

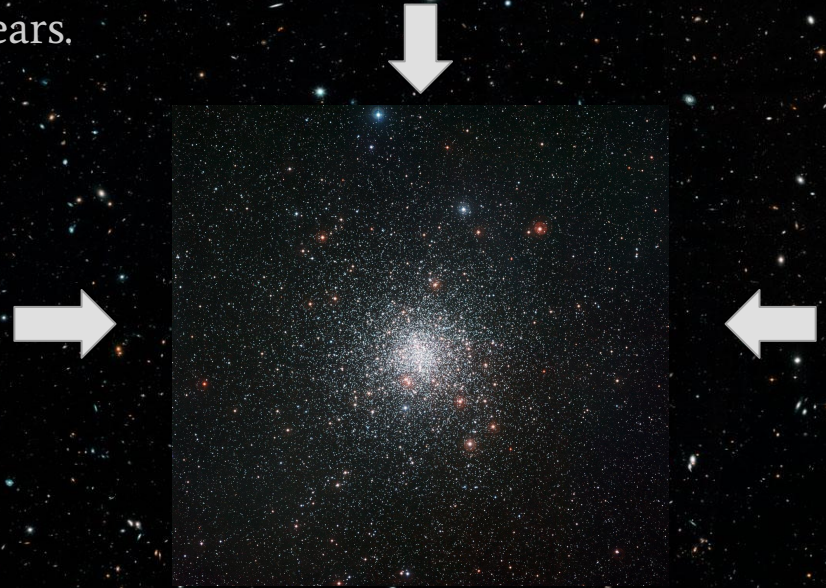
Animated 3D Color-Magnitude Diagram of Messier 4 Globular Cluster

...

By Kingsley Ehrich and Riley Clark

About Messier 4

Messier 4 is a globular cluster in the constellation of Scorpius. You can see it as a fuzzy ball of light even with the smallest telescopes. It is about 75 light-years across, and is approximately 7,200 light-years away, making it the closest globular cluster to the Solar System. It has an estimated age of 12.2 billion years.



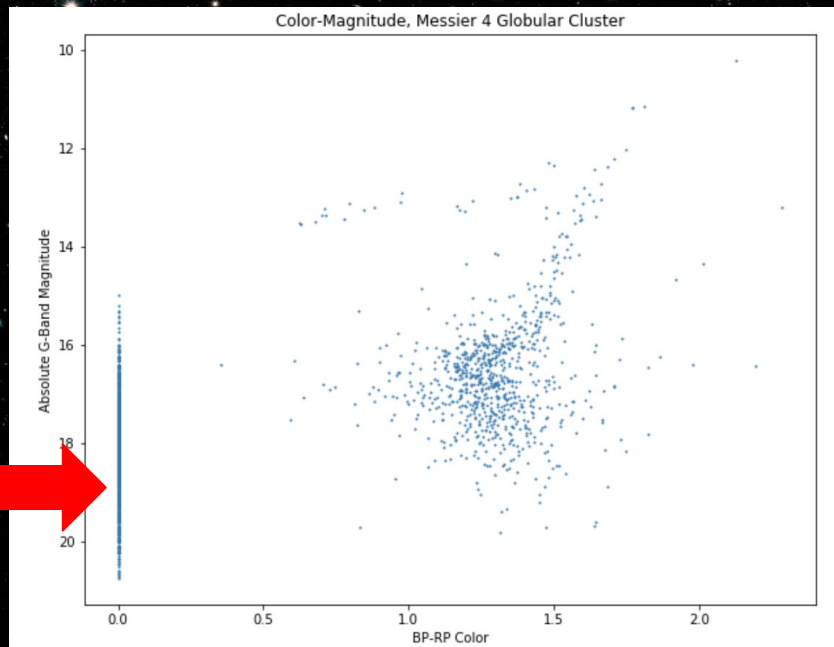
Getting the Data!

- Very helpful instructor Yilun showed us a site called <https://vizier.u-strasbg.fr/viz-bin/VizieR-2> that allowed us to download data from the European Space Agency's Gaia Mission
- We used the Messier 4 data from the second Gaia data release and selected the parameters for which we wanted data.
- The main parameters we were interested in were Effective Temperature (Teff), BP-RP Color, and G Magnitude.
- It should also be noted that we inverted the y-axis, corresponding to G Magnitude, in all of our plots because of the way the magnitude scale works.

Cleaning Up the Data

- When we first graphed the normal CMD, using BP-RP Color and G Magnitude, we noticed there was a line of outlier data points:

WACK!



Cleaning Up the Data

- To eliminate this blasphemous false data, we used the following code:

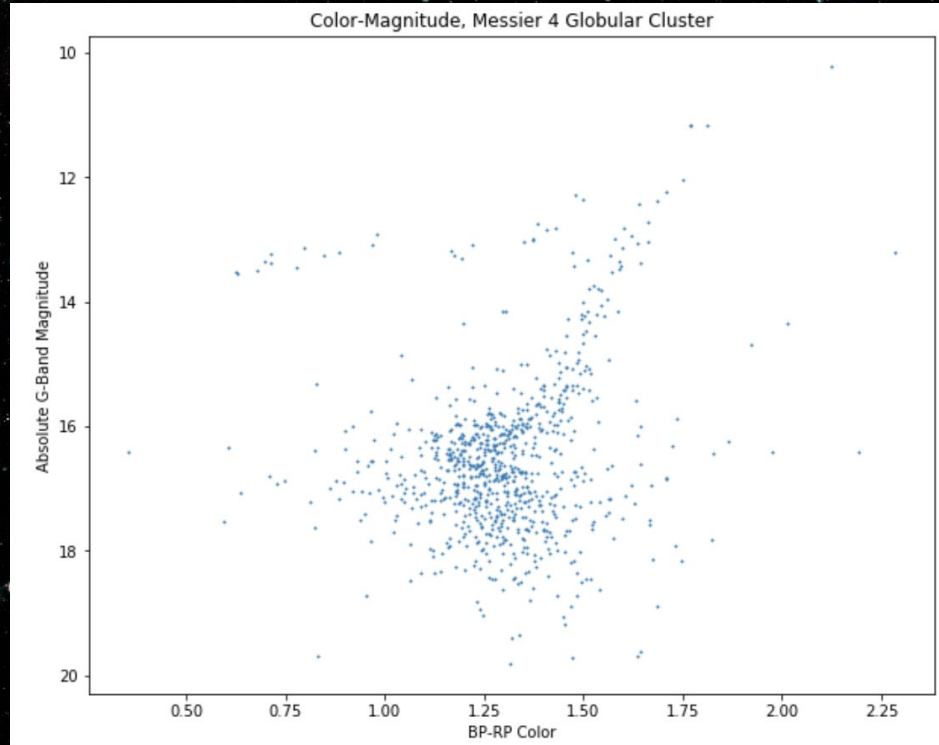
```
bprp_array = np.array([])
gmag_array = np.array([])
lum_array = np.array([])
teff_array = np.array([])
for i in data['BP-RP']:
    if i != 0.0:
        bprp_array = np.append(bprp_array, i)

for k in np.arange(0, 1913):
    if bprp_arr1[k] != 0.0:
        gmag_array = np.append(gmag_array, dat_arr[k])
        lum_array = np.append(lum_array, lum_arr[k])
        teff_array = np.append(teff_array, teff_arr[k])
```

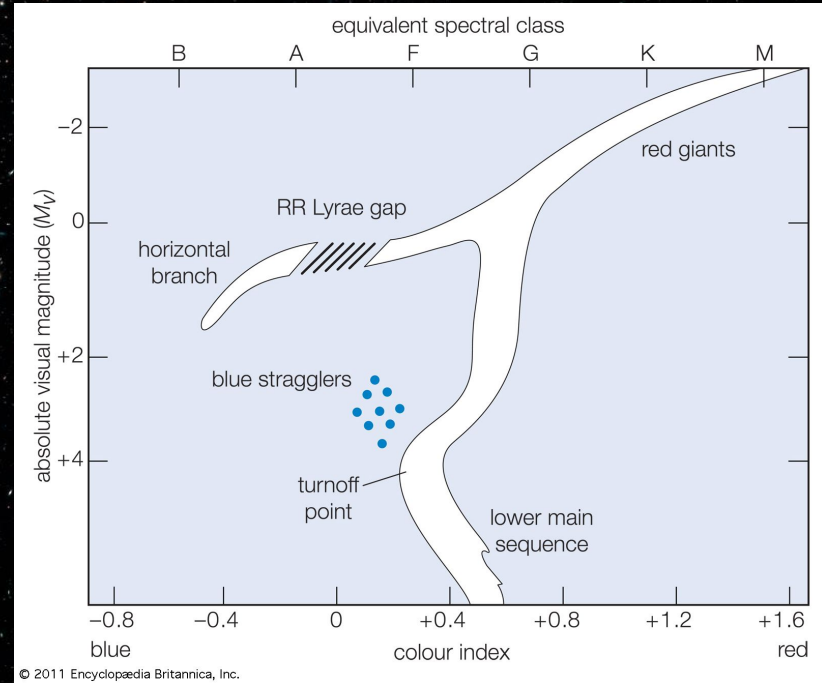
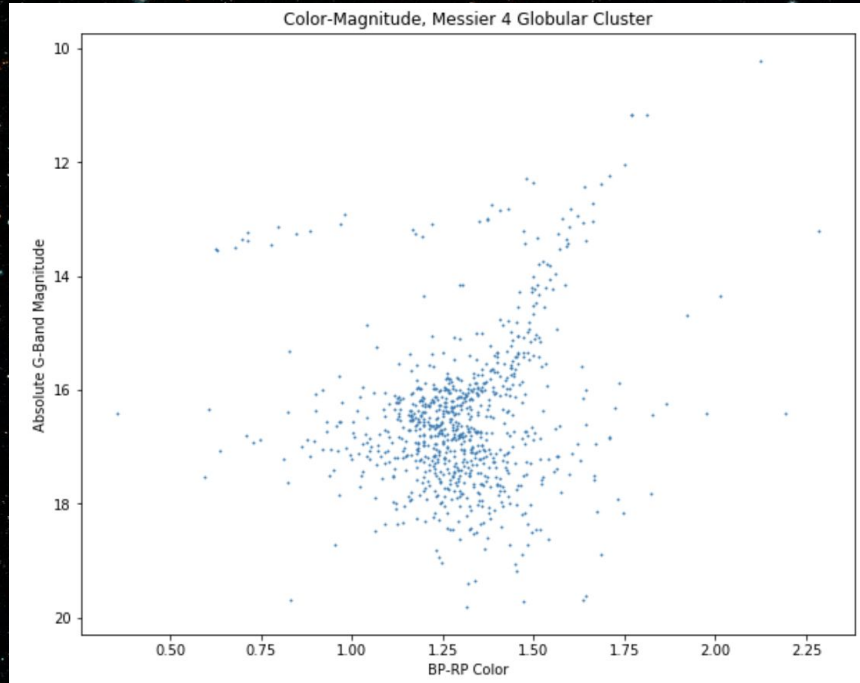
NOTE: We didn't actually end up using luminosity, but we made the arrays for it in case we needed it.

Prettier 2D CMD

- After eliminating the false zero points, we made a new and improved CMD:



What Does It Mean?



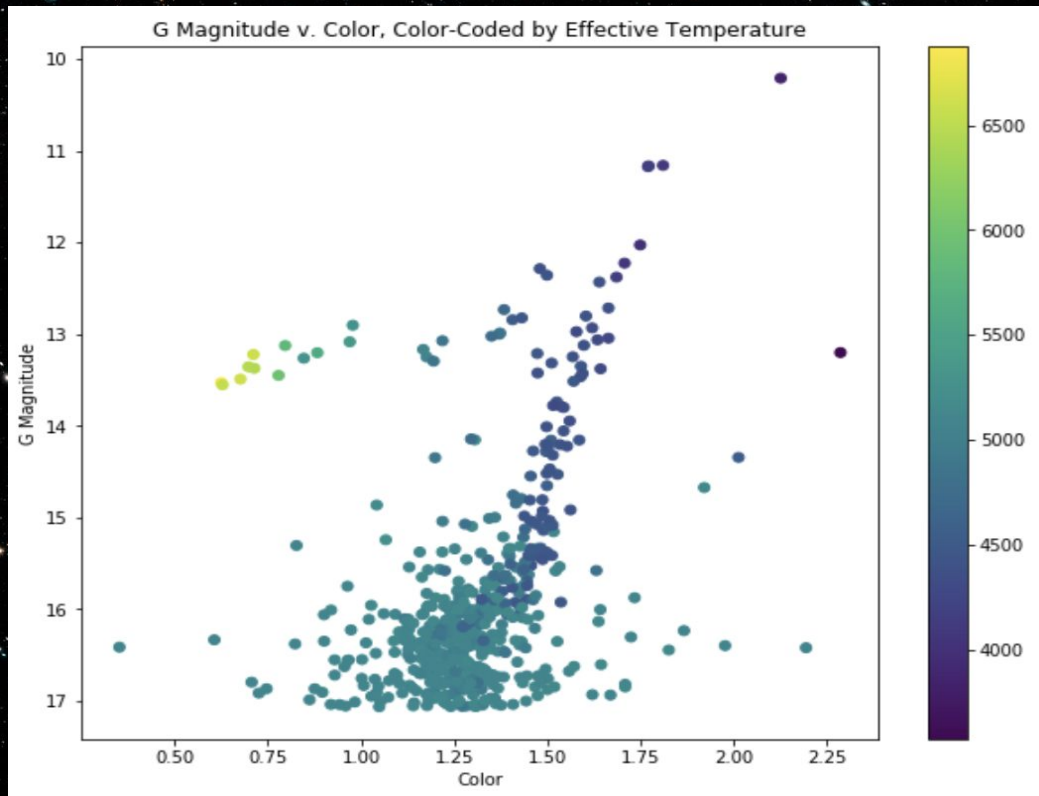
Now, Let's Spice It Up

- After successfully making a 2D CMD, we spiced it up by color-coding it by effective temperature using the following code:

```
finalteff_array = np.array([])
finalbprp_array = np.array([])
finalgmag_array = np.array([])
finallum_array = np.array([])
for k in np.arange(0, 841):
    if teff_array[k] != 0.0:
        finalgmag_array = np.append(finalgmag_array, gmag_array[k])
        finallum_array = np.append(finallum_array, lum_array[k])
        finalteff_array = np.append(finalteff_array, teff_array[k])
        finalbprp_array = np.append(finalbprp_array, bprp_array[k])
```


Sick Color-Coded CMD

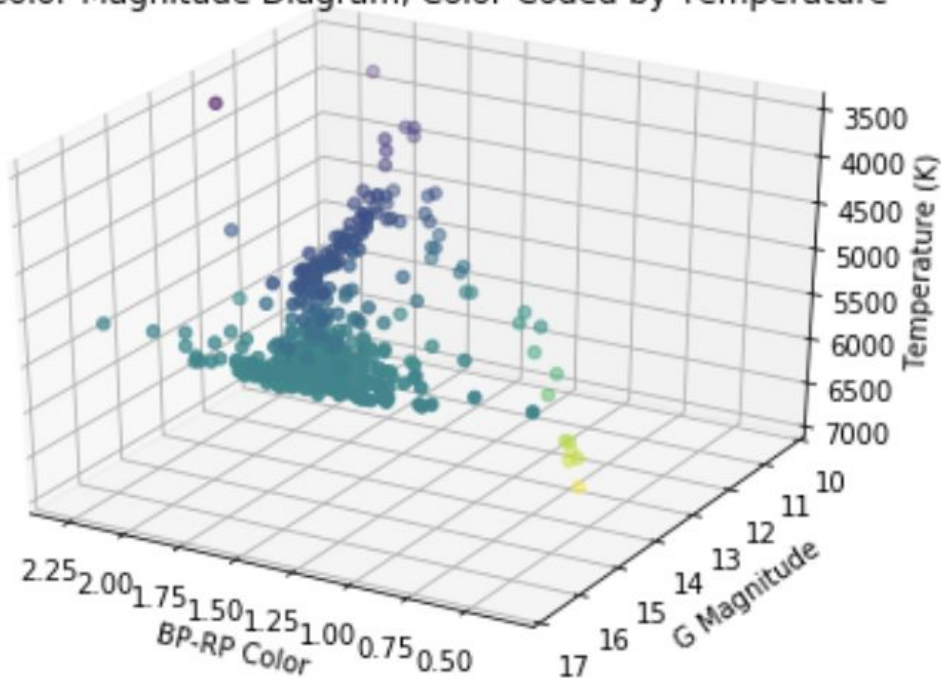
- We passed in the Teff array for c (color) to color-code our CMD by temperature



Now.... Make It 3D!!!

