trial1

January 15, 2024

[ ]:
```
from astropy.timeseries import LombScargle
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[ ]:
```
data = pd.read_csv('./data/sksolartimevariation5804d.txt', skiprows=13,␣
↪sep='\s+', names=['t_mean(s)', 't_mean-t_start(s)', 't_end-t_mean(s)',␣
↪'nu_flux(1e6cm-2s-1)', 'flux_up_error(1e6cm-2s-1)',␣
↪'flux_down_error(1e6cm-2s-1)'])
```

[ ]:
```
data
```

[ ]:

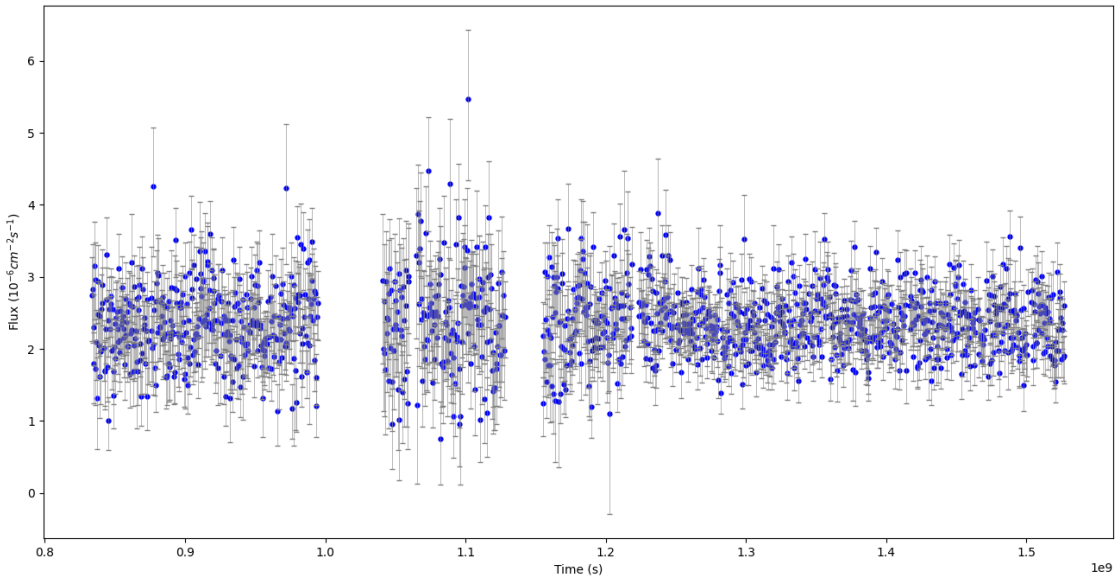
	t_mean(s)	t_mean-t_start(s)	t_end-t_mean(s)	nu_flux(1e6cm-2s-1)	\
0	833654760	170100	277380	2.74	
1	834127080	175500	210060	2.83	
2	834550800	213180	230160	2.30	
3	834997020	199380	212640	1.79	
4	835380420	170520	265680	3.15	
…	…	…	…	…	
1338	1525315550	172739	172774	2.36	
1339	1525703838	215064	215054	2.26	
1340	1526138206	216970	216028	1.88	
1341	1526588224	232102	226109	1.90	
1342	1527014775	199299	208324	2.60	

	flux_up_error(1e6cm-2s-1)	flux_down_error(1e6cm-2s-1)
0	0.63	0.53
1	0.75	0.62
2	0.53	0.45
3	0.55	0.44
4	0.74	0.61
…	…	…
1338	0.36	0.33
1339	0.31	0.29
1340	0.33	0.29
1341	0.38	0.28
1342	0.35	0.33

```
[1343 rows x 6 columns]
```

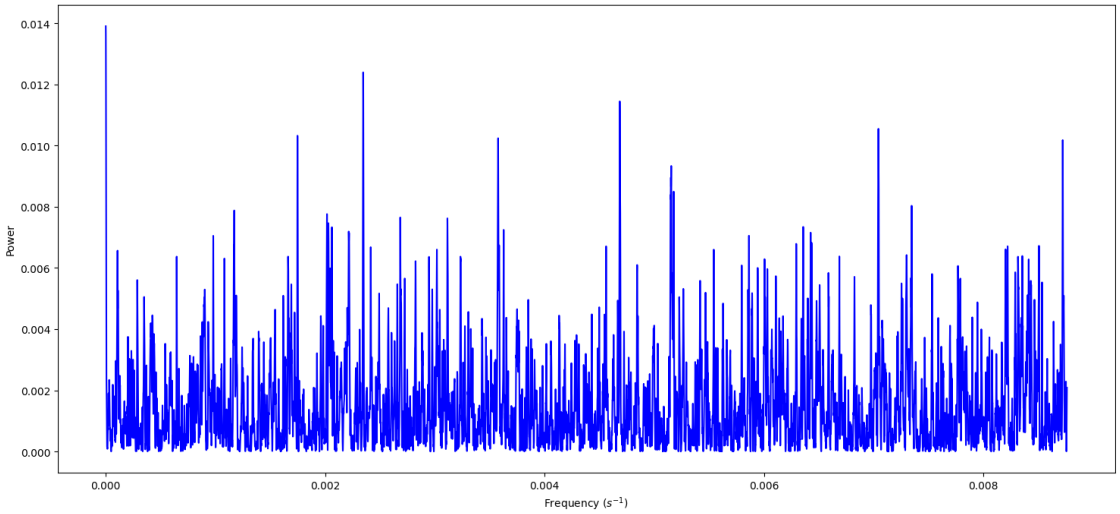
[ ]:
```
plt.figure(figsize=(16, 8))


plt.errorbar(data['t_mean(s)'], data['nu_flux(1e6cm-2s-1)'], yerr=np.
 ↪array(list(zip(data['flux_up_error(1e6cm-2s-1)'].values,␣
 ↪data['flux_down_error(1e6cm-2s-1)'].values))).T, fmt='none', ecolor='gray',␣
 ↪elinewidth=0.4, capsize=2)
# plt.hlines([data['nu_flux(1e6cm-2s-1)'].mean(), data['nu_flux(1e6cm-2s-1)'].
 ↪mean() + data['nu_flux(1e6cm-2s-1)'].std(), data['nu_flux(1e6cm-2s-1)'].
 ↪mean() - data['nu_flux(1e6cm-2s-1)'].std()], xmin=data['t_mean(s)'].min(),␣
 ↪xmax=max(data['t_mean(s)']), color='red', linestyles='dashed', label='mean␣
 ↪flux')
plt.scatter(data['t_mean(s)'], data['nu_flux(1e6cm-2s-1)'], color='blue',␣
 ↪marker='.', s=50)
plt.xlabel('Time (s)')
plt.ylabel('Flux ($10^{-6}cm^{-2}s^{-1}$)')
plt.show()
```



[ ]:
```
lsp = LombScargle(data['t_mean-t_start(s)'].values, data['nu_flux(1e6cm-2s-1)'].
 ↪values, (data['flux_up_error(1e6cm-2s-1)'].values +␣
 ↪data['flux_down_error(1e6cm-2s-1)'].values)/2)
freql, powerl = lsp.autopower(samples_per_peak=10, nyquist_factor=6)
```

[ ]:
```
plt.figure(figsize=(18, 8))
plt.plot(freql, powerl, color='blue', ls='-')
plt.xlabel('Frequency ($s^{-1}$)')
plt.ylabel('Power')
plt.show()
```



[ ]:
```
fapr = lsp.false_alarm_probability(powerl.max(), method='bootstrap')
```

[ ]:
```
fapr
```

[ ]: 0.351

[ ]: