Lab1_114415543

February 1, 2024

1 Astronomy 443 - Lab 1

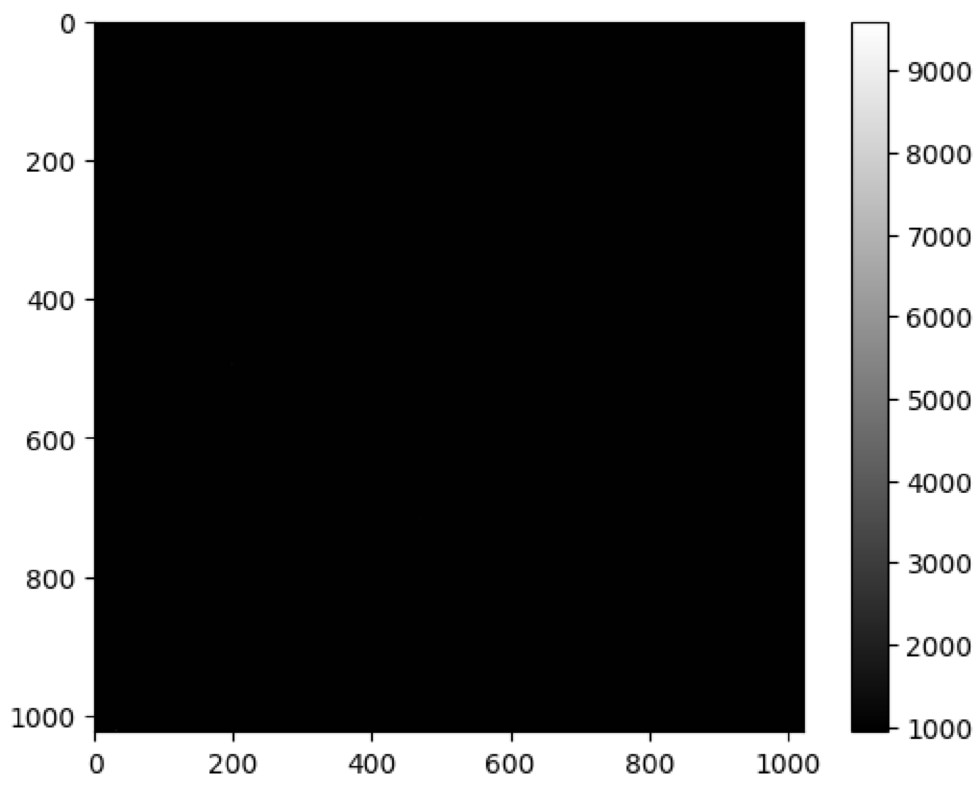
1.1 Purpose

The objectives of this lab are to learn how to use the equipment needed to conduct observations during the semester. This equipment includes the STL-1001E, DADOS spectrographs, and the Mt. Stony Brook 14-inch telescope. Sets of calibration data for imaging and spectroscopic observations were taken, and strategies on analyzing said data were developed.

1.2 4.1 Bias Frames

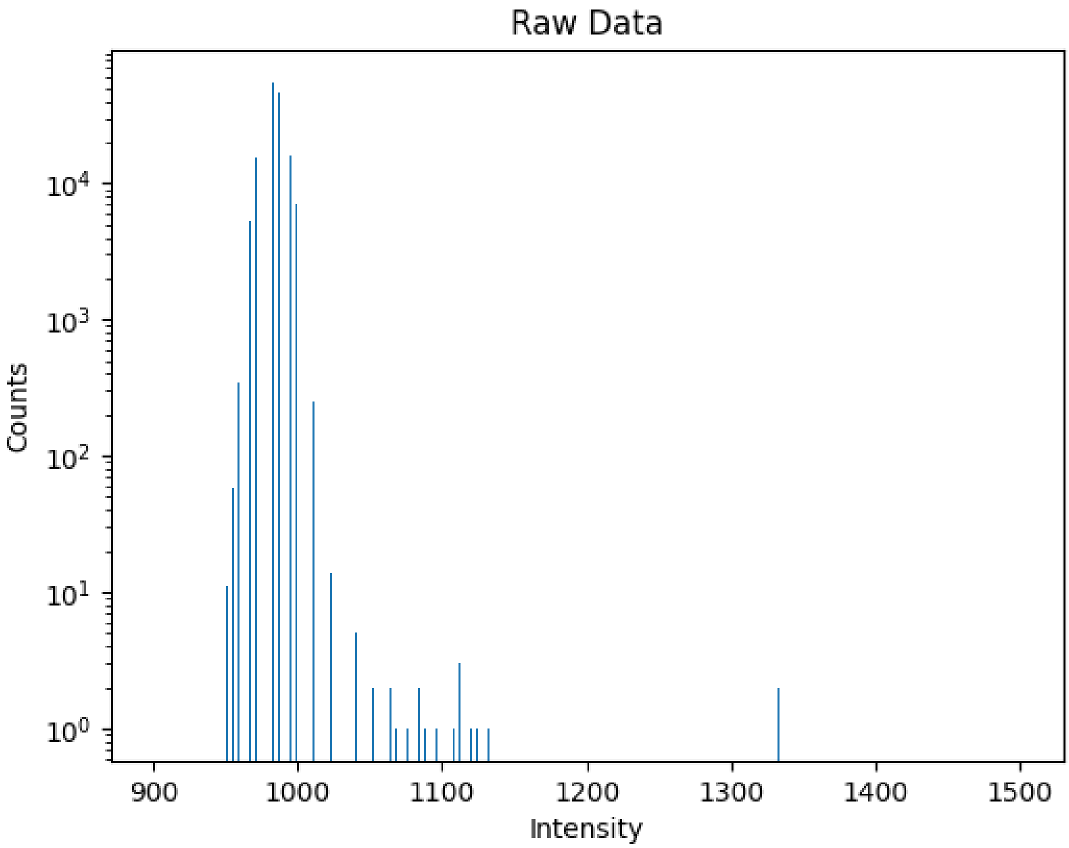
A bias frame is an exopsure of 0 seconds, with the camera shutter closed, at the temperature of observation. They are used to determine the bias level of every pixel and weed out faulty ones (e.g. hot pixels). The below image is a bias frame at the temperature of observation, -5 Celsius.

[1]: <matplotlib.colorbar.Colorbar at 0x116e224f0>

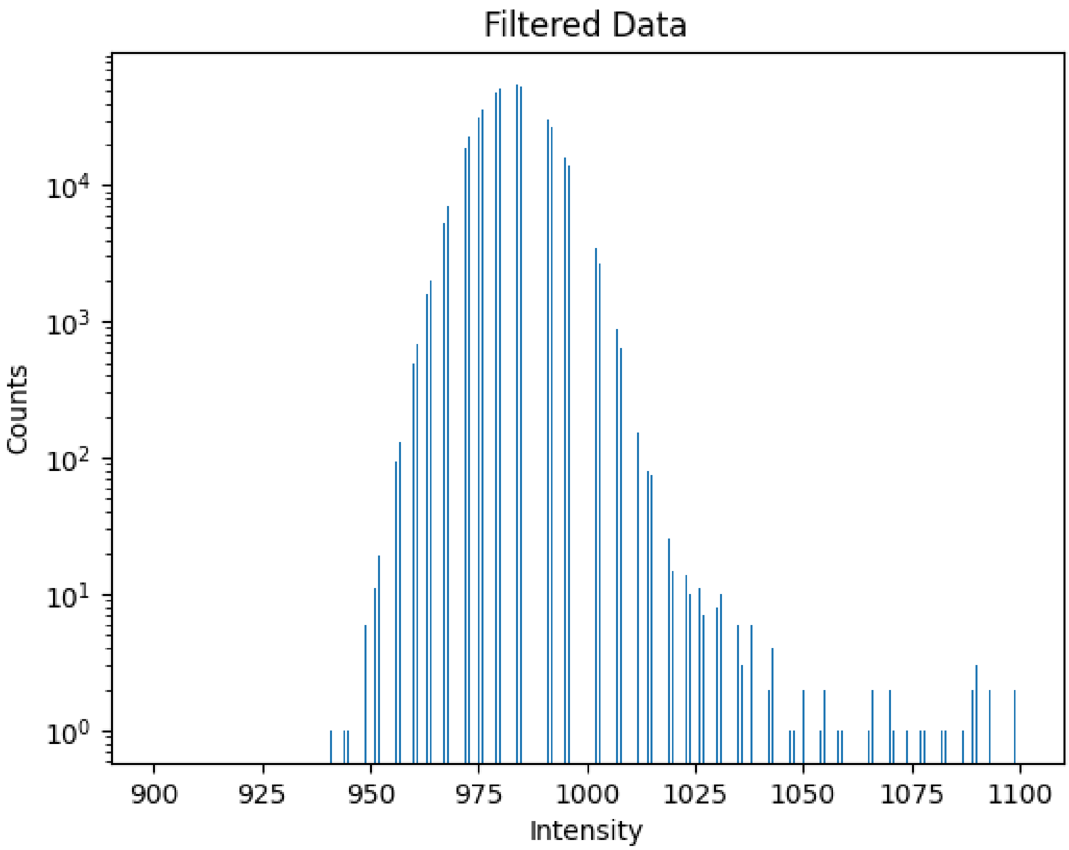


Astropy's unique FITs file handling functions make it quick and easy to plot the count levels of the intensities of pixels.

```
[112]: Text(0.5, 1.0, 'Raw Data')
```



It can be seen in the graph above that there are some pixels with a large dark current, causing them to become saturated. To filter the data, an intensity threshold is applied to the original data. Essentially, any pixels with a count greater than the threshold (1250 in this case) will be discarded from all of the data.




```
number cut from original data: 9
fraction cut from original data: 0.000858306884765625
```

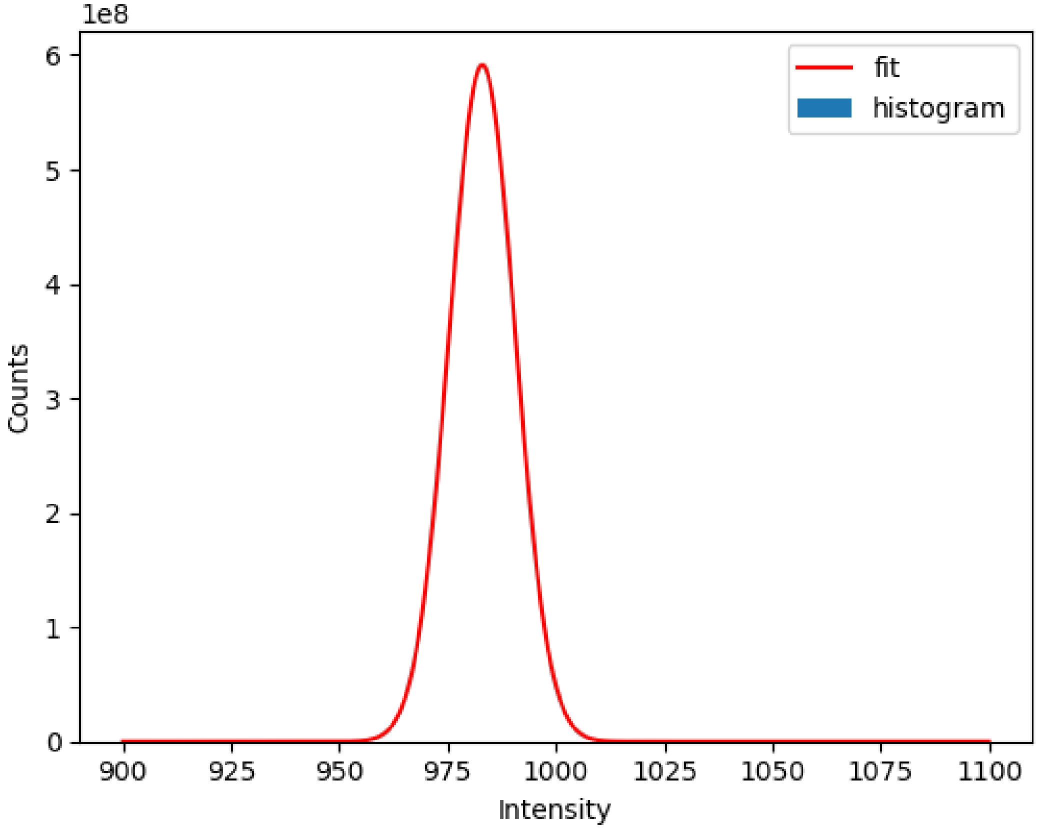
The graph of Filtered Data provides a more even distribution of counts, as there are no more erronious outliers. By subtracting the number of counts of the filtered data from the number of counts of the raw data, it is determined that there were 9 hot pixels (about 0.000858% of pixels).

Using Numpy's mean and standard deviation functions, the following were calculated:

```
Mean: 983.3301152906777
Stdev: 7.936998018975151
```

Using the calculated mean and standard deviation values, Astropy was once again utalized to plot an overlaying Gaussian. Astropy contains a variety of models and fitting algorithms, of which Gaussian1D and LevMarLSQFitter are used here to produce the desired fit.

```
[109]: <matplotlib.legend.Legend at 0x36a7f9220>
```



The gain, number of electrons, and number of counts are related by:

$$gain = \frac{N_{electrons}}{N_{counts}}$$

Using the spec sheet for the STL1001E, the actual gain is 2 and the number of electrons is 14.8e RMS. With a read noise equal to the standard deviation of counts, 7.94, the number of electrons was found to be:

$$2 \times 7.94 = 15.88e^{-}$$

While this is approximately one electron higher than the spec sheet, it is also relatively accurate.