Importing the required packages

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from astropy.io import fits
from astropy.table import Table
from astropy import units as u
plt.ion()
import os
```

Finding the mean of an array

In this cell we test basic python commands like finding the mean of an array

We then print the result

In [2]:
```
# finding the mean of an array
a = [23, 56, 2, 4, 3]
b = np.mean(a)
print(b)
```

```
17.6
```

Commands with fits files

1. Opening the file

The first step when working with fits files is to open them from the correct directory

In [4]:
```
# opening the fits file
hdu = fits.open("./data/hst_results_nd.fits")
```

2.Getting the header and the data

Once we have the fits file, we write the header and the data in two separate variables

In [5]:
```
# getting the header and the data from the fits file
hdr = hdu[0].header
data = hdu[0].data
```

3. Printing

We print the header to examine its contents

In [6]:
```
print(hdr)
```

```
SIMPLE  =                    T / Standard FITS format
BITPIX  =                    8 / Character data
NAXIS   =                    1 / Text string
NAXIS1  =                15419 / Number of characters
VOTMETA =                    T / Table metadata in VOTable format
EXTEND  =                    T / There are standard extensions
COMMENT
COMMENT The data in this primary HDU consists of bytes which
COMMENT comprise a VOTABLE document.
COMMENT The VOTable describes the metadata of the table contained
COMMENT in the following BINTABLE extension.
COMMENT Such a BINTABLE extension can be used on its own as a perfectly
COMMENT good table, but the information from this HDU may provide some
COMMENT useful additional metadata.
COMMENT There is one following BINTABLE.
NTABLE  =                    1 / Number of following BINTABLE HDUs
END
```

4. Creating a table from the fits data

5. Write the relevant star parameters into separate variables

In this tutorial we want to generate a color-magnitude diagram using the data from the **F475W_VEGA** and **F814W_VEGA** filters

6. Print column names

In [1]:
```python
# creating a table from the data
t = Table.read("./data/hst_results_nd.fits")
ra = t["RA"]
dec = t["DEC"]
av = t["Av_p50"]
age = t["logA_p50"]
mass = t["M_ini_p50"]
f475w = t["F475W_VEGA"]
f814w = t["F814W_VEGA"]
#t[0].colnames
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[1], line 2
      1 # creating a table from the data
----> 2 t = Table.read("./data/hst_results_nd.fits")
      3 ra = t["RA"]
      4 dec = t["DEC"]

