars from the catalog. Each star is plotted as a small red dot in Right Ascension (RA) and Declination (DEC) coordinates. The small marker size helps to distinguish individual stars, especially in dense regions of the field.

0.0.5 Plotting Stars Colored by Extinction (A_v)

In this step, we will create a scatter plot where the color of each star corresponds to its extinction value (A_v).

```
plt.figure()
plt.scatter(t['RA'], t['DEC'], s=3, c=t['Av_p50'], cmap='magma', vmin=0, vmax=1)
plt.xlabel('RA')
plt.ylabel('DEC')
```

```
plt.title('All stars in the table')
plt.colorbar()
plt.savefig('lmc_av_spacial.png')
plt.show()
```

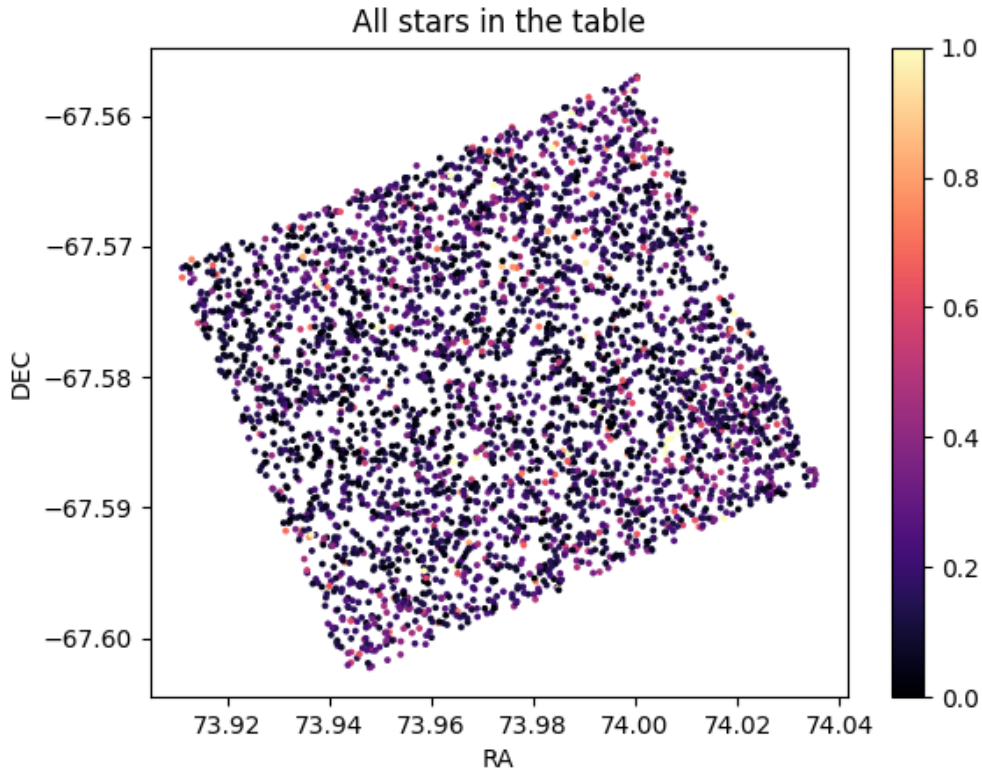


Figure: Spatial distribution of stars color-coded by extinction (A_{v_p50}). The colorbar indicates extinction values, scaled between 0 and 1 using the “magma” colormap.

0.0.6 Analyzing the Distance Distribution of Stars

Next, we will plot a histogram showing how stars are distributed by distance (in kiloparsecs). The plot includes annotations for both the mean and median distance values.

```
dist = t['distance_p50']
plt.figure()
plt.hist(dist / 1000, bins=25)
plt.xlabel('Distance (kpc)', fontsize=12)
plt.ylabel('Number of stars')
plt.title('Distance distribution of stars')

d_mean = np.mean(dist)
d_median = np.median(dist)
d_std = np.std(dist)
```

```

plt.axvline(d_mean / 1000, color='r', linestyle='dashed', linewidth=1)
plt.axvline(d_median / 1000, color='gray', linestyle='dashed', linewidth=1)
plt.text(0.5, 0.9, 'Mean: {:.2f} kpc'.format(d_mean / 1000),
        color='r', fontsize=12, transform=plt.gca().transAxes)
plt.text(0.5, 0.85, 'Median: {:.2f} kpc'.format(d_median / 1000),
        color='g', fontsize=12, transform=plt.gca().transAxes)
plt.savefig('lmc_hist_dist.png')
plt.show()

```

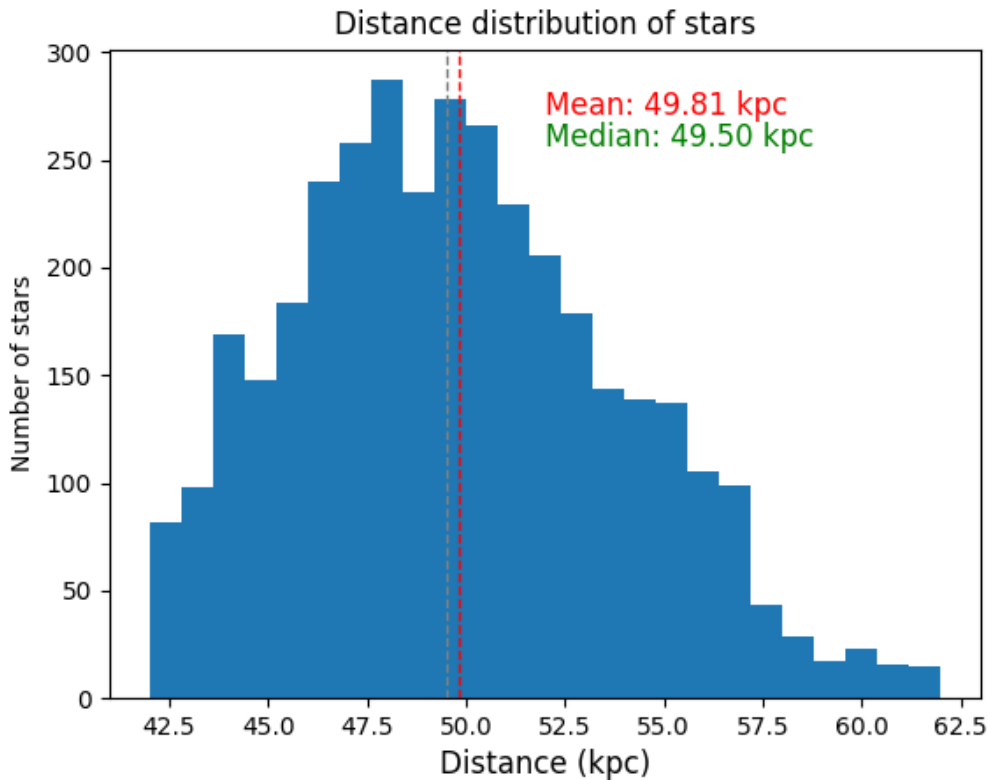


Figure: Histogram showing the distance distribution of stars in kiloparsecs. The red dashed line marks the mean distance, while the gray dashed line indicates the median distance. Annotations display the exact values of both metrics on the plot.

0.0.7 Creating a Color-Magnitude Diagram from External Catalog

Finally, we will load an external catalog and create a CMD using F475W and F814W magnitudes.

```
cat = Table.read('hlsp_scylla_hst_wfc3_lmc-04_multi_v1_st.fits')
```

```
f475 = cat["F475W_VEGA"]
```

```
f814 = cat["F814W_VEGA"]
```

```
n = len(f475)
```

```
col = f475 - f814
```

```

mag = f475

plt.figure()
plt.plot(col, mag, '.b', ls='None', markersize=1, label='N = %s' % n)
plt.legend()
plt.xlabel('f475w - f814w')
plt.ylabel('f475w')
plt.title('hlsp_scylla_hst_wfc3_lmc-04_multi_v1_st')
plt.xlim(-3, 6)
plt.ylim(30, 17)
plt.savefig('lmc_color_f475.png')
plt.show()

```

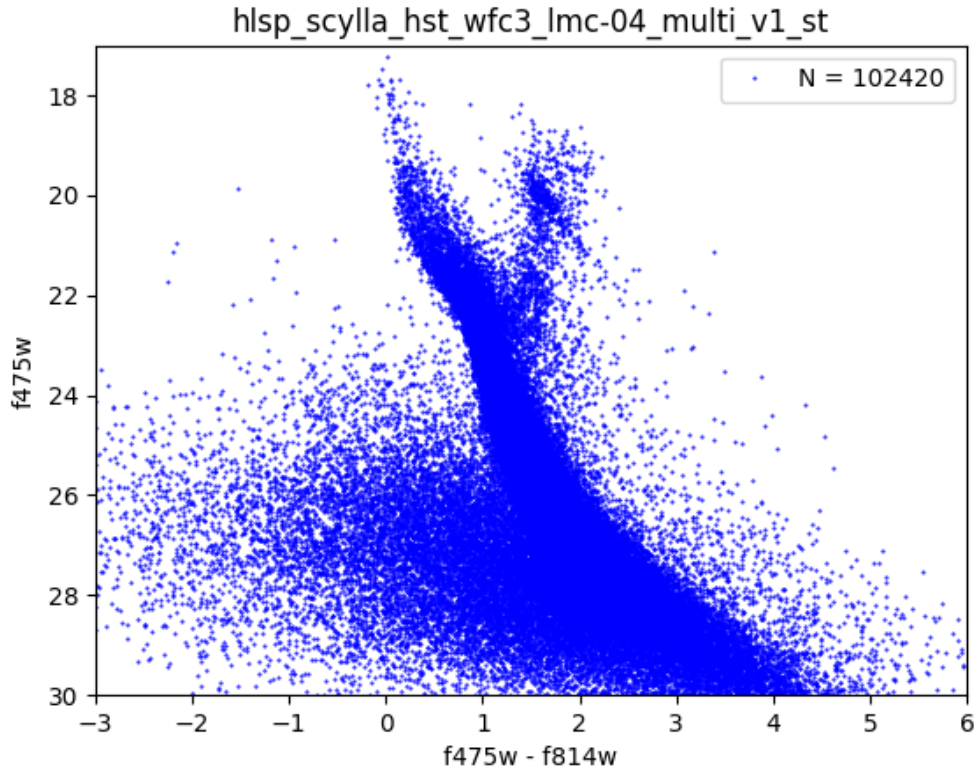


Figure: Color-magnitude diagram (CMD) showing the relation between color ($F475W - F814W$) and the $F475W$ magnitude for stars from the catalog `hlsp_scylla_hst_wfc3_lmc-04_multi_v1_st.fits`. The plot displays a total of N stars (indicated in the legend) with blue markers.