In [1]:
```
!pip show astropy
```

```
Name: astropy
Version: 7.0.1
Summary: Astronomy and astrophysics core library
Home-page:
Author:
Author-email: The Astropy Developers <astropy.team@gmail.com>
License: BSD-3-Clause
Location: C:\Users\btcya\Desktop\astronomy-course-programming\astrocourse\Lib\sit
e-packages
Requires: astropy-iers-data, numpy, packaging, pyerfa, PyYAML
Required-by:
```

In [2]:
```
# Import the package
import astropy

# Check the installed version
print(astropy.__version__)
```

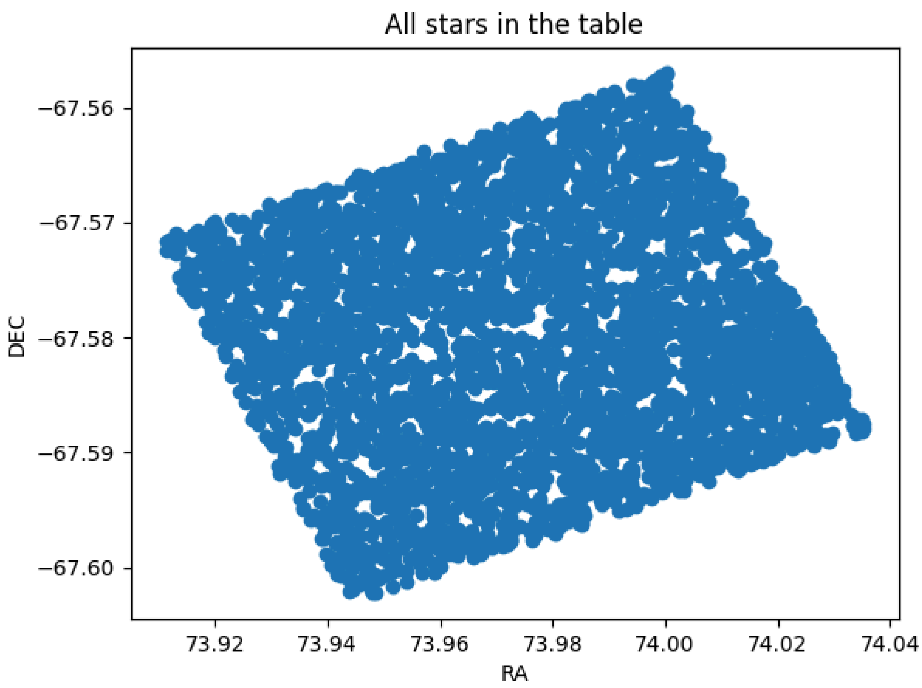
```
7.0.1
```

Explanation of the Code

1. **Import Required Libraries**:
   - `numpy` : For numerical operations.
   - `matplotlib.pyplot` : For plotting data.
   - `astropy.io.fits` : For reading FITS (Flexible Image Transport System) files.
   - `astropy.wcs` : For handling World Coordinate System (WCS) information.
   - `astropy.table` : For working with tabular data.
   - `astropy.visualization.make_lupton_rgb` : For creating RGB images (not used in this code).
   - `plt.ion()` : Enables interactive mode for Matplotlib, allowing plots to update dynamically.
2. **Open and Read FITS File**:
   - The FITS file `hst_results_nd.fits` is opened using `fits.open` .
   - The header (`hdr`) is extracted from the first HDU (Header Data Unit) to access metadata.
   - A WCS object (`wcs`) is created from the header for celestial coordinate transformations.
   - The data (`data`) is extracted from the first HDU.
3. **Read the Table from FITS File**:
   - The FITS file is read as an Astropy table (`t`), which contains tabular data such as star positions and magnitudes.
4. **Access and Print Table Information**:
   - The Right Ascension (RA) and Declination (DEC) of the first row in the table are extracted.
   - The column names of the table are retrieved and printed.

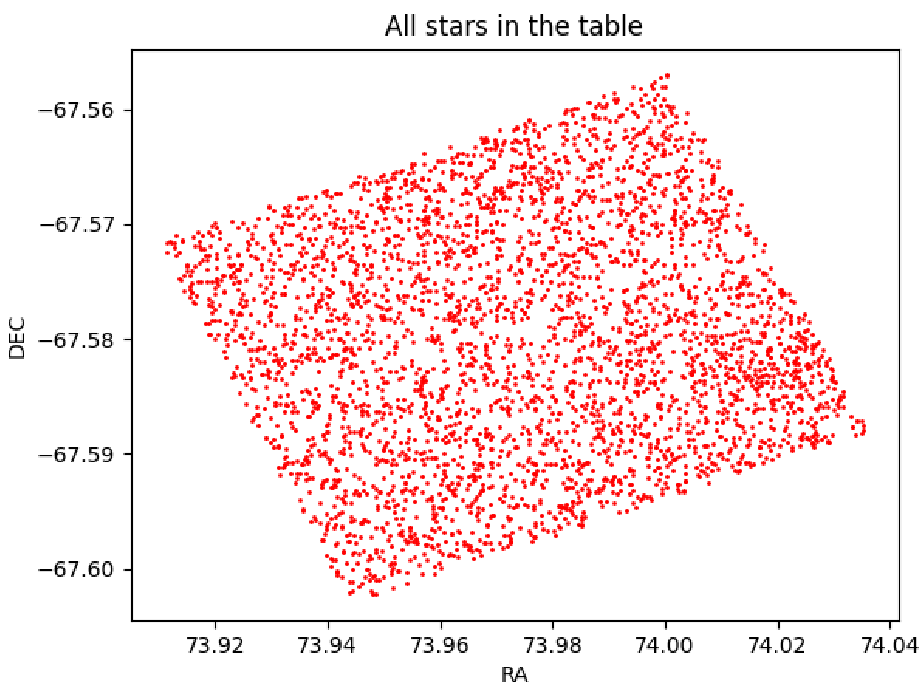
5. **Plot All Stars in the Table**:

- A scatter plot is created using the RA and DEC values of all stars in the table.
- The x-axis is labeled as "RA", and the y-axis is labeled as "DEC".
- The plot is titled "All stars in the table".



