Binning (how to do HW#3, problem #3)

Say you have a dataset of noisy (x,y) values. You want to make a binned plot of the median value of y, in bins of x. You also want to estimate the sum of all the y values if they had followed the median values (instead of their actual noisy values) as a function of x.

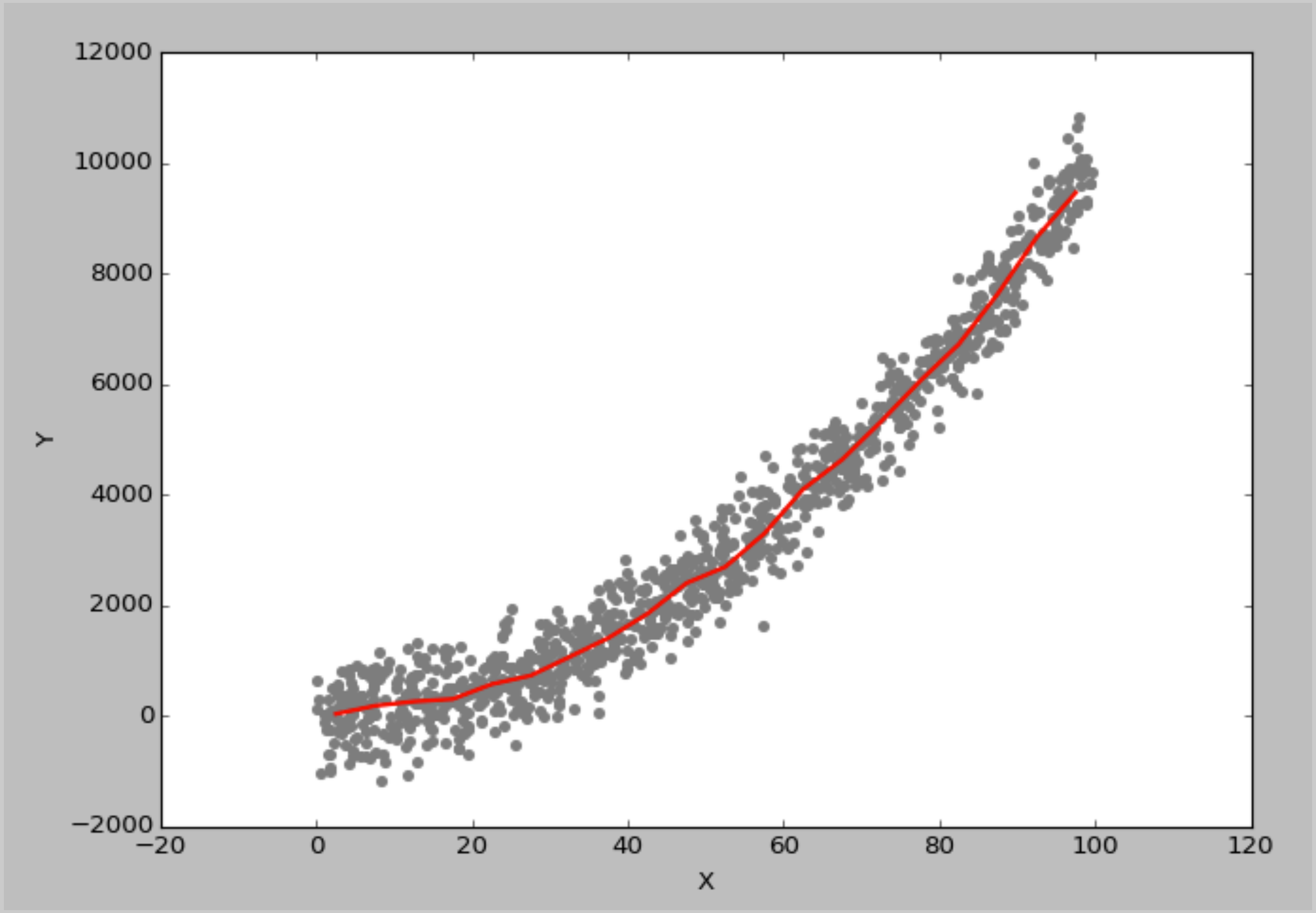
Lets do 20 bins of x, where each bin has a width dx.

```
nbin = 20
dx = 5
inbin = np.trunc(x/dx)
bin_x = np.zeros(nbin)
bin_med = np.zeros(nbin)
bin_medsum = np.zeros(nbin)
for i in range(0,nbin):
    want = (inbin==i)
    bin_med[i] = np.median(y[want])
    bin_medsum[i] = np.median(y[want])*len(y[want])
    bin_x[i] = (i+0.5)*dx

plt.scatter(x,y)
plt.scatter(bin_x,bin_med)
print np.sum(bin_medsum)
```

