Plotting a Light Curve Example

K2 data of HIP 41378, AKA EPIC ID number 211311380

In [19]:
```
#plot a light curve that displays a planet, using plotly

#LHW: hiya, this is what I *think* the goal is, but I 100% may be wrong.
#I've added a ton of stuff (edits with LHW). Again, this is
#how *I* would have coded things, definitely not gospel!!! Hopefully
#the python functions will at least be helpful.

#CAN
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import plotly
import plotly.graph_objects as go
import plotly.express as px
from scipy.ndimage import gaussian_filter1d
import altair as alt


from scipy import interpolate #LHW
from scipy.interpolate import UnivariateSpline #LHW
%matplotlib notebook #LHW
```

```
UsageError: unrecognized arguments: #LHW
```

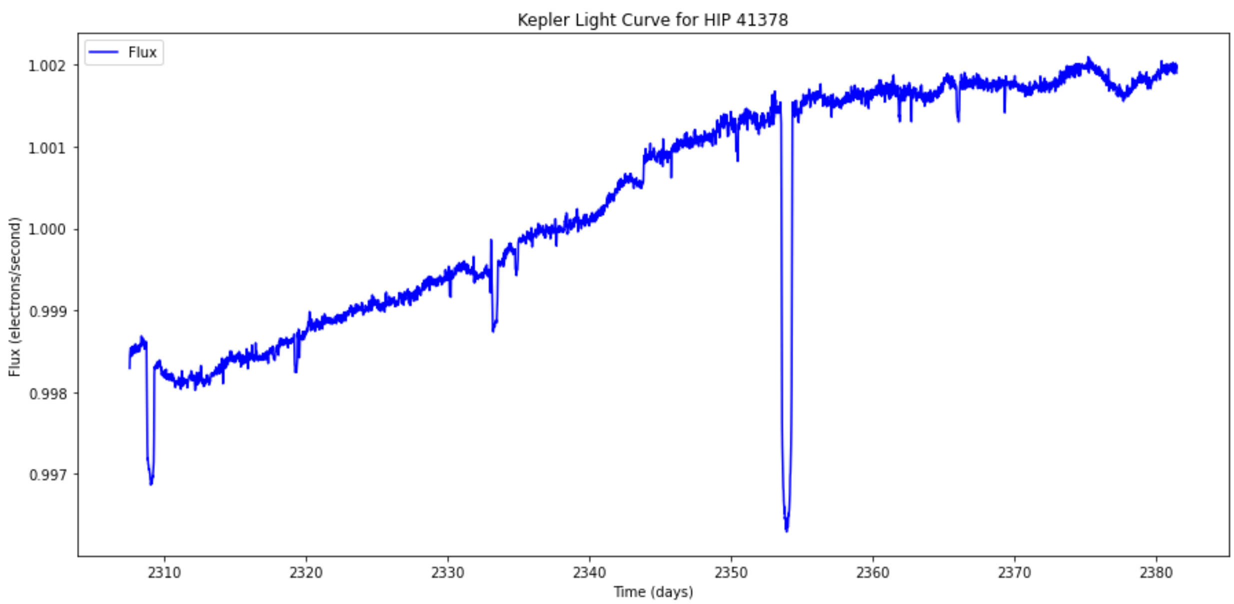
In [20]:
```
#Light Curve, Data from Hubble Space Telescope, HIP 41378
lightcurve = pd.read_csv('Data/HIP_41378 .csv')
lightcurve = lightcurve.rename(columns={"BJD - 2454833" : "Time"})
lightcurve
```

Out[20]:

	Time	Flux
0	2307.550487	0.998291
1	2307.570919	0.998452
2	2307.591351	0.998505
3	2307.611783	0.998540
4	2307.632215	0.998534
...	...	...
3397	2381.309196	1.001959
3398	2381.350060	1.002019
3399	2381.370492	1.001949
3400	2381.390924	1.002003
3401	2381.411356	1.001900

3402 rows × 2 columns

In [21]:
```
# fig = go.Figure()
# fig.add_trace(go.Scatter(x = lightcurve['Time'], y = lightcurve['Flux'], mode
# fig
plt.figure(figsize=(15,7))

original = plt.plot(lightcurve['Time'], lightcurve['Flux'], '-b', label='Flux'
plt.title('Kepler Light Curve for HIP 41378')
plt.xlabel('Time (days)')
plt.ylabel('Flux (electrons/second)')
plt.legend()
plt.show()
```



In the above plot, we see multiple dips in the spectra that could indicate multiple planet transits. However, the main dip would be located around the 2355 mark on the x-axis. I would assume this is the primarily planet being observed.

Kepler Dataset, Object name HAT-P-7 b resolved by SANTA resolver

In [73]:
```
#read in fits file
#Object name HAT-P-7 b resolved by SANTA resolver to HAT-P-7 b ( ), Kepler Data
import os
import glob
import numpy as np
import matplotlib.pyplot as plt
from astropy.io import fits
from astropy.table import Table
```

In [74]:
```
kep = fits.open("Data/hat-p-7.fits")
kep.info()
```

```
Filename: Data/hat-p-7.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    155   476R x 20C   [D, E, J, E, E, E, E,
E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (12, 10)   int32
```

In [75]:
```
object_name = kep[0].header['OBJECT']
object_name
```

Out[75]:
```
'KIC 10666592'
```

In [76]:
```
header1 = kep[1].header
print(repr(header1[0:24]))
```

```
XTENSION= 'BINTABLE'           / marks the beginning of a new HDU
BITPIX  =                    8 / array data type
NAXIS   =                    2 / number of array dimensions
NAXIS1  =                  100 / length of first array dimension
NAXIS2  =                  476 / length of second array dimension
PCOUNT  =                    0 / group parameter count (not used)
GCOUNT  =                    1 / group count (not used)
TFIELDS =                   20 / number of table fields
TTYPE1  = 'TIME    '           / column title: data time stamps
TFORM1  = 'D       '           / column format: 64-bit floating point
TUNIT1  = 'BJD - 2454833'      / column units: barycenter corrected JD
TDISP1  = 'D14.7   '           / column display format
TTYPE2  = 'TIMECORR'           / column title: barycenter - timeslice correcti
on
TFORM2  = 'E       '           / column format: 32-bit floating point
TUNIT2  = 'd       '           / column units: day
TDISP2  = 'E13.6   '           / column display format
TTYPE3  = 'CADENCENO'          / column title: unique cadence number
TFORM3  = 'J       '           / column format: signed 32-bit integer
TDISP3  = 'I10     '           / column display format
TTYPE4  = 'SAP_FLUX'           / column title: aperture photometry flux
TFORM4  = 'E       '           / column format: 32-bit floating point
TUNIT4  = 'e-/s    '           / column units: electrons per second
TDISP4  = 'E14.7   '           / column display format
TTYPE5  = 'SAP_FLUX_ERR'       / column title: aperture phot. flux error
```

