```
pip install astropy
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: astropy in /usr/local/lib/python3.10/dist-packages (5.2.2)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from astropy) (1.22.4)
Requirement already satisfied: pyerfa>=2.0 in /usr/local/lib/python3.10/dist-packages (from astropy) (2.0.0.3)
Requirement already satisfied: PyYAML>=3.13 in /usr/local/lib/python3.10/dist-packages (from astropy) (6.0)
Requirement already satisfied: packaging>=19.0 in /usr/local/lib/python3.10/dist-packages (from astropy) (23.1)
```

```
import numpy as np
import matplotlib.pyplot as plt
from astropy.io import fits
```

Now we are ready to load some FITS file.We will open the M13 FITS file. To load one of these in to Python we use the open function, and pass the location of the file we want as the first argument. We took the fits file from "https://dr12.sdss.org/fields/"

```
M42 = fits.open('/content/frame-u-006073-4-0063.fits')
```

Printing Information Of the FITS File

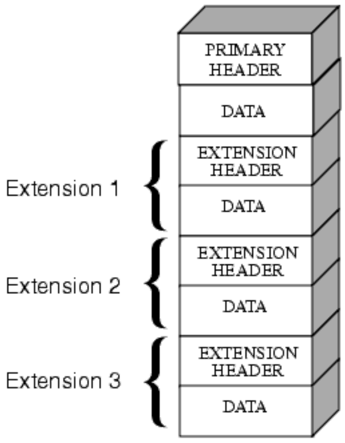
```
M42
```

```
[<astropy.io.fits.hdu.image.PrimaryHDU object at 0x7f3e8f56e680>, <astropy.io.fits.hdu.image.ImageHDU object at
0x7f3e8f56fd00>, <astropy.io.fits.hdu.table.BinTableHDU object at 0x7f3e8f56fee0>,
<astropy.io.fits.hdu.table.BinTableHDU object at 0x7f3e8d870100>]
```

```
M42.info()
```

```
Filename: /content/frame-u-006073-4-0063.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      96   (2048, 1489)   float32
  1                1 ImageHDU         6   (2048,)   float32
  2                1 BinTableHDU     27   1R x 3C   [49152E, 2048E, 1489E]
  3                1 BinTableHDU     79   1R x 31C   [J, 3A, J, A, D, D, 2J, J, D, D, D, D, D, D, D, D, D, D, D, D, D, [
```