6. **Plot Stars with Smaller Red Markers**:

- Another scatter plot is created with red circular markers (`'ro'`) and a smaller marker size (`markersize=1`) for better distinction.
- The x-axis is labeled as "RA", and the y-axis is labeled as "DEC".
- The plot is titled "All stars in the table".



Summary:

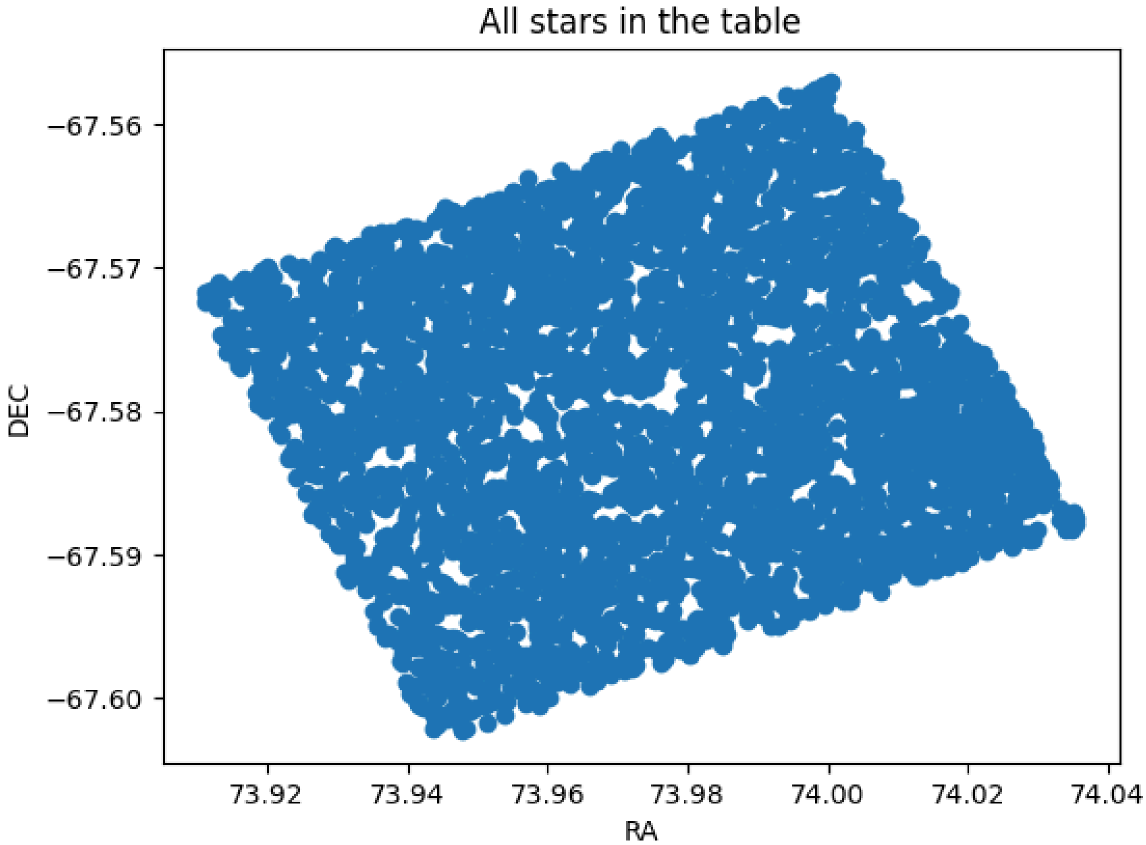
- The code reads a FITS file containing astronomical data.
- It extracts the RA and DEC of stars from the table and plots them in two ways:
    1. A default scatter plot.
    2. A scatter plot with smaller red markers for better distinction.
- The WCS object (`wcs`) is initialized but not used in this code. It could be used for coordinate transformations if needed.