In [77]:
```
datatext = kep[1].data
data = Table(datatext)
data[:5]
```

Out[77]: *Table length=5*

TIME	TIMECORR	CADENCENO	SAP_FLUX	SAP_FLUX_ERR	SAP_BKG	SAI
float64	float32	int32	float32	float32	float32	
120.53881583872862	0.00066341873	568	1027193.4	26.107775	3764.0305	
120.55925017884147	0.00066415884	569	1027220.8	26.108118	3797.0823	
120.57968441895355	0.00066489895	570	1027189.94	26.109035	3783.3223	
120.60011885905988	0.00066563906	571	1027253.06	26.109783	3858.511	
120.62055319923093	0.00066637923	572	1027207.9	26.10963	3843.1775	

In [78]:
```
fi = kep[1].header['BJDREFI']
ff = kep[1].header['BJDREFF']

times = kep[1].data['time']
sap_fluxes = kep[1].data['SAP_FLUX'] #Simple Aperture Photometry (SAP)
pdcsap_fluxes = kep[1].data['PDCSAP_FLUX'] #Pre-search Data Conditioning SAP f
```

In [79]:
```
#CAN
# Convert the time array to full BJD by adding the offset back in.
#this value was derived from the estimated difference between transits, howeve
#perhaps by using a loop that detects gaps
#to do this, I would have needed to already removed the slope of the light cur


#LHW: this may be difficult, it doesn't look like the transits follow an easy
```

```
#like a gaussian, it is possible your estimate is pretty good, since they line
#plot. just state that in your presentation and maybe show that its not the sa
#gaussian, voigt, etc

#LHW: you may want to grab the fourth transit as well in the average, up to yo
#remove all the things with 4 if you don't want that transit included
```

In [80]:

```
#extracting the times
bjd = times + fi + ff
bjd2 = bjd - 2.205
bjd3 = bjd2 - 2.205
bjd4 = bjd3 - 2.205 #LHW
```

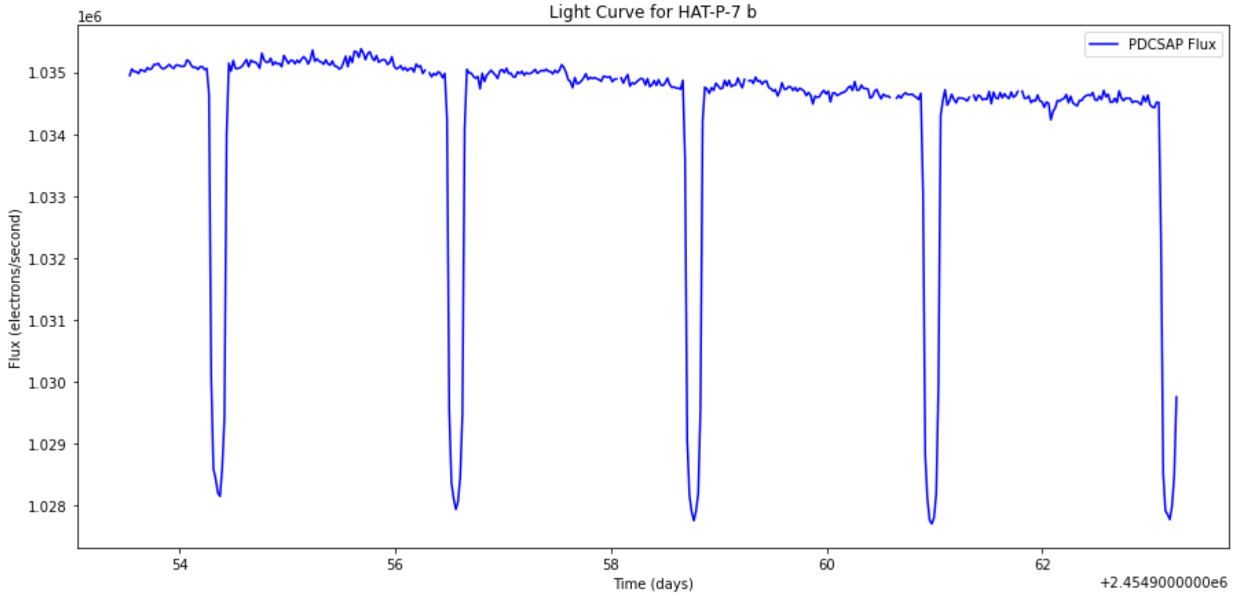
In [81]:

```
fig = go.Figure()
original = fig.add_trace(go.Scatter(x = bjd, y = pdcsap_fluxes, name="Original
fig.update_layout(title = 'Kepler Light Curve',
                xaxis_title="Time (Days)",
                yaxis_title="Flux (Electrons/Second",
                legend_title="Light Curves",)
fig.show()
```

In [82]:

```
# original plot
plt.figure(figsize=(15,7))
original = plt.plot(bjd, (pdcsap_fluxes), '-b', label='PDCSAP Flux') #correcte
plt.title('Light Curve for HAT-P-7 b')
plt.xlabel('Time (days)')
```

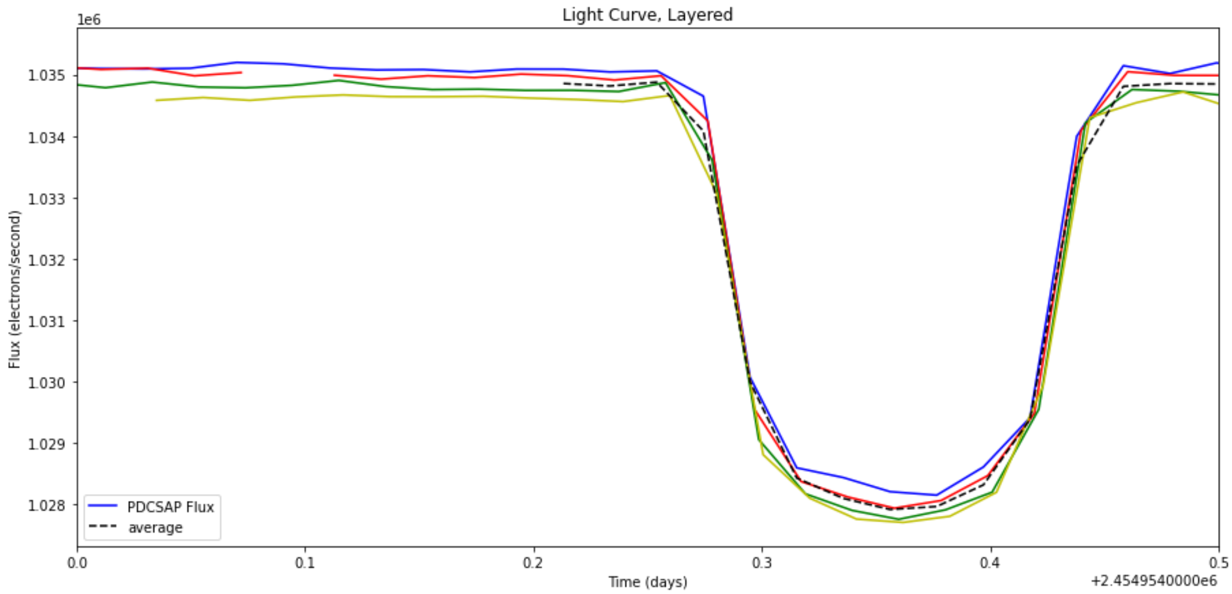
```
plt.ylabel('Flux (electrons/second)')
plt.legend()
plt.show()
```



In [83]:
```
#LHW:
#get stack transit curves
curve_x = [t for t in bjd if 54.2+2.4549e6 < t < 54.5+2.4549e6]
#are these numbers hard coded, analyze how i can do this computationally
f_y1 = interpolate.interp1d(bjd, pdcsap_fluxes)
f_y2 = interpolate.interp1d(bjd2, pdcsap_fluxes)
f_y3 = interpolate.interp1d(bjd3, pdcsap_fluxes)
f_y4 = interpolate.interp1d(bjd4, pdcsap_fluxes)
curve_y1 = f_y1(curve_x)
curve_y2 = f_y2(curve_x)
curve_y3 = f_y3(curve_x)
curve_y4 = f_y4(curve_x)

all_curves = np.vstack((curve_y1, curve_y2, curve_y3, curve_y4))
mean_curve_y = np.nanmean(all_curves, axis=0)
```

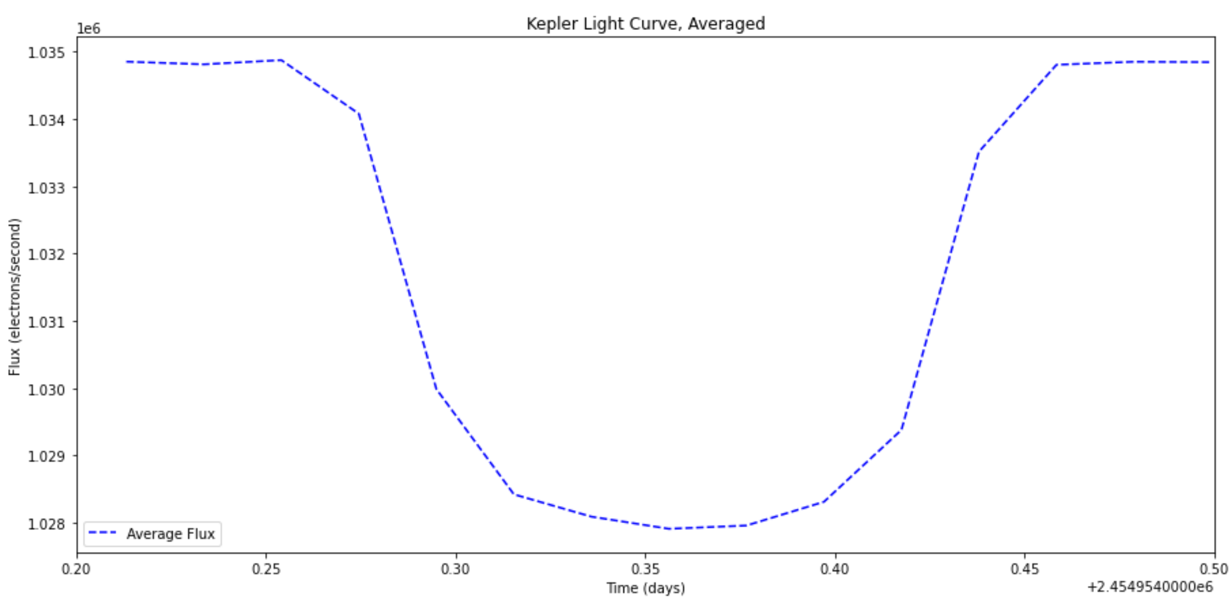
In [84]:
```
plt.figure(figsize=(15,7))

#plt.plot(bjd, sap_fluxes, '-k', label='SAP Flux') #original flux
plt.plot(bjd, pdcsap_fluxes, '-b', label='PDCSAP Flux') #corrected flux
plt.plot(bjd2, pdcsap_fluxes, '-r')
plt.plot(bjd3, pdcsap_fluxes, '-g')
plt.plot(bjd4, pdcsap_fluxes, '-y') #LHW
plt.plot(curve_x, mean_curve_y, '-k', ls='--', label='average') #LHW


plt.xlim((54+2.4549e6), (54.5+2.4549e6)) #hard coded for this plot
plt.title('Light Curve, Layered')
plt.xlabel('Time (days)')
plt.ylabel('Flux (electrons/second)')
plt.legend()
plt.show()
```

```
/var/folders/m2/1g4yxt715m54blcw4245j4ch0000gn/T/ipykernel_96087/619321262.py:
8: UserWarning:

linestyle is redundantly defined by the 'linestyle' keyword argument and the f
mt string "-k" (-> linestyle='-'). The keyword argument will take precedence.
```



In [85]:

```
#average plot alone
plt.figure(figsize=(15,7))
plt.plot(curve_x, mean_curve_y, 'blue', ls='--', label='Average Flux') #LHW
plt.xlim(((54.2)+2.4549e6), (54.5+2.4549e6))
#plt.ylim(1.02e6, 1.04e6)
plt.title('Kepler Light Curve, Averaged')
plt.xlabel('Time (days)')
plt.ylabel('Flux (electrons/second)')
plt.legend()
plt.show()

plt.savefig('average.png')
```




```
<Figure size 432x288 with 0 Axes>
```

In [86]:
```
#zoom in and fit
fig = go.Figure()
fig.add_trace(go.Scatter(x = curve_x, y = mean_curve_y, mode='lines' , ))
#fig.update_traces(marker=dict(color='black'),)
fig.update_layout(title = "Average Light Curve for HAT-P-7", xaxis_title = "Tir
fig
```

In [87]:
```
#What can you derive from an average transit such as the one plotted above? You
```

In [88]:
```
#secondary eclipse data
# During the secondary eclipse, thermal emission emitted from the planet does n
# spectrum of light seen from both the host star and the exoplanet (see the spe

# #The light detected during the eclipse can be compared with the total light c
# eclipse to reveal the contribution from the planet’s surface and atmosphere.
# Astronomers can’t detect a secondary eclipse for every exoplanet;
# the smaller and cooler the planet, the harder the eclipse is to identify.
```

Data Analysis

In [89]:
```
#turning arrays into a dataframe to manipulate data, original data
lightcurves = pd.DataFrame((zip(bjd, pdcsap_fluxes)))
```

```
lightcurves.columns =['Time', 'Flux'] #renaming columns
lightcurves.dropna() #removing nan values
```

Out[89]:

	Time	Flux
0	2.454954e+06	1.034945e+06
1	2.454954e+06	1.035051e+06
2	2.454954e+06	1.035009e+06
3	2.454954e+06	1.035010e+06
4	2.454954e+06	1.034975e+06
...	...	...
471	2.454963e+06	1.027846e+06
472	2.454963e+06	1.027766e+06
473	2.454963e+06	1.027986e+06
474	2.454963e+06	1.028483e+06
475	2.454963e+06	1.029748e+06

469 rows × 2 columns

In [90]:

```
#creating a duplicate dataframe that contains the reduced light curve
reduced = pd.DataFrame((zip(bjd, pdcsap_fluxes)))
reduced.columns =['Time', 'Flux'] #renaming columns
reduced.dropna() #removing nan values
reduced.head()
```

