Untitled0

November 4, 2014

```
In [30]: import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         from __future__ import division
```

1 Problem 1

According to Snell's Law:

$$\sin\alpha_0 = n\sin\alpha \tag{1}$$

the refraction angle:

$$\Delta\alpha \approx -\tan\alpha_0 \cdot \frac{\Delta n}{n^2} \propto \Delta n \tag{2}$$

According to Cauchy Formula,

$$\Delta n(\lambda = 0.6) = 276.0 \times 10^6 \Delta n(\lambda = 0.4) = 280.2 \times 10^6 \Delta n(\lambda = 0.8) = 274.5 \times 10^6 \tag{3}$$

Thus, the refraction angle:

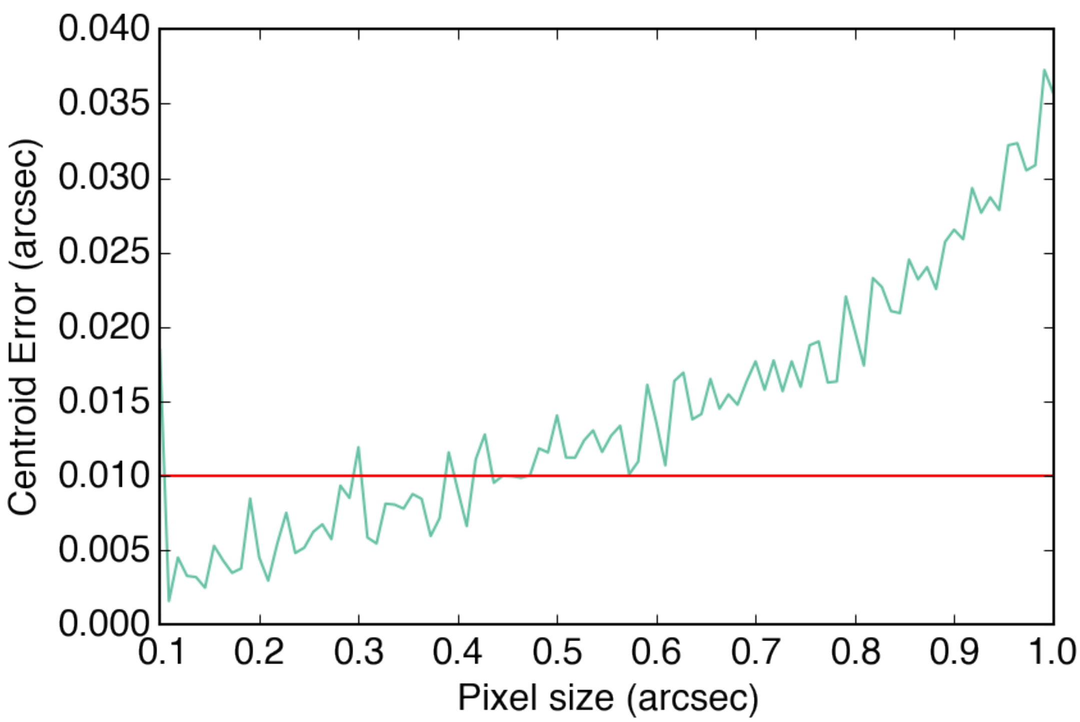
$$\Delta = \Delta\alpha(\lambda = 0.4) - \Delta\alpha(\lambda = 0.8) = (\frac{280.2}{276.0} - \frac{274.5}{276.0})\text{arcmin} = 0.053\text{arcmin} \tag{4}$$

2 Problem 2

```
In [60]: def Gauss(x, FWHM):
             sigma = 1./2.3548 * FWHM
             return np.exp(-x**2/(2*sigma**2))

         x = np.linspace(-10, 10, 2000) #length in arcsec, oversample
         y = Gauss(x, 1.)

         def centroidErr(pixelSize):
             pixelCenter = np.sort(np.concatenate((np.arange(-0.75*pixelSize, -10+pixelSize,-pixelSize)
                                          np.arange(0.25*pixelSize, 10 -pixelSize, pixelSize))))
             pixelFlux = []
             for pxCenter in pixelCenter:
                 # flux for one pixel can be devided into three part
                 flux1 = np.sum(0.9 * y[(x > pxCenter - 0.5 * pixelSize)&(x < pxCenter - 0.45 * pixelSi
                 flux2 = np.sum(y[(x > pxCenter - 0.45 * pixelSize)&(x < pxCenter + 0.45 * pixelSize)])
                 flux3 = np.sum(0.9 * y[(x > pxCenter + 0.45 * pixelSize)&(x < pxCenter + 0.45 * pixelSi
                 pixelFlux.append(flux1 + flux2 + flux3)
```

```
    pixelFlux = np.array(pixelFlux)
    cMeasured = np.sum(pixelCenter * pixelFlux)/np.sum(pixelFlux)
    return abs(cMeasured)

psList = np.linspace(0.1, 1, 100)
errList = map(centroidErr, psList)
plt.plot(psList,errList)
plt.axhline(0.01, color = 'r')
plt.xlabel('Pixel size (arcsec)')
plt.ylabel('Centroid Error (arcsec)')
```

Out[60]: <matplotlib.text.Text at 0x10c7fc250>



Above script calculates the centroid measurement error as a function of pixel size. According to the plot, the sample size should be smaller than 0.4 arcsec so that the centroid can be measured as precise as 0.01 arcsec

3 Problem 3

table assembled as below

```
In [75]: df0 = pd.read_csv('table0.csv', names = ['name','type','Jmag','Kmag'], usecols = (0,1,2,3))
         df1 = pd.read_csv('table1.csv', names = ['name', 'Jmag', 'Kmag'], usecols = (0,3,5))
         df0['name'] = [string.strip() for string in df0['name'].values]
         df1['name'] = [string.strip() for string in df1['name'].values]
         total =pd.merge(df0, df1, on = 'name')
         total['J-K(KM)'] = total['Jmag_y'] - total['Kmag_y']
         total
```

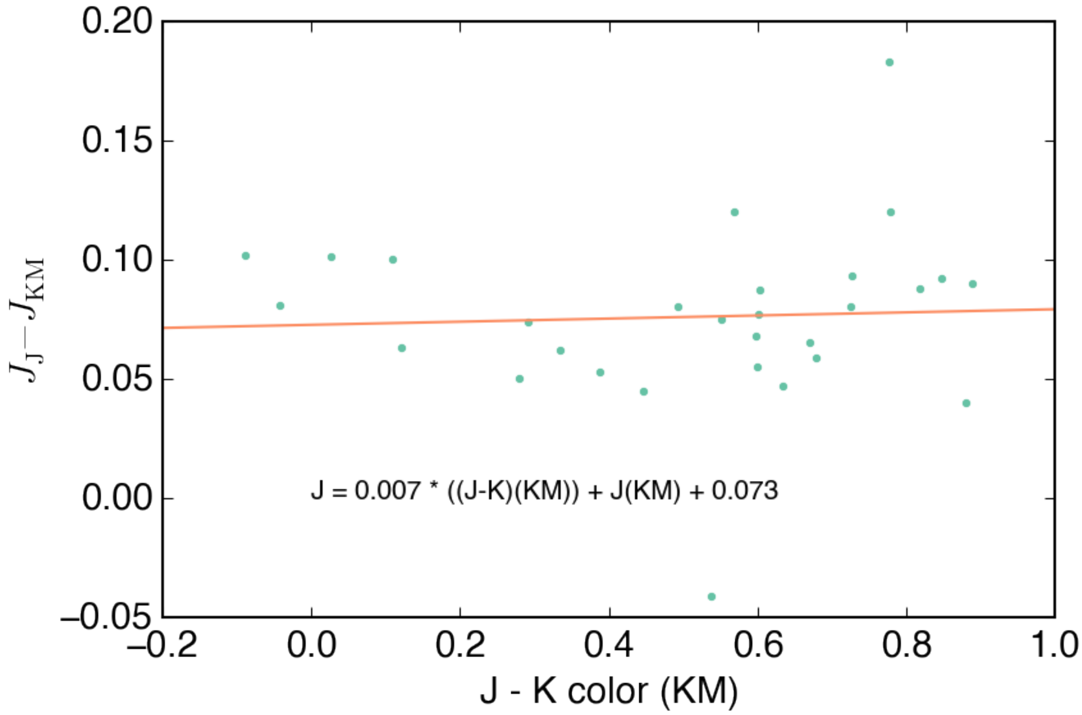
Out[75]:

	name	type	Jmag_x	Kmag_x	Jmag_y	Kmag_y	J-K(KM)
0	HR 1256	K0III	2.63	1.97	2.575	1.9750	0.6000
1	HR 1286	K1II-III	3.36	2.49	3.177	2.4000	0.7770
2	HR 1791	B7III	1.97	2.05	1.868	1.9560	-0.0880
3	HR 1907	K0IIIb	2.35	1.69	2.282	1.6840	0.5980
4	HR 1963	K1III	2.89	2.11	2.797	2.0700	0.7270
5	HR 2077	K0III	2.09	1.46	1.970	1.4000	0.5700
6	HR 2427	K3Iab	2.79	2.05	2.731	2.0520	0.6790
7	HR 2560	G5III-IV	2.89	2.33	2.810	2.3160	0.4940
8	HR 3003	K4III	2.28	1.33	2.190	1.3000	0.8900
9	HR 4335	K1III	1.16	0.44	1.095	0.4240	0.6710
10	HR 4377	K3III	1.18	0.31	1.092	0.2740	0.8180
11	HR 4608	G8IIIa	2.48	1.90	2.435	1.9877	0.4473
12	HR 4737	K1III	2.55	1.90	2.473	1.8710	0.6020
13	HR 4983	F9.5V	3.24	2.90	3.166	2.8730	0.2930
14	HR 5107	A3V	3.20	3.11	3.099	3.0720	0.0270
15	HR 5854	K2IIIb	0.76	0.06	0.713	0.0790	0.6340
16	HR 5947	K2III	2.09	1.30	2.010	1.2840	0.7260
17	HR 6623	G5IV	2.18	1.77	2.127	1.7390	0.3880
18	HR 6698	G9III	1.68	1.12	1.721	1.1830	0.5380
19	HR 6705	K5III	-0.39	-1.34	-0.430	-1.3100	0.8800
20	HR 6707	F2II	3.55	3.23	3.500	3.2200	0.2800
21	HR 7236	B9Vn	3.64	3.67	3.559	3.6000	-0.0410
22	HR 7525	K3II	0.30	-0.59	0.208	-0.6400	0.8480
23	HR 7557	A7V	0.39	0.26	0.327	0.2050	0.1220
24	HR 7615	K0III	2.28	1.67	2.205	1.6530	0.5520
25	HR 7949	K0III	0.77	0.11	0.683	0.0800	0.6030
26	HR 8143	B9Iab	3.95	3.79	3.850	3.7400	0.1100
27	HR 8632	K2III	2.36	1.51	2.240	1.4600	0.7800
28	HR 8905	F8IV	3.37	3.02	3.308	2.9730	0.3350

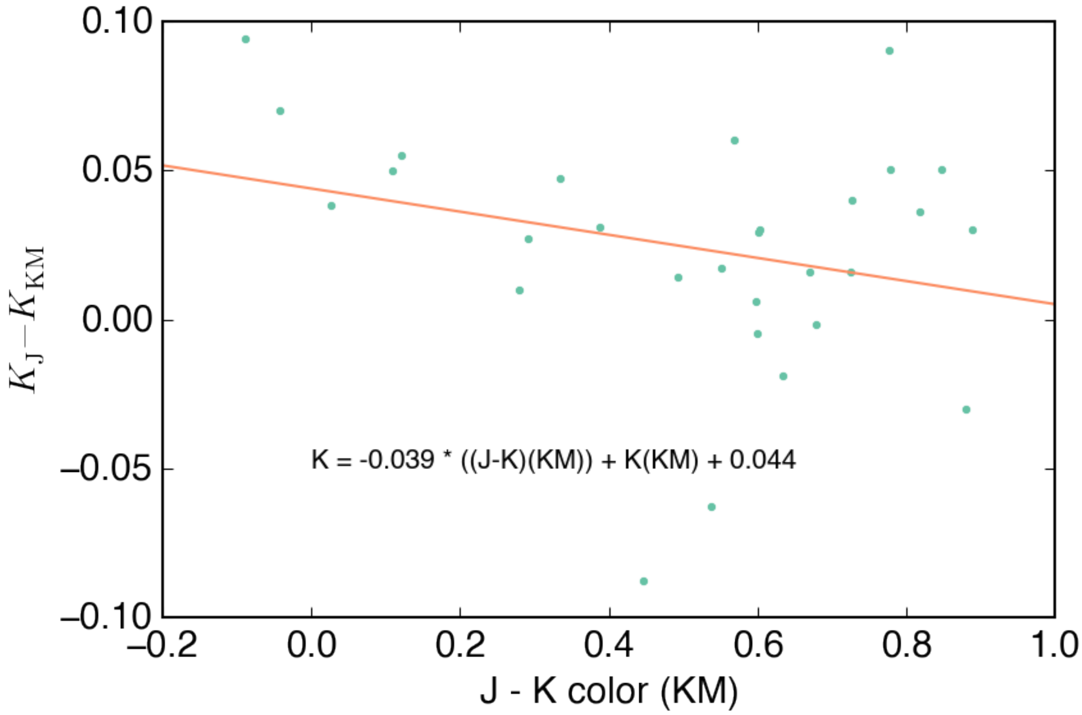
In [79]:

```
#Linear Fit
#For J
JK = np.linspace(-0.2, 1.0, 100)
mJ, bJ = np.polyfit( total['J-K(KM)'], total['Jmag_x']-total['Jmag_y'], 1)
plt.plot(total['J-K(KM)'], total['Jmag_x'] - total['Jmag_y'], '.')
plt.plot(JK, mJ*JK + bJ)
plt.xlabel('J - K color (KM)')
plt.ylabel('$J_\mathrm{J} - J_\mathrm{KM}$')
plt.text(0, 0, 'J = {0:.3f} * ((J-K)(KM)) + J(KM) + {1:.3f}'.format(mJ, bJ))
```

Out[79]: <matplotlib.text.Text at 0x10c84d210>




```
In [81]: mK, bK = np.polyfit( total['J-K(KM)'], total['Kmag_x']-total['Kmag_y'], 1)
         plt.plot(total['J-K(KM)'], total['Kmag_x'] - total['Kmag_y'], '.')
         plt.plot(JK, mK*JK + bK)
         plt.xlabel('J - K color (KM)')
         plt.ylabel('$K_\mathrm{J} - K_\mathrm{KM}$')
         plt.text(0, -0.05, 'K = {0:.3f} * ((J-K)(KM)) + K(KM) + {1:.3f}'.format(mK, bK))

Out[81]: <matplotlib.text.Text at 0x10fd8bf10>
```



With linear fit of J and K magnitude difference with KM's J-K color, Johnson's magnitude can be expressed as above.

4 Problem 4

a) According to the plot in Problem 3, the precision can be around 0.1 magntiude.

b) For vega, J-K color equals 0. According to the plot above, Vega has a J magnitude of -0.05 and K magnitude of -0.02. Vega changes because it is a fasting rotate star and varies a lot.

5 Problem 5

- determined by integrating the source spectrum over the filter passband(if $f = a\lambda^{-4}d\lambda$):

$$
\begin{aligned}
I &= a\int_{\lambda_1}^{lambda_2} \lambda^{-4} d\lambda \\
&= \frac{a}{3}\left(\frac{1}{(\lambda-\Delta/2)^3} - \frac{1}{(\lambda_2+\Delta/2)^3}\right) \\
&= \frac{a}{3\lambda^3}\left[1 + 3\left(\frac{\Delta}{2\lambda}\right) + \frac{3\times 4}{2!}\left(\frac{\Delta}{2\lambda}\right)^2 + \frac{3\times 4\times 5}{3!}\left(\frac{\Delta}{2\lambda}\right)^3 + \cdots\right] - \frac{a}{3\lambda^3}\left[1 - 3\left(\frac{\Delta}{2\lambda}\right) + \frac{3\times 4}{2!}\left(\frac{\Delta}{2\lambda}\right)^2 - \frac{3\times 4\times 5}{3!}\left(\frac{\Delta}{2\lambda}\right)^3 + \right. \\
&= a\frac{\Delta}{\lambda^4}\left[1 + \frac{5}{6}\left(\frac{\Delta}{\lambda}\right)^2\right] + O\left(\frac{\Delta^4}{\lambda^4}\right)
\end{aligned}
\tag{5}
$$

6 Problem 6

B-V color is 0.68, thus

- $T_{\text{eff}} = 5560K$
- absolute V band magnitude is $M_{\text{V}} = 5.1$
- Distance:

$$D = 10^{\frac{m_V - M_V + 5}{5}} = 10^{(7.02-5.1+5)/5} = 24.2\text{pc} \tag{6}$$

- Bolometric magnitude: $M_{\text{bol}} = M_{\text{V}} + BC = 4.89$
- Luminosity:

$$L = 10^{(M_{\text{bol},\odot} - M_{\text{bol}})/2.5} L_{\odot} = 10^{(4.75-4.89)/2.5} L_{\odot} = 0.88 L_{\odot} \tag{7}$$

7 Problem 7

use two half wave plate would rotate the position angle with 180°, that is two say it would be the same as there is no half wave plate

```
In [ ]:
```