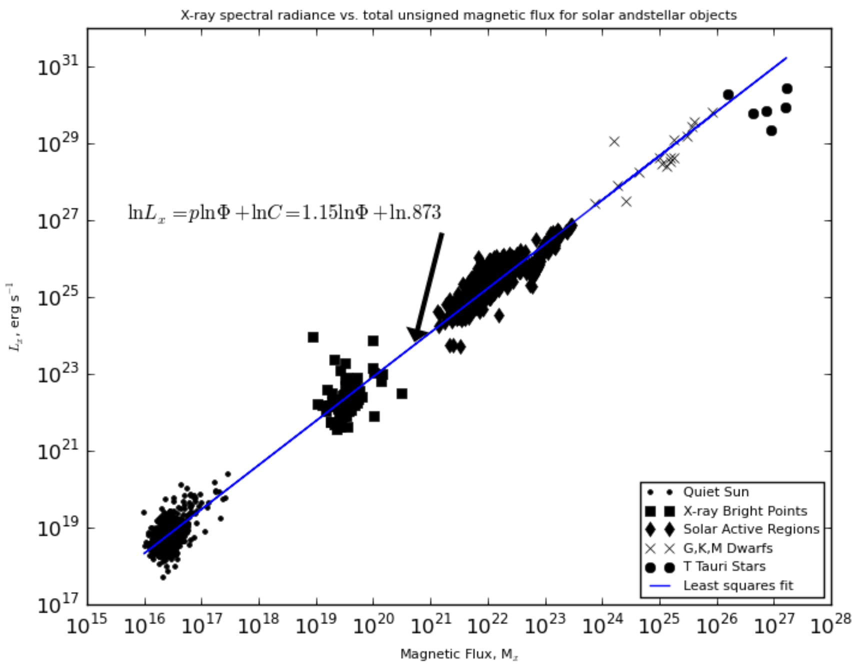
Problem Set 3: Fitting Solar/Stellar data without error



1.

2. $L_x = C * \Phi^p \Rightarrow \ln(L_x) = p\ln(\Phi) + \ln(C)$
   For n unknowns, the system of equations, $A\mathbf{x} = \mathbf{b}$ holds, where for an unknown n $\mathbf{b_n}$ is $L_{x,n}$, x_1 is p, x_2 is $\ln(C)$, A_{n1} is $\ln\Phi$ and A_{n2} is 1.

3. The `mflux_lx_all.txt` file is 1316 lines long, so $n = 1316$. A is 2 by n, $\mathbf{x}$ is 1 by 2, and $\mathbf{b}$ is 1 by n. The system of equations looks like,

$$\begin{pmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{21} & \mathrm{A}_{22} \\ \vdots & \vdots \\ \mathrm{A}_{n1} & \mathrm{A}_{n2} \end{pmatrix} \begin{pmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \end{pmatrix} = \begin{pmatrix} \mathrm{b}_1 \\ \mathrm{b}_2 \\ \vdots \\ \mathrm{b}_n \end{pmatrix}$$

4.

$$\mathrm{Ax{=}b}$$
$$\mathrm{A}^T A\mathbf{x} = A^T\mathbf{b}$$
$$\mathrm{x{=}(A}^T A)^{-1}A^T\mathbf{b}$$

The value of $\mathbf{x}$ found above minimizes $(A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b})$ because it is the solution to the equation $(A\mathbf{x} - \mathbf{b}) = 0$ and $(A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b})$ is effectively "$(A\mathbf{x} - \mathbf{b})^2$", so with $\mathbf{x}$ as above $(A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b}) = 0$, which is the minimum value for a real squared number.

5. Solving for x yields: $p = 1.1492$, $C = 8.7255 \cdot 10^{-1}$
My p does agree with Pevtsov et al. (2003).

6. Overplotted in the figure in question 1. The mean absolute error in logspace of the fit is .546. If the mean absolute error was calculated in non-logspace, it would be huge, because the linear fit optimizes the fit to have approximately the same error for all points, and the scale of the error for the points with higher M_x would suddenly be much greater than that of points with lower M_x.

Code involved in this assignment:

```
#!/usr/bin/python3.4

import numpy as np
from numpy import linalg as la
from matplotlib import pyplot as plt
from copy import deepcopy

###PLOT###GENERATION#############################################################

#Reading in data and obtaining indices of different objects
data = np.genfromtxt('mflux_lx_all.txt').transpose() #Col 1: Mx, Col2: Lx
qsun_inds = np.where(data[0] < 1e18) # Quiet Sun
XBP_inds = np.where((data[0] > 1e18) & (data[0] < 1e21)) # Xray bright pts
asun_inds = np.where((data[0] > 1e21) & (data[1] < 1e27)) # Active Sun
dwf_inds =  np.where((data[1] > 1e27) & (data[0] < 1e26)) # G,K,M dwarfs
TT_inds =  np.where(data[0] > 1e26) # T Tauri stars


#Plotting everything as in Fig1 of Pevtsov et al 2003
fig = plt.figure()  ;  ax = fig.add_subplot(1,1,1)
ax.loglog(data[0][qsun_inds],data[1][qsun_inds],'k.',label='Quiet Sun') # points
ax.loglog(data[0][XBP_inds],data[1][XBP_inds],'ks',label='X-ray Bright Points') # squares
ax.loglog(data[0][asun_inds],data[1][asun_inds],'kd',label='Solar Active Regions') # diamonds
ax.loglog(data[0][dwf_inds],data[1][dwf_inds],'kx',label='G,K,M Dwarfs') # Xs
ax.loglog(data[0][TT_inds],data[1][TT_inds],'ko',label='T Tauri Stars') # circles

#ax.legend(loc='lower right', fontsize= 8)
#fig.savefig('hw3_fig.png')
#plt.show()


##FITTING##COEFFICIENTS##########################################################
A = np.array([np.log(data[0]),np.ones(len(data[0]))])
```

```
b = deepcopy(np.log(data[1]))

x = np.linalg.solve(np.dot(A,A.transpose()), np.dot(b,A.transpose()))
p = x[0]   ;   C = np.exp(x[1])
print('p = '+ str(p))
print('C = '+ str(C))

fit = np.dot(x,A)
avg_abs_err = np.mean(np.abs(b - fit))
print('Mean absolute error of fit is: '+str(avg_abs_err))


##PLOTTING##OUT##THE##FINAL##PLOT
ax.loglog(data[0],np.exp(fit),label='Least squares fit')

ax.set_xlabel('Magnetic Flux, M$_x$',fontsize=8)
ax.set_ylabel('$L_x$, erg s$^{-1}$', fontsize=8)
ax.set_title('X-ray spectral radiance vs. total unsigned magnetic flux for solar andstellar objects', fontsize=8)
ax.legend(loc='lower right', fontsize= 8)
ax.annotate('$\ln{L_x} = p\ln{\Phi}+\ln{C} = 1.15\ln{\Phi}+\ln{.873}$',xy=(5e20,5e23),xytext=(5e15,1e27), arrowprops=dict(facecolor='black',shrink=.05,width=2))
fig.savefig('hw3_fig.png')
plt.show()
```