AstroBytes, Data Analytics Report, NSSC 2021

Rohan Kumar[1,*] and Ankush Kumar Singh[2,!]

[1] *IISER, Kolkata*
[2] *IIT(ISM), Dhanbad*
[*] *Corresponding author: Rohan Kumar, rohankumarprasad@yahoo.com*
[!] *Team Leader: Ankush Kumar Singh, 20je0155@fme.iitism.ac.in*

December 13, 2021

Problem 1

To begin the analysis we will first import the standard library *Lightkurve*.

```
import lightkurve as lk
```

We can now download the lightcurve data for *Kepler-17*[1] and plot it.

```
lc_KIC=lk.search_lightcurve("KIC 10619192")[38].download()
lc_KIC.plot()
```

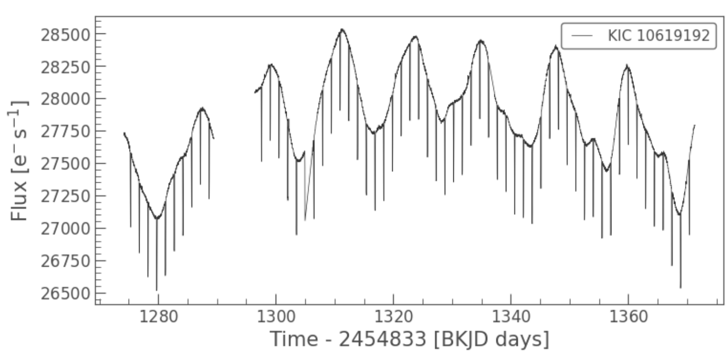


Figure 1: Kepler-17 Lightcurve

[1] TIC 273874849, KIC 10619192

To make the lightcurve usable we will remove the the outliers and NAN-values and plot the clean lightcurve.

```
lc_KIC=lc_KIC.remove_outliers()
lc_KIC=lc_KIC.remove_nans()
lc_KIC.plot()
```

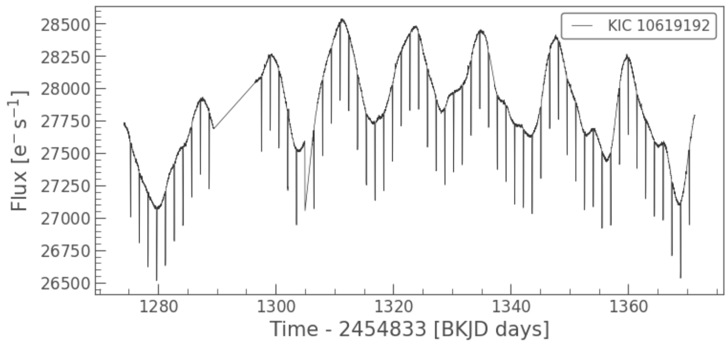


Figure 2: Clean lightcurve

To remove the long term trends we will faltten[2] out the lightcurve, doing this will in a sense magnify the local trends in which we are interested in for exo-planets study.

```
flat_lc=lc_KIC.flatten()
flat_lc.plot()
```

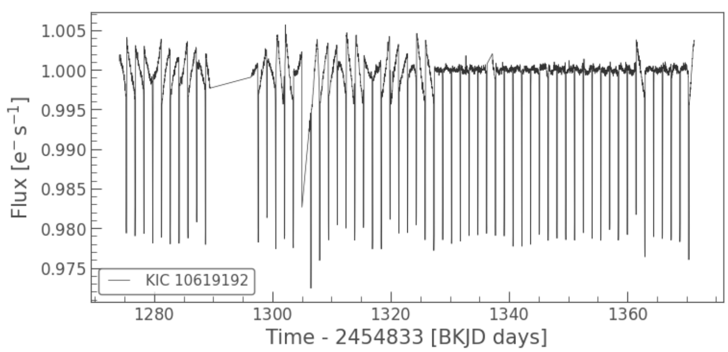


Figure 3: Flat Lightcurve

[2]normalize

We can now convert the `flat_lc` to an astropy table and store the flux values in new variable and calculate the mean and standard deviation using *NumPy*. For CDPP[3] we will use the `lightkurve.estimate_cdpp()`.

```
flux=flat_lc.to_table()['flux']
import numpy as np
std=np.std(flux)
mean=np.mean(flux)
cdpp=flat_lc.estimate_cdpp()
```

$$\boxed{\sigma = 0.0045\ es^{-1}}$$

$$\boxed{\text{mean flux} = \bar{x} = 0.9989\ es^{-1}}$$

$$\boxed{\text{CDPP} = 1271.0180 \text{ ppm}}$$

Problem 2

We will now calculate and plot the BLS[4] periodogram of the `flat_lc`,

```
bls=flat_lc.to_periodogram(method='bls')
bls.plot()
```

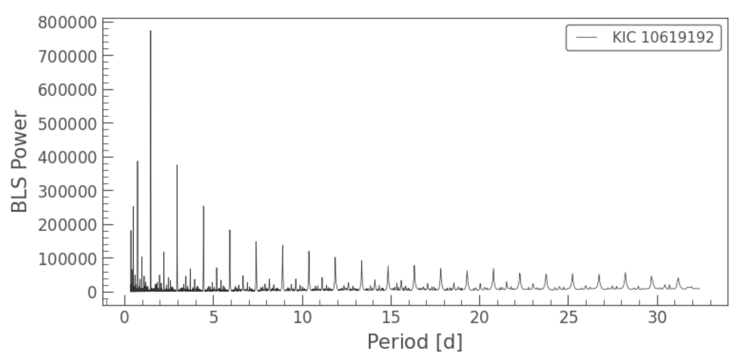


Figure 4: BLS Periodogram

We know that BLS Periodogram is a power spectrum which shows the dominant periods in the time series data. So we can use this fact to extract the transit duration and time period of revolution.

```
planet_period = bls.period_at_max_power
planet_dur = bls.duration_at_max_power
```

$$\boxed{\text{Transit Duration} = 0.1 \text{ d}}$$

$$\boxed{\text{Time Period} = 1.4857 \text{ d}}$$

[3]Combined Differential Photometric Precision
[4]Box Least Square

To verify whether the time period obtained above is correct or not we will fold the light curve with stated period and plot the folded lightcurve,

```
folded_lc = flat_lc.fold(period=1.4856943)
folded_lc.plot()
```

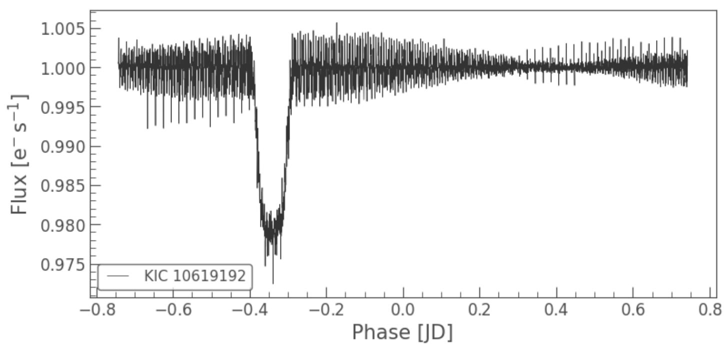


Figure 5: Folded Lightcurve with Period=1.4857 d

We can see from the above figure a distinct dip around -0.4 which means we have correctly determined the period of the planet.

We can now make a model curve by using the BLS[5] and giving it the same time period and transit time to get a model lightcurve.

```
planet_t0 = bls.transit_time_at_max_power
planet_model = bls.get_transit_model(period=planet_period,
    transit_time=planet_t0,duration=planet_dur)
ax = flat_lc.fold(planet_period, planet_t0).scatter()
planet_model.fold(planet_period, planet_t0).plot(ax=ax, c='r', lw
    =1)
```

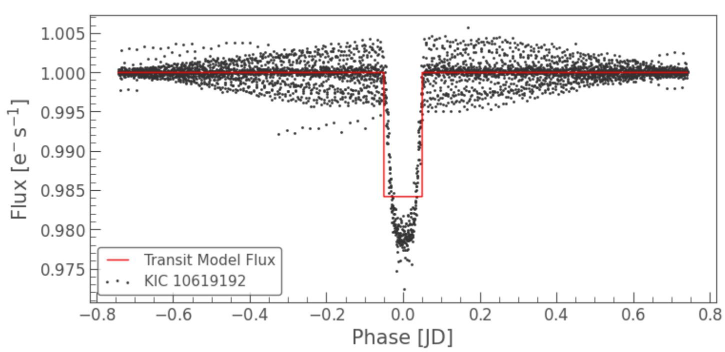


Figure 6: Model Lightcurve

[5] Box-fitting Least Squares

To calculate the relative flux dip($\frac{\Delta F}{F}$),

```
model_flux=planet_model.fold(planet_period, planet_t0)['flux']
min=np.min(model_flux)
max=np.max(model_flux)
rfd=(max-min)/max
```

$$\boxed{\text{Relative Flux Dip} = \frac{\Delta F}{F} = 0.015807545862905243 \approx 0.0158}$$

Problem 3

To start the analysis of the star *Kepler-21*[6] in quarter 2,5,6,7 we will first download the short cadence lightcurve data and plot the lightcurve,

```
search_result = lk.search_lightcurve("Kepler-21", author="Kepler",
    cadence="short",quarter=[2,5,6,7])
search_result
lc_collection=search_result.download_all()
for i in range(len(search_result)):
    lc_collection[i].plot()
```

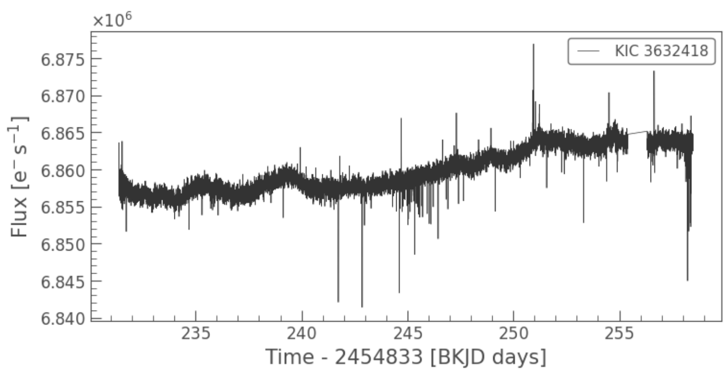
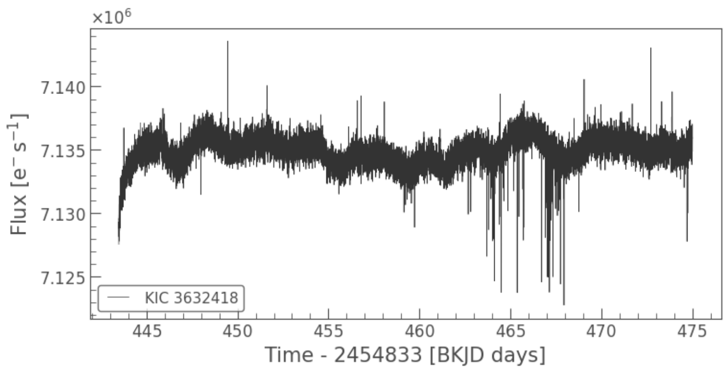
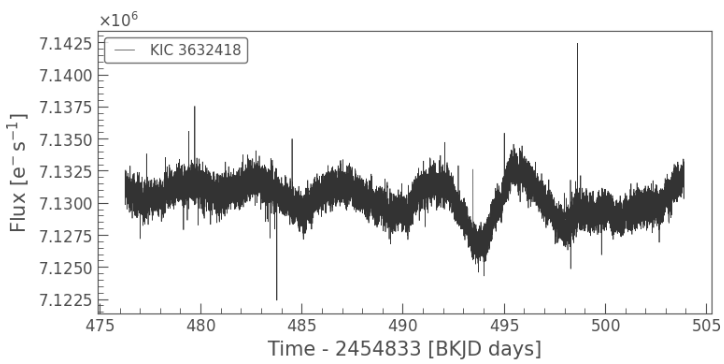


Figure 7: Quarter 2 Time[BKJD] 231-258

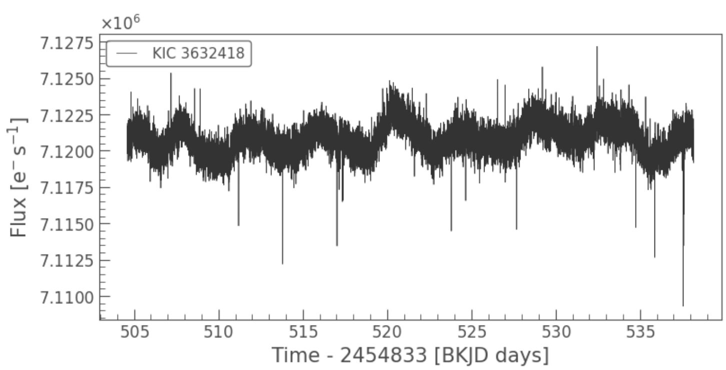
[6]KIC 3632418, TIC 121214185



(a)

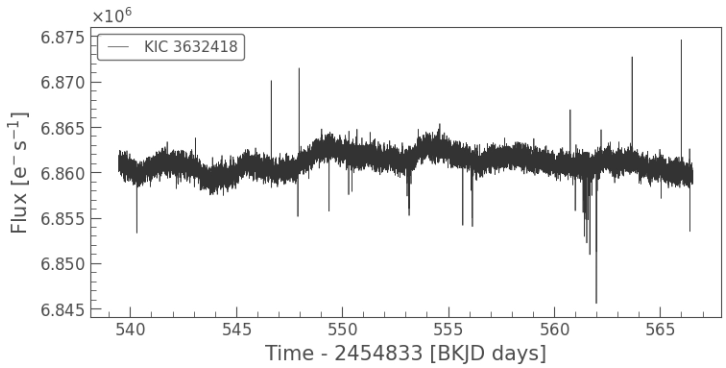


(b)

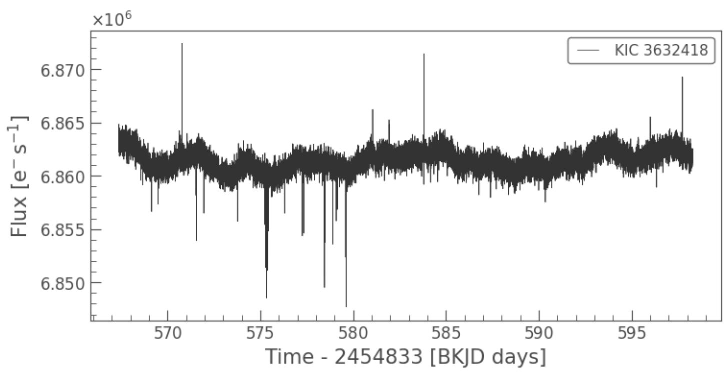


(c)

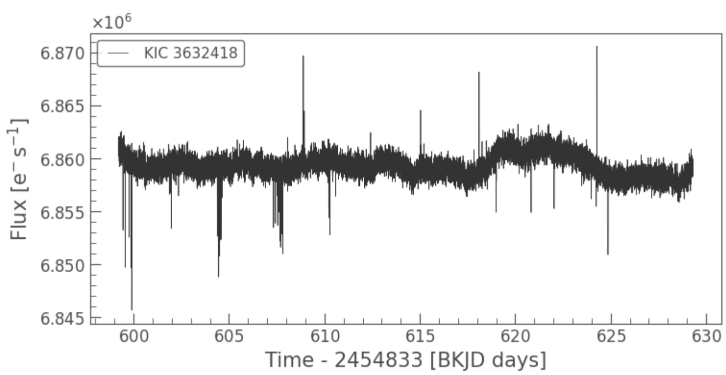
Figure 8: Quarter 5 Time[BKJD] (a)443-475, (b)476-503, (c)504-538



(a)

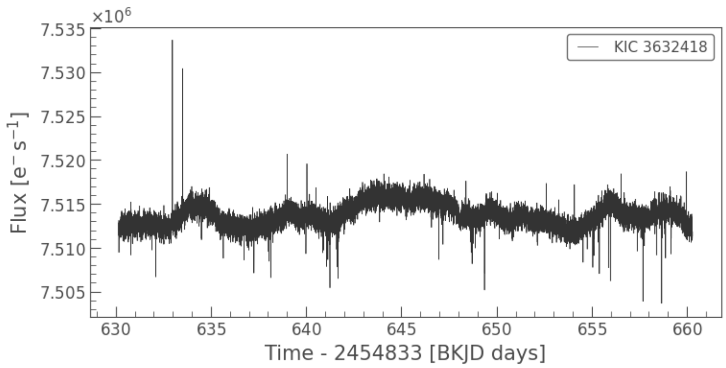


(b)

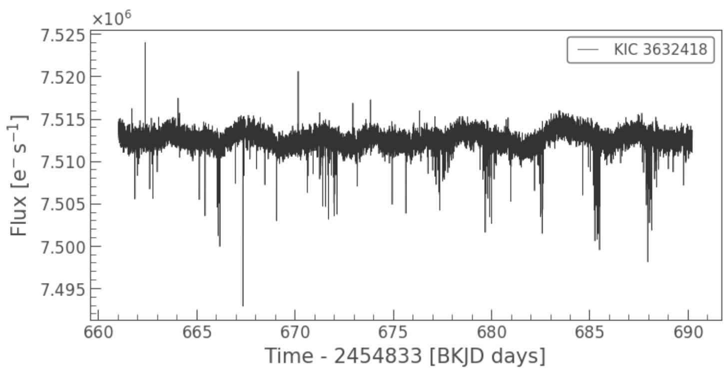


(c)

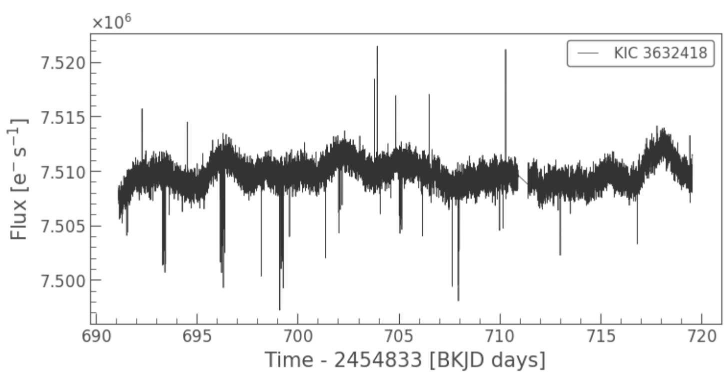
Figure 9: Quarter 6 Time[BKJD] (a)539-566, (b)567-598, (c)599-629



(a)



(b)



(c)

Figure 10: Quarter 7 Time[BKJD] (a)630-660, (b)661-690, (c)691-719

Now we will stitch the lightcurve of different quarter to get the long term behaviour of the star over all the quarters.

```
lc_stitched = lc_collection.stitch()
lc_stitched.plot()
```

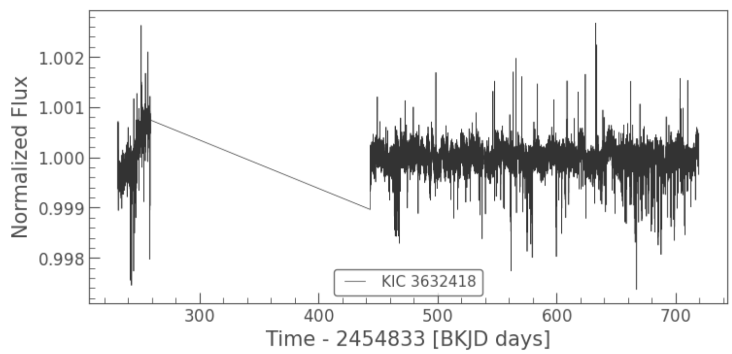


Figure 11: Stitched lightcurve over quarters 2,5,6,7

Now we will remove the outliers and the NAN-values from the composite lightcurve to make the result more accurate.

```
clean_lc=lc_stitched.remove_outliers()
clean_lc=lc_stitched.remove_nans()
clean_lc.plot()
```

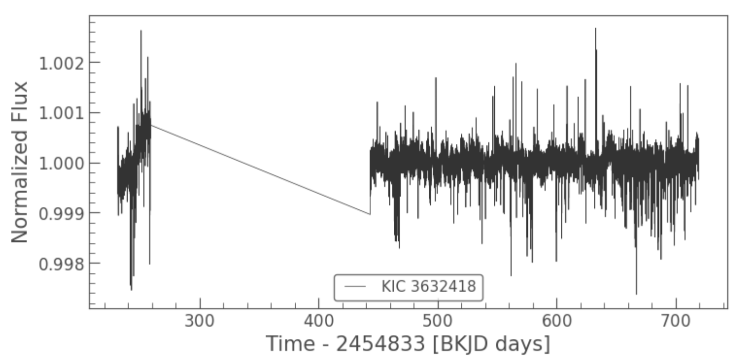


Figure 12: Clean lightcurve

As we are interested in the short term details of this composite lightcurve[7] we can flatten the lightcurve to remove the long term behaviour of the star and plot the flattened lightcurve.

[7]Here we are interested in the stellar osscilations which happens at very high frequencies compared to the long the term behaviour we can see here.

```
lc_stitched_flat=clean_lc.flatten()
lc_stitched_flat.plot()
```

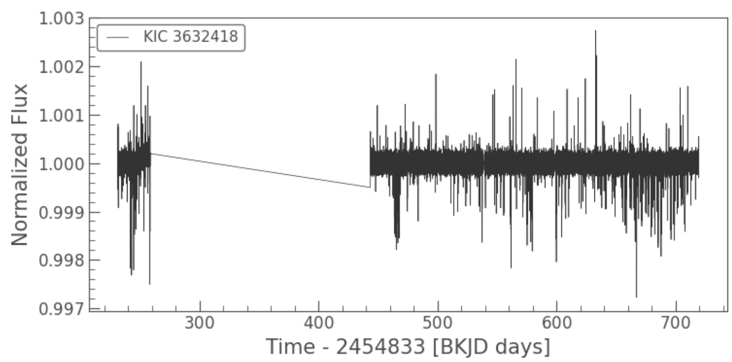


Figure 13: Flattened lightcurve

We will now plot the periodogram to see the data in frequency domain.

```
pg=lc_stitched_flat.to_periodogram(normalization='psd')
pg.plot(scale='log')
```

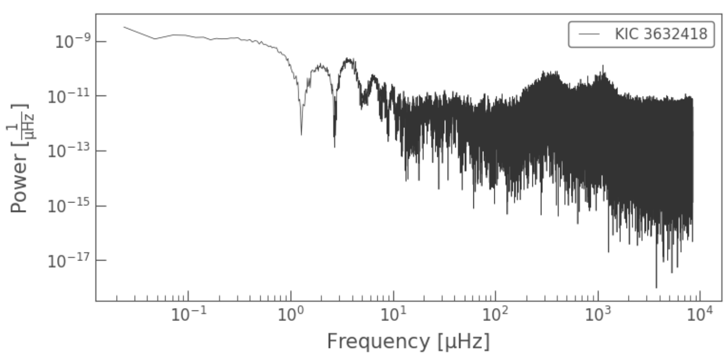


Figure 14: Periodogram in frequency domain

From the given document[8] we know that when we draw the periodogram of a solar-type oscillator star, we get a peak around 10^3 μHz. Therefore we zoomed into the region within the range of 700-1500 μHz and found a peak around 10^3 μHz as mentioned above.

```
pg=lc_stitched_flat.to_periodogram(normalization='psd',
    minimum_frequency=700,maximum_frequency=1500)
pg.plot()
```

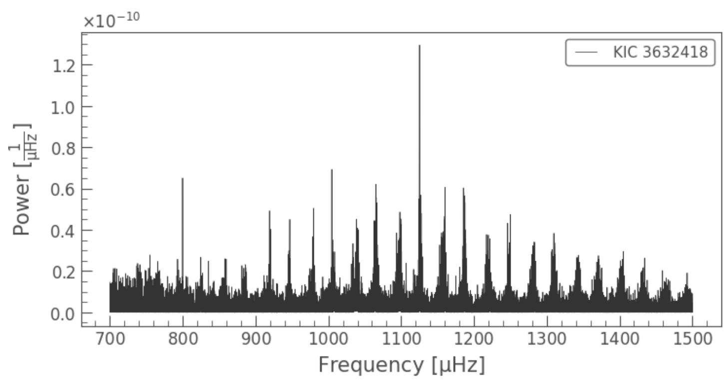


Figure 15: Periodogram zoomed near power excess region around $10^3 \mu$Hz

As the above plot is very noisy thus to make it usable we will smooth it out using `scipy.signal.medfilt()`,

```
freq=pg.to_table()['frequency']
pw=pg.to_table()['power']
from scipy.signal import medfilt
smooth=medfilt(pw)
fig =plt.figure(figsize=(12.5,7),facecolor='white')
plt.plot(freq,smooth,'k')
plt.xlabel(r'Frequency [$\mu$Hz]',fontsize=20)
plt.ylabel(r'Power [$\frac{1}{\mu Hz}$]',fontsize=20)
plt.legend(['KIC 3632418'])
plt.show()
```

[8] Astroseismology

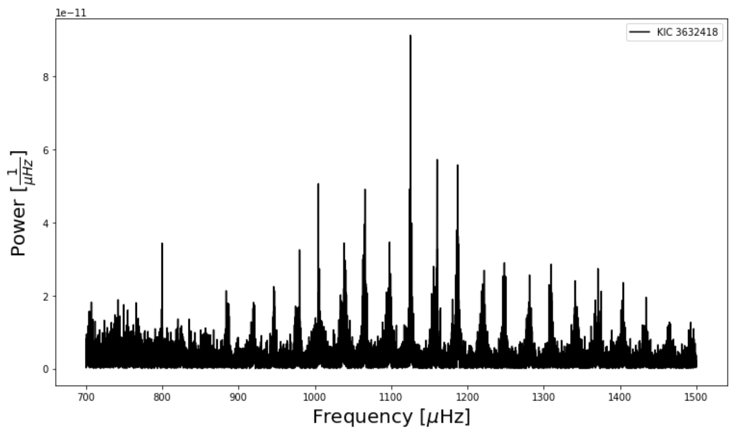


Figure 16: Smoothed out Periodogram

For the *Gaussian* enevlope we have taken the distinct peaks with power greater than $4 \times 10^{-11} Hz^{-1}$ were selected and stored in an array which was then used for the *Gaussian* fitting to get the envelope, for the fitting process we used `lmfit.Model()`.

```
x=[]
y=[]
for i in range(len(freq)):
    if (smooth[i]>4e-11):
        x.append(freq[i])
        y.append(smooth[i])

from lmfit import Model

def gaussian(x, amp, cen, wid):
    "1-d gaussian: gaussian(x, amp, cen, wid)"
    return (amp/(np.sqrt(2*np.pi)*wid)) * np.exp(-(x-cen)**2 /(2*
    wid**2))

gmod = Model(gaussian)

result = gmod.fit(y, x=x, amp=9e-11, cen=1150, wid=1000)
```

We then use `print(result.fit_report())` to get the parameters for the *Gaussian* envelope,

```

[[Model]]
    Model(gaussian)
[[Fit Statistics]]
    # fitting method   = leastsq
    # function evals   = 123
    # data points      = 13
    # variables        = 3
    chi-square         = 2.5757e-21
    reduced chi-square = 2.5757e-22
    Akaike info crit   = -643.650708
    Bayesian info crit = -641.955860
[[Variables]]
    amp:  2.5869e-08 +/- 1.6467e-08 (63.65%) (init = 9e-11)
    cen:  1125.47295 +/- 56.1484650 (4.99%) (init = 1150)
    wid:  168.600608 +/- 116.586197 (69.15%) (init = 1000)
[[Correlations]] (unreported correlations are < 0.100)
    C(amp, wid) = 0.993
    C(amp, cen) = 0.680
    C(cen, wid) = 0.674
```

We can now use the calculated parameters of the *Gaussian* envelope to plot it.

```
gauss=gaussian(freq,2.5869e-08,1125.47295,168.600608)
fig =plt.figure(figsize=(12.5,7),facecolor='white')
plt.plot(freq,smooth,'k')
plt.plot(freq,gauss,'r')
plt.xlabel(r'Frequency [$\mu$Hz]',fontsize=20)
plt.ylabel(r'Power [$\frac{1}{\mu Hz}$]',fontsize=20)
plt.legend(['KIC 3632418','Gaussian Fit'])
plt.show()
```

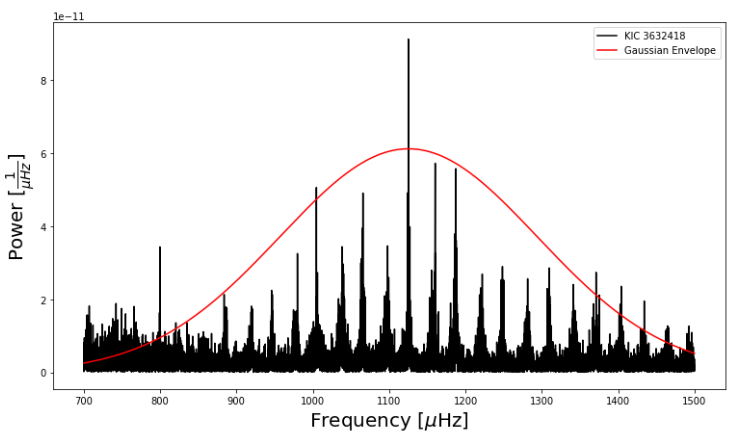


Figure 17: Periodogram with *Gaussian* envelope

From the *Gaussian* envelope parameter we can get the ν_{max} and for the $\Delta\nu$ we can use the following code.

```
seismology = pg.flatten().to_seismology() #using seismology
    function of Lightkurve library
seismology.estimate_numax(method='acf2d')
seismology.estimate_deltanu()
```

$$\boxed{\nu_{\text{max}} = 1125.2730\ \mu Hz}$$

$$\boxed{\Delta\nu = 60.72\ \mu Hz}$$

Now we will use above calculated values to get the radius(R) and mass(M) of *Kepler-21* using the following expressions,

$$\frac{M}{M_{\odot}} \simeq \left(\frac{\nu_{max}}{\nu_{max,\odot}}\right)^{3} \left(\frac{\Delta\nu}{\Delta\nu_{\odot}}\right)^{-4} \left(\frac{T_{eff}}{T_{eff,\odot}}\right)^{1.5}$$

$$\frac{R}{R_{\odot}} \simeq \left(\frac{\nu_{max}}{\nu_{max,\odot}}\right) \left(\frac{\Delta\nu}{\Delta\nu_{\odot}}\right)^{-2} \left(\frac{T_{eff}}{T_{eff,\odot}}\right)^{0.5}$$

where $\nu_{\text{max},\odot} = 3090\mu Hz$, $\Delta\nu_{\text{max},\odot} = 135.1\mu Hz$, $T_{\text{eff},\odot} = 5777.2K$, temperature of the star $T_{\text{eff}} = 6200K$,

```
M = ((1125.273/3090)**3)*((60.72/135.1)**(-4))*((6200/5777.2)
    **(3/2))
R = (1125.273/3090)*((60.72/135.1)**(-2))*((6200/5777.2)**(1/2))
```

$$\boxed{R = 1.8676 R_{\odot}}$$

$$\boxed{M = 1.3158 M_{\odot}}$$

Problem 4

We will first import the standard libraries,

```
import numpy as np
import astropy.constants as const
```

To calculate the radius of *Kepler-17 b* we will use,

$$\frac{\Delta F}{F} = \left(\frac{R_p}{R_s}\right)^2 \tag{1}$$

We are given the value of $R_s = 1.01R_\odot$ and we already calculated the value of $\frac{\Delta F}{F}$ here, we can now calulate R_p which is the radius of *Kepler-17 b*.[9]

```
R_sun = const.R_sun
R_s = 1.01*R_sun   #converting radius of star given in solar radius
     to metre
delF_F = 0.015807545862905243
R_p = (R_s)*np.sqrt(delF_F)
```

$$\boxed{R_p = 1.2357R_J = 13.8511R_\oplus = 88343703.6673 \text{ m}}$$

Since the time period of the planet and mass of star is known, we can use Kepler's third law to compute the orbital radius of the planet.

$$T^2 = \frac{4\pi^2 a^3}{G(M_s + M_p)}$$

We can make the following assumptions $a = R_{\text{orbit}}$[10] and $M_s + M_p \approx M_s$[11] we already have the value of $M_s = 1.04M_\odot$, these assumptions simplifies the above equation.

$$T^2 = \frac{4\pi^2 R_{orbit}^3}{G(M_s)}$$

```
M_s = 1.04*const.M_sun        #converting mass of star given in
    solar mass to kg
T = 1.4856943*24*3600
R_cube = (const.G*M_s*(T**2))/(4*(np.pi**2))
R_orbit = R_cube**(1/3)
```

$$\boxed{R_{orbit} = 0.0258 \text{ AU} = 3862103236.9458 \text{ m}}$$

[9] Here $\odot$, $\oplus$ and J stands for solar, earth and jupiter parameters respectively.
[10] It's given in the problem statement to assume that the orbit is circular.
[11] Here we are talking about a star planet binary system which means that $M_s >> M_p$.

Mass of *Kepler-17 b* can be found using conservation of angular momentum.

$$M_s V_s = M_p V_p$$

We already have the values of $M_s = 1.04 M_{\odot}$ and $V_s = 0.228$ km/s, V_p can be calculated by,

$$V_p = \frac{D}{T}$$

where D is the circumference of the orbit $2\pi R_{orbit}$ and T is the time period,

$$V_p = \frac{2\pi R_{orbit}}{T}$$

This means the expression for M_p becomes.

$$M_p = \frac{M_s V_s T}{2\pi R_{orbit}}$$

```
V_s = 0.228*1000              #conversion to m/s
M_p = (M_s*V_s*T)/(2*np.pi*R_orbit)
vol_planet = (4*np.pi*(R_p**3))/3
D = M_p/vol_planet
```

$$\boxed{M_p = 1.3137 M_J = 417.6202 M_{\oplus} = 2.4941 \times 10^{27} kg}$$

$$\boxed{\rho = 863.5712 \text{ kg } m^{-3}}$$

Now we will do the classification of *Kepler-17 b* according to the above calculated parameters.

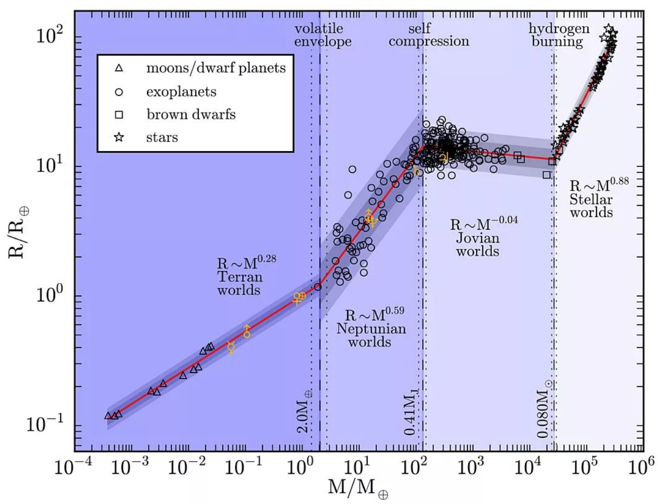


Figure 18: Classification of Exo-Planets[12]

Coordinate of *Kepler-17 b* in this plot will be (M_p, R_p) and this comes out to be $(417.6202M_{\bigoplus}, 0.0258AU)$, we can clearly see that it is in the *Jovian worlds* region. This means that *Kepler-17 b* is a gas giant. We can also say that it's a *Jupiter* like planet because it's mass is $1.3137M_J$.

Now to comment whether the planet is *Hot Jupiter* or *Cold Jupiter* we will use the surface temperature of the host star *Kepler-17* which is $5781 \pm 85K$[13] which is very close to the surface temperature of the *Sun*. We also know that $R_{orbit} = 0.0258AU$ and for a *Sun* like star *Continued Habitable Zone* is around $0.95AU - 1.15AU$ and we can see that R_{orbit} is much less than $1AU$, this means that the planet is very close to the host star which implies that *Kepler-17 b* is a ***Hot Jupiter***.

[12]Plot used is taken from https://arxiv.org/abs/1603.08614
[13]This data is taken from https://arxiv.org/abs/1110.5462

Error Analysis

Kepler-17 b			
Parameter	Standard data	Calculated data	Error(percentage)
Orbital Period[d]	1.4857	1.4857	0.00
Transit Duration[h]	2.276	2.4	5.45
Orbital Semi-Major Axis[AU]	0.0268	0.0258	3.73
Radius [R_J]	1.33	1.2357	7.09
Mass[M_J]	2.47	1.3137	46.81
Density[g cm^{-3}]	1.3	0.8636	33.57
Kepler-21			
Mass[$M_{\odot}$]	1.408	1.3158	6.55
Radius[$R_{\odot}$]	1.902	1.8676	1.81

In the above table for the standard data column we have used the values from here for *Kepler-17 b* and from here for *Kepler-21*.

We can observe large error in the measurment of mass of *Kepler-17 b*, hence a large difference in its density, from the standard data. This *could* be due to incorrect measurement of stellar velocity.