ASTR3800

May 16, 2015

1 ASTR3800: Analyzing Galactic SDSS Spectras

1.1 By: Paul Salminen

In [2]:
```
#Import wanted packages
%matplotlib -- inline
import matplotlib.pyplot as plt
import numpy as np
import pyfits as pf
from sklearn.gaussian_process import GaussianProcess
```

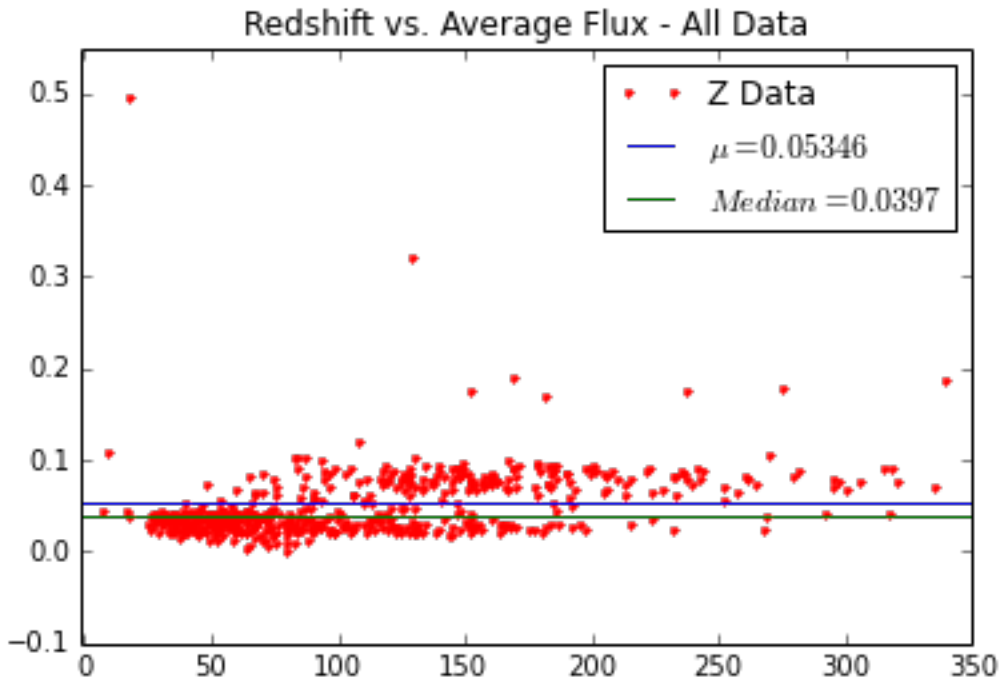
In [3]:
```
#Import Data from analysis done earlier
Zed = np.genfromtxt('zInfo.csv', delimiter=' ')
F = [y for x, y in Zed]
Z = [x for x, y in Zed]
dats = np.column_stack((F, Z))

#Find raw mean, median, and standard deviation
mu = np.mean(Z)
sig = np.std(Z)
med = np.median(Z)
```

1.2 Analysis of Raw data

This first graph is showing galaxy redshift versus the average flux. Over this data, in blue, is a line showing the average (mean) redshift for the data. We see that there That there are a few large outliers in this data, and that most lies around an area where flux is less that 100 and the Redshift is less than 0.05, and lies fairly close to the mean. Away from there, the data is more scattered.

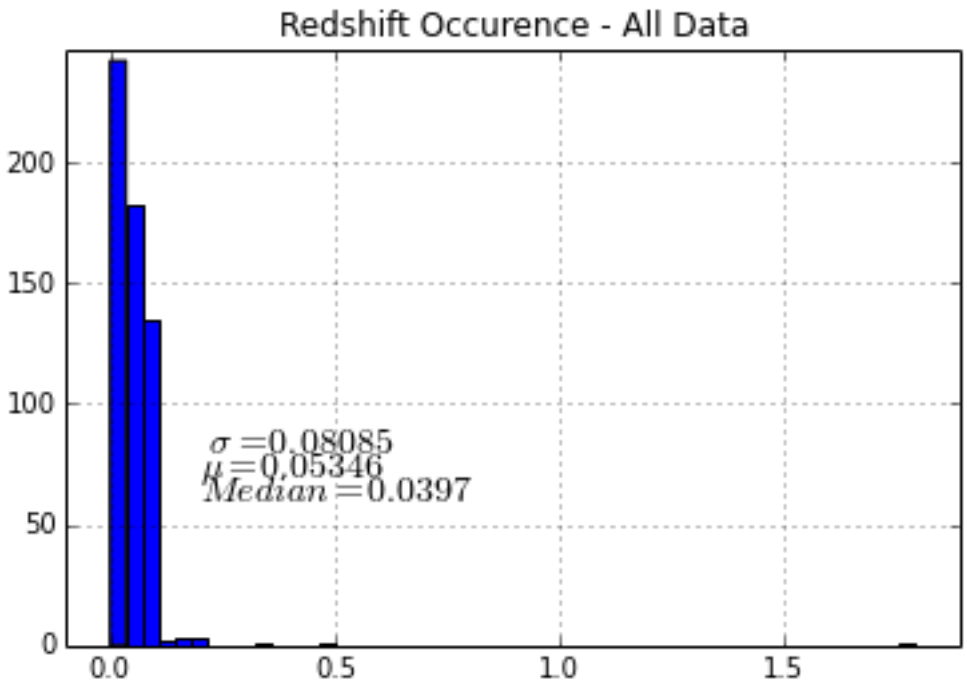
In [4]:
```
#Create mean and median line data
x = np.arange(-2, 350, 1)
y = np.zeros(x.shape[0])
z= np.zeros(x.shape[0])
y.fill(mu)
z.fill(med)

#Plot the Data
plt.plot(F, Z, 'r.', label='Z Data')
plt.plot(x, y, linewidth=1.0, label=r'$\mu= %s$' %(round(mu, 5)), color = 'b')
plt.plot(x, z, linewidth=1.0, label=r'$Median= %s$' %(round(med, 5)), color = 'g')
plt.legend()
plt.xlim(-2, 350)
plt.ylim(-.1, 0.55)
plt.title('Redshift vs. Average Flux - All Data')
```

```
plt.show()
plt.close()
```



In the histogram below, we see that there are a few very common ranges right around the mean. This representation also shows the bias of the outliers, and how must it skews the data.

```
In [6]: #Plot histogram of Z frequency
        n, bins, patches = plt.hist(Z, bins=50)
        plt.text(.2, 80, r'$\ \sigma=%s $' %(round(sig, 5)), fontsize=14)
        plt.text(.2, 70, r'$\mu=%s $' %(round(mu, 5)), fontsize=14)
        plt.text(.2, 60, r'$Median=%s $' %(round(med, 5)), fontsize=14)
        plt.grid(True)
        plt.legend()
        plt.ylim(0, max(n)+5)
        plt.xlim(-.1, max(bins)+0.1)
        plt.title('Redshift Occurence - All Data')
        plt.show()
        plt.close()
```