The image information is located in the PRIMARY block. We can see that it's made up of two HDU (Header Data Unit) objects. A HDU is a high level FITS file component, typically containing data and an associated header.



```
# Printing Information Of Primary Block
M42[0]
```

```
<astropy.io.fits.hdu.image.PrimaryHDU at 0x7f3e8f56e680>
```

```
M42[0].header
```

```
SCDOFSET=                  410 / scdDisplayOffset
SCDDYNTH=                  -21 / scdDynamicThresh
SCDSTTHL=                   30 / scdStaticThreshL
SCDSTTHR=                   30 / scdStaticThreshR
SCDREDSZ=                  527 / scdReduceSize
SCDSKYL =                 1247 / scdSkyLeft
SCDSKYR =                 1214 / scdSkyRight
COMMENT  CCD-specific parameters
CAMROW  =                    3 / Row in the imaging camera
BADLINES=                    0 / Number of bad lines in frame
EQUINOX =              2000.00 /
SOFTBIAS=                 1000 / software "bias" added to all DN
BUNIT   = 'nanomaggy'          / 1 nanomaggy = 3.631e-6 Jy
FILTER  = 'u       '           / filter used
CAMCOL  =                    4 / column in the imaging camera
VERSION = 'v5_6_3  '
DERV_VER= 'NOCVS:v8_23'
ASTR_VER= 'NOCVS:v5_24'
ASTRO_ID= '2010-03-29T20:33:41 25639'
BIAS_ID = 'PS      '
FRAME_ID= '2010-08-10T09:06:37 03057'
KO_VER  = 'devel   '
PS_ID   = '2010-03-29T20:12:50 25067 camCol 4'
ATVSN   = 'NOCVS:v5_24'        / ASTROTOOLS version tag
RADECSYS= 'ICRS    '           / International Celestial Ref. System
CTYPE1  = 'RA---TAN'           /Coordinate type
CTYPE2  = 'DEC--TAN'           /Coordinate type
CUNIT1  = 'deg     '           /Units
CUNIT2  = 'deg     '           /Units
CRPIX1  =        1025.00000000 /X of reference pixel
CRPIX2  =        745.000000000 /Y of reference pixel
CRVAL1  =        83.9233250659 /RA of reference pixel (deg)
CRVAL2  =       -5.31904299495 /Dec of reference pixel (deg)
CD1_1   =    0.000110087467634 /RA deg per column pixel
CD1_2   =    2.52333879844E-08 /RA deg per row pixel
CD2_1   =    4.95111108581E-08 /Dec deg per column pixel
CD2_2   =   -0.000110024887054 /Dec deg per row pixel
HISTORY GSSSPUTAST: Aug 10 09:06:42 2010
COMMENT  Calibration parameters
COMMENT  Floats truncated at 10 binary digits with FLOATCOMPRESS
NMGY    =            0.0137521 / Calibration factor [nMgy per count]
NMGYIVAR=             0.155281 / Calibration factor inverse variance
VERSIDL = '7.0     '           / Version of IDL
VERSUTIL= 'v5_5_5  '           / Version of idlutils
VERSPOP = 'v1_11_1 '           / Version of photoop product
PCALIB  = '/clusterfs/riemann/raid006/dr8/groups/boss/calib/dr8_final' / Value o
PSKY    = '/clusterfs/riemann/raid006/dr8/groups/boss/photo/sky' / Value of PHOT
RERUN   = '301     '           / rerun
HISTORY SDSS_FRAME_ASTROM: Astrometry fixed for dr9 Sun Jun 24 23:13:07 2012
```

```
# We are only interested in primary block and will extract its data
data = M42[0].data
```

```
type(data)
```

```
numpy.ndarray
```

Our Image is stored as 2D array

```
data.shape
```

```
(1489, 2048)
```

```
data.dtype.name
```

```
'float32'
```

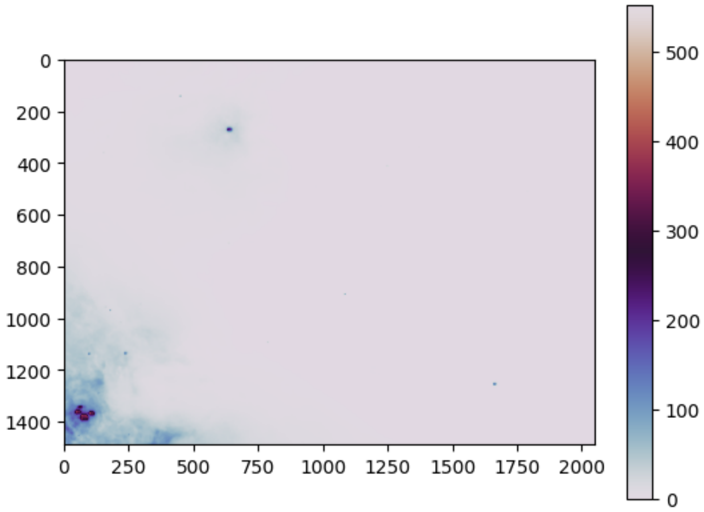
```
# Printing some statistics of the Image/2D Array
print(data)
print('Min:', np.min(data))
print('Max:', np.max(data))
print('Mean:', np.mean(data))
print('Stdev:', np.std(data))
```

```
[[ 2.4609375e-01  1.9140625e-01  9.5703125e-01 ...  2.1923828e-01
   1.2008667e-02  6.7260742e-02]
 [ 1.0937500e-01  1.9140625e-01  1.0390625e+00 ...  1.1993408e-02
   6.7260742e-02  1.3623047e-01]
 [ 1.0937500e-01  2.1875000e-01  1.1074219e+00 ... -4.3273926e-02
  -2.9449463e-02  1.1978149e-02]
 ...
 [ 1.8750000e-01  2.1508789e-01  9.2375000e+01 ...  2.4291992e-01
   1.6015625e-01  2.2900391e-01]
 [ 2.4218750e-01  2.8320312e-01  8.9750000e+01 ...  1.7382812e-01
```

```
   2.7050781e-01  1.6015625e-01]
 [ 1.8750000e-01  6.4575195e-02  8.7500000e+01 ...  1.4624023e-01
   1.3256836e-01  1.7382812e-01]]
Min: -0.34472656
Max: 552.0
Mean: 5.3398747
Stdev: 15.16415
```