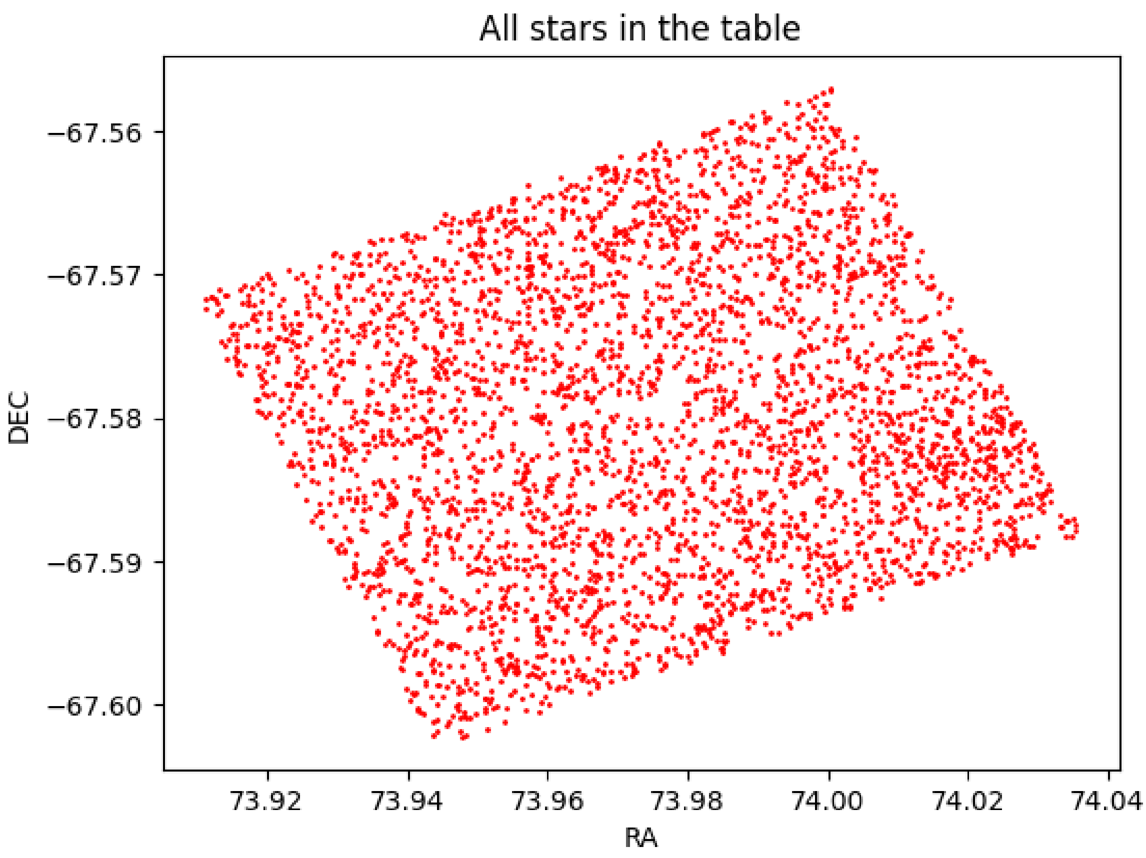
In [ ]:

```
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
plt.ion()

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')

# check all stars in the table which are the first row of the table
ra = t[0]['RA']
dec = t[0]['DEC']
columns = t[0].colnames
print(columns)

# plot the table of all stars
plt.figure()
plt.plot(t['RA'], t['DEC'], 'o')
plt.xlabel('RA')
plt.ylabel('DEC')
plt.title('All stars in the table')
plt.savefig('all_stars.png')
plt.show()


# plot the start with smaller red spots so that they can be distinguished
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')
plt.savefig('all_stars_distinguished.png')
plt.show()
```

['Name', 'RA', 'DEC', 'HST_WFC3_F225W', 'HST_WFC3_F275W', 'HST_WFC3_F336W', 'HST_WFC3_F475W', 'HST_WFC3_F814W', 'HST_WFC3_F110W', 'HST_WFC3_F160W', 'Av_Best', 'Av_Exp', 'Av_p16', 'Av_p50', 'Av_p84', 'Rv_Best', 'Rv_Exp', 'Rv_p16', 'Rv_p50', 'Rv_p84', 'Rv_A_Best', 'Rv_A_Exp', 'Rv_A_p16', 'Rv_A_p50', 'Rv_A_p84', 'f_A_Best', 'f_A_Exp', 'f_A_p16', 'f_A_p50', 'f_A_p84', 'distance_Best', 'distance_Exp', 'distance_p16', 'distance_p50', 'distance_p84', 'radius_Best', 'radius_Exp', 'radius_p16', 'radius_p50', 'radius_p84', 'logL_Best', 'logL_Exp', 'logL_p16', 'logL_p50', 'logL_p84', 'logg_Best', 'logg_Exp', 'logg_p16', 'logg_p50', 'logg_p84', 'mbolmag_Best', 'mbolmag_Exp', 'mbolmag_p16', 'mbolmag_p50', 'mbolmag_p84', 'logA_Best', 'logA_Exp', 'logA_p16', 'logA_p50', 'logA_p84', 'logT_Best', 'logT_Exp', 'logT_p16', 'logT_p50', 'logT_p84', 'M_ini_Best', 'M_ini_Exp', 'M_ini_p16', 'M_ini_p50', 'M_ini_p84', 'M_act_Best', 'M_act_Exp', 'M_act_p16', 'M_act_p50', 'M_act_p84', 'Z_Best', 'Z_Exp', 'Z_p16', 'Z_p50', 'Z_p84', 'logHST_WFC3_F225W_nd_Best', 'logHST_WFC3_F225W_nd_Exp', 'logHST_WFC3_F225W_nd_p16', 'logHST_WFC3_F225W_nd_p50', 'logHST_WFC3_F225W_nd_p84', 'logHST_WFC3_F275W_nd_Best', 'logHST_WFC3_F275W_nd_Exp', 'logHST_WFC3_F275W_nd_p16', 'logHST_WFC3_F275W_nd_p50', 'logHST_WFC3_F275W_nd_p84', 'logHST_WFC3_F336W_nd_Best', 'logHST_WFC3_F336W_nd_Exp', 'logHST_WFC3_F336W_nd_p16', 'logHST_WFC3_F336W_nd_p50', 'logHST_WFC3_F336W_nd_p84', 'logHST_WFC3_F475W_nd_Best', 'logHST_WFC3_F475W_nd_Exp', 'logHST_WFC3_F475W_nd_p16', 'logHST_WFC3_F475W_nd_p50', 'logHST_WFC3_F475W_nd_p84', 'logHST_WFC3_F814W_nd_Best', 'logHST_WFC3_F814W_nd_Exp', 'logHST_WFC3_F814W_nd_p16', 'logHST_WFC3_F814W_nd_p50', 'logHST_WFC3_F814W_nd_p84', 'logHST_WFC3_F110W_nd_Best', 'logHST_WFC3_F110W_nd_Exp', 'logHST_WFC3_F110W_nd_p16', 'logHST_WFC3_F110W_nd_p50', 'logHST_WFC3_F110W_nd_p84', 'logHST_WFC3_F160W_nd_Best', 'logHST_WFC3_F160W_nd_Exp', 'logHST_WFC3_F160W_nd_p16', 'logHST_WFC3_F160W_nd_p50', 'logHST_WFC3_F160W_nd_p84', 'logGALEX_FUV_nd_Best', 'logGALEX_FUV_nd_Exp', 'logGALEX_FUV_nd_p16', 'logGALEX_FUV_nd_p50', 'logGALEX_FUV_nd_p84', 'logF_QION228_nd_Best', 'logF_QION228_nd_Exp', 'logF_QION228_nd_p16', 'logF_QION228_nd_p50', 'logF_QION228_nd_p84', 'logHST_WFC3_F225W_wd_Best', 'logHST_WFC3_F225W_wd_Exp', 'logHST_WFC3_F225W_wd_p16', 'logHST_WFC3_F225W_wd_p50', 'logHST_WFC3_F225W_wd_p84', 'logHST_WFC3_F275W_wd_Best', 'logHST_WFC3_F275W_wd_Exp', 'logHST_WFC3_F275W_wd_p16', 'logHST_WFC3_F275W_wd_p50', 'logHST_WFC3_F275W_wd_p84', 'logHST_WFC3_F336W_wd_Best', 'logHST_WFC3_F336W_wd_Exp', 'logHST_WFC3_F336W_wd_p16', 'logHST_WFC3_F336W_wd_p50', 'logHST_WFC3_F336W_wd_p84', 'logHST_WFC3_F475W_wd_Best', 'logHST_WFC3_F475W_wd_Exp', 'logHST_WFC3_F475W_wd_p16', 'logHST_WFC3_F475W_wd_p50', 'logHST_WFC3_F475W_wd_p84', 'logHST_WFC3_F814W_wd_Best', 'logHST_WFC3_F814W_wd_Exp', 'logHST_WFC3_F814W_wd_p16', 'logHST_WFC3_F814W_wd_p50', 'logHST_WFC3_F814W_wd_p84', 'logHST_WFC3_F110W_wd_Best', 'logHST_WFC3_F110W_wd_Exp', 'logHST_WFC3_F110W_wd_p16', 'logHST_WFC3_F110W_wd_p50', 'logHST_WFC3_F110W_wd_p84', 'logHST_WFC3_F160W_wd_Best', 'logHST_WFC3_F160W_wd_Exp', 'logHST_WFC3_F160W_wd_p16', 'logHST_WFC3_F160W_wd_p50', 'logHST_WFC3_F160W_wd_p84', 'logGALEX_FUV_wd_Best', 'logGALEX_FUV_wd_Exp', 'logGALEX_FUV_wd_p16', 'logGALEX_FUV_wd_p50', 'logGALEX_FUV_wd_p84', 'logF_QION228_wd_Best', 'logF_QION228_wd_Exp', 'logF_QION228_wd_p16', 'logF_QION228_wd_p50', 'logF_QION228_wd_p84', 'symlogHST_WFC3_F225W_wd_bias_Best', 'symlogHST_WFC3_F225W_wd_bias_Exp', 'symlogHST_WFC3_F225W_wd_bias_p16', 'symlogHST_WFC3_F225W_wd_bias_p50', 'symlogHST_WFC3_F225W_wd_bias_p84', 'symlogHST_WFC3_F275W_wd_bias_Best', 'symlogHST_WFC3_F275W_wd_bias_Exp', 'symlogHST_WFC3_F275W_wd_bias_p16', 'symlogHST_WFC3_F275W_wd_bias_p50', 'symlogHST_WFC3_F275W_wd_bias_p84', 'symlogHST_WFC3_F336W_wd_bias_Best', 'symlogHST_WFC3_F336W_wd_bias_Exp', 'symlogHST_WFC3_F336W_wd_bias_p16', 'symlogHST_WFC3_F336W_wd_bias_p50', 'symlogHST_WFC3_F336W_wd_bias_p84', 'symlogHST_WFC3_F475W_wd_bias_Best', 'symlogHST_WFC3_F475W_wd_bias_Exp', 'symlogHST_WFC3_F475W_wd_bias_p16', 'symlogHST_WFC3_F475W_wd_bias_p50', 'symlogHST_WFC3_F475W_wd_bias_p84', 'symlogHST_WFC3_F814W_wd_bias_Best', 'symlogHST_WFC3_F814W_wd_bias_Exp', 'symlogHST_WFC3_F814W_wd_bias_p16', 'symlogHST_WFC3_F814W_wd_bias_p50', 'symlogHST_WFC3_F814W_wd_bias_p84', 'symlogHST_WFC3_F110W_wd_bias_Best', 'symlogHST_WFC3_F110W_wd_bias_Exp', 'symlogHST_WFC3_F110W_wd_bias_p16', 'symlogHST_WFC3_F110W_wd_bias_p50', 'symlogHST_WFC3_F110W_wd_bias_p84', 'symlogHST_WFC3_F160W_wd_bias_Best', 'symlogHST_WFC3_F160W_wd_bias_Exp', 'symlogHST_WFC3_F160W_wd_bias_p16', 'symlogHST_WFC3_F160W_wd_bias_p50', 'symlogHST_WFC3_F160W_wd_bias_p84', 'chi2min', 'Pmax', 'Pmax_indx', 'total_log_norm', 'best_gridsub_tag', 'reorder_tag', 'X', 'Y', 'F110W_RATE', 'F110W_RATERR', 'F110W

```
_VEGA', 'F110W_STD', 'F110W_ERR', 'F110W_CHI', 'F110W_SNR', 'F110W_SHARP', 'F110W
_ROUND', 'F110W_CROWD', 'F110W_FLAG', 'F160W_RATE', 'F160W_RATERR', 'F160W_VEGA',
'F160W_STD', 'F160W_ERR', 'F160W_CHI', 'F160W_SNR', 'F160W_SHARP', 'F160W_ROUND',
'F160W_CROWD', 'F160W_FLAG', 'F225W_RATE', 'F225W_RATERR', 'F225W_VEGA', 'F225W_S
TD', 'F225W_ERR', 'F225W_CHI', 'F225W_SNR', 'F225W_SHARP', 'F225W_ROUND', 'F225W_
CROWD', 'F225W_FLAG', 'F275W_RATE', 'F275W_RATERR', 'F275W_VEGA', 'F275W_STD', 'F
275W_ERR', 'F275W_CHI', 'F275W_SNR', 'F275W_SHARP', 'F275W_ROUND', 'F275W_CROWD',
'F275W_FLAG', 'F336W_RATE', 'F336W_RATERR', 'F336W_VEGA', 'F336W_STD', 'F336W_ER
R', 'F336W_CHI', 'F336W_SNR', 'F336W_SHARP', 'F336W_ROUND', 'F336W_CROWD', 'F336W
_FLAG', 'F475W_RATE', 'F475W_RATERR', 'F475W_VEGA', 'F475W_STD', 'F475W_ERR', 'F4
75W_CHI', 'F475W_SNR', 'F475W_SHARP', 'F475W_ROUND', 'F475W_CROWD', 'F475W_FLAG',
'F814W_RATE', 'F814W_RATERR', 'F814W_VEGA', 'F814W_STD', 'F814W_ERR', 'F814W_CH
I', 'F814W_SNR', 'F814W_SHARP', 'F814W_ROUND', 'F814W_CROWD', 'F814W_FLAG', 'HST_
WFC3_F336W_MD', 'reliable']
```

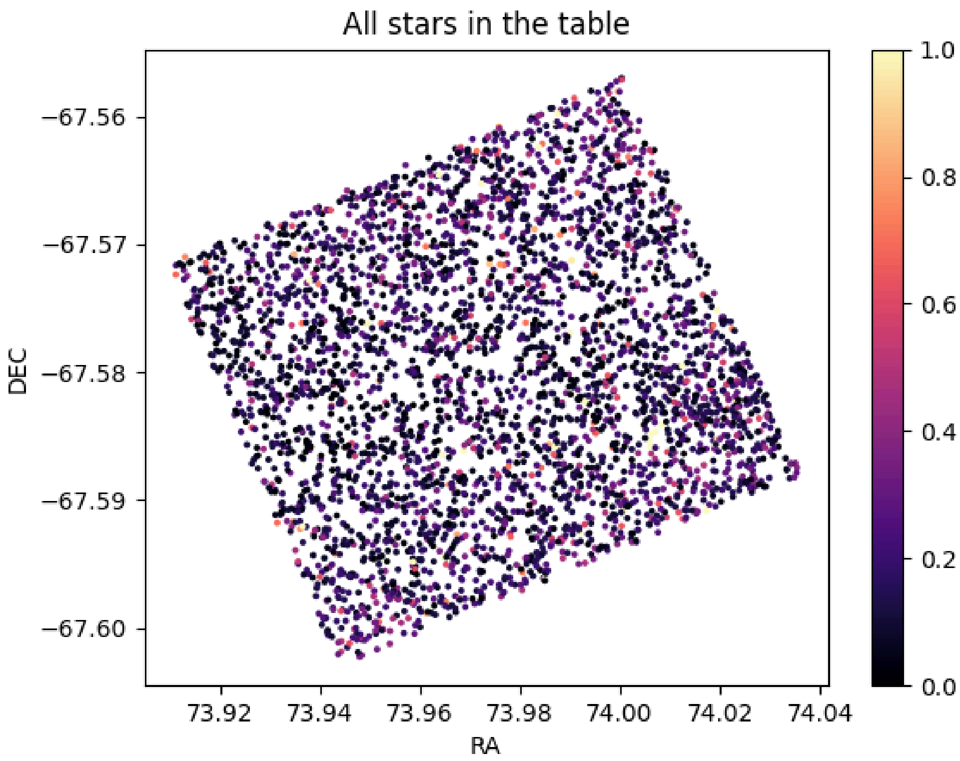




Summary:

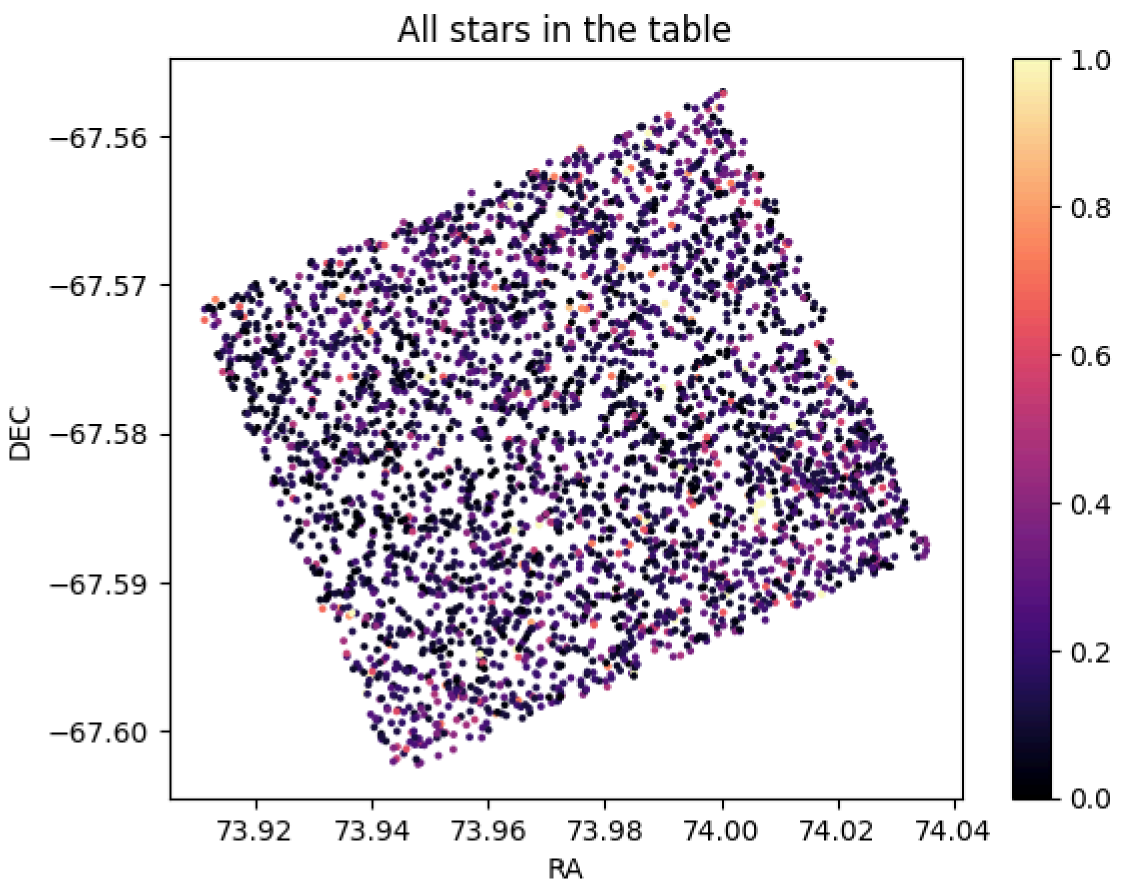
This code creates a scatter plot of stars with their positions (`RA` , `DEC`) and colors mapped to the `Av_p50` extinction values. The plot is saved as a file `lmc_av_spacial.png` and includes a color bar for reference.

- The scatter plot shows the spatial distribution of stars in the table, with their colors representing the `Av_p50` values.
- The color bar provides a reference for interpreting the color scale, with values ranging from 0 to 1.
- This visualization is useful for studying the spatial variation of extinction (`Av_p50`) across the stars in the dataset. It helps identify regions with higher or lower extinction values.



In [4]:

```
# use 'Av_p50' to plot the color of the stars
plt.figure()
plt.scatter(t['RA'], t['DEC'], s=3, c = t['Av_p50'], cmap = 'magma', vmin=0, vma
plt.xlabel('RA')
plt.ylabel('DEC')
plt.title('All stars in the table')
plt.colorbar()
plt.savefig('lmc_av_spacial.png')
plt.show()
```



1. **Purpose**:

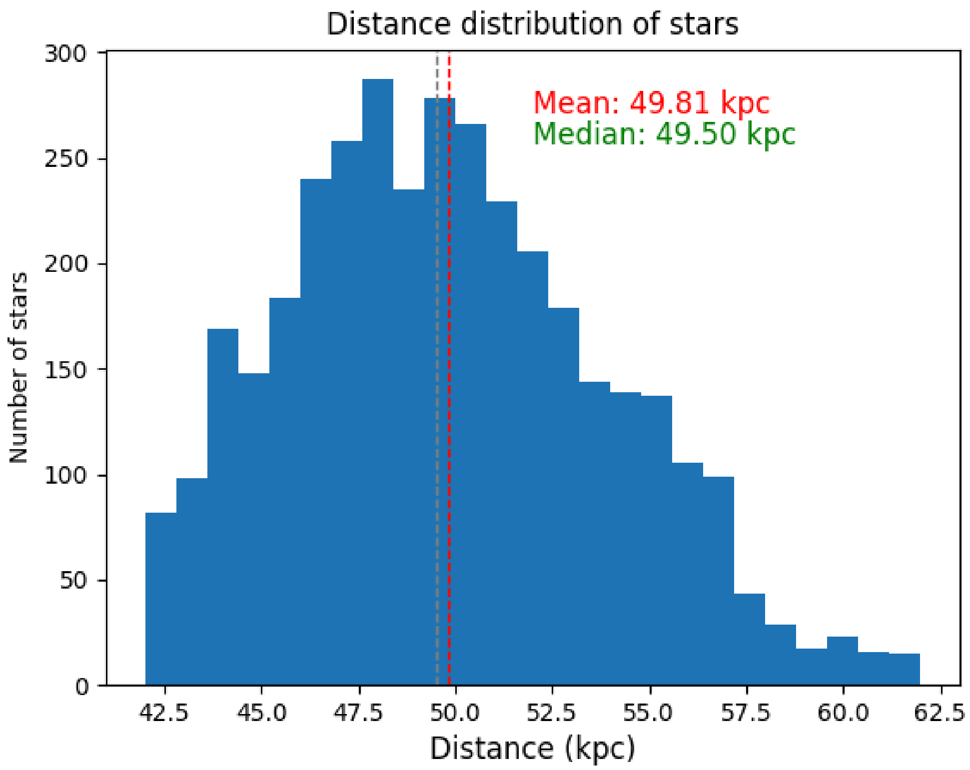
   - This code creates a histogram to visualize the distribution of distances (`distance_p50`) of stars in the dataset.
   - It calculates and displays the mean and median distances on the plot for reference.

2. **Statistical Calculations**:

   - `d_mean = np.mean(dist)` : Calculates the mean distance.
   - `d_median = np.median(dist)` : Calculates the median distance.
   - `d_std = np.std(dist)` : Calculates the standard deviation of the distances (not directly used in the plot).

Output:

- A histogram showing the distribution of distances (in kiloparsecs) of stars.
- Red dashed line: Indicates the mean distance.
- Gray dashed line: Indicates the median distance.
- Text annotations: Display the mean and median values on the plot.

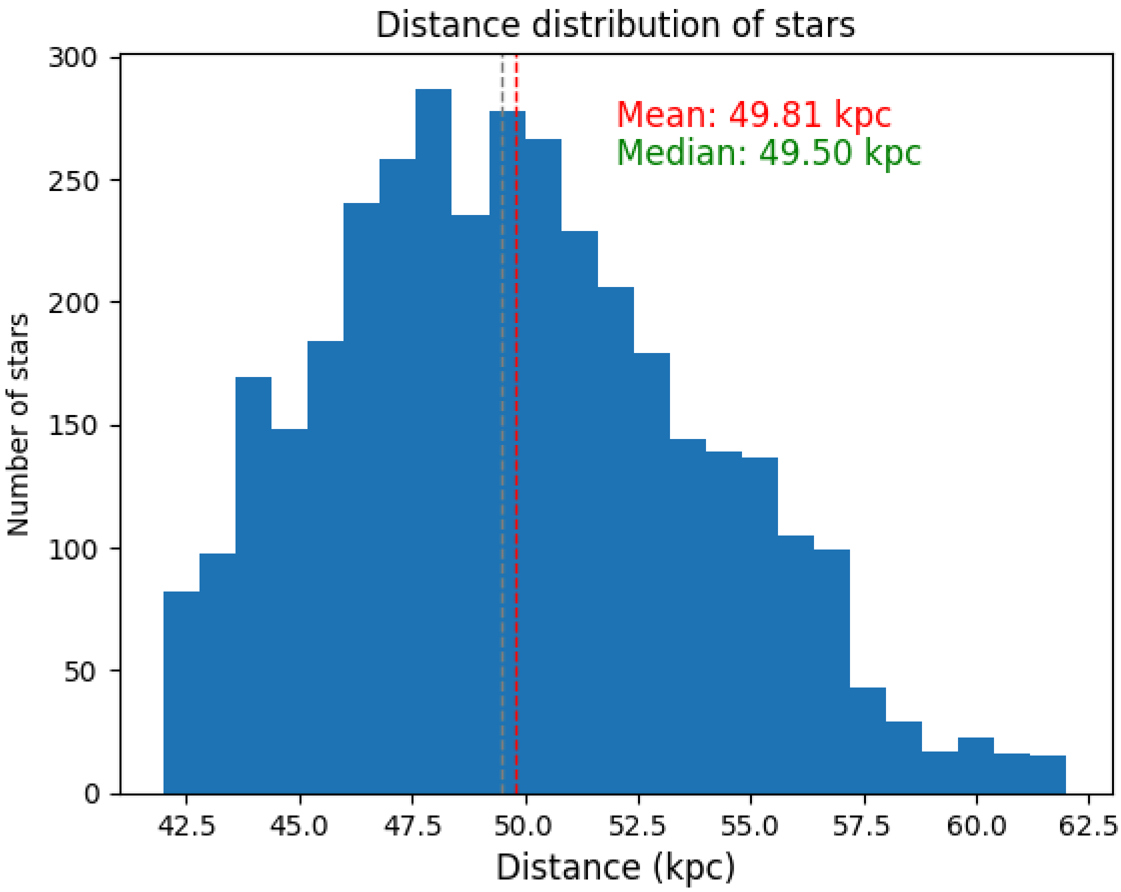


Use Case:

This visualization helps analyze the distribution of distances of stars in the dataset, providing insights into the spread and central tendency (mean and median) of the distances.

In [5]:

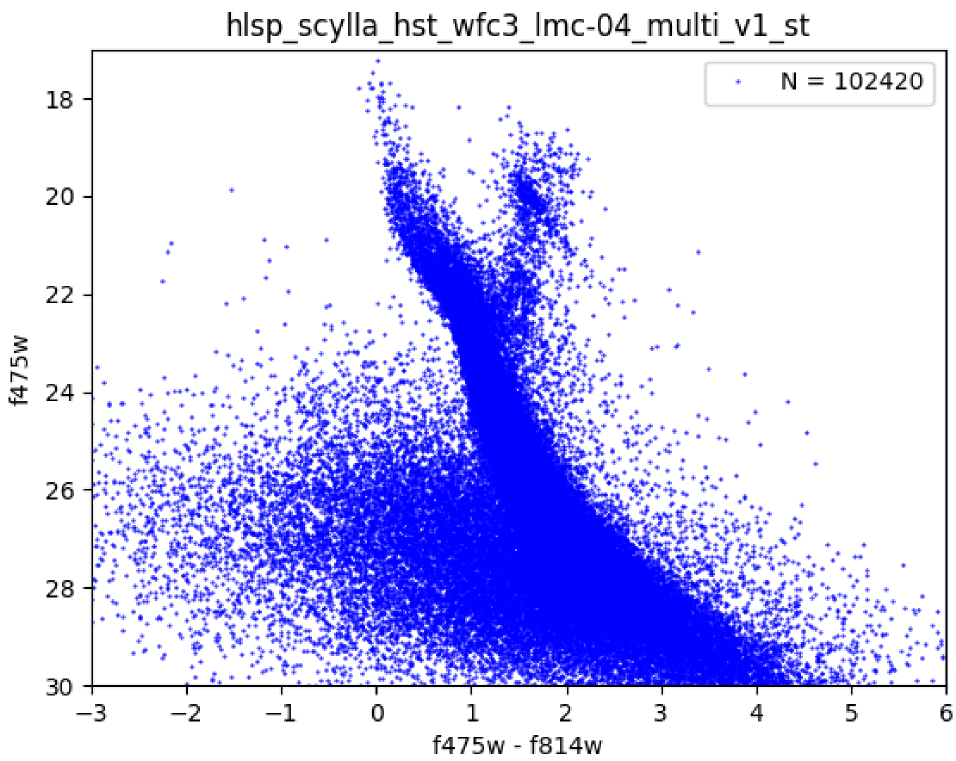
```
dist=t['distance_p50']
plt.figure()
plt.hist(dist/1000, bins=25)
plt.xlabel('Distance (kpc)', fontsize=12)
plt.ylabel('Number of stars')
plt.title('Distance distribution of stars')
d_mean = np.mean(dist)
d_median = np.median(dist)
d_std = np.std(dist)
plt.axvline(d_mean/1000, color='r', linestyle='dashed', 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=1
plt.text(0.5, 0.85, 'Median: {:.2f} kpc'.format(d_median/1000), color='g', fonts
plt.savefig('lmc_hist_dist.png')
plt.show()
```



Explanation of the Code

Output:

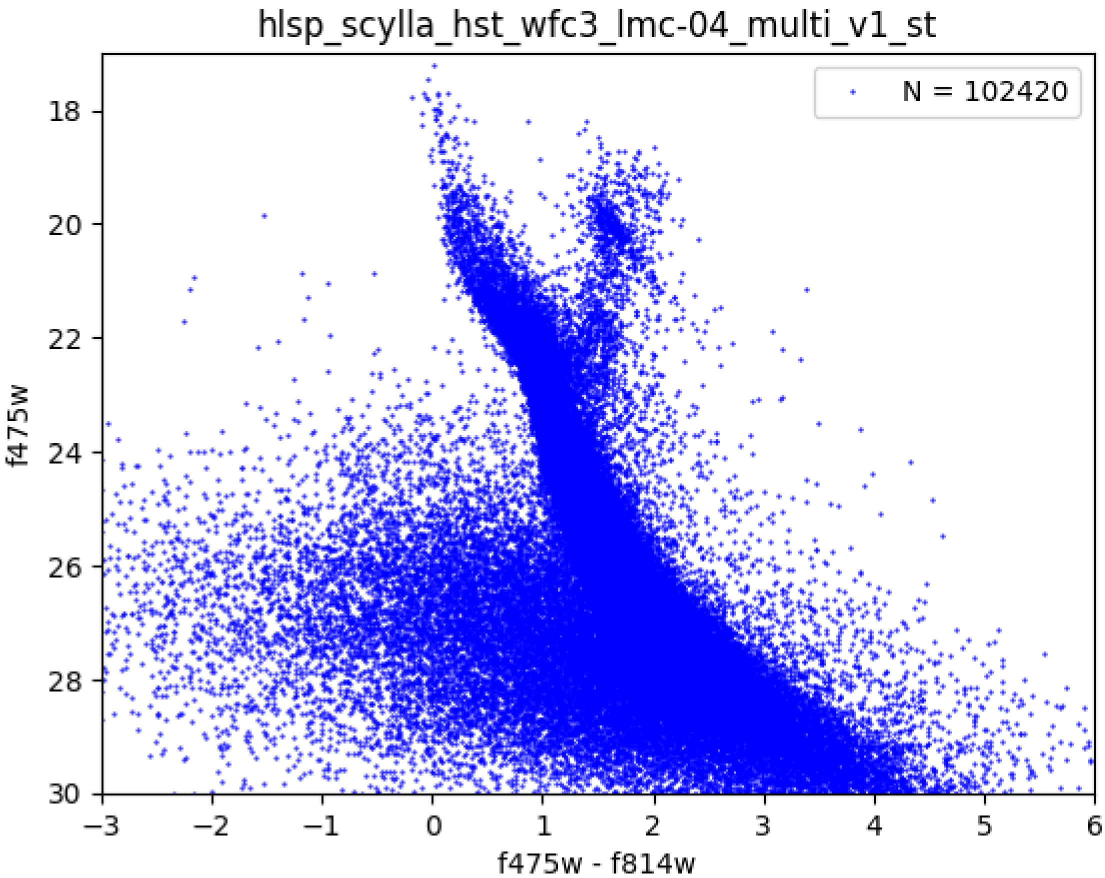
- A **color-magnitude diagram** (CMD) is generated, showing the relationship between the color index (`f475w - f814w`) and the magnitude (`f475w`) for stars in the dataset.
- The x-axis represents the color index, while the y-axis represents the magnitude (inverted for brighter stars).
- The plot is saved as `lmc_color_f475.png` .



Use Case:

This CMD is useful for studying the stellar population of the dataset, identifying different types of stars, and analyzing their properties based on their color and brightness.

In [6]:
```
# Read the FITS file into an Astropy table
cat = Table.read('hlsp_scylla_hst_wfc3_lmc-04_multi_v1_st.fits')

# Extract the F475W and F814W magnitudes in the Vega system
f475 = cat["F475W_VEGA"]
f814 = cat["F814W_VEGA"]

# Calculate the number of stars in the dataset
n = len(f475)

# Compute the color (f475w - f814w) and magnitude (f475w)
col = f475 - f814
mag = f475

# Create a color-magnitude diagram
plt.figure()
plt.plot(col, mag, '.b', ls='None', markersize=1, label='N = %s' % n)  # Plot st
plt.legend()  # Add a legend showing the number of stars
plt.xlabel('f475w - f814w')  # Label for the x-axis (color)
plt.ylabel('f475w')  # Label for the y-axis (magnitude)
plt.title('hlsp_scylla_hst_wfc3_lmc-04_multi_v1_st')  # Title of the plot
plt.xlim(-3, 6)  # Set x-axis limits for the color range
plt.ylim(30, 17)  # Set y-axis limits for the magnitude range (inverted for brig
plt.savefig('lmc_color_f475.png')  # Save the plot as an image file
plt.show()  # Display the plot
```


hlsp_scylla_hst_wfc3_lmc-04_multi_v1_st
N = 102420
f475w
18
20
22
24
26
28
30
−3
−2
−1
0
1
2
3
4
5
6
f475w - f814w

```
In [7]:
```

pepega