Out[90]:

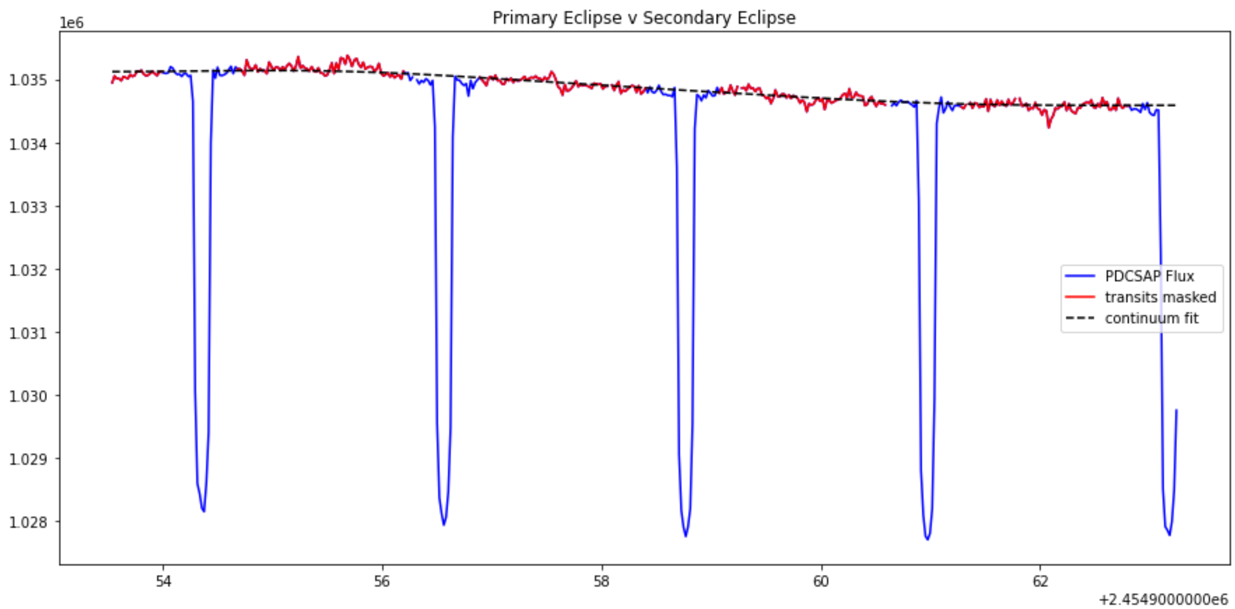
	Time	Flux
0	2.454954e+06	1.034945e+06
1	2.454954e+06	1.035051e+06
2	2.454954e+06	1.035009e+06
3	2.454954e+06	1.035010e+06
4	2.454954e+06	1.034975e+06

In [91]:

```
#LHW:
#get and fit a continuum level to the data
#transits 2.205 apart (from above...keep in mind this is estimate)
#first dip appears at x~2454954.365, mask out +/- 0.300
#mask out each dip (4)
masked_ydata = []
for x, y in zip(bjd, pdcsap_fluxes):
    if 2454954 < x < 2454954.665:
        masked_ydata.append(np.nan)
        continue
    if 2454954+2.205 < x < 2454954.665+2.205:
        masked_ydata.append(np.nan)
        continue
    if 2454954+(2.205*2) < x < 2454954.665+(2.205*2):
        masked_ydata.append(np.nan)
        continue
```

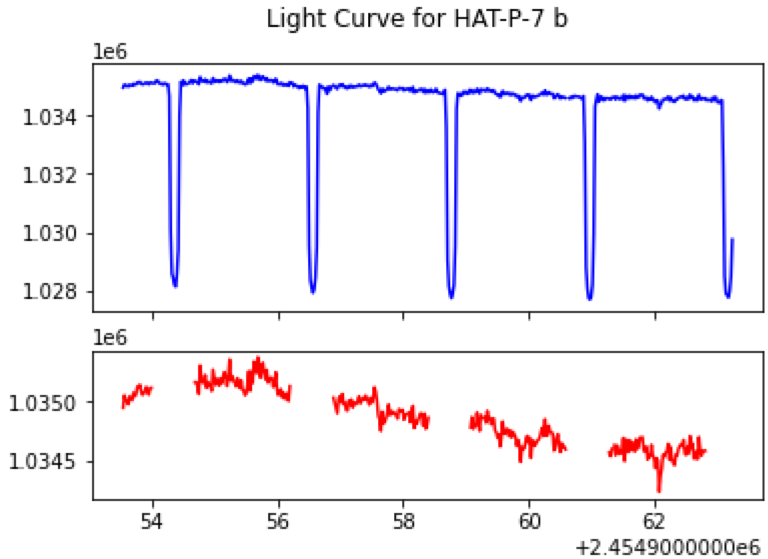
```
    if 2454954+(2.205*3) < x < 2454954.665+(2.205*3):
        masked_ydata.append(np.nan)
        continue
    if 2454954+(2.205*4) < x < 2454954.665+(2.205*4):
        masked_ydata.append(np.nan)
        continue
    masked_ydata.append(y)

masked_ydata = np.asarray(masked_ydata)
#fit curve through data to subtract off/normalize continuum level
#just trust me on this part, it's a pain
masked_xdata2 = [b for b, myd in zip(bjd, masked_ydata) if np.isnan(myd)==Fals
masked_ydata2 = [myd for b, myd in zip(bjd, masked_ydata) if np.isnan(myd)==Fa
spl = UnivariateSpline(masked_xdata2, masked_ydata2, s=1,k=1)
spline = spl(bjd)
continuum = np.asarray(gaussian_filter1d(spline, 50))

plt.figure(figsize=(15,7))
plt.plot(bjd, pdcsap_fluxes, '-b', label='PDCSAP Flux')
plt.plot(bjd, masked_ydata, '-r', label='transits masked')
plt.plot(bjd, continuum, 'k', ls='--', label='continuum fit')

plt.title("Primary Eclipse v Secondary Eclipse")

plt.legend()#first dip appears at x~2454954.365, mask out +/- 0.300
```