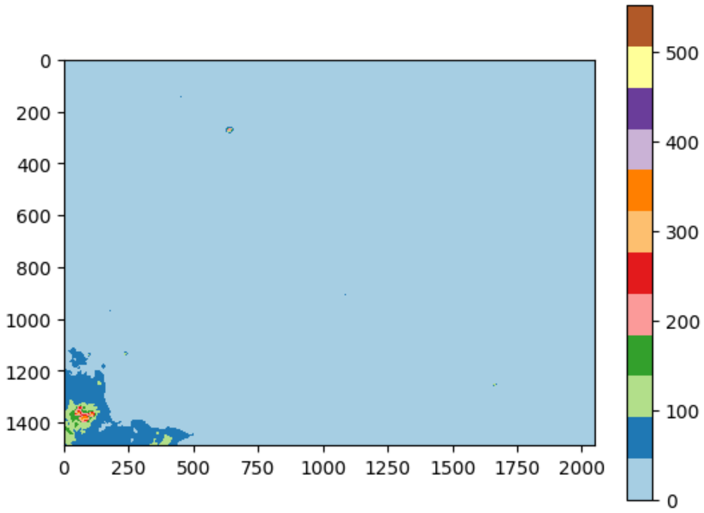
```
# Printing the Image
plt.imshow(data, cmap='twilight')
plt.colorbar()
```

```
<matplotlib.colorbar.Colorbar at 0x7f3e8d8737c0>
```



```
plt.imshow(data, cmap='Paired')
plt.colorbar()
```

```
<matplotlib.colorbar.Colorbar at 0x7f3e8c88b520>
```

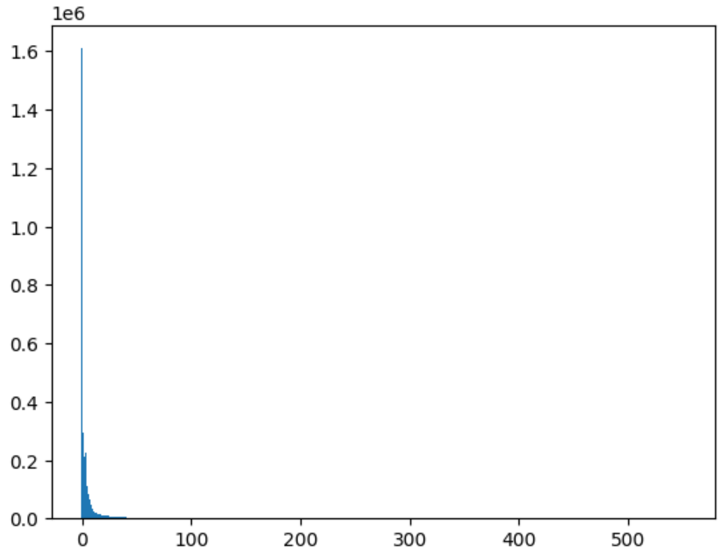


```
plt.imshow(data, cmap='gist_heat')
plt.colorbar()
```

```
<matplotlib.colorbar.Colorbar at 0x7f3e8c79ece0>
```

Now we will be printing Histogrm , which will help us to understand the Values of **Brighteness for Each Pixel**. For printing the histogram we need to flaten the array

```
histogram = plt.hist(data.flat, bins=500)
```

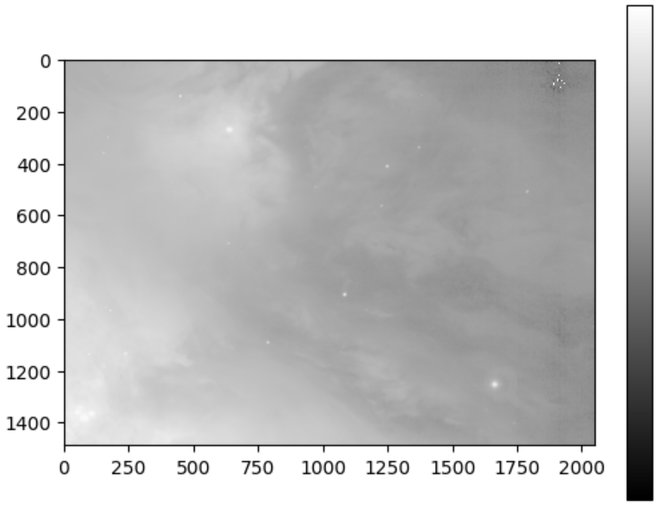


As we Can observe that most of the High Brighteness values are from 0 to 100, so we will scale the image to that values.

```
from matplotlib.colors import LogNorm
# Scaling the Vaues to Logarethemic values

plt.imshow(data, cmap='gray', norm=LogNorm(),)
cbar = plt.colorbar(ticks=[4.e3,1.e4,2.e4])
cbar.ax.set_yticklabels(['5,000','10,000','20,000'])
```

```
[Text(1, 4000.0, '5,000'),
 Text(1, 10000.0, '10,000'),
 Text(1, 20000.0, '20,000')]
```



Now we will Produce an **RGB IMAGE OF M42**

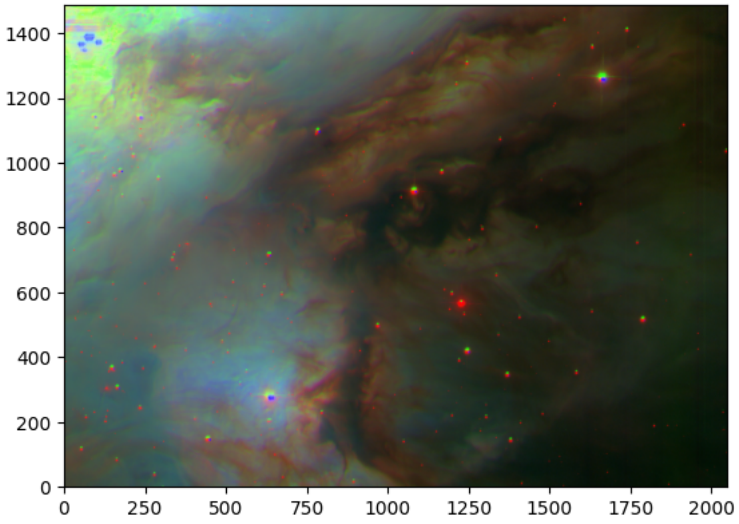
```
M42u = fits.open('frame-u-006073-4-0063.fits')
M42i = fits.open('frame-i-006073-4-0063.fits')
M42g = fits.open('frame-g-006073-4-0063.fits')
```

```
# Function used for making RGB image
from astropy.visualization import make_lupton_rgb
g = M42g[0].data
```

```
i = M42i[0].data
u = M42u[0].data
```

```
#it has the following value   R G B
rgb_default = make_lupton_rgb(i,g,u,stretch=2.5,Q=10)


plt.imshow(rgb_default, origin='lower')
plt.savefig("rgbM42.jpg")
```



✓ 2s completed at 12:07 AM ● ×