Afrah Ali

Physics 139

Lab 6

```
pip install astropy
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-whe
Requirement already satisfied: astropy in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/di
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: pyerfa>=1.7.3 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packag
```

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

```
from astropy.modeling import models, fitting
```

PART ONE

```
# Combining flat images into one master array
filestart = "s03"
flat_holder = np.zeros((2048,2048,11))

for i in list(range(10,20)):
  file = filestart + str(i) + ".fits"
  flat_holder[:,:,(i-10)] = fits.open(file)[0].data[:,:]
```

```
# Combining flat subtracted dark images into one master array
dark1 = fits.open("s0347.fits")[0].data[:,:]
dark2 = fits.open("s0348.fits")[0].data[:,:]
dark3 = fits.open("s0349.fits")[0].data[:,:]
dark4 = fits.open("s0350.fits")[0].data[:,:]
dark5 = fits.open("s0351.fits")[0].data[:,:]
dark6 = fits.open("s0352.fits")[0].data[:,:]
dark7 = fits.open("s0353.fits")[0].data[:,:]
dark8 = fits.open("s0354.fits")[0].data[:,:]
dark9 = fits.open("s0355.fits")[0].data[:,:]
dark10 = fits.open("s0356.fits")[0].data[:,:]
dark11 = fits.open("s0357.fits")[0].data[:,:]
```

```
dark12 = fits.open("s0358.fits")[0].data[:,:]
dark13 = fits.open("s0359.fits")[0].data[:,:]
dark14 = fits.open("s0360.fits")[0].data[:,:]
dark15 = fits.open("s0796.fits")[0].data[:,:]
dark16 = fits.open("s0797.fits")[0].data[:,:]
dark17 = fits.open("s0798.fits")[0].data[:,:]
dark18 = fits.open("s0799.fits")[0].data[:,:]
dark19 = fits.open("s0800.fits")[0].data[:,:]
dark20 = fits.open("s0801.fits")[0].data[:,:]
dark21 = fits.open("s0802.fits")[0].data[:,:]
```

```
dark1.shape
```

```
(2048, 2048)
```

```
dark_list = ([dark1,dark2,dark3,dark4,dark5,dark6,dark7,dark8,dark9,dark10,dark11,d
med = np.median(dark_list)
print(med)
```

```
3.375
```

```
dark_holder = np.zeros((2048,2048,11))
for i in list(range(0,11)):
  arr = flat_holder[:,:,i] - 3.375
  dark_holder[:,:,(i-0)] = arr
```

```
normalized_flat = dark_holder/np.median(dark_holder)
```

```
# Using one of the individual dark-subtracted, flat-field corrected images of
# the star, measure the FWHM of the star in both the x and y direction.
```

```
## The image produced in the last part of lab 5 (if you even produced one)
## is incorrect. You can try using it since the general shape of the image is
## correct; it just doesn't have a discernable star in it. If not, just use one
## of the raw science frames without subtracting the dark from it and dividing
## it by the normalized flat image. The general process for finding the image
## that you need is (raw science image - dark image)/normalized flat image, but
## instead of using this process, you can just use the raw science image to
## start off the lab.
```

```
# Subtracting the dark and normalized flat from raw target image
```

```
rawimg0 = fits.open("s0576.fits")
```

```
rawimg_subset = rawimg0[0:1200, 0:1400]
```