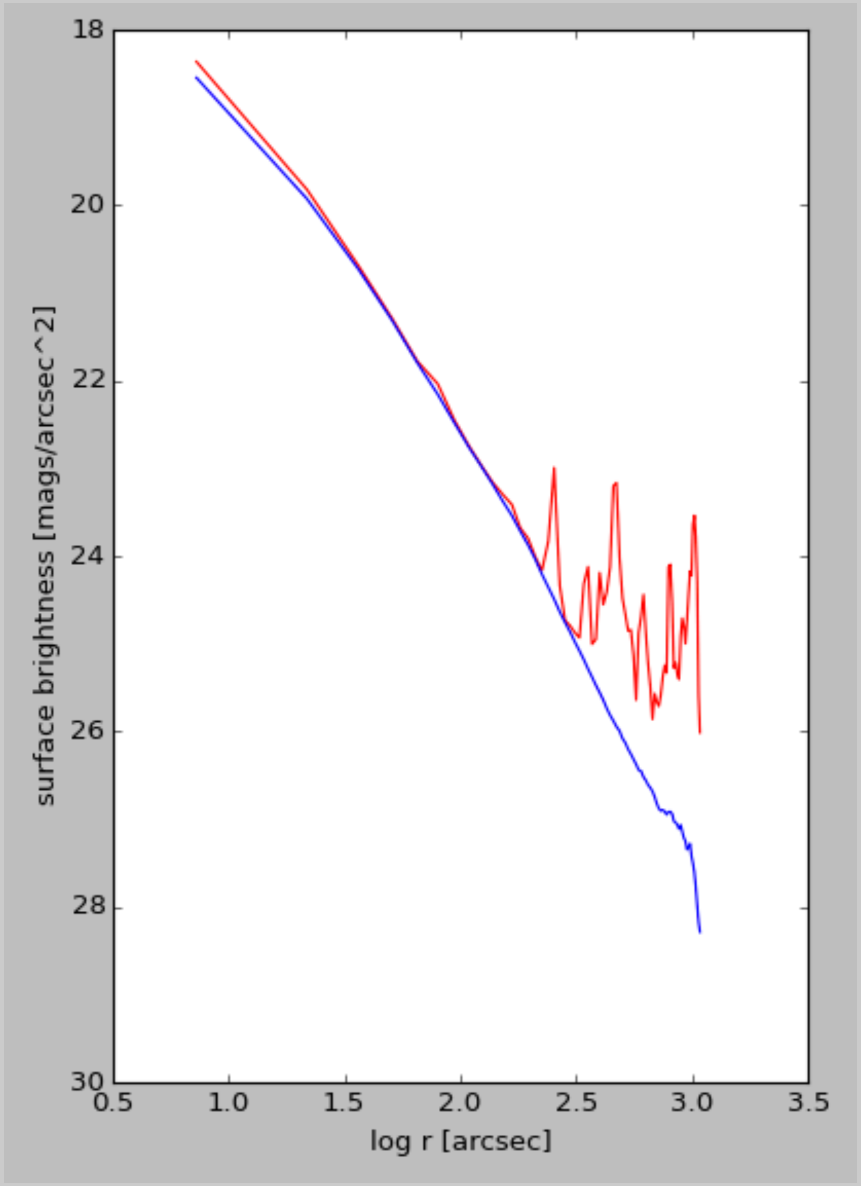
Example binning

grey dots: raw unbinned noisy data
red line: median of y values in 20 bins of x with bin size dx=5.



Surface Brightness Profile of M84 (how to do HW#3, problem #3)

Your plot for M84's surface brightness profile should look a lot like this.



Combining Images

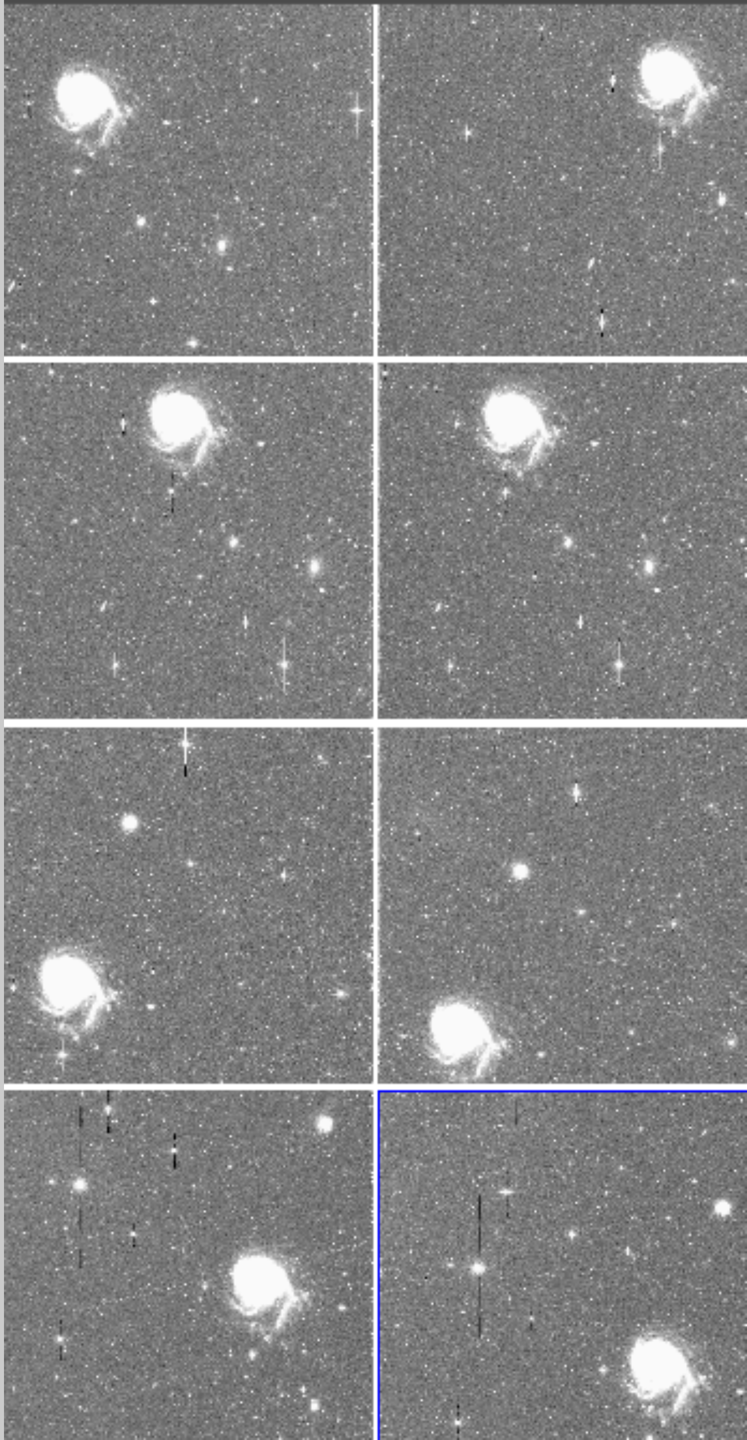
We can digitally combine N individual images into one master image of much better quality.

Advantage #1: Increase exposure time and signal-to-noise.

Advantage #2: Correct for image contaminants (cosmic rays, satellite trails, scattered light)

Advantage #3: Correct for detector problems (bad columns, flat fielding variations, etc)

Advantage #4: Reduce observing risk.



Combining Images

Problem #1: Different Photometric Zeropoints

Images were taken at different airmasses (and sometimes on different nights) so they have different photometric zeropoints. The same star will produce fewer counts when observed at greater airmass. We can't just average all the images together, we have to scale them to correct for the different zeropoints.

Method #1: Observe standard stars, work out photometric solution, apply to individual images.

$$m_V = m_{inst} + K_V * sec(z) + C_V * (B-V) + ZP_V$$

Method #2: Measure stars of known brightness on the image, calibrate zeropoints directly.

$$m_V = m_{inst} + C_V * (B-V) + ZPFRAME \text{ (where } ZPFRAME = K_V * sec(z) + ZP_V)$$

Each star on the image gives a value for $m_V - m_{inst}$ and (B-V), plot those values and work out C_V and ZPFRAME.

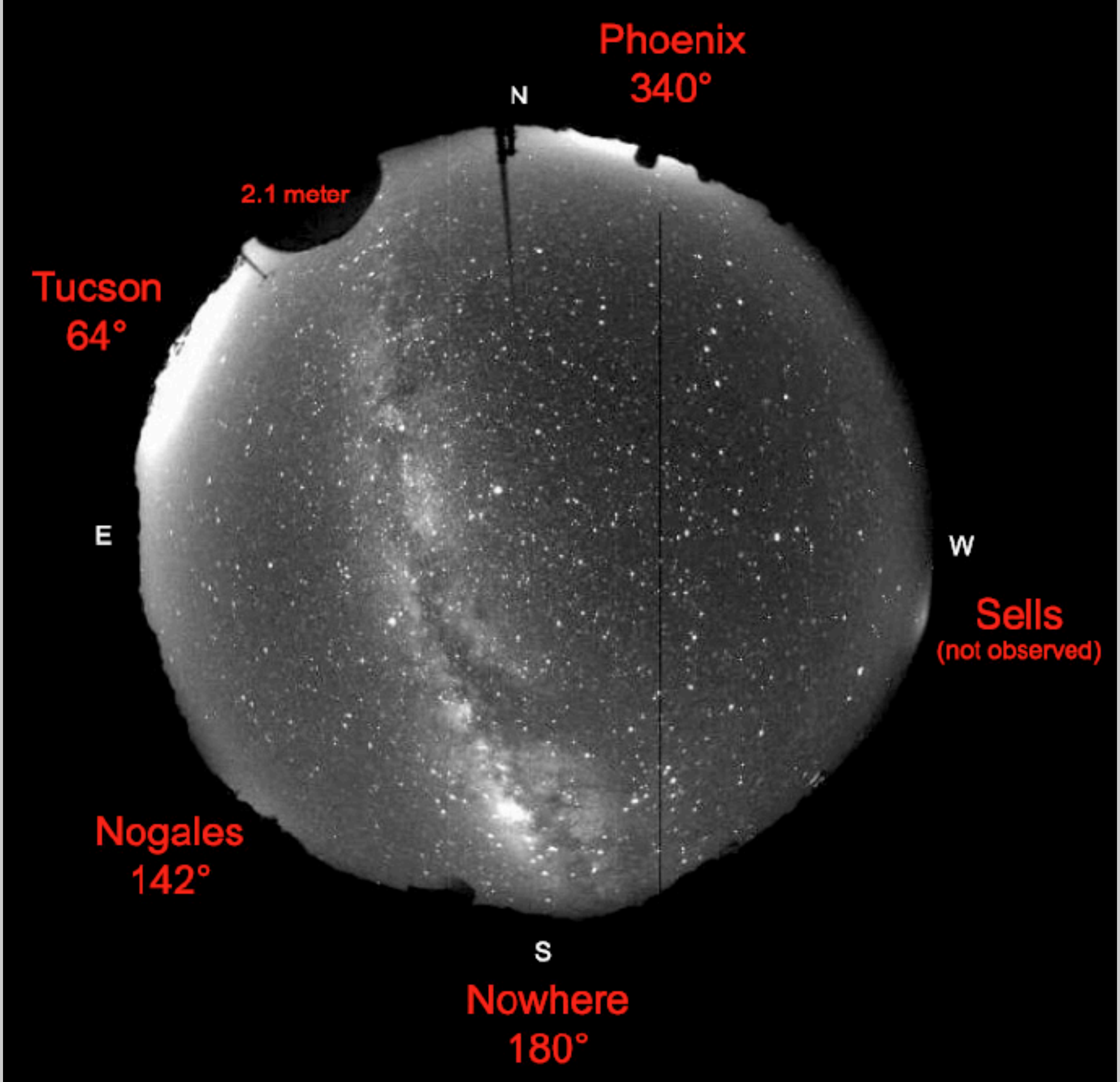
Combining Images

Problem #2: Different Sky values

Sky brightness can change over the course of a night, and also depends on airmass and direction you are observing. So the images all have different sky levels and we have to subtract off this sky level *before* combining.

Method #1: Measure sky at many spots across the image, work out an average value, subtract it off.