Out[91]: <matplotlib.legend.Legend at 0x7fab912a9d90>



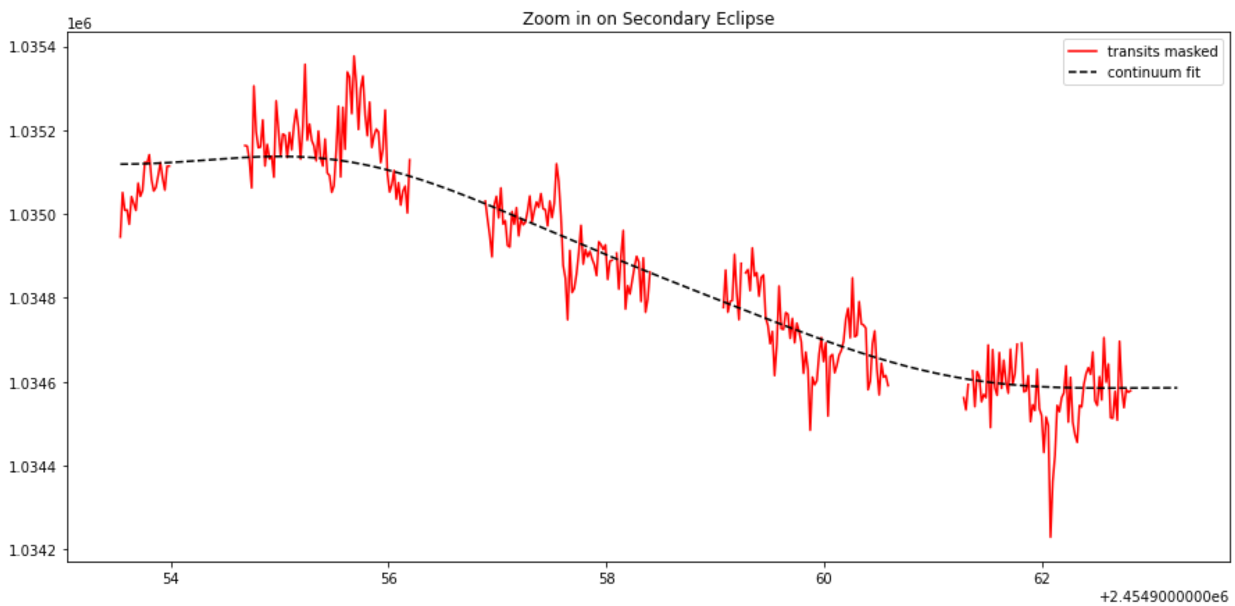
In [92]:
```
fig, (ax1, ax2) = plt.subplots(2, 1, gridspec_kw={'height_ratios': [5, 3]}, sha
fig.suptitle('Light Curve for HAT-P-7 b')
ax1.plot(bjd, pdcsap_fluxes, '-b', label='PDCSAP Flux')
ax2.plot(bjd, masked_ydata, '-r', label='transits masked')
```

Out[92]: [<matplotlib.lines.Line2D at 0x7fab85b4dee0>]



In [93]:
```
plt.figure(figsize=(15,7))
#plt.plot(bjd, pdcsap_fluxes, '-b', label='PDCSAP Flux')
plt.plot(bjd, masked_ydata, '-r', label='transits masked')
plt.plot(bjd, continuum, 'k', ls='--', label='continuum fit')

plt.title("Zoom in on Secondary Eclipse")

plt.legend()
```

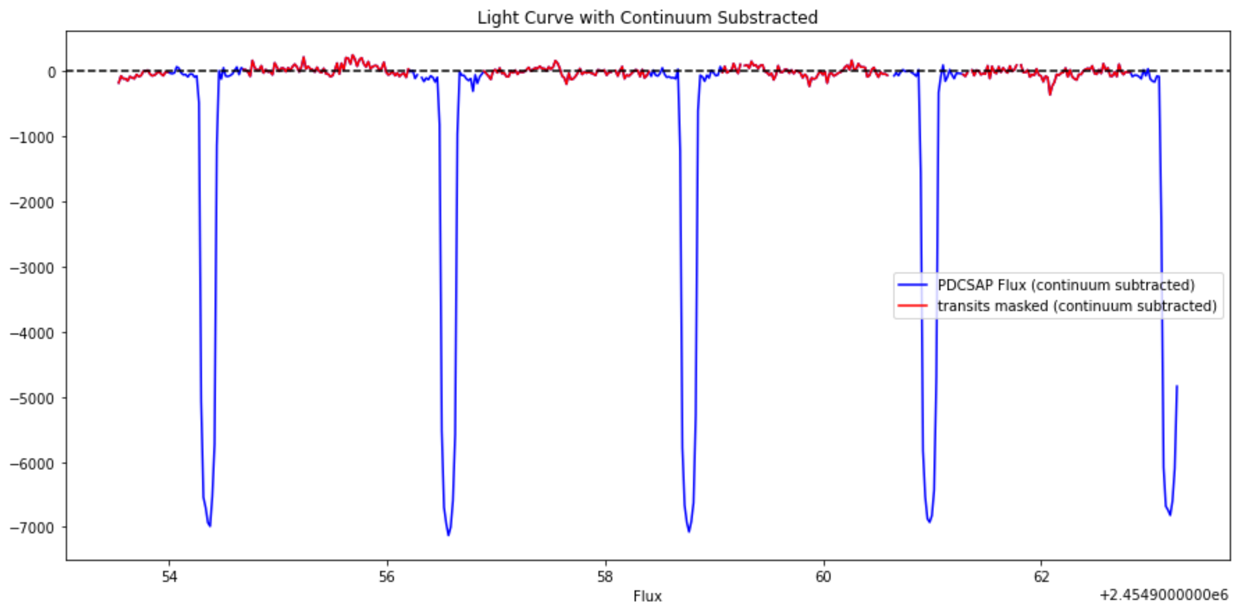
Out[93]:
```
<matplotlib.legend.Legend at 0x7fab866a86d0>
```



In [94]:
```
#LHW:
#subtract off continuum, fit not too bad. will work for now
pdcsap_fluxes_contsub = pdcsap_fluxes - continuum
masked_ydata_contsub = masked_ydata - continuum
plt.figure(figsize=(15,7))
plt.plot(bjd, pdcsap_fluxes_contsub, '-b', label='PDCSAP Flux (continuum subtra
plt.plot(bjd, masked_ydata_contsub, '-r', label='transits masked (continuum sul
plt.axhline(y=0, color='k', ls='--')
plt.legend()
```

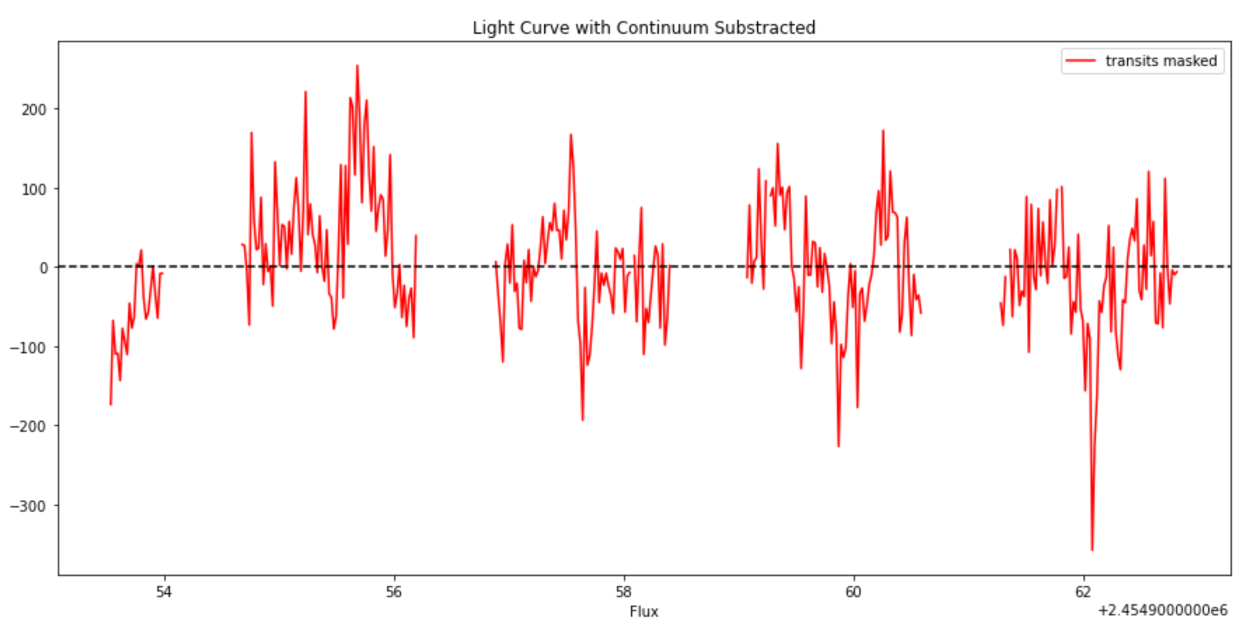
```
plt.title("Light Curve with Continuum Substracted")
plt.xlabel("Flux")
```

Out[94]: Text(0.5, 0, 'Flux')



In [95]:
```
time_diff = 2.4549e6
pdcsap_fluxes_contsub = pdcsap_fluxes - continuum
masked_ydata_contsub = masked_ydata - continuum
plt.figure(figsize=(15,7))
#plt.plot(bjd, pdcsap_fluxes_contsub, '-b', label='PDCSAP Flux (continuum subt
plt.plot(bjd, masked_ydata_contsub, '-r', label='transits masked')
plt.axhline(y=0, color='k', ls='--')
plt.title("Light Curve with Continuum Substracted")
plt.xlabel("Flux")
plt.legend()

#plt.ylim(-200, 200)
#plt.xlim(56.5+time_diff, 60.5+time_diff)
```

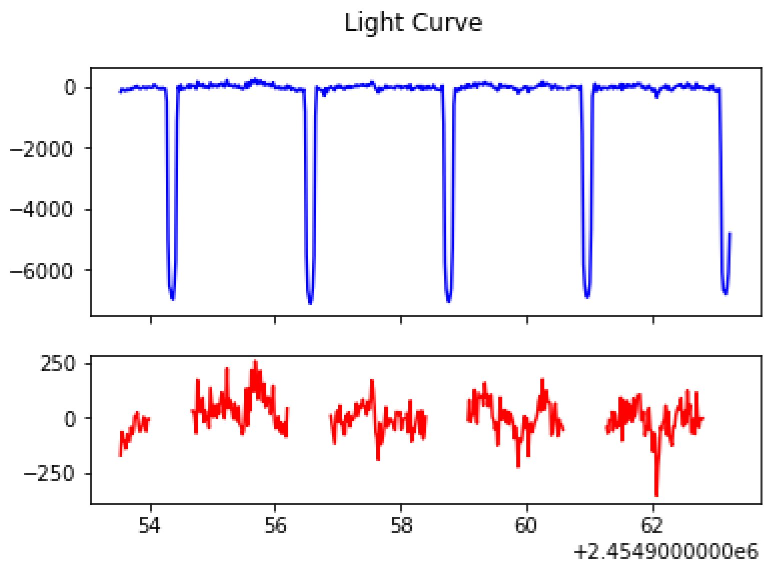
Out[95]: <matplotlib.legend.Legend at 0x7fab8a9bb1f0>



In [96]:
```
#concat plots
plt.figure(figsize=(15,7))
fig, (ax1, ax2) = plt.subplots(2, 1, gridspec_kw={'height_ratios': [5, 3]}, sha
fig.suptitle('Light Curve')
ax1.plot(bjd, pdcsap_fluxes_contsub, '-b', label='PDCSAP Flux')
ax2.plot(bjd, masked_ydata_contsub, '-r', label='transits masked')
```

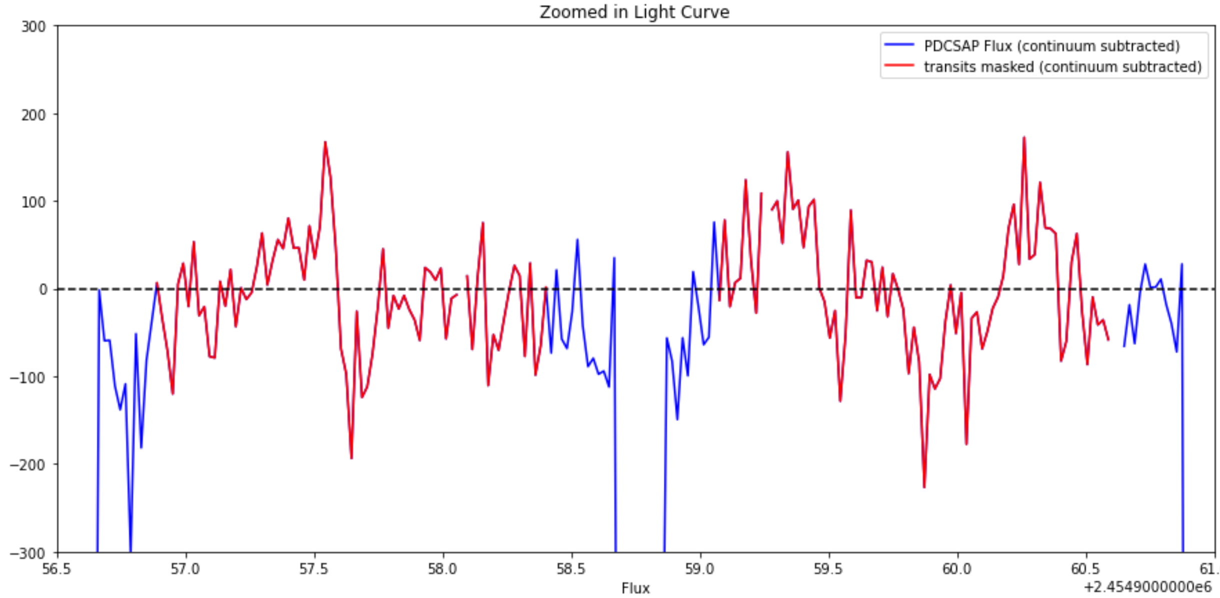
Out[96]:
```
[<matplotlib.lines.Line2D at 0x7faba04cba60>]
```

```
<Figure size 1080x504 with 0 Axes>
```



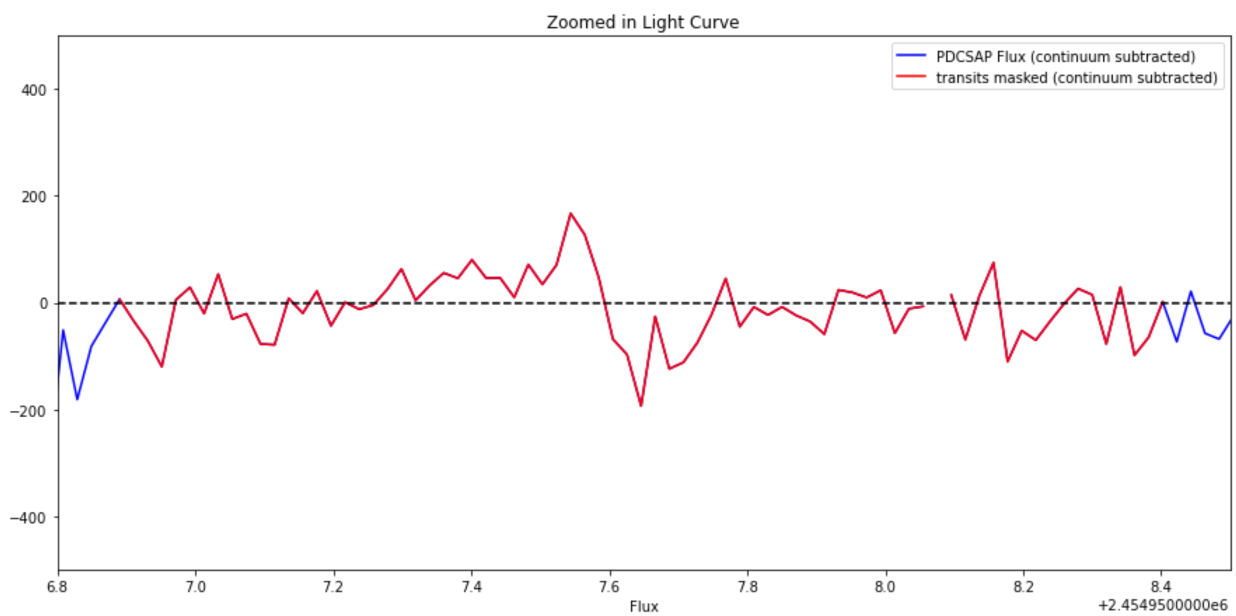
In [102…
```
time_diff = 2.4549e6
pdcsap_fluxes_contsub = pdcsap_fluxes - continuum
masked_ydata_contsub = masked_ydata - continuum
plt.figure(figsize=(15,7))
plt.plot(bjd, pdcsap_fluxes_contsub, '-b', label='PDCSAP Flux (continuum subtra
plt.plot(bjd, masked_ydata_contsub, '-r', label='transits masked (continuum sul
plt.axhline(y=0, color='k', ls='--')
plt.legend()

plt.ylim(-300, 300)
plt.xlim(56.5+time_diff, 61+time_diff)

plt.title("Zoomed in Light Curve")
plt.xlabel("Flux")
```

Out[102]:
```
Text(0.5, 0, 'Flux')
```



In [52]:
```
time_diff = 2.4549e6
pdcsap_fluxes_contsub = pdcsap_fluxes - continuum
masked_ydata_contsub = masked_ydata - continuum
plt.figure(figsize=(15,7))
plt.plot(bjd, pdcsap_fluxes_contsub, '-b', label='PDCSAP Flux (continuum subtra
plt.plot(bjd, masked_ydata_contsub, '-r', label='transits masked (continuum sul
plt.axhline(y=0, color='k', ls='--')
plt.legend()

plt.ylim(-500, 500)
plt.xlim(56.8+time_diff, 58.5+time_diff)

plt.title("Zoomed in Light Curve")
plt.xlabel("Flux")
```

Out[52]:
```
Text(0.5, 0, 'Flux')
```



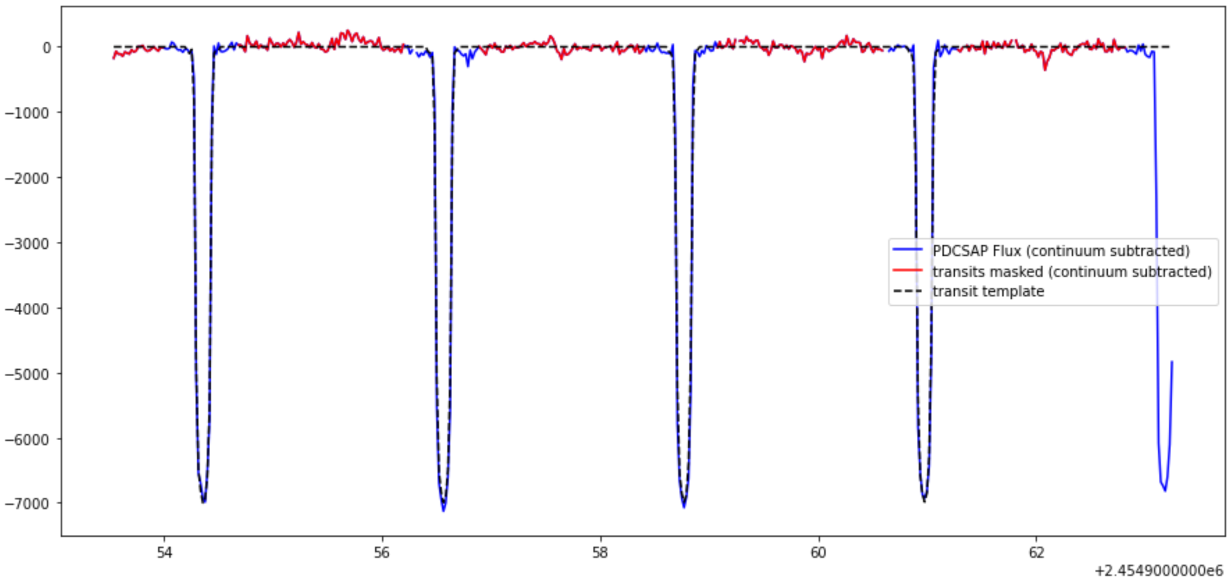
In [51]:
```
#LHW:
#average transits
#we'd techincally want to do this here instead, with continuum subtracted curve
```

```
curve_x = np.asarray([t for t in bjd if 54.2+2.4549e6 < t < 54.5+2.4549e6])
f_y1 = interpolate.interp1d(bjd, pdcsap_fluxes_contsub)
f_y2 = interpolate.interp1d(bjd2, pdcsap_fluxes_contsub)
f_y3 = interpolate.interp1d(bjd3, pdcsap_fluxes_contsub)
f_y4 = interpolate.interp1d(bjd4, pdcsap_fluxes_contsub)
curve_y1 = f_y1(curve_x)
curve_y2 = f_y2(curve_x)
curve_y3 = f_y3(curve_x)
curve_y4 = f_y4(curve_x)

all_curves = np.vstack((curve_y1, curve_y2, curve_y3, curve_y4))
mean_curve_y = np.nanmean(all_curves, axis=0)

#now we need to make a composite template curve with the average dip at each ti
#since we will be interpolating outside of original range (bjd), need a fill va
#points, send it to 0(see graph)
oringinal_range = bjd
curve_x1 = curve_x
curve_x2 = curve_x1 + 2.205
curve_x3 = curve_x1 + (2.205*2)
curve_x4 = curve_x1 + (2.205*3)
f_mean1 = interpolate.interp1d(curve_x1, mean_curve_y, bounds_error=False, fil
f_mean2 = interpolate.interp1d(curve_x2, mean_curve_y, bounds_error=False, fil
f_mean3 = interpolate.interp1d(curve_x3, mean_curve_y, bounds_error=False, fil
f_mean4 = interpolate.interp1d(curve_x4, mean_curve_y, bounds_error=False, fil
curve_mean1 = f_mean1(bjd)
curve_mean2 = f_mean2(bjd)
curve_mean3 = f_mean3(bjd)
curve_mean4 = f_mean4(bjd)

#now add them together to make composite (this works bc everything except trans
template = np.sum((curve_mean1, curve_mean2, curve_mean3, curve_mean4), axis=0

plt.figure(figsize=(15,7))
plt.plot(bjd, pdcsap_fluxes_contsub, '-b', label='PDCSAP Flux (continuum subtra
plt.plot(bjd, masked_ydata_contsub, '-r', label='transits masked (continuum sub
plt.plot(bjd, template, '-k', ls='--', label='transit template')
plt.legend()
#great, composite looks good (again, not exact because things are just shifted
#was your estimate. will work for now. in the future can be more rigorous with
#but that will take a lot more python/curve_fit)
#remember to ignore that last transit since it was not completed
```

```
/var/folders/m2/1g4yxt715m54blcw4245j4ch0000gn/T/ipykernel_96087/852644289.py:
40: UserWarning:

linestyle is redundantly defined by the 'linestyle' keyword argument and the f
mt string "-k" (-> linestyle='-'). The keyword argument will take precedence.
```

Out[51]:

```
<matplotlib.legend.Legend at 0x7fab8a34a850>
```

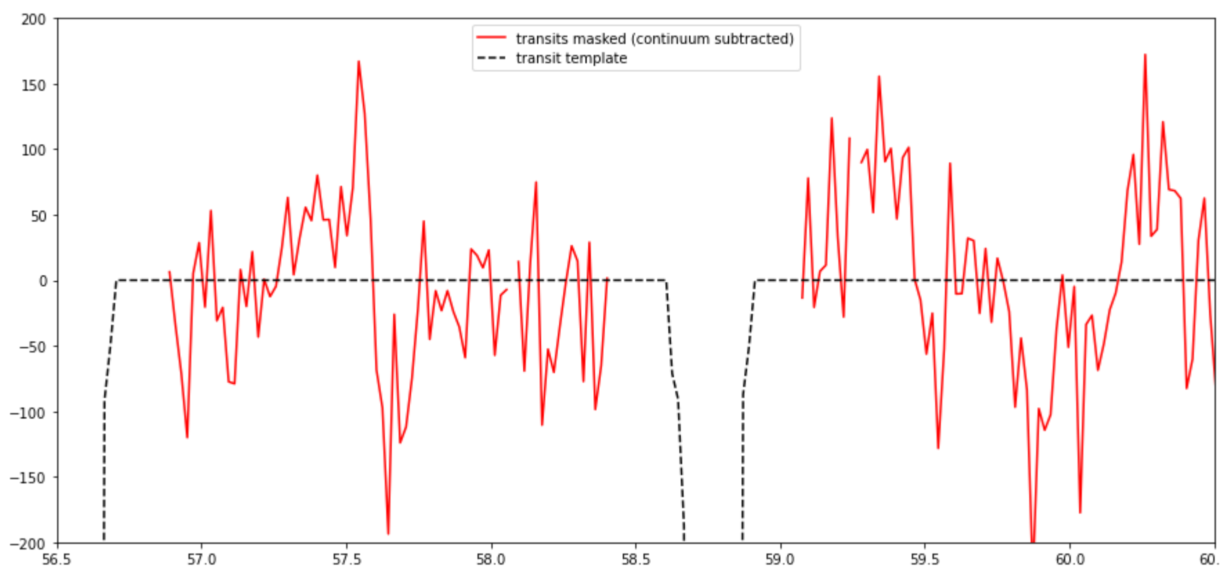


In [367…

```
plt.figure(figsize=(15,7))
#plt.plot(bjd, pdcsap_fluxes_contsub, '-b', label='PDCSAP Flux (continuum subt
plt.plot(bjd, masked_ydata_contsub, '-r', label='transits masked (continuum su
plt.plot(bjd, template, '-k', ls='--', label='transit template')
plt.legend()

plt.ylim(-200, 200)
plt.xlim(56.5+time_diff, 60.5+time_diff)
```

```
/var/folders/m2/1g4yxt715m54blcw4245j4ch0000gn/T/ipykernel_86263/3209319169.p
y:4: UserWarning:

linestyle is redundantly defined by the 'linestyle' keyword argument and the f
mt string "-k" (-> linestyle='-'). The keyword argument will take precedence.
```

Out[367]: (2454956.5, 2454960.5)

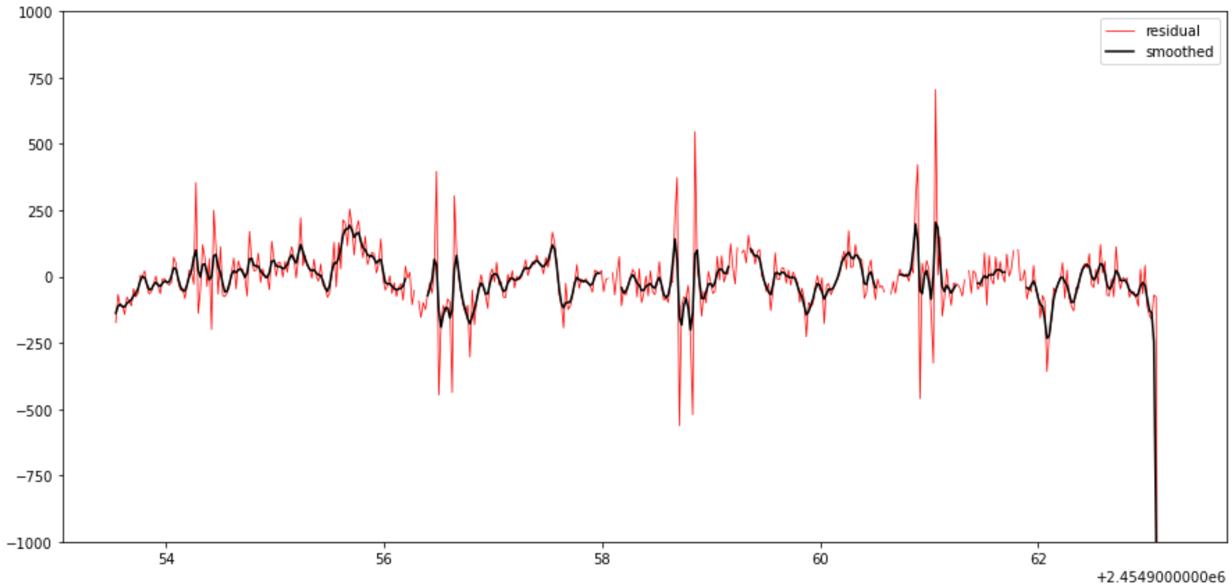


In [368…

```
#LHW:
#subtract off two curves

#subtract off the template from the original light curve --> residual
```

```
residual = pdcsap_fluxes_contsub - template