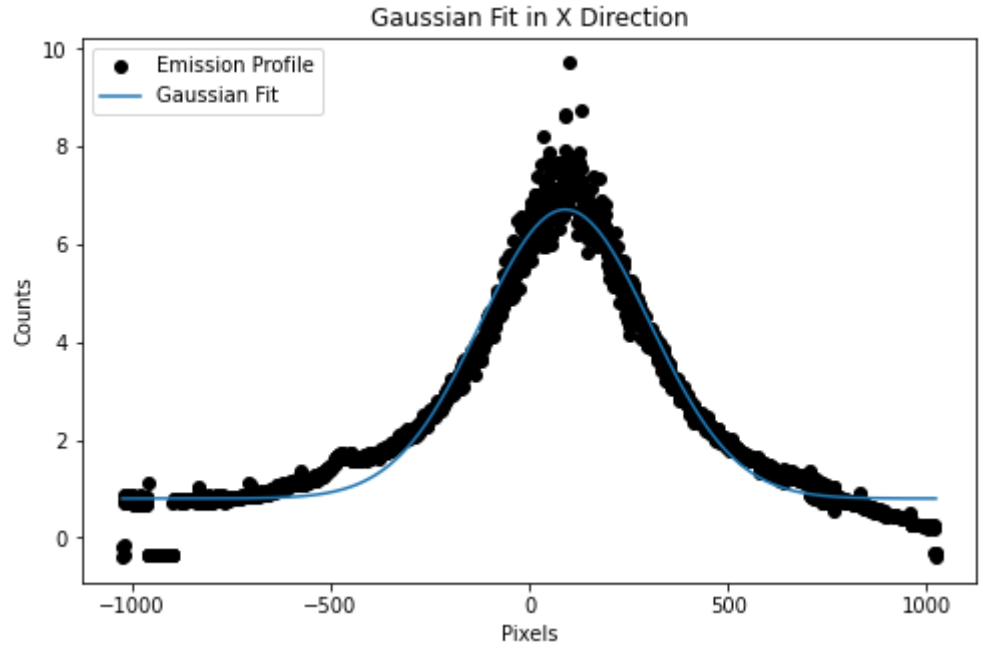
```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-15-d45fc96003b9> in <module>
----> 1 rawimg_subset = rawimg0[0:1200, 0:1400]

                                 2 frames
/usr/local/lib/python3.7/dist-packages/astropy/io/fits/hdu/hdulist.py in
index_of(self, key)
    740                 '{} indices must be integers, extension names as
strings, '
    741                 'or (extname, version) tuples; got {}'
--> 742                 ''.format(self.__class__.__name__, _key))
    743
    744         _key = (_key.strip()).upper()

KeyError: 'HDUList indices must be integers, extension names as strings, or
(extname, version) tuples; got slice(0, 1200, None)'
```

```
plt.figure()
plt.imshow(rawimg)
```

```
# Finding FWHM in x direction
```

```
# Initializing X and Y values from the image
x = np.median(normalized_flat[:,:,0], axis = 0)
y = np.median(normalized_flat[:,:,0], axis=1)

np.random.seed(0)
xdata = np.arange(-1024,1024)
ydata = x # Focusing on x direction so set y to x to standardize

# Initalizing Gaussian Fit
g_init = models.Gaussian1D(amplitude = 1., mean = 0, stddev = 1.) + models.Const1D(
fit_g = fitting.LevMarLSQFitter()
g = fit_g(g_init, xdata, ydata)

xstd = np.std(ydata)
xamp = np.max(ydata)
xmean = np.mean(ydata)
print(xstd, xamp, xmean)

print(g)

plt.figure(figsize=(8,5))
plt.plot(xdata, ydata, "ko", label="Emission Profile")
plt.plot(xdata, g(xdata), label="Gaussian Fit")
plt.title("Gaussian Fit in X Direction")
plt.xlabel("Pixels")
plt.ylabel("Counts")
plt.legend(loc=2)
```

```
2.058170681065363 9.713375796178344 2.3410939988057327
Model: CompoundModel
Inputs: ('x',)
Outputs: ('y',)
Model set size: 1
Expression: [0] + [1]
Components:
    [0]: <Gaussian1D(amplitude=5.91397243, mean=90.49736243, stddev=213.25882492

    [1]: <Const1D(amplitude=0.79746256)>
Parameters:
       amplitude_0           mean_0            stddev_0         amplitude_1
    ------------------ ----------------- ----------------- ------------------
    5.9139724293440565 90.49736243203705 213.2588249165479 0.7974625600893013
<matplotlib.legend.Legend at 0x7fa5d00e7fd0>
```



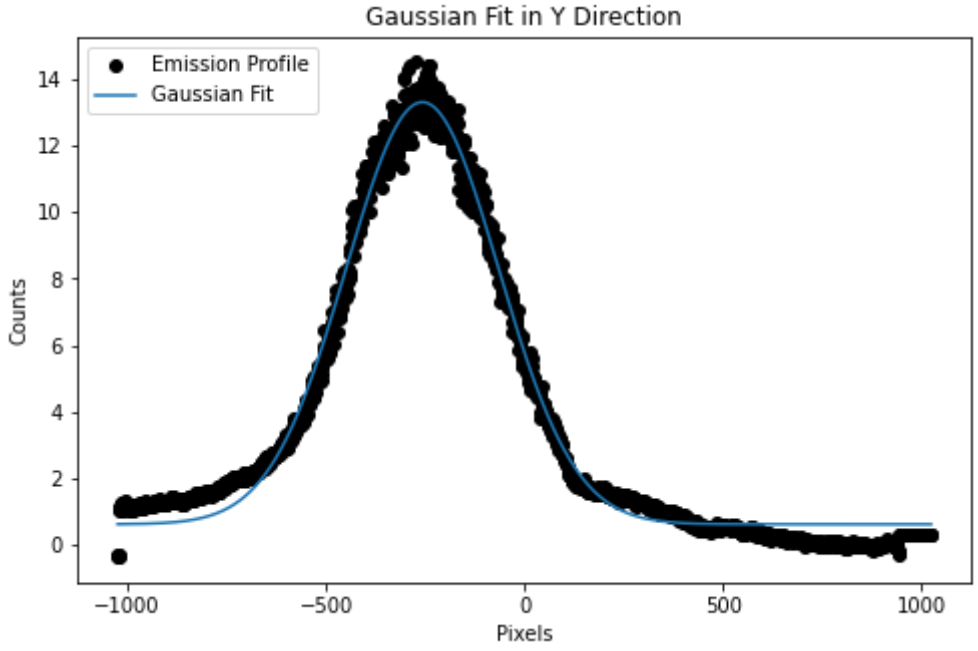

```
1
```

```
2.058170681065363 9.713375796178344 2.3410939988057327
```

```
# Initializing X and Y values from the image
x = np.median(normalized_flat[:,:,0], axis = 0)
y = np.median(normalized_flat[:,:,0], axis=1)

np.random.seed(0)
xdata = np.arange(-1024,1024)
ydata = y # Focusing on y direction

# Initalizing Gaussian Fit
g_init = models.Gaussian1D(amplitude = 1., mean = 0, stddev = 1.) + models.Const1D(
fit_g = fitting.LevMarLSQFitter()
g = fit_g(g_init, xdata, ydata)

```

```
xstd = np.std(ydata)
xamp = np.max(ydata)
xmean = np.mean(ydata)
print(xstd, xamp, xmean)

print(g)

plt.figure(figsize=(8,5))
plt.plot(xdata, ydata, "ko", label="Emission Profile")
plt.plot(xdata, g(xdata), label="Gaussian Fit")
plt.title("Gaussian Fit in Y Direction")
plt.xlabel("Pixels")
plt.ylabel("Counts")
plt.legend(loc=2)
```

```
4.250301118174297 14.512738853503185 3.590966485867834
Model: CompoundModel
Inputs: ('x',)
Outputs: ('y',)
Model set size: 1
Expression: [0] + [1]
Components:
    [0]: <Gaussian1D(amplitude=12.68967184, mean=-257.29416844, stddev=191.58301

    [1]: <Const1D(amplitude=0.61550808)>
Parameters:
       amplitude_0           mean_0            stddev_0         amplitude_1
    ------------------ ------------------ ------------------ ------------------
    12.689671838192545 -257.2941684445349 191.58301459729475 0.6155080787851588
<matplotlib.legend.Legend at 0x7fa5d0267050>
```



Colab paid products - Cancel contracts here

✓ 0s completed at 2:16 PM