SKY = average sky


Phoenix
340°
N
2.1 meter
Tucson
64°
E
W
Sells
(not observed)
Nogales
142°
S
Nowhere
180°

Combining Images

Problem #2: Different Sky values

Sky brightness can change over the course of a night, and also depends on airmass and direction you are observing. So the images all have different sky levels and we have to subtract off this sky level *before* combining.

Method #1: Measure sky at many spots across the image, work out an average value, subtract it off.

SKY = average sky

Method #2: Measure sky at many spots across the image, fit a plane to the sky level as a function of X,Y position on the image.

SKY = X*dx + Y*dy + const

Sky correction

Last time: fit a plane to the sky to subtract: SKY = X*dx + Y*dy + const

After subtracting this sky model, we can look at the residual sky values around each star we did photometry on.

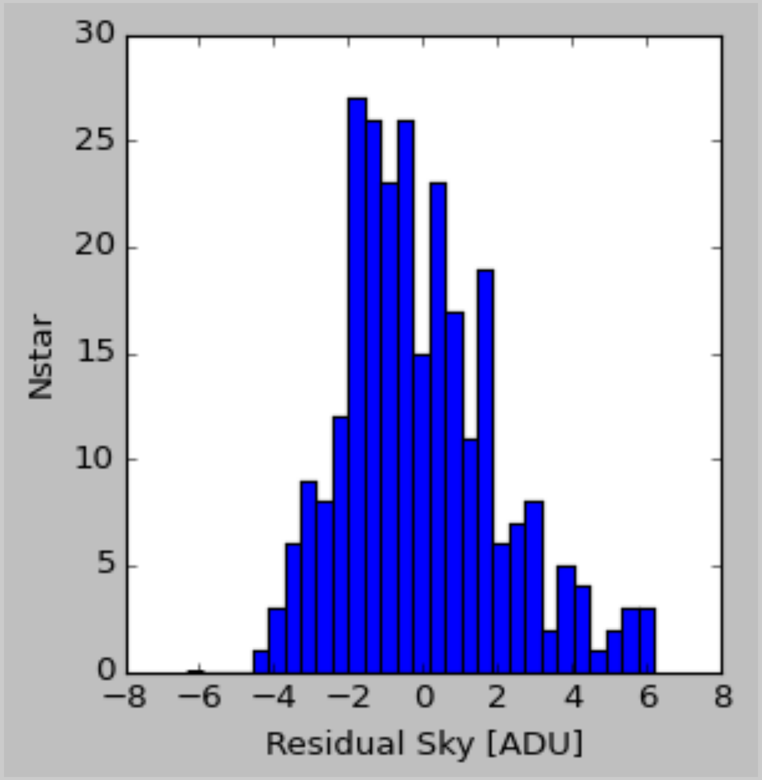
It looks asymmetric, skewed towards positive values.

- Mean = −0.05 ADU
- Median = −0.43 ADU

Why would it be skewed positive?

So the constant sky value is biased to be too bright. We will correct for this by reducing the fitted const value by the median offset.

SKY0 = const − 0.43



Photometric Scaling

The zeropoint converts counts (in ADU) to calibrated magnitude. A star of given magnitude m will have different counts in images with different zeropoints. Since zeropoints are in magnitudes, we can say

$ZP_1 – ZP_2 = −2.5*\log(COUNTS_1/COUNTS_2)$

So we can pick a “master zeropoint” and rescale all the images in intensity so that they have that zeropoint:

$COUNTS\_MASTER = COUNTS * 10^{+0.4*(ZP – MASTERZP)}$

That 10^ term is the photometric scaling we multiply each image by to get them on the master zeropoint.

What do we chose for the MASTERZP?   MASTERZP = np.average(ZP)

Applying corrections to data

imexpression: pyraf command to do complex image math. Conceptually, we want to do the following:

corrected image = (flattened image – sky model) * photometric scaling

using imexpr we say:

I means CCD X coordinate
J means CCD Y coordinate

```
imexpr
    expr=“(a-(I*b+J*c+d+e))*f”
    output = c<imagename>
    a = <imagename>
    b = <dx value for image>
    c = <dy value for image>
    d = <const sky value for image>
    e = <median sky residual for image>
    f = <photometric scaling>
```

Grab and run the scripts:

in pyraf, in your Vdata directory:

```
!cp /astroweb_data/ASTR306/Vcalib.cl .
cl < Vcalib.cl
```

And then do the same thing to the B data with Bcalib.cl

Now you have sky subtracted, photometrically scaled data. We are almost ready to combine!