NameError: name 'Table' is not defined
```

7. Sort the stars based on age

We sort the stars into two groups based on their age. The first group ('young stars') contains all stars with *log(age) < 8.5*. The second group ('old stars') contains all stars with *log(age) > 10.0*

8. Print the number of stars in each group

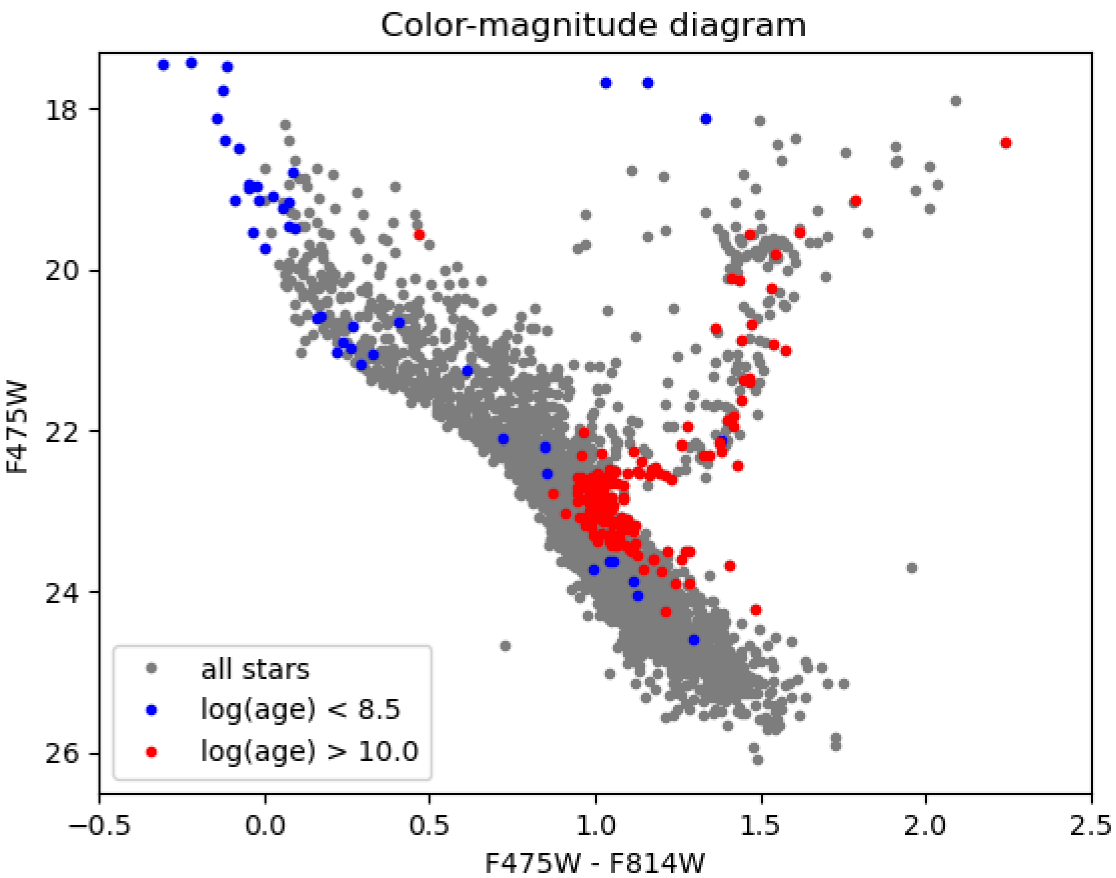
9. Generate color-magnitude diagram

The generated color-magnitude diagram compares the young stars, the old stars and all stars from the data.

10. Save the figure as a pdf

In [9]:
```
young_stars = t[age < 8.5]
old_stars = t[age > 10.0]
#print(old_stars[0:100])
print(len(young_stars))
print(len(old_stars))
plt.figure()
plt.plot(f475w-f814w, f475w, ls = "", color='gray', marker='.', label = 'al
plt.ylim(26.5, 17.3)
plt.xlim(-0.5, 2.5)
plt.plot(young_stars["F475W_VEGA"]-young_stars["F814W_VEGA"],young_stars["F4
plt.plot(old_stars["F475W_VEGA"]-old_stars["F814W_VEGA"],old_stars["F475W_VE
plt.xlabel("F475W - F814W")
plt.ylabel("F475W")
plt.title("Color-magnitude diagram")
plt.legend()
plt.savefig("cmd_by_age.pdf")
```

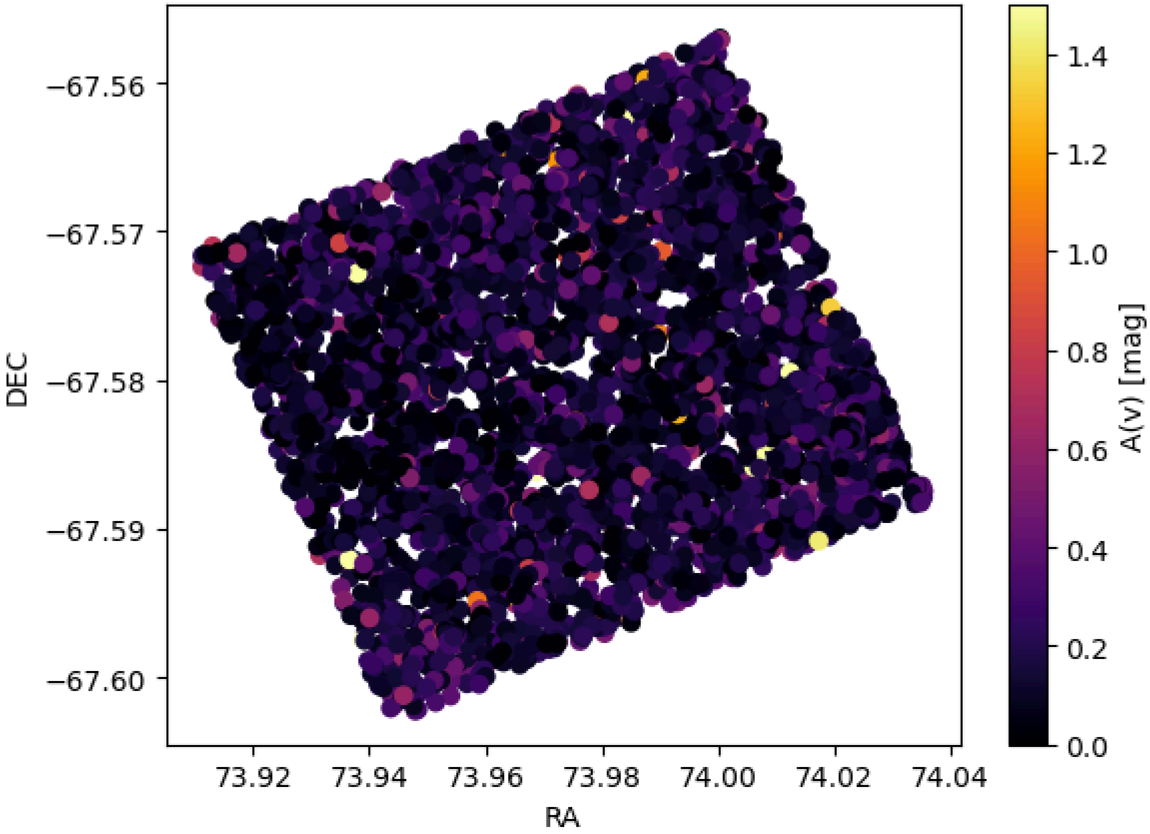
```
44
170
```



11. Generate an extinction plot

This plot shows the extinction of stars in different parts of the sky

In [10]:
```
plt.figure()
#plt.plot(ra,dec, 'r.', ls='')
# Av max = 5 ---> limit Av to 1.5
cb = plt.scatter(ra, dec, c=av, marker = 'o', cmap='inferno', vmin = 0, vma
plt.xlabel("RA")
plt.ylabel("DEC")
plt.colorbar(cb, label="A(v) [mag]")
plt.savefig("lmc_av_spatial.pdf")
```



12. Collect distance data

In [11]:
```
dist = t["distance_p50"]
```

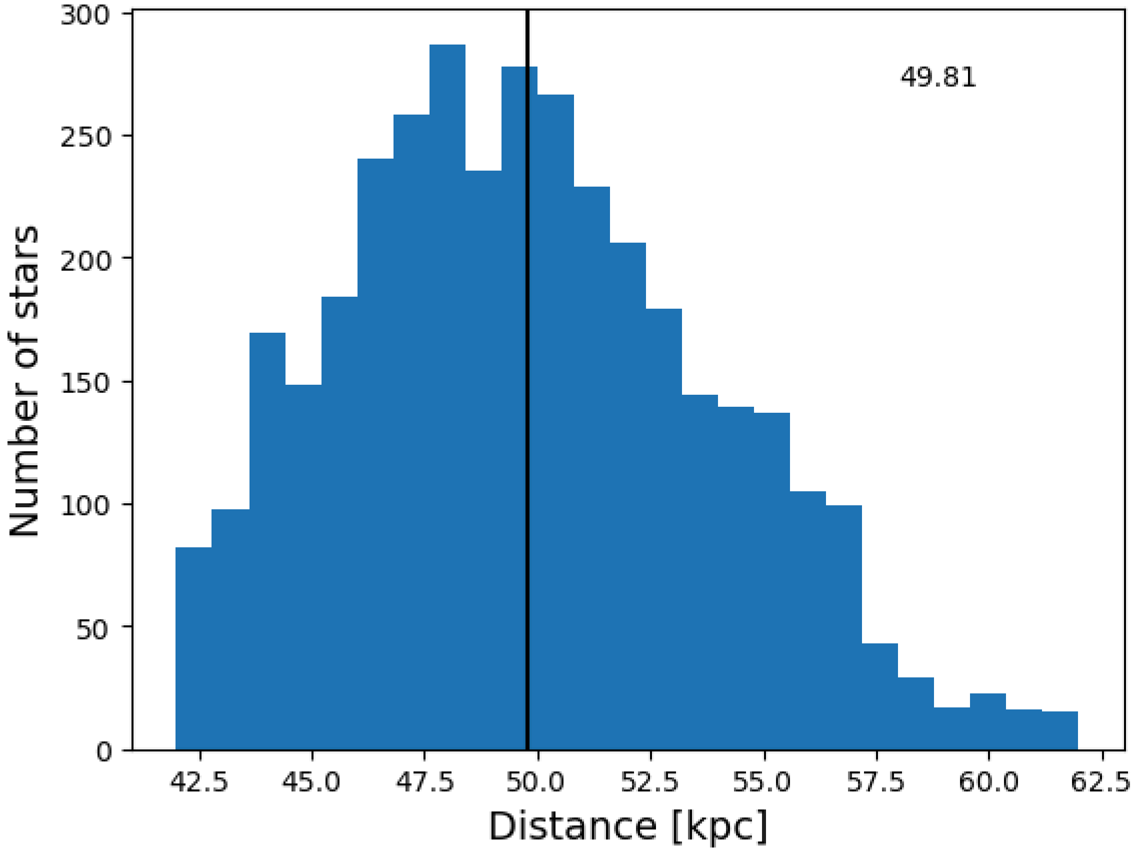
13. Create a histogram

This histogram shows the number of stars in different distance intervals, as well as the mean distance

14. Save the histogram as pdf

In [12]:
```
plt.figure()
plt.hist(dist/1000, bins = 25)
plt.xlabel("Distance [kpc]", fontsize = 14)
plt.ylabel("Number of stars", fontsize = 14)
d_mean = np.mean(dist/1000)
print(d_mean)
plt.axvline(d_mean, c = "k")
plt.text(58, 270, '%s' % np.around(d_mean, decimals = 2))
plt.savefig("lmc_hist_distance.pdf")
```

```
49.806221103883765
```



15. Read data from HST for the lmc_04 field

In [13]:
```
# read data from HST for the lmc_04 field
cat = Table.read("./hlsp_scylla_hst_wfc3_lmc-04_multi_v1_st.fits")
```

16. Get values of the magnitude for two filters

Here we write the magnitude data for the **F475W_VEGA** and the **F814W_VEGA** filters in two separate variables

In [14]:
```
# get values of the magnitude for two filters
cat[0].colnames
f475 = cat["F475W_VEGA"]
f814 = cat["F814W_VEGA"]
```

17. Generate the color-magnitude diagram for the two filters

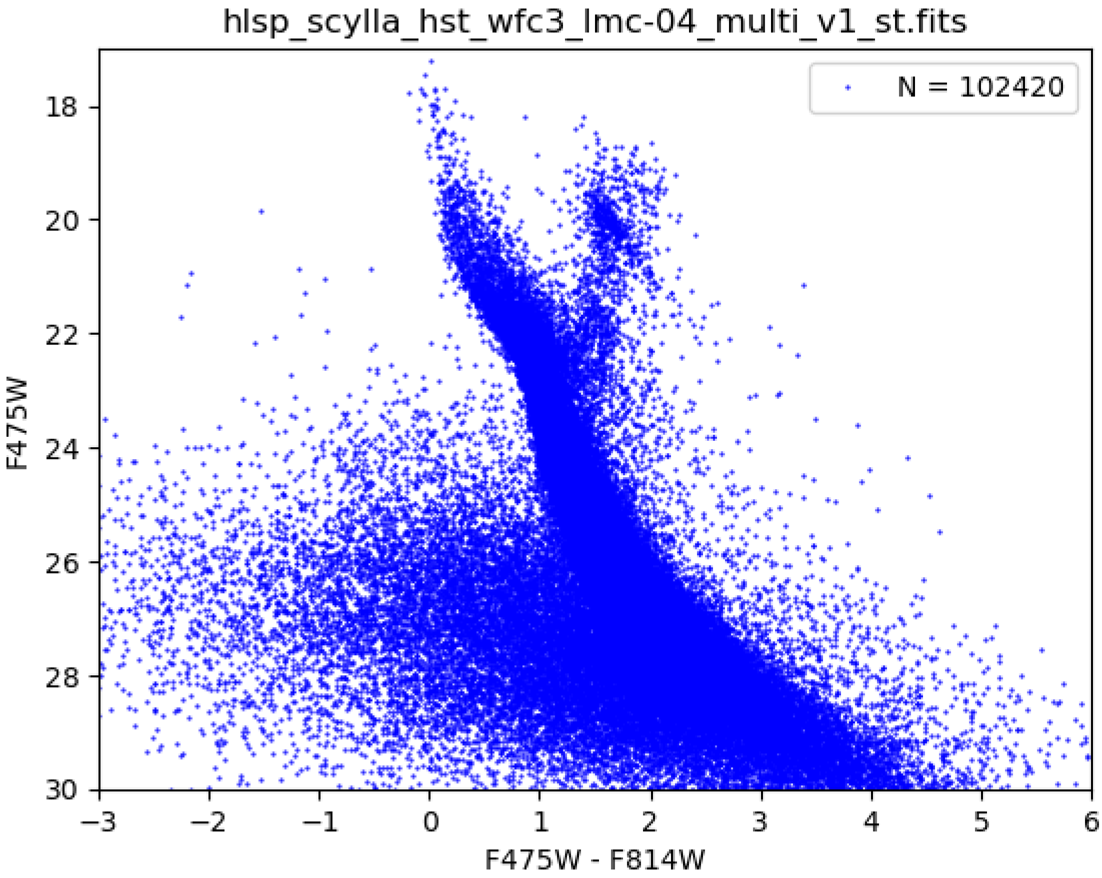
18. Save the diagram as pdf

In [20]:
```
# generate the color-magniture diagram for the two filters
col = f475 - f814
mag = f475
f = "hlsp_scylla_hst_wfc3_lmc-04_multi_v1_st.fits"
n = len(f475)

plt.figure()
plt.plot(col, mag, '.b', markersize = 1, ls=' ', label='N = %s' % n)

plt.legend()
plt.xlim(-3,6)
plt.ylim(30,17)
plt.xlabel("F475W - F814W")
plt.ylabel("F475W")
```

```
plt.title("%s" % f)
plt.savefig("cmd_lmc_04.pdf")
```



In [ ]: