Coursework_A_v2

October 27, 2023

1 Coursework A - Observation Planning, Data Analysis & Photometry

This coursework should take you approximately 1-2h to complete (after working through the chapter 1 & 2 notebooks) and is worth 20% of your grade for this unit. Ensure that any calculations run correctly, i.e. make sure there are no typos in any Python code you include. Check this by running all cells before submission. Watch the short video that explains how to use this notebook.

For answering the questions, you will be provided with images just like the ones in Chapter2_AperturePhotometry. These are simulated specifically for you given your Student ID. You can use the functions .get_data, .plot_x, .plot_y and .show_ima to interrogate the data. The functions that allow you to check your answer are however disabled.

Answer all questions in the cells provided, you can also add additional cells.

You can add additional cells to import any functions you wrote for other notebooks.

Add your Candidate Number below:

IMPORTANT: DO NOT CHANGE WHILE WORKING ON YOUR COURSEWORK, IT IS USED TO CREATE A RANDOM DATASET BASED ON YOUR ID!

[106]:
```
student_id = 25381
```

[107]:
```
%load_ext autoreload
from image_simulator import ImageSimulator
import LightCurveSimulator
import numpy as np
%autoreload
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

[108]:
```
#Photutils is the model we will use for aperture photometry.
#The documentation is available here: https://photutils.readthedocs.io/en/
↪stable/index.html
from photutils.aperture import aperture_photometry
from photutils.aperture import CircularAperture
import photutils
```

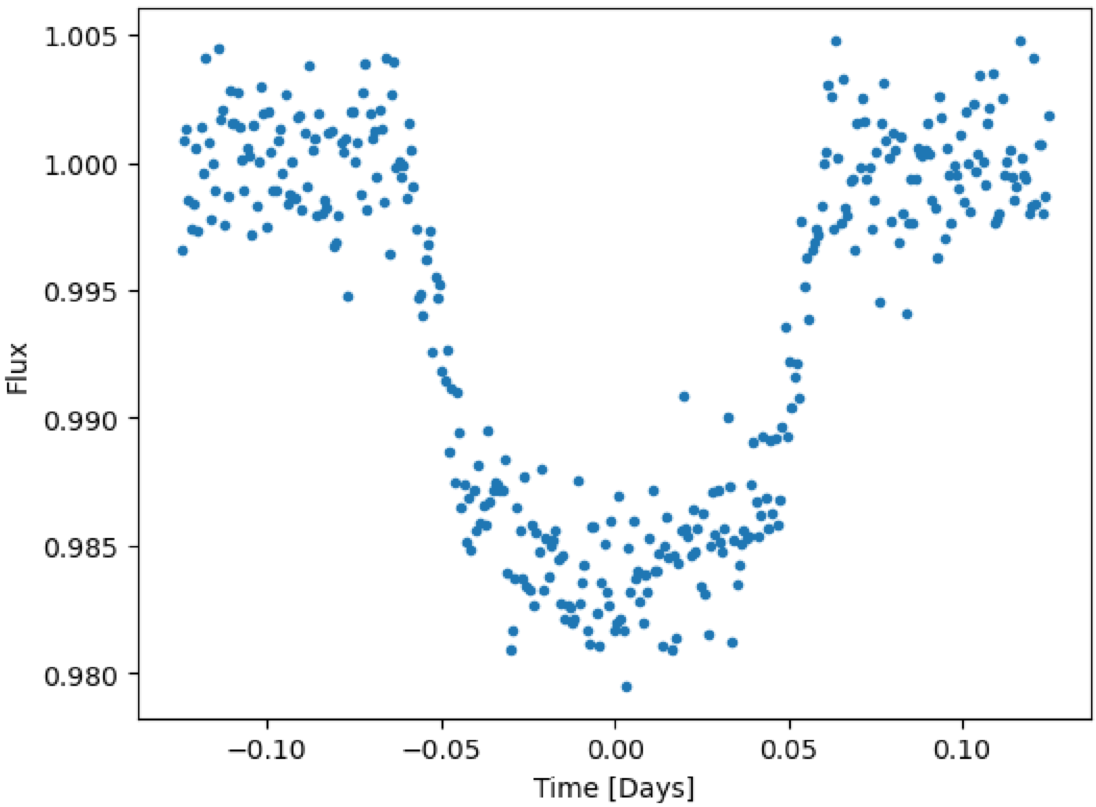
2 Section 1 : Observations

2.1 Question 1

Given a lightcvurve with signal-to-noise 500 and a host star that has the same radius as the sun, estimate the size of the smallest detectable planet? Describe how you derived your answer. [2 marks]

2.2 Answer 1

[109]:
```
import LightCurveSimulator
lc = LightCurveSimulator.ShortTransit()
lc.add_noise(sn=500)
lc.plotlc()
```



With a signal to noise of 500 the smallest possible dip that we can still reasonably detect corresponds to a change in flux of about (1 - 0.994) 0.006. As this is as far as the noise extends to and where we should reasonably be able to detect a dip despite the noise. The above light curve is normalised and so we can estimate the fractional change in flux as follows.

$\frac{\Delta F}{F} = \frac{R^2_{planet}}{R^2_{star}}$
$\Delta F = F_{star} - F_{final} = 1 - 0.994 = 0.006$

$R_{star} = 6.96 \times 10^8 m$

Carrying out the calculation we get:

$R_{planet} = 5.4 \times 10^7 m$

This is the same order of magnitude as a Jupiter sized planet.

2.3 Question 2

What changes would need to be made to the observations to be able to observe a planet that has an semi major axis that is 10 times smaller? Qualitatively describe your answer. Assume the stellar mass does not change and all planets transit the star centrally. [4 marks]

2.4 Answer 2

The transit probability of a star is given by:
$\frac{R_{star}+R_{planet}}{a} \approx \frac{R_{star}}{a}$ Since the radius of the planet can be assumed to be much smaller than the radius of the star. This equation tells us that if the semi-major axis gets 10 times smaller, then in theory the probability of detecting the planet will be 10 times greater. However if the planet is small and it gets too close to the star, then you run the risk of missing the planet if the planet is engulfed by the star light, making the dip in flux hard to notice and measure. This means that the telescope needs to be able to resolve the small dip in flux so a telescope would need to have a high sensitivity and resolution. Additionally, the aperture of the telescope could be made smaller so that less light reaches the optics so theimage is not over exposed. If the semi-major axis is 10 times smaller then the transit time is likely to also be smaller, therefore observations should be made frequently in order to catch the transit. The observations should be taken high up and on a clear day, in order to reduce light attenuation.

2.5 Question 3

What are the potential issues when using ground based telescopes for detecting transiting exoplanets?

2.6 Answer 3

There are a few issues when using ground based telescopes. - The light from the star has to pass through the atmosphere of the earth which will attenuate the light giving a weaker signal and a lower signal to noise ratio. - Observation times will be limited to the night time as the sun will overexpose any images taken. This means that there will be periodic gaps in the data, and it will be quite hard to analyse the data if the period is of the order of a few days. - Weather conditions and cloud cover can affect the measurements taken - an additional loss of data - Light pollution on earth may pollute the data

2.7 Question 4

You are planning to observe the radial velocity signature of an exoplanet. Which characteristics of the spectrograph used will be important to ensure a suitable dataset?

2.8 Answer 4

The radial velocity method uses the doppler shift of the spectral lines emitted from the star as the planet and the star both rotate around a common centre of mass. This means that the spectral resolution of the spectrograph must be good in order to measure the tiny shifts in the absorption lines. The sensitivity of the spectrograph should be high. The spectrogram should be stable and not suffer from any vibrations that could affect the accuracy of the measurements. The spectrogram should contain a wide enough range of wavelengths to observe the absorption lines.

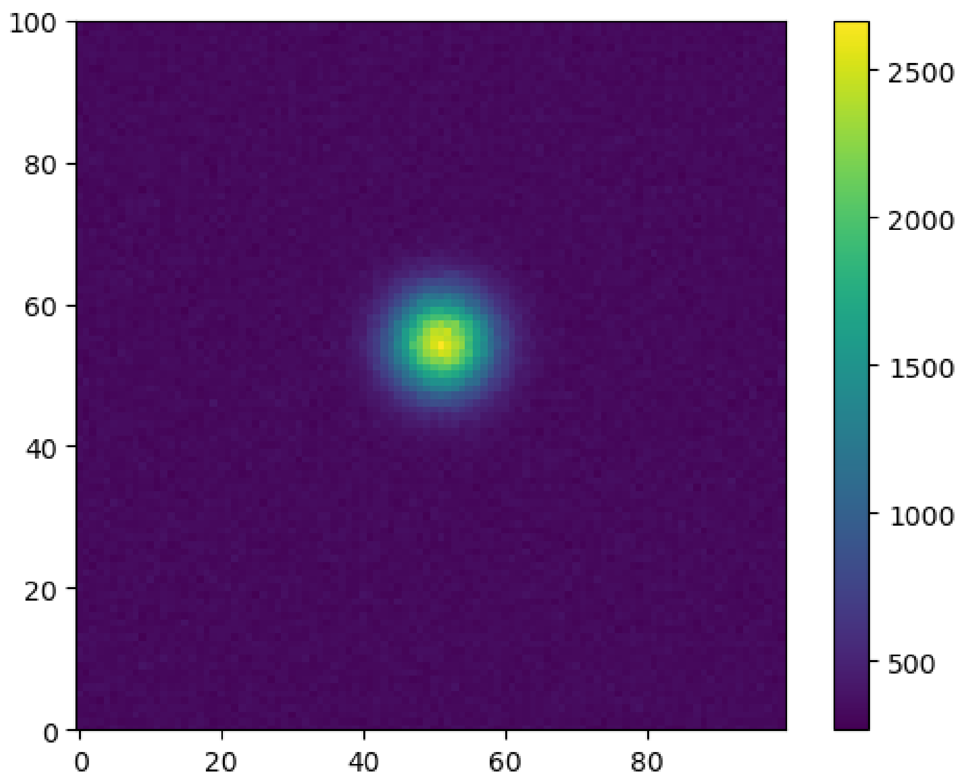
3 Section 2 : Data Analysis & Photometry

3.1 Question 1

What is the flux of the following star? Give your uncertainty. [5 marks]

[110]:
```
Q1 = ImageSimulator(student_id=student_id, assessmentQ=1, size=(100, 100))
```

```
RON is 3.47
```



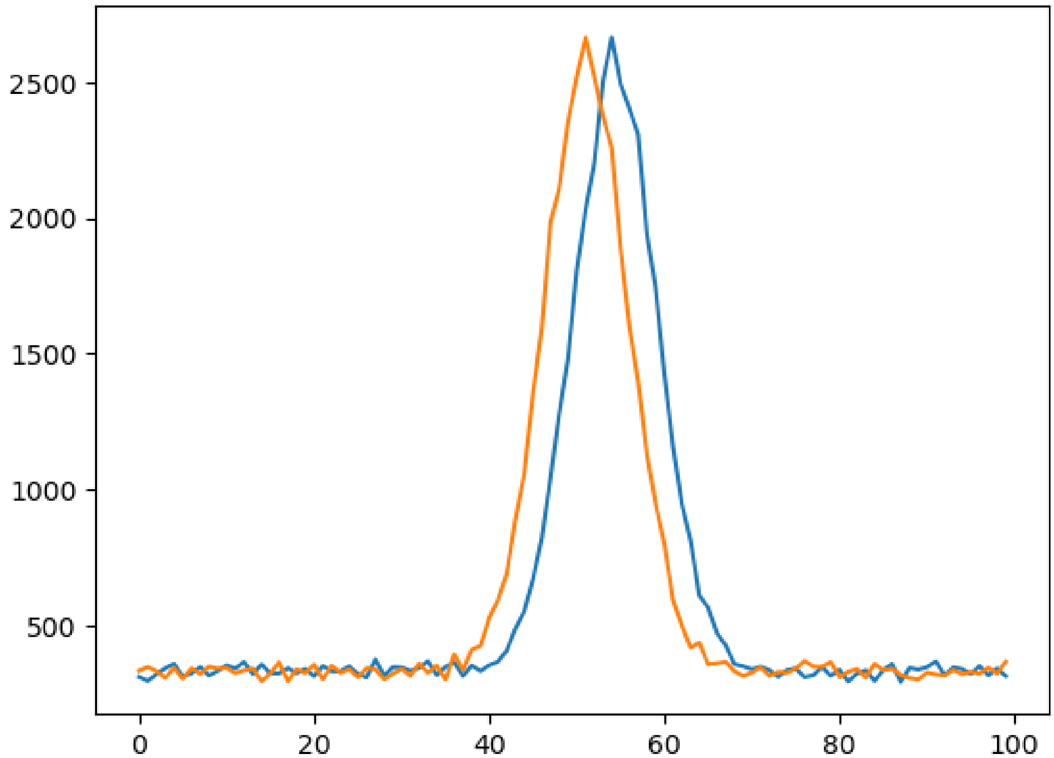
[111]:
```
print(Q1.plot_x)
```

```
<bound method ImageSimulatorBase.plot_x of <image_simulator.ImageSimulator
object at 0x7fa1ce740f10>>
```

3.2 Answer 1:

In order to get the flux of the star we first need to identify the location of the star in the image and how much area the star flux is incident on the image in order to add up the components of light intensity in that area.

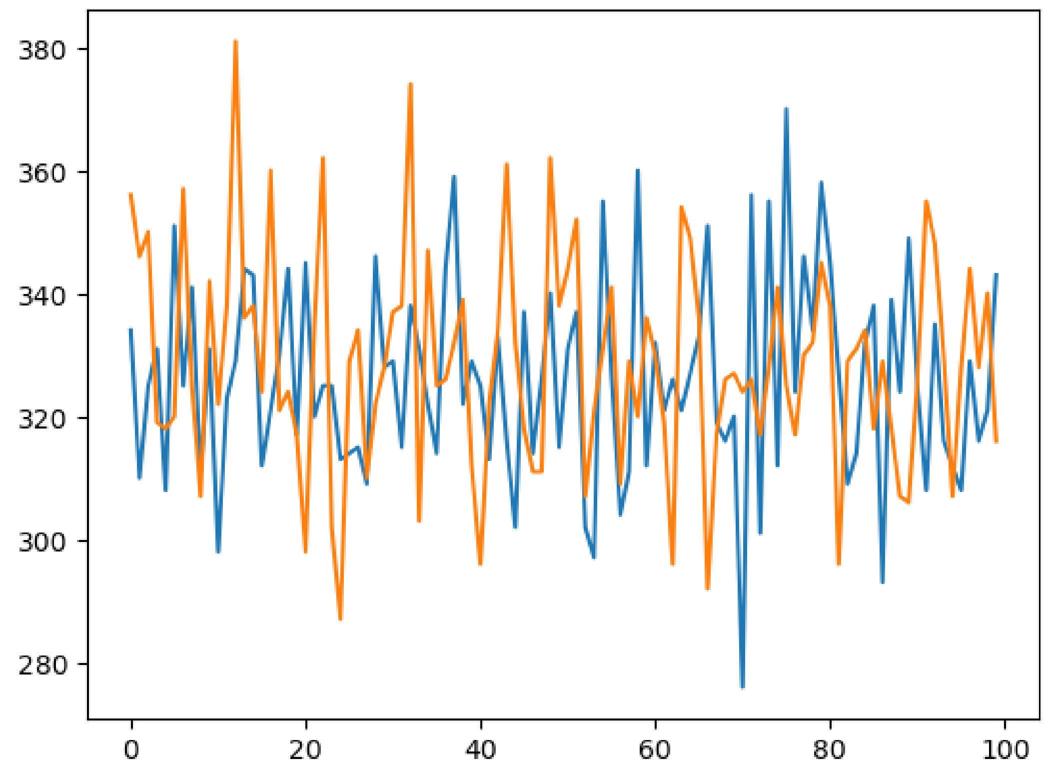
[112]:
```
# code to identify the location of the centre of the star
# ie. The area with the highest intensity
Q1.plot_x(51)
Q1.plot_y(54)
```



With the code above, we can slice the data at different y and x values and we find that there is a maximum peak in intensity at x = 51 and y = 54. We can also estimate the diameter of the source using the slices, the aperture radius should encompass the whole light source. The diameter is around 25 pixels long, so the radius can be estimated to be 13.

We can also use the slicing tool to check if there are any other unexpected sources of light in the image:

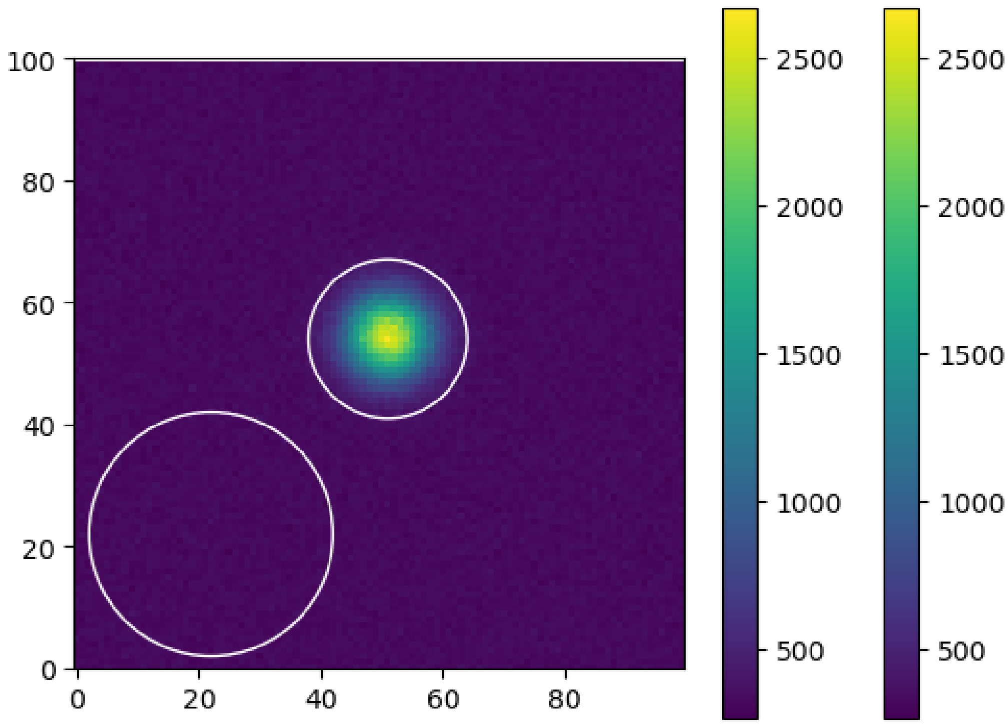
[113]:
```
Q1.plot_x(20)
Q1.plot_y(10)
```



Now we can estimate a suitable aperture size for the image.

[114]:
```
def ShowAperture(xpos, ypos, rad, image):
    aperture1 = photutils.aperture.CircularAperture((xpos, ypos), rad)
    phot = aperture_photometry(image.get_data(), aperture1)
    return image.show_ima(), aperture1.plot(ec='white'), phot
ShowAperture(51, 54, 13, Q1)
ShowAperture(22, 22, 20, Q1)
```

[114]:
```
(None,
 (<matplotlib.patches.Circle at 0x7fa1ce3491c0>,),
 <QTable length=1>
   id  xcenter ycenter    aperture_sum
         pix     pix
 int64 float64 float64      float64
 ----- ------- ------- -----------------
     1    22.0    22.0 411526.9326761081)
```



Now we can use the below function to calculate the total flux received from the source.

[115]:
```
def ap_phot(image, x, y, radius, skyx, skyy, skyinner, skyouter, ron):
    #Defining the object aperture
    ap = photutils.aperture.CircularAperture((x, y), radius)
    #measuring the flux in the object aperture
    fl_ap = aperture_photometry(image, ap)
    fl_ap = fl_ap['aperture_sum'][0]
    #Defining the sky perture, if syinner=0, create a circular ap, otherwise,␣
↪create an annulus
    if skyinner == 0:
        ap_sky = photutils.aperture.CircularAperture((skyx, skyy), skyouter)
    else:
        ap_sky = photutils.CircularAnnulus((skyx, skyy), skyinner, skyouter)
    #measure the flux in the sky
    fl_sky = aperture_photometry(image, ap_sky)
    fl_sky = fl_sky['aperture_sum'][0]
    #the areas of each of the apertures are
    obj_area = ap.area
    sky_area = ap_sky.area

    #Write down the flux in the oject given
    #fl_ap: the flux in the aperture
    #fl_sky: the flux in the sky aperture
```

```
    #obj_area: area of object aperture
    #sky_area: area of sky aperture

    fl_object = (fl_ap)-((fl_sky)*(obj_area/sky_area))

    #Errors

    #the shot noise in the main aperture:
    shot_ap = np.sqrt(fl_ap)
    #uncertainty in the sky level
    unc_sky = np.sqrt(fl_sky/sky_area)/np.sqrt(sky_area)
    # This is so  that we get the sky error per pixel (first factor)

    #add the other sources of error
    err_obj = np.sqrt(shot_ap**2 +  obj_area * unc_sky**2 + obj_area * ron**2)

    return fl_object, err_obj
```

[116]:
```
#ap_phot(image, x, y, radius, skyx, skyy, skyinner, skyouter, ron)
flux = ap_phot(Q1.get_data(), 51, 54, 13, 22, 22, 0, 20, 3.47)[0]
flux_err = ap_phot(Q1.get_data(), 51, 54, 13, 22, 22, 0, 20, 3.47)[1]

print('Total Flux: ', flux, '+/- ', flux_err)
```

```
Total Flux:  318556.86621758784 +/-  706.3697485239446
```

3.3 Question 2

Explain your reasoning for choosing your aperture size and sky region in Question 1.

3.4 Answer 2:

As described in question 1, I chose the shot aperture so that it should include almost all the light emitted by the source. I did this by estimating the radius from the slices and noting when the light from the source reached the same level as the background sky level. The aperture was not made any bigger than it needed to be though as this would have increased the amount of sky noise in the shot aperture and also increased the error in the measurement.

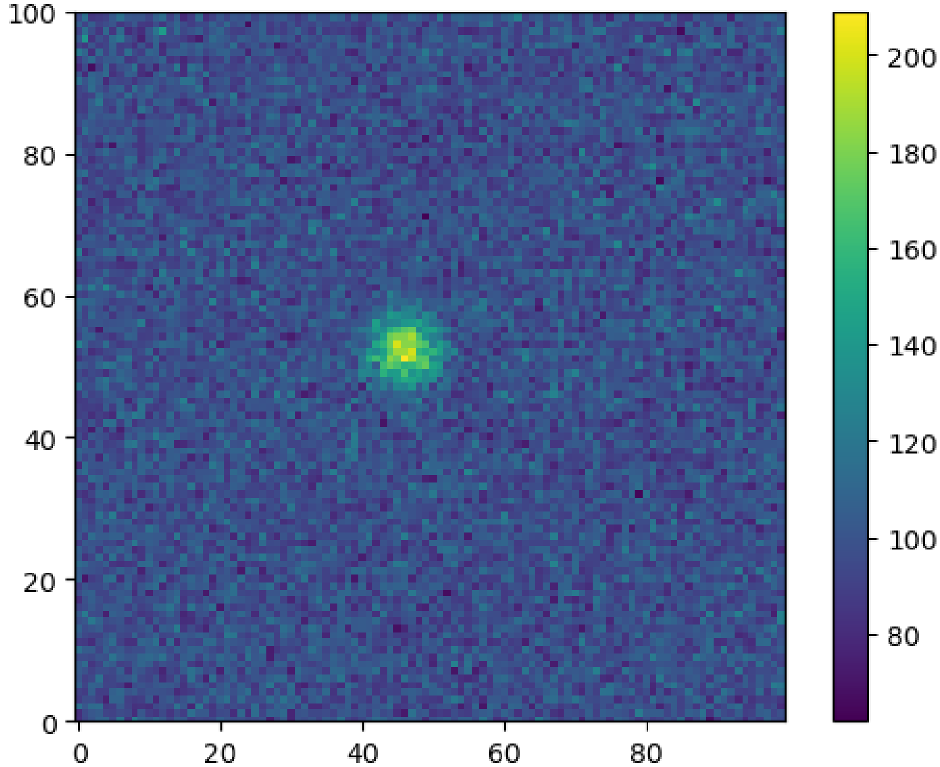
The sky aperture is located in an area away from the source and I checked that there was no unexpected peaks in this area indicating an extra source. The sky aperture was made to be big in order to get an accurate reading of the sky noise and reduce the sky level error.

3.5 Question 3

Below, you see an image displaying a star. Calculate separately each of the individual factors that contribute to the uncertainty in the flux emission from this star, and thus comment on what the dominant source of uncertainty is.

[117]:
```
Q3 = ImageSimulator(student_id=student_id, assessmentQ=2, size=(100, 100))
```

```
RON is 1.43
```



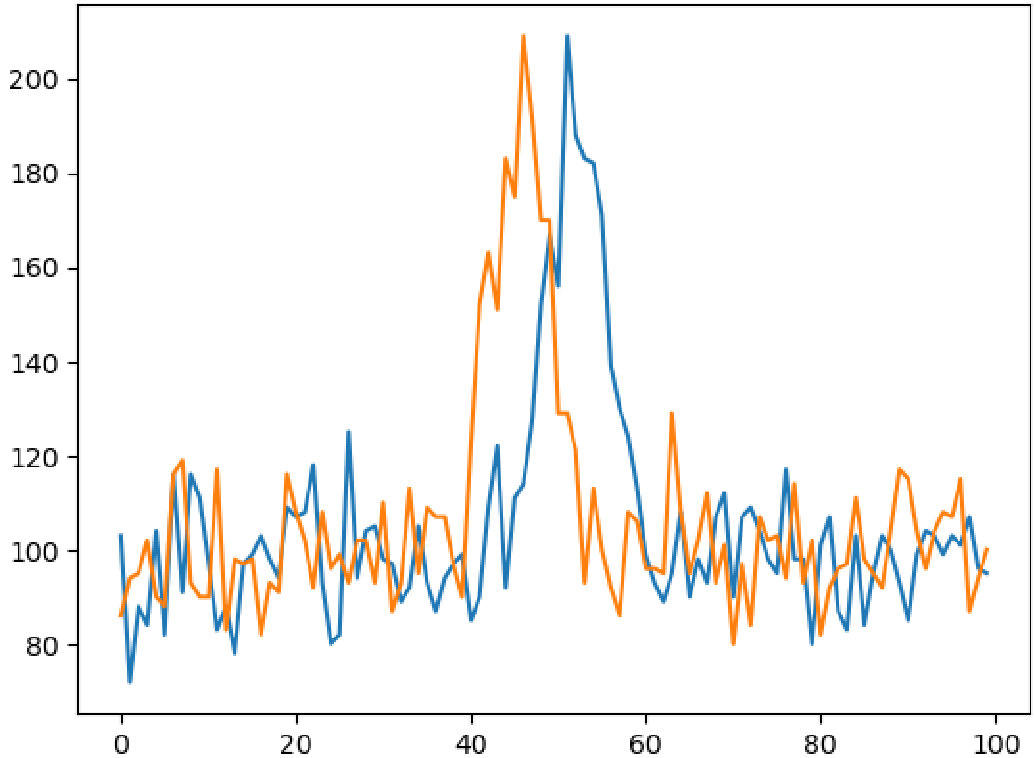
[ ]:

```

```

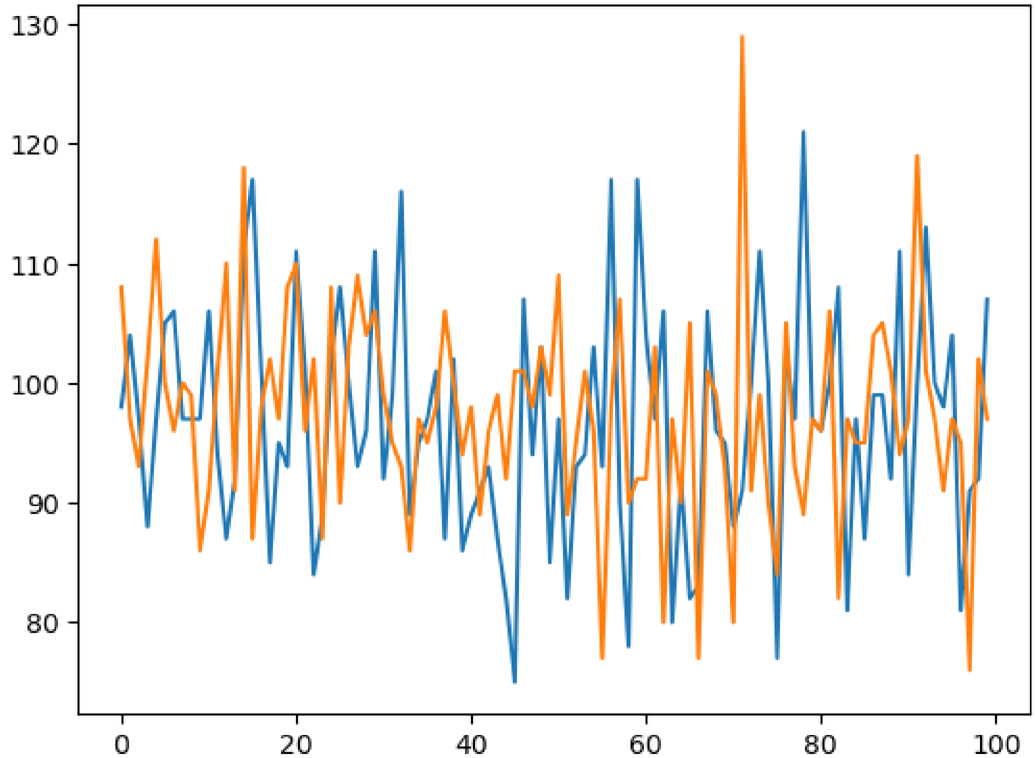
3.6 Answer 3:

[118]:

```
Q3.plot_x(46)
Q3.plot_y(51)
#y = 51
#x = 46

#ap_phot(image, x, y, radius, skyx, skyy, skyinner, skyouter, ron)
```

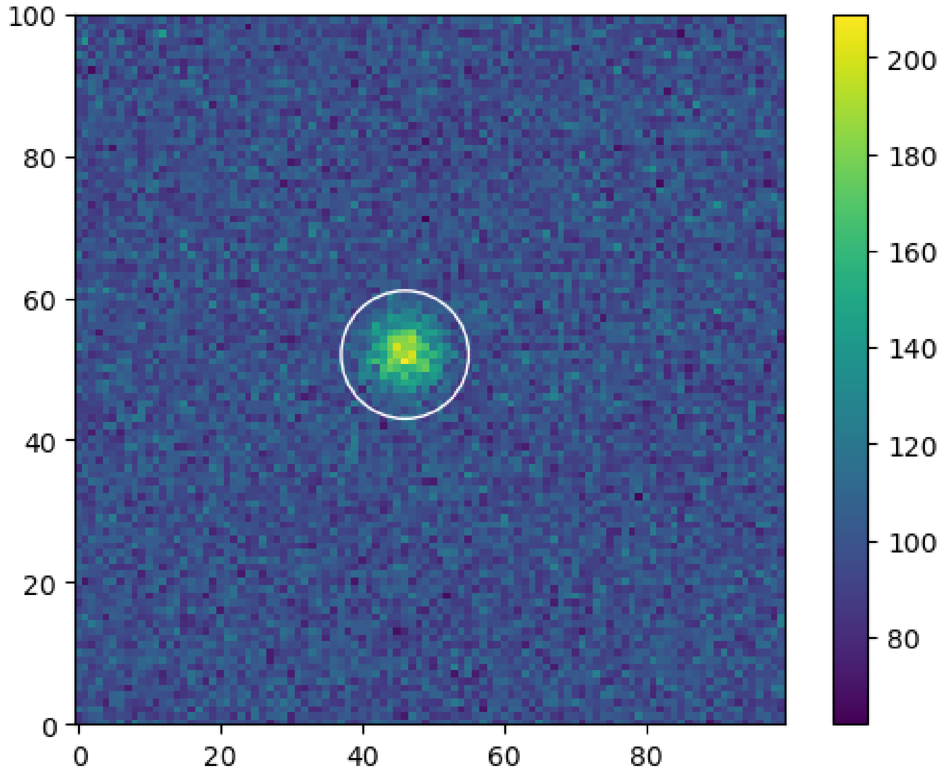


[119]:
```
Q3.plot_x(80)
Q3.plot_y(80)
```



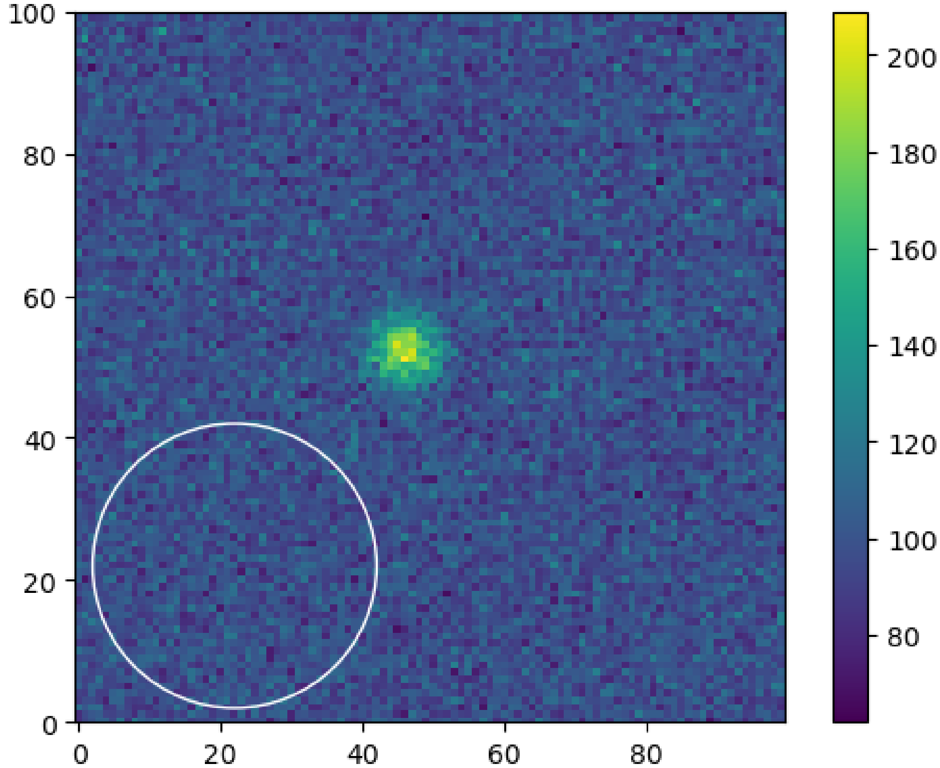

```
[120]: #ShowAperture(xpos, ypos, rad, image)
y = 52
x = 46
#star looks more centred at y = 52
rad = 9
ShowAperture(x, y, rad, Q3)
```

```
[120]: (None,
 (<matplotlib.patches.Circle at 0x7fa1ce10fbb0>,),
 <QTable length=1>
   id  xcenter ycenter    aperture_sum
         pix     pix
 int64 float64 float64      float64
 ----- ------- ------- -----------------
     1    46.0    52.0 31288.80091900924)
```



[121]:
```
ShowAperture(22, 22, 20, Q3)
```

[121]:
```
(None,
 (<matplotlib.patches.Circle at 0x7fa1ce75c640>,),
 <QTable length=1>
   id  xcenter ycenter    aperture_sum
         pix     pix
 int64 float64 float64      float64
 ----- ------- ------- ------------------
     1    22.0    22.0 123298.45019609356)
```



Calculating the different sources of uncertainty.

[122]:
```
def errors(image, x, y, radius, skyx, skyy, skyinner, skyouter, ron):
    #Defining the object aperture
    ap = photutils.aperture.CircularAperture((x, y), radius)
    #measuring the flux in the object aperture
    fl_ap = aperture_photometry(image, ap)
    fl_ap = fl_ap['aperture_sum'][0]
    #Defining the sky perture, if syinner=0, create a circular ap, otherwise,␣
 ↪create an annulus
    if skyinner == 0:
        ap_sky = photutils.aperture.CircularAperture((skyx, skyy), skyouter)
    else:
        ap_sky = photutils.CircularAnnulus((skyx, skyy), skyinner, skyouter)
    #measure the flux in the sky
    fl_sky = aperture_photometry(image, ap_sky)
    fl_sky = fl_sky['aperture_sum'][0]
    #the areas of each of the apertures are
    obj_area = ap.area
    sky_area = ap_sky.area

    #Write down the flux in the oject given
```

```python
    #fl_ap: the flux in the aperture
    #fl_sky: the flux in the sky aperture
    #obj_area: area of object aperture
    #sky_area: area of sky aperture

    fl_object = (fl_ap)-((fl_sky)*(obj_area/sky_area))

    #Errors

    #the shot noise in the main aperture:
    shot_err = np.sqrt(fl_ap)

    #uncertainty in the sky level
    unc_sky = np.sqrt(fl_sky/sky_area)/np.sqrt(sky_area)

    #err_obj = np.sqrt(shot_ap**2 +  obj_area * unc_sky**2 + obj_area * ron**2)

    return shot_err**2, obj_area * unc_sky**2, obj_area* ron**2

y = 52
x = 46
#star looks more centred at y = 52
rad = 9
skyx = 22
skyy = 22
skyouter = 20

poisson_noise = errors(Q3.get_data(), x, y, rad, skyx, skyy, 0, skyouter, 1.
 ↪43)[0]
sky_error = errors(Q3.get_data(), x, y, rad, skyx, skyy, 0, skyouter, 1.43)[1]
RON_err = errors(Q3.get_data(), x, y, rad, skyx, skyy, 0, skyouter, 1.43)[2]

#proportions of the contribution of each error is shown below
print("Poisson Noise Error: ", poisson_noise)
print("Sky Level Error: ", sky_error)
print("RON Error: ", RON_err)
```

```
Poisson Noise Error:  31288.80091900924
Sky Level Error:  19.868852297081638
RON Error:  520.3636682033871
```

Looking at the above analysis we can conclude that the dominant source of uncertainty comes from the poisson noise from the source and sky within the shot aperture. Additionally looking at the images and the slices we can see that the sky counts per pixel is centred around 100 and the maximum peak is around 200 (sky level is at 50% of the peak of the source which is a significant percentage). Therefore we can conclude that the measurement is sky noise dominated.

3.7 Question 4:

Below is a crowded field. If you perform aperture photometry on Star 1, is the flux of star 1 likely to be:

a) measured correctly

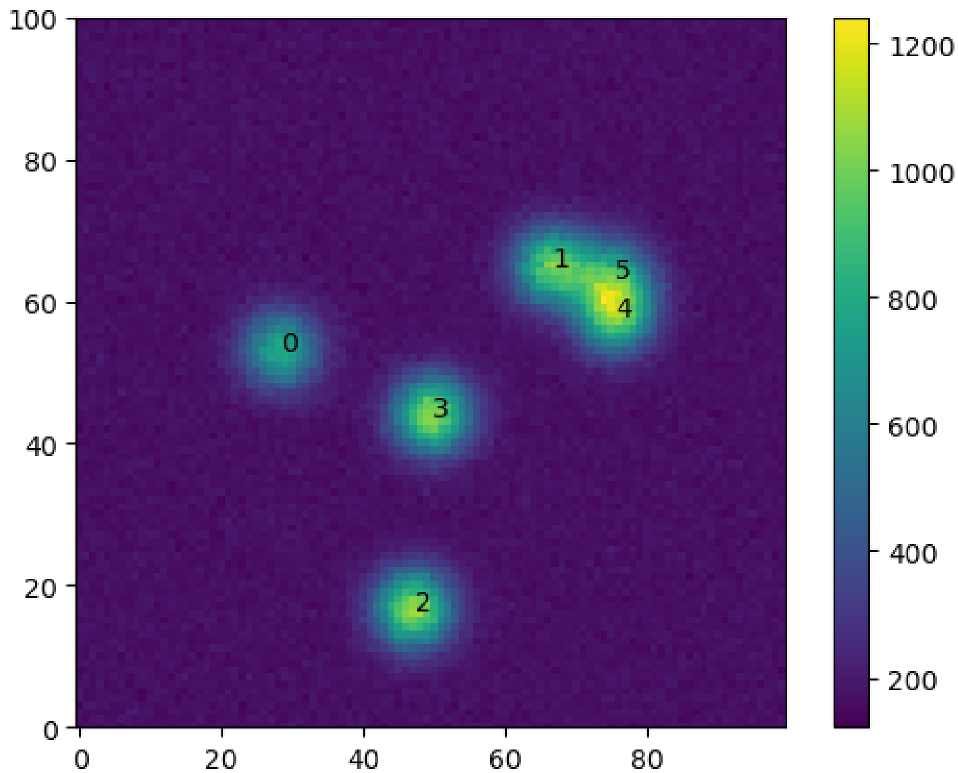
b) overestimated or

c) underestimated?

Explain your reasoning? You can use other stars in the field to support your argument.

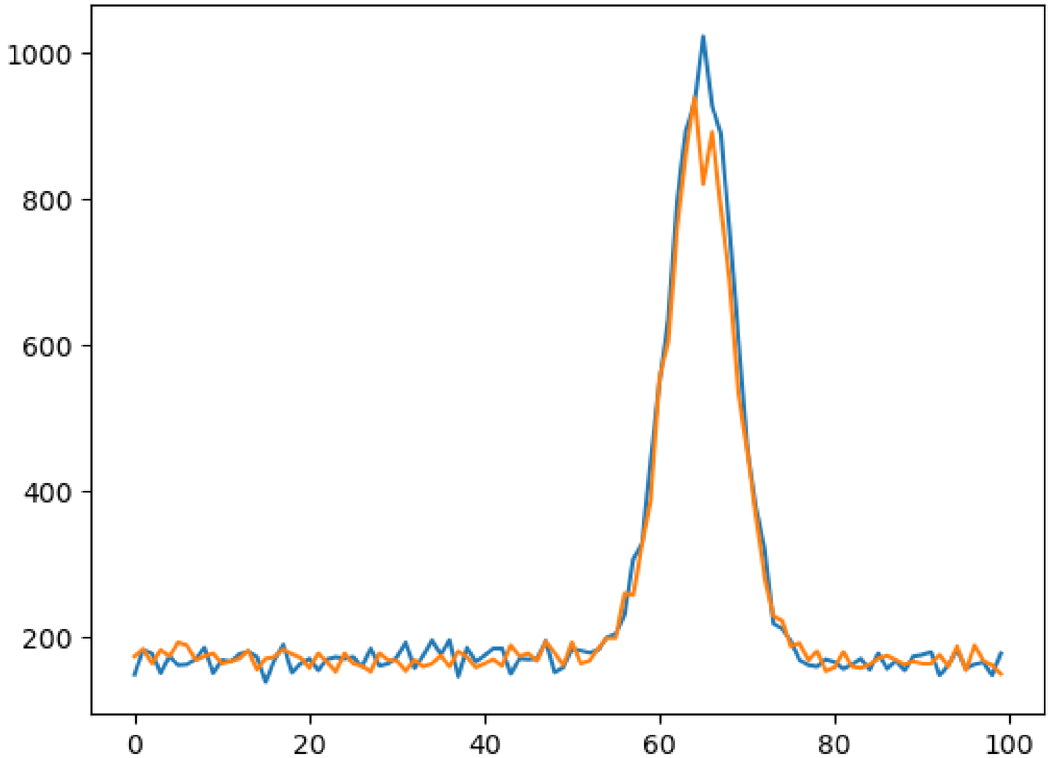
[123]:
```
Q4 = ImageSimulator(student_id=student_id, assessmentQ=4, size=(100, 100))
```

```
RON is 2.04
```

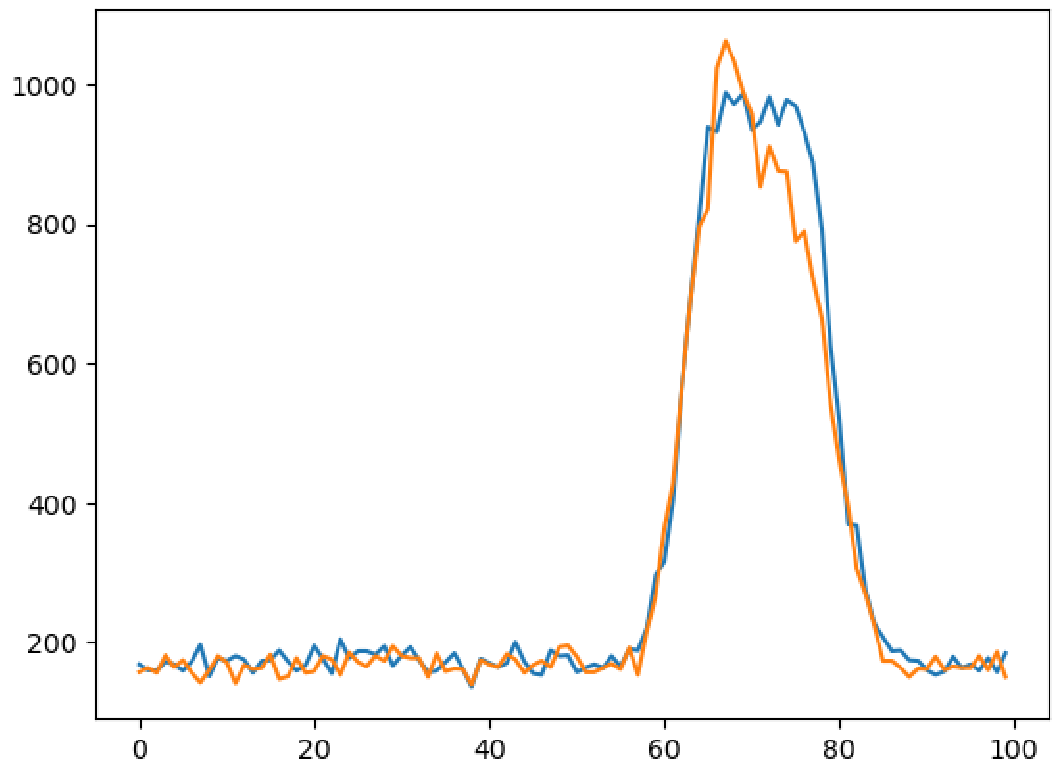


3.8 Answer 4:

[124]:
```
Q4.plot_x(66)
Q4.plot_x(65)
```

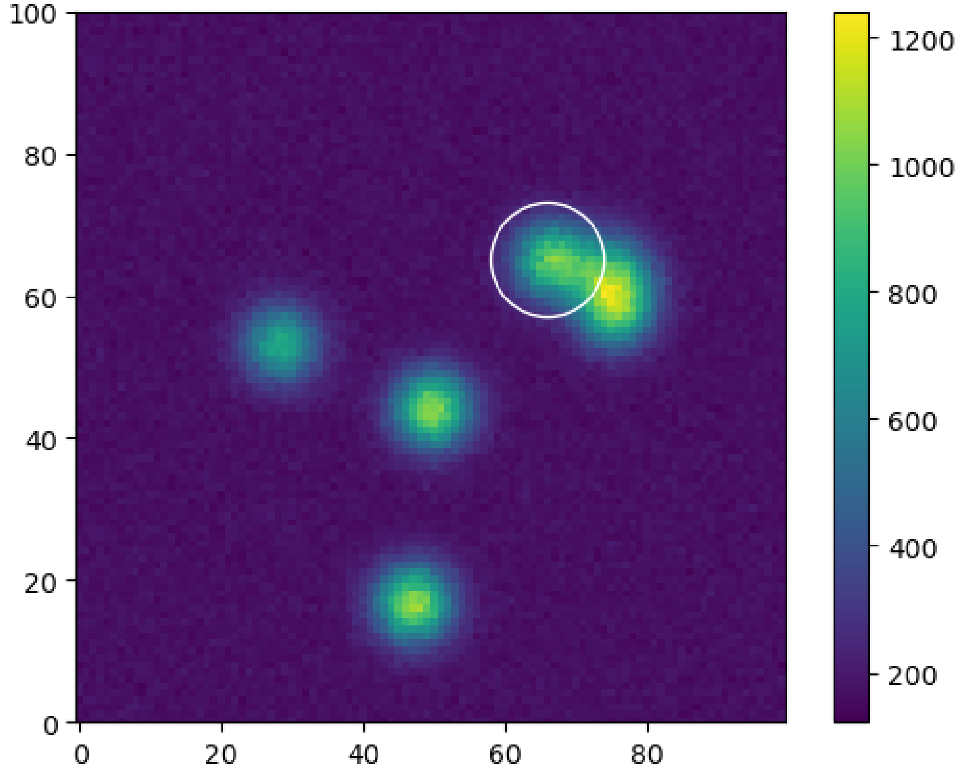


[125]:
```
Q4.plot_y(64)
Q4.plot_y(65)

#x = 66, y = 65
```



[126]:
```
ShowAperture(66, 65, 8, Q4)
```

[126]:
```
(None,
 (<matplotlib.patches.Circle at 0x7fa1cf392ca0>,),
 <QTable length=1>
   id  xcenter ycenter    aperture_sum
         pix     pix
 int64 float64 float64      float64
 ----- ------- ------- ------------------
     1    66.0    65.0 117798.64758467033)
```

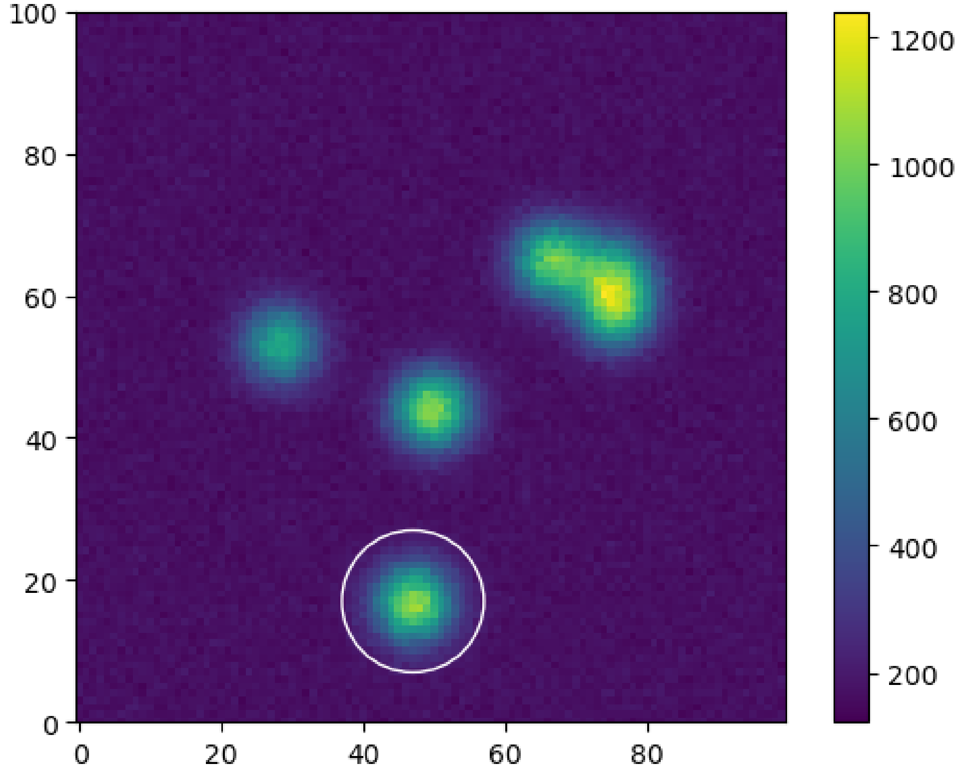


[127]:
```
#ap_phot(image, x, y, radius, skyx, skyy, skyinner, skyouter, ron)
flux = ap_phot(Q4.get_data(), 66, 65, 8, 80, 20, 0, 25, 2.04)[0]
flux_err = ap_phot(Q4.get_data(), 66, 65, 8, 80, 20, 0, 25, 2.04)[1]
print('Flux = ', flux, '+/- ', flux_err)
```

```
Flux =  87383.0209099928 +/-  344.4573666846361
```

[128]:
```
#estimating the position of star 2 for comparison
Q4.plot_x(47)
#Q4.plot_x(48)

#Q4.plot_y(18)
Q4.plot_y(17)

#x = 47, y = 17
ShowAperture(47, 17, 10, Q4)

flux = ap_phot(Q4.get_data(), 47, 17, 10, 80, 20, 0, 25, 2.04)[0]
flux_err = ap_phot(Q4.get_data(), 47, 17, 10, 80, 20, 0, 25, 2.04)[1]
print('Flux = ', flux, '+/- ', flux_err)
```

```
Flux =  83023.87289125158 +/-  363.1527209808784
```



The flux from star one is likely to be overestimated as the spatial resolution of the image is not high enough to separate out one star from the other. This means that we will also be getting flux from the 2 nearby stars (star 5 and star 4). We can try to make the photometry measurements more accurate by using a smaller shot aperture but this will mean that we will miss some of the flux from the star which will be an issue if we are interested in obtaining the absolute flux of star 1. Applying aperture photometry techniques on star 2 which looks as though it should have a similar brightness to that of star 1 we can estimate that the flux has been overestimated by 5%.

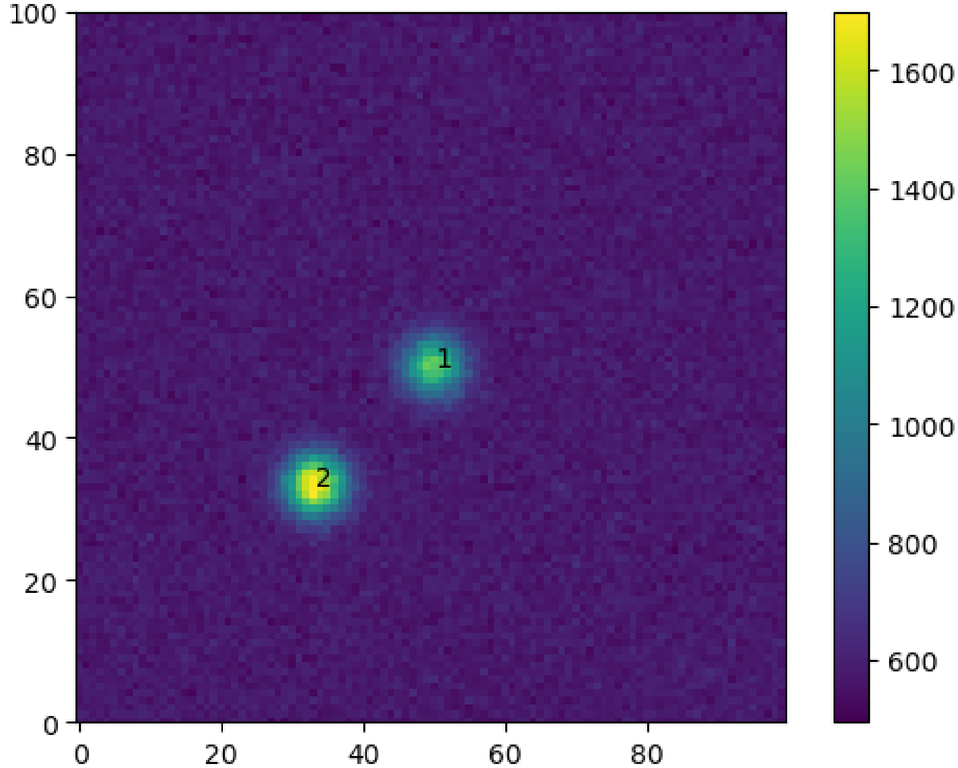
3.9 Question 5

The star indicated in the center of image Q5 (Star 1) has an apparent r-band magnitude of r=19.37 +/- 0.04 mag.

Given this information, what is the image zeropoint? Show and document your full working.

[129]:
```
Q5 = ImageSimulator(student_id=student_id, assessmentQ=5, size=(100, 100))
```

```
RON is 3.80
```

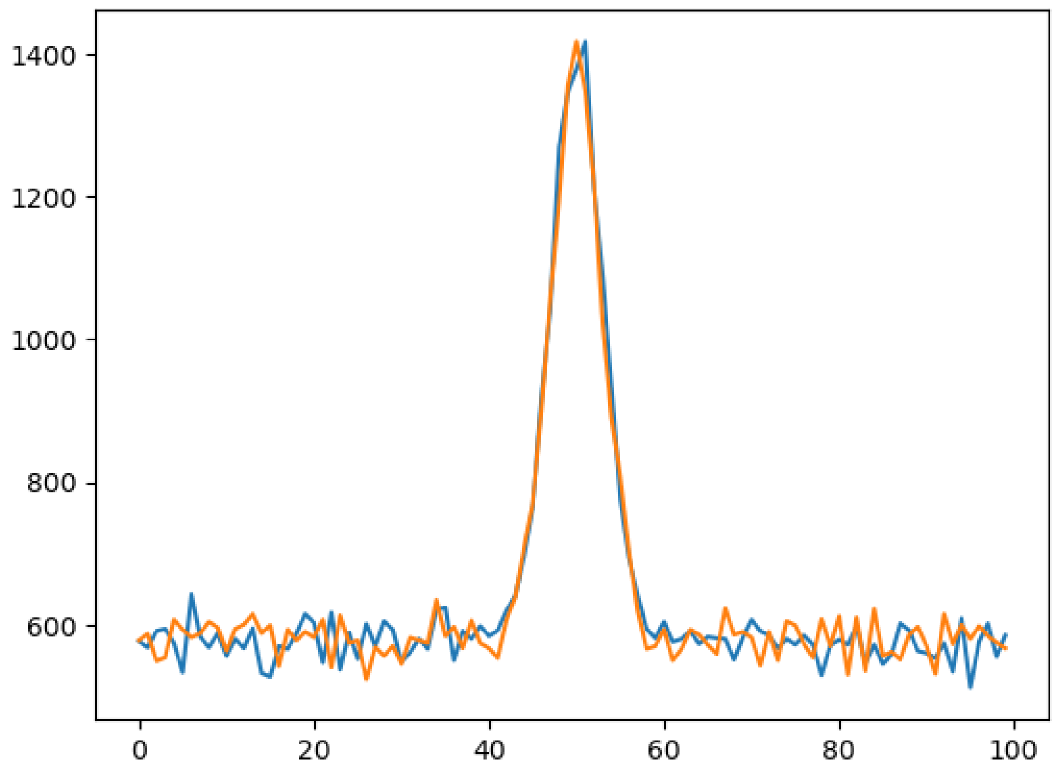


3.10 Answer 5:

The zero-point of the image is given by:
$Zeropoint = Mag + 2.5log(Flux)$

[130]:
```
#This function calculates the zero-point of the image.
def zeropoint(Mag, Flux, Mag_err, Flux_err):
    zeropoint = Mag + 2.5* np.log10(Flux)
    #error calculated using error propagation with partial derivatives
    zeropoint_err= np.sqrt(Mag_err**2 + ((1/(Flux*np.log(10)))*2.5*Flux_err)**2)
    return zeropoint, zeropoint_err
```

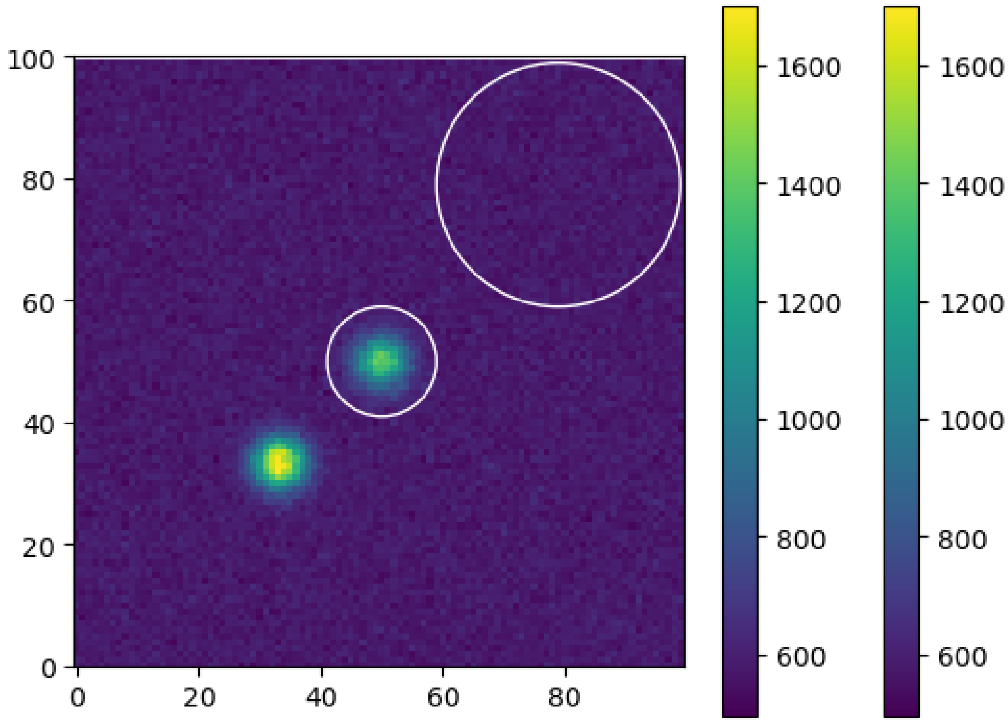
[131]:
```
Q5.plot_x(50)
Q5.plot_y(51)
```



[132]:

```
#x = 50
#Y = 50
#ShowAperture(xpos, ypos, rad, image)
radius = 9
ShowAperture(50, 50, radius, Q5)
ShowAperture(79, 79, 20, Q5)

flux_1 = ap_phot(Q5.get_data(), 50, 50, radius, 79, 79, 0, 20, 3.8)[0]
flux_err_1 =  ap_phot(Q5.get_data(), 50, 50, radius, 79, 79, 0, 20, 3.8)[1]

#print(flux_1)
#print(flux_err_1)

Mag = 19.37
Mag_err = 0.04

zp = zeropoint(Mag, flux_1, Mag_err, flux_err_1)[0]
zp_err = zeropoint(Mag, flux_1, Mag_err, flux_err_1)[1]

print("zeropoint = ", zp, " +/- ", zp_err)
```

```
zeropoint =  31.052322379155207  +/-  0.04129577849712782
```



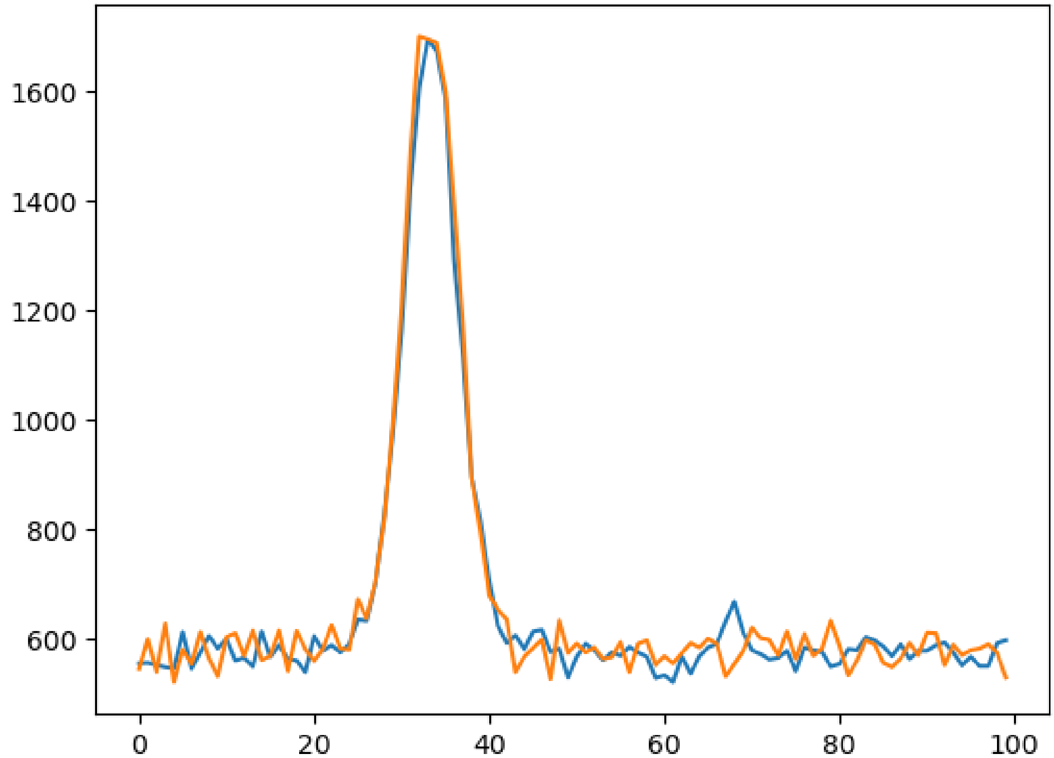
3.11 Question 6

Use the zeropoint that you derived in Question 5 to determine the apparent magnitude of the star indicated in the image above. Provide the associated statistical and systematic magnitude uncertainty in your answer.

3.12 Answer 6:

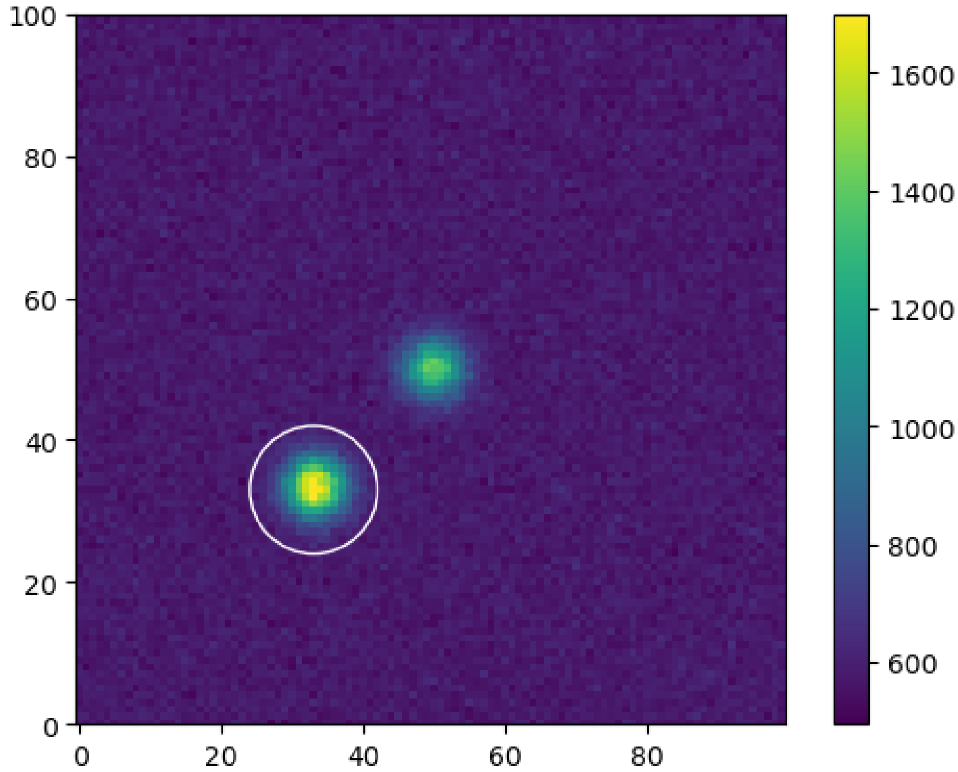
[133]:
```
def magnitude(zeropoint, Flux, zeropoint_err, Flux_err):
    mag = zeropoint - 2.5*np.log10(Flux)
    mag_err= np.sqrt(zeropoint_err**2 + ((1/(Flux*np.log(10)))*2.5*Flux_err)**2)
    return mag, mag_err
#random uncertainty = flux err
#systematic uncertainty = zero-point err.
```

[134]:
```
Q5.plot_y(33)
Q5.plot_x(33)
```



[135]:
```
ShowAperture(33, 33, 9, Q5)
#Aperture should be the same for both stars
```

[135]:
```
(None,
 (<matplotlib.patches.Circle at 0x7fa1ce289f70>,),
 <QTable length=1>
   id  xcenter ycenter    aperture_sum
         pix     pix
 int64 float64 float64      float64
 ----- ------- ------- ------------------
     1    33.0    33.0 211944.39371450763)
```



[136]:
```
star2_flux = ap_phot(Q5.get_data(), x = 33, y = 33, radius = 9, skyx = 79, skyy␣
 ↪= 79,\
 skyinner = 0,skyouter = 20, ron = 3.8)[0]
star2_flux_err = ap_phot(Q5.get_data(), x = 33, y = 33, radius = 9, skyx = 79,␣
 ↪skyy = 79,\
 skyinner = 0,skyouter = 20, ron = 3.8)[1]
print('Flux Uncertainty: ', star2_flux_err)
```

```
Flux Uncertainty:  464.47402893381786
```

[137]:
```
mag = magnitude(zp,star2_flux ,zp_err,star2_flux_err)[0]
mag_err = magnitude(zp,star2_flux ,zp_err,star2_flux_err)[1]
print("apparent mag = ", mag, " +/- ", mag_err)
```

```
apparent mag =  19.025568800882308  +/-  0.042025592581555186
```

The random uncertainty is defined as unpredictable changes in the photon flux that is measured. Read out noise and sky background are examples of random noise that affects each datapoint differently. Systematic uncertainties is the uncertainty in the magnitude of the standard star used to calculate the zero-point.

Zero Point Error is the systematic uncertainty which has a value of plus or minus 0.04 as shown in question 5. The statistical random uncertainty comes from the measurement of the flux which has a value of 465. Error propogation of all the errors is shown in the 'magnitude' function.

3.13 Question 7

How would the zeropoint that you measured above change if the observations were taken in better observing conditions (e.g. less air turbulance, few clouds or cirrus) ?

3.14 Answer 7:

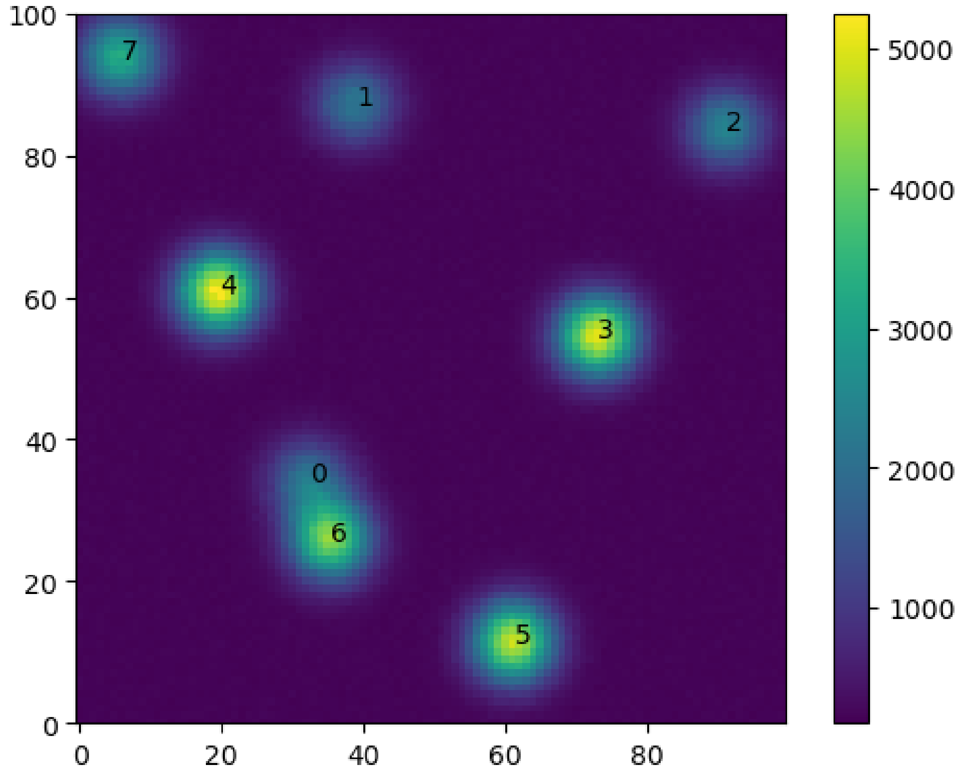
If the observations were taken in better observing conditions, then the measured flux of the standard star would be larger and hence the zero-point would also be bigger. The better observing conditions means that the random uncertainty per pixel would be lower so the calculated zero-point would be more accurate.

3.15 Question 8

The apparent magnitude of all sources in the image below is known to an accuracy of \$±\$10% from previous surveys. Which of the numbered sources in the image would be appropriate, and conversely, inappropriate for deriving the image zeropoint? Provide reasons for your response.

[138]:
```
Q8 = ImageSimulator(student_id=student_id, assessmentQ=8, size=(100, 100))
```

```
RON is 2.22
```



3.16 Answer 8:

Star 3, 4 and 5 would be suitable for measurement, these are bright sources that are spatially resolved making them a good choice for the zero-point. Choosing a bright star decreases the fractional uncertainty in the measurement of it's brightness. Stars 1, 2 and 7 are quite faint making them not as great choices. The star should be spatially resolved as well to get an accurate reading for the flux of the star. The above points are important as getting a good reading of the flux from the instrument of a star of known magnitude will mean better calibration.

3.17 Question 9

Which of the science objectives listed below require absolute flux calibration, and which are achievable with relative flux calibration?

a) Change in flux of a star due to transiting exoplanet.

b) The absolute magnitude of the host of an exoplanet.

c) The difference in the B-band and V-band magnitude (i.e. the colour) of the host star.

Provide your reasoning in your answers.

3.18 Answer 9:

a) This requires relative flux calibration as we are only interested in the change in flux as the question states. This means that even if the absolute values of the flux are incorrect as long as the changes in flux is correct, this is satisfactory for transit dip measurements.

b) This requires absolute flux calibration. The relative flux method will not be satisfactory as this method takes the errors due to the atmosphere as being the same as another nonvarying star and adjustments are made based on this star and the points are moved up or down. The points themselves may be too bight or too dim but the change will be correct.

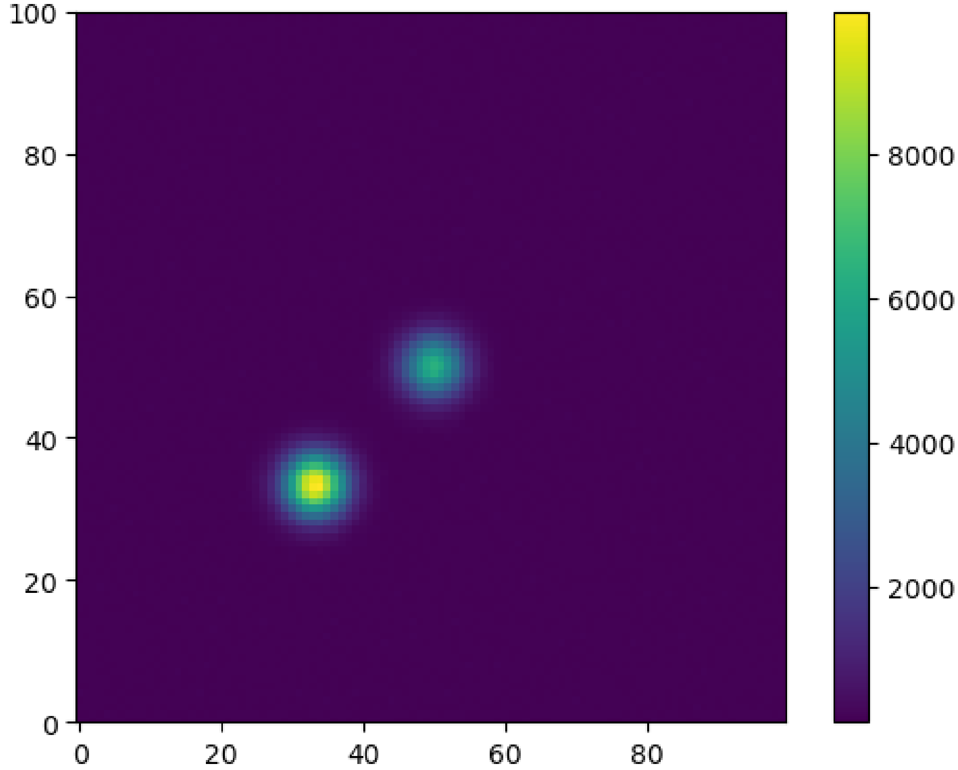
c) Relative flux is required for this as the ratio of luminosity in the B band and the V band need to be correct but the actual value of the luminosity does not need to be.

3.19 Question 10

The two images below (Q10a, Q10b) show a star targeted for an exoplanet transit in the centre of the image. The second star is the standard star with constant flux. Apply relative flux calibration to determine which of the two images shows the planet in transit.

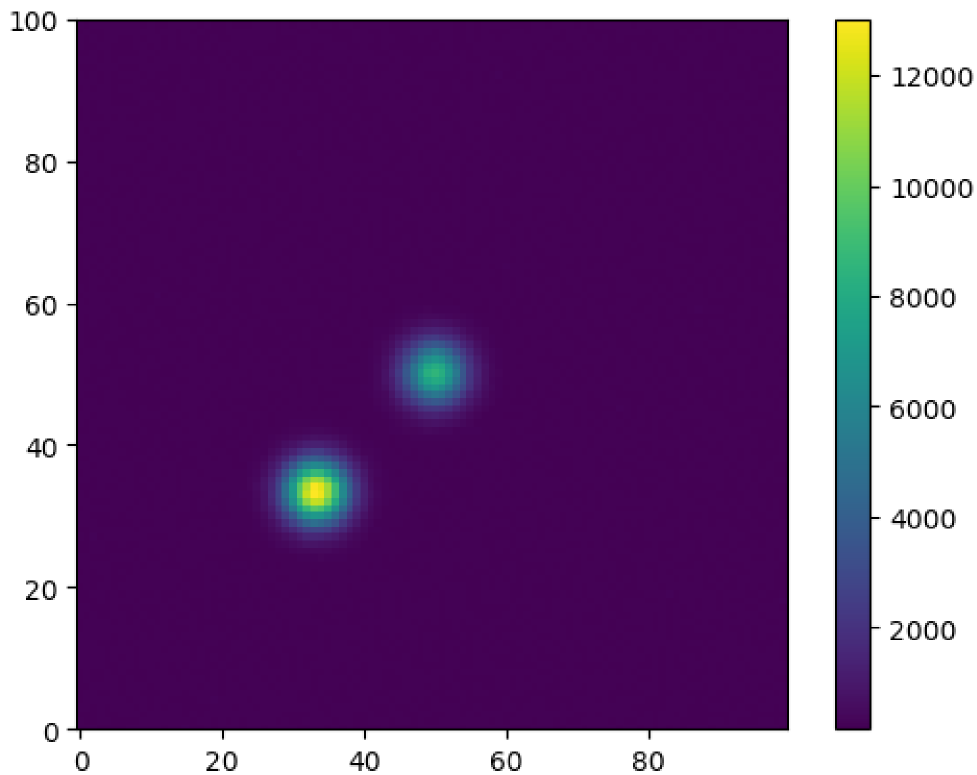
[139]:
```
Q10a = ImageSimulator(student_id=student_id, assessmentQ=101, size=(100, 100))
```

```
RON is 2.50
```



[140]:
```
Q10b = ImageSimulator(student_id=student_id, assessmentQ=102, size=(100, 100))
```

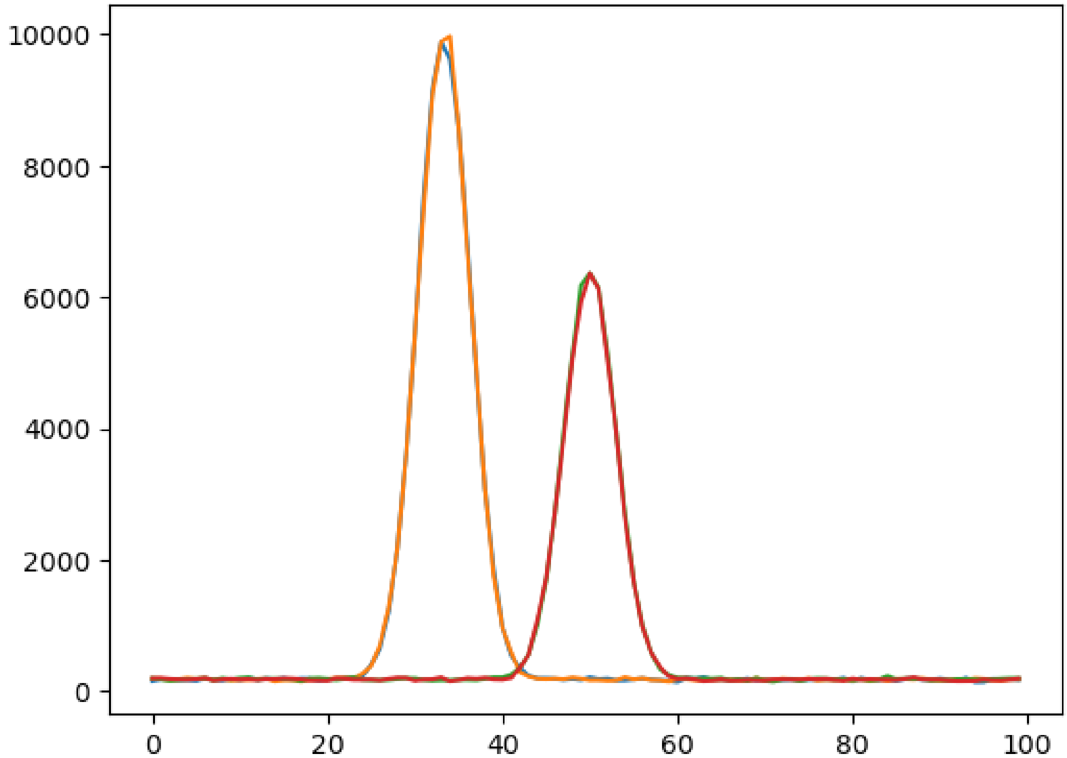
```
RON is 2.50
```



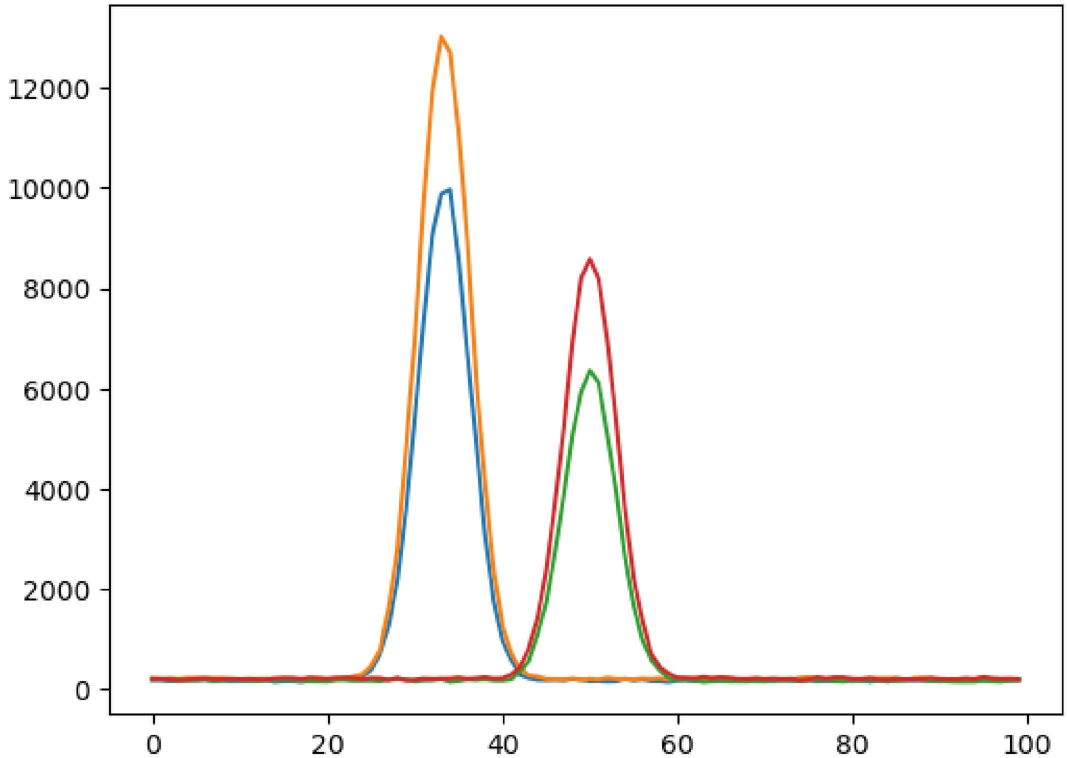

```
[ ]:
```

3.20 Answer 10:

```
[141]: Q10a.plot_x(33) #constant star
       Q10a.plot_y(33)
       Q10a.plot_x(50) #transit star
       Q10a.plot_y(50)
```



[142]:
```
Q10a.plot_y(33)
Q10b.plot_y(33) #the flux has increased

Q10a.plot_y(50) # transit
Q10b.plot_y(50) # the flux has also increased for the transit star
```



The first image (10a) shows the planet in transit as the flux of this star is lower than the flux in 10 b. Relative flux calibration is shown in the following blocks:

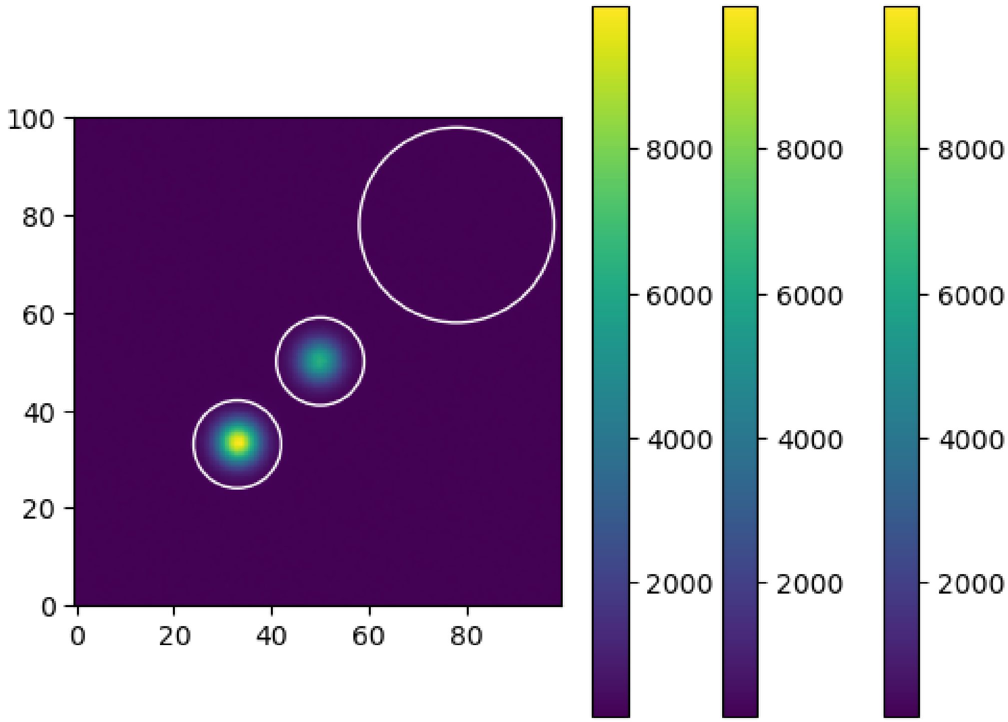
Steps:

- Take the change in flux of the non-varying star. Calculate the percentage change.
- Make corrections for the change in flux of transit star and then see which image has lower flux (indicating transit)

Flux of standard - non varying star:

[143]:
```
ShowAperture(33, 33, 9, Q10a)
ShowAperture(78, 78, 20, Q10a)
ShowAperture(50, 50, 9 , Q10a)
```

[143]:
```
(None,
 (<matplotlib.patches.Circle at 0x7fa1cdd887f0>,),
 <QTable length=1>
   id  xcenter ycenter   aperture_sum
         pix     pix
 int64 float64 float64     float64
 ----- ------- ------- ----------------
     1    50.0    50.0 391076.275777236)
```



[144]:
```python
#Calculating the flux of each star from each image:
flux_s_1 = ap_phot(Q10a.get_data(), 33, 33, 9, 78, 78, 0, 20, 2.5)[0]
flux_s_2 = ap_phot(Q10b.get_data(), 33, 33, 9, 78, 78, 0, 20, 2.5)[0]
print(flux_s_1, flux_s_2  )

flux_t_1 = ap_phot(Q10a.get_data(), 50, 50, 9, 78, 78, 0, 20, 2.5)[0]
flux_t_2 = ap_phot(Q10b.get_data(), 50, 50, 9, 78, 78, 0, 20, 2.5)[0]

print(flux_t_1, flux_t_2  )

#Applying relative flux calibration:
perc_change = (flux_s_2 - flux_s_1)/flux_s_1
print(perc_change)
print(f"The brightness has increased by %.1f%%"% (perc_change*100))
#The actual change in flux of transit star from 1 to 2:

new_flux_t_2 = flux_t_2 * (1-perc_change)
#print(new_flux_t_2)
delta_flux = new_flux_t_2 - flux_t_1
print("The actual change in flux of the transiting star is: ", delta_flux)
```

```
553198.6076026437 722289.5926397601
345452.65273654606 468143.53513890295
0.30566053983738983
```

```
The brightness has increased by 30.6%
The actual change in flux of the transiting star is:  -20402.123269584263
```

From the analysis above, we can see that eventhough it looks as though the stars in image 10b have a higher flux than 10b, in actuality the flux of the transit star has decreased. This means that the planet is transiting in image 10b, accounting for the smaller flux. The flux fractional drop in flux is around 6%.

[145]:
```
fractional_drop = ((-delta_flux)/flux_t_1)*100
print(fractional_drop)
```

```
5.905910146576184
```

3.21 Question 11

What possible reasons are there for the change in flux in the constant source in the two images above?

3.22 Answer 11:

The change in flux of the star refects the changes in attenuation of the star's light due to atmospheric variability. This may be because of cloud cover in the first image (10a) making the light from the source seem smaller than what is actually the case. Another possibility is dust in the optics when the first image was taken but not in the second. Alternatively this could be to do with the properties of the star itself, it may have properties analogous to sun spots on our sun, which cause a temporary dip in the flux emitted.