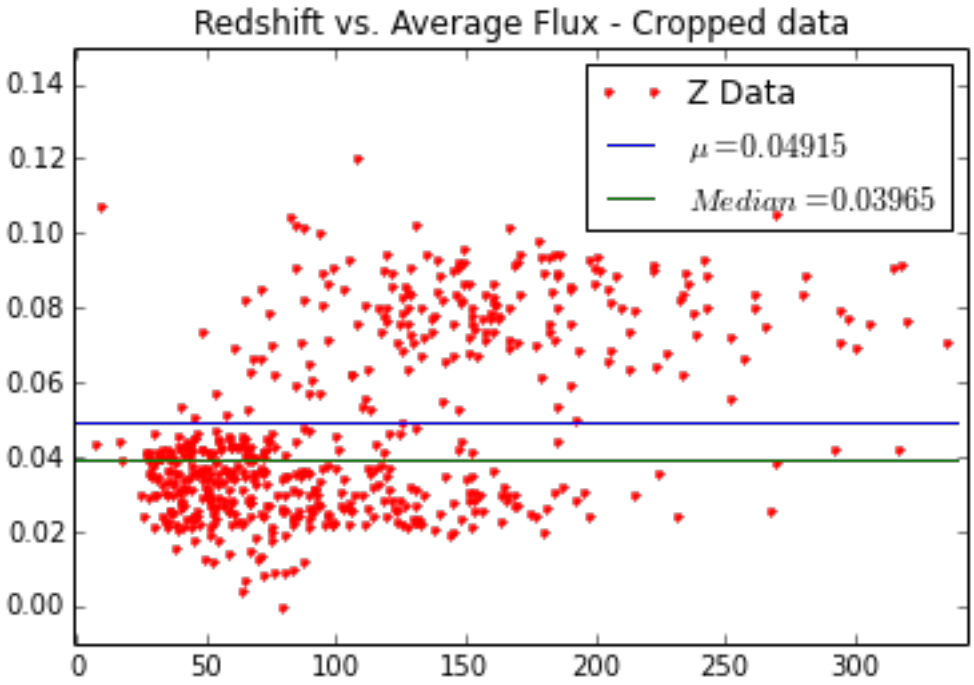
1.3 Data without the Outliers

```
In [7]: #Create new info for Z and F without the outliers
        finds = [q for q, w in enumerate(Z) if w<mu+3*sig]
        z_noOutliers = [f for j, f in enumerate(Z) if f<mu+3*sig]
        F_noOutliers = [j for i, j in enumerate(F) if i in finds]

        #Find new mean, median, and standard deviation
        mu_noOut = np.mean(z_noOutliers)
        med_noOut = np.median(z_noOutliers)
        sig_noOut = np.std(z_noOutliers)
        s_noOut = np.sqrt(np.var(z_noOutliers))
        print('We lost %s data points by erasing outliers' %(len(Z) - len(z_noOutliers)))
```

```
We lost 3 data points by erasing outliers
```

```
In [8]: #Create mean and median line data
        x = np.arange(-1, max(F_noOutliers)+5, 10)
        y = np.zeros(x.shape[0])
        y.fill(mu_noOut)
        z = np.zeros(x.shape[0])
        z.fill(med_noOut)

        #Plot the Data
        plt.plot(F_noOutliers, z_noOutliers, 'r.', label='Z Data')

        #Plot average lines
        plt.plot(x, y, linewidth=1.0, label=r'$\mu= %s$' %(round(mu_noOut, 5)), color = 'b')
        plt.plot(x, z, linewidth=1.0, label=r'$Median= %s$' %(round(med_noOut, 5)), color = 'g')
```

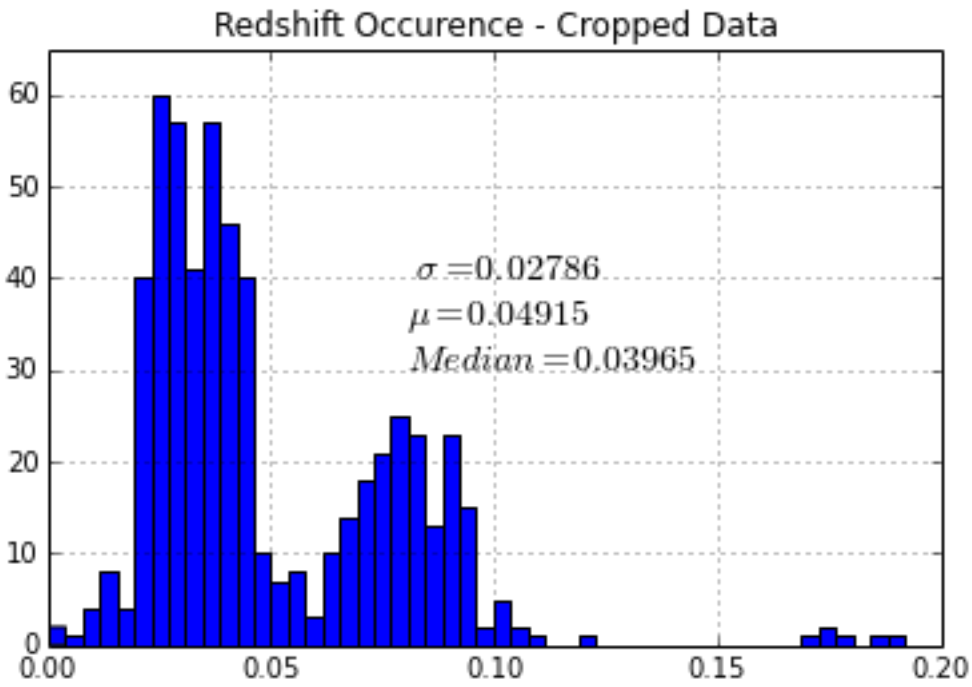
```
#Make it nice
plt.title('Redshift vs. Average Flux - Cropped data')
plt.ylim(-.01, 0.15)
plt.xlim(-1, max(F_noOutliers)+5)
plt.legend()
plt.show()
plt.close()
```



Due to the way this data was was calculated, there may be some possibility for outliers. Furthermore, looking at the discrepency between the median and the mean along with the standard deviation, it is clear there are outliers way out from the norm.

This new image gives us a much better idea of what our data actually looks like.

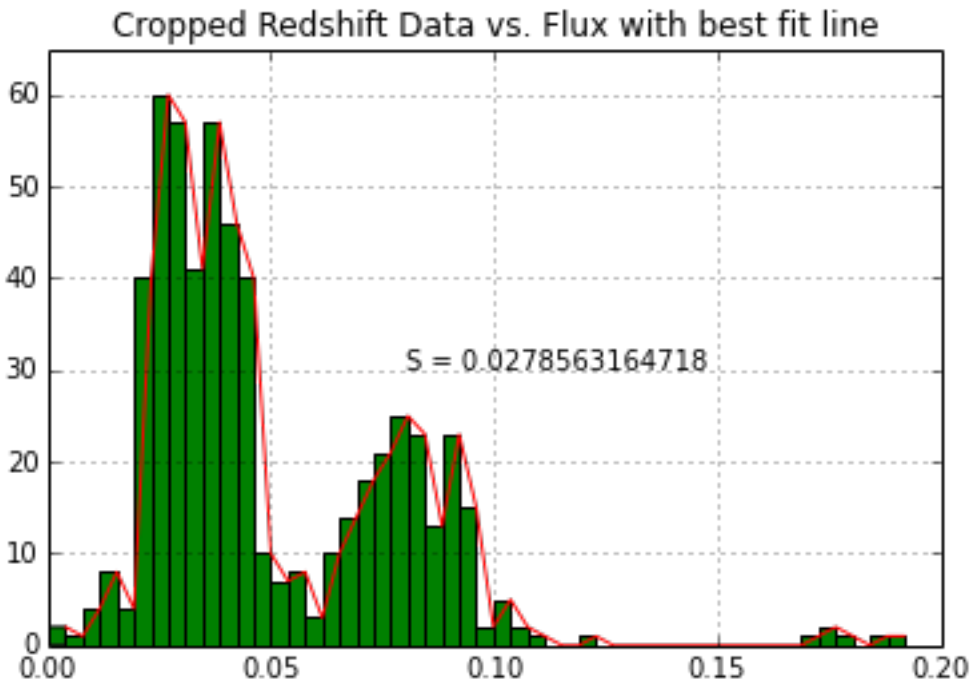
In [10]:
```
#Plot a histogram of the cropped data
n, bins, patches = plt.hist(z_noOutliers, bins=50)
plt.legend()
plt.xlabel=('Redshift')
plt.ylabel =('Frequency')
plt.title('Redshift Occurence - Cropped Data ')
plt.text(.08, 40, r'$\ \sigma=%s $' %(round(sig_noOut, 5)), fontsize=14)
plt.text(.08, 35, r'$\mu=%s $' %(round(mu_noOut, 5)), fontsize=14)
plt.text(.08, 30, r'$Median=%s $' %(round(med_noOut, 5)), fontsize=14)
plt.ylim(0, max(n)+5)
plt.grid(True)
plt.show()
plt.close()
```



The new graph shows us a very different, much more detailed image of the data.

Well, this data looks pretty good. It seems that there are two, somewhat gaussian peaks. One at around F=0.03, and the other around 0.08.

```
In [11]: #Make best fit line for data, Gaussian Process from Sci-kit Learn
         fitLine = GaussianProcess(np.vstack([bins[1:,], n]))

         #Graph data with line
         n, binsNew, patches = plt.hist(z_noOutliers, bins=50, facecolor='g')
         plt.plot(fitLine.regr[0], fitLine.regr[1], color='r')
         plt.text(.08, 30, 'S = %s' %(s_noOut))
         plt.grid(True)
         plt.ylim(0, max(n)+5)
         plt.title('Cropped Redshift Data vs. Flux with best fit line')
         plt.savefig('RedshiftvsFluxBestFit.png')
```



Based on this evidence, there appears to be two different clumps of data

1.4 Find Distance

```
In [12]: Dist = np.array([x*(3e5/71) for y, x in enumerate(z_noOutliers)])
         print("Mean Distance: %s Mpc" %(np.average(Dist)))
         print("Median Distance: %s Mpc" %(np.median(Dist)))
```

```
Mean Distance: 207.662862928 Mpc
Median Distance: 167.539941601 Mpc
```