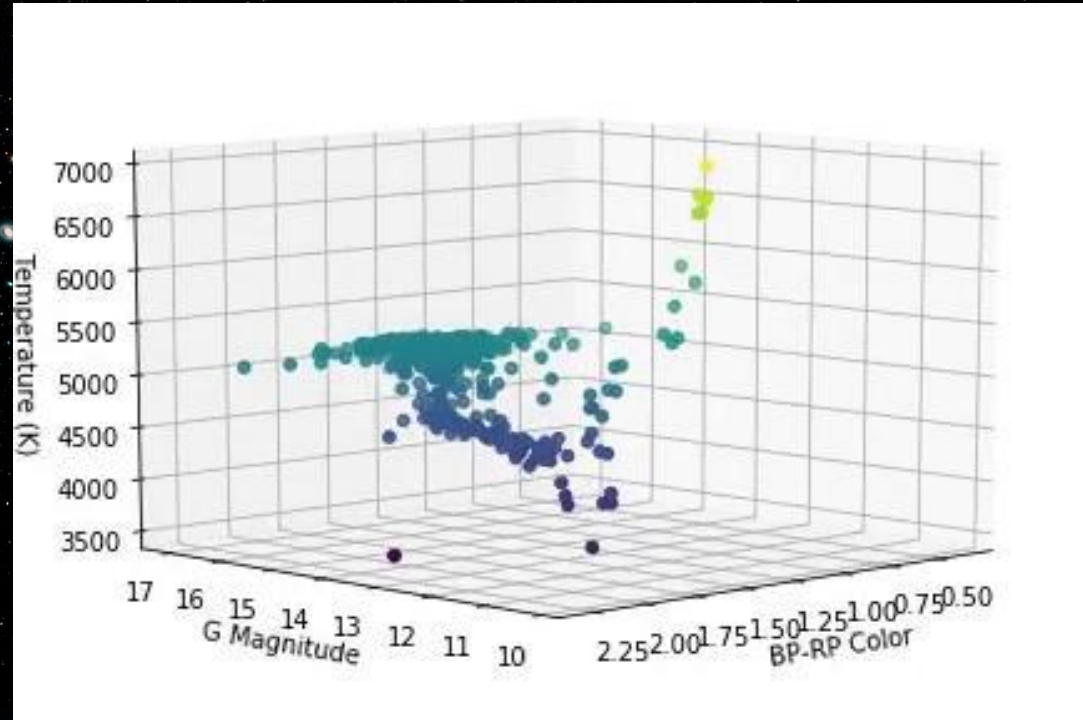
- Then instead of simply color-coding it, we made the CMD 3D!!
- We used matplotlib Axes3D to make this bad boy

3D Color-Magnitude Diagram, Color-Coded by Temperature



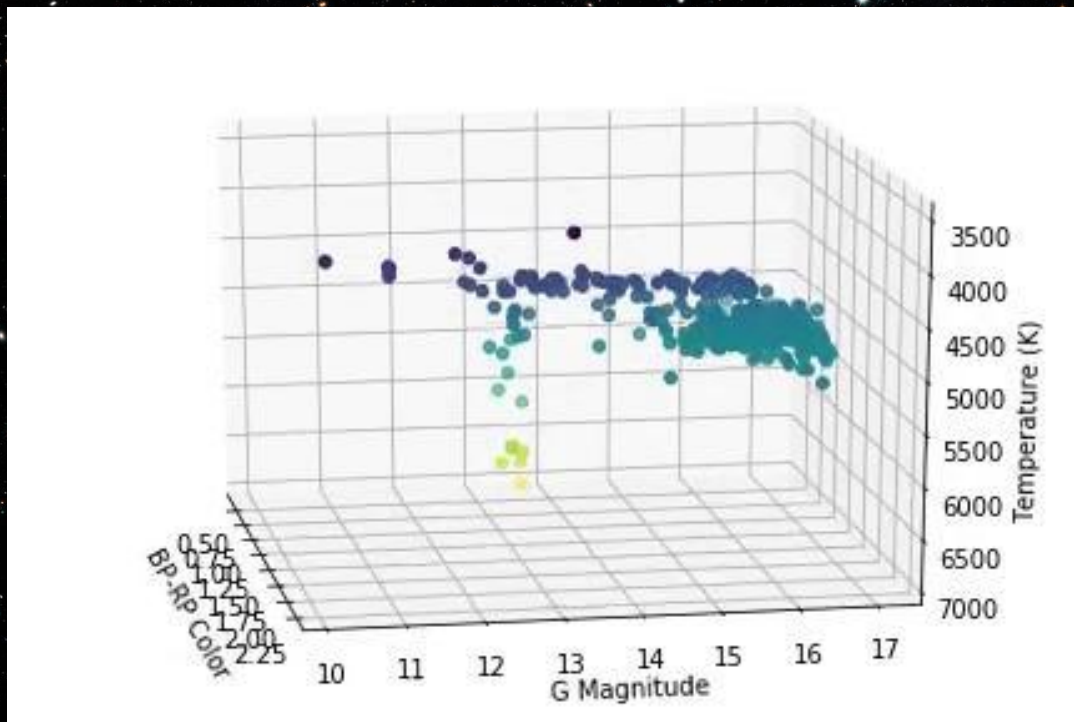
ANIMATE

- JUST WHEN YOU THOUGHT IT COULDN'T GET BETTER
- We used matplotlib animation to make this. It allows you to pass in elevation and azimuth, which dictates the angle(s) around which the figure is rotated.



ANIMATE PT 2

- MORE ANGLES.



ANIMATE PT 3

- Because we know you're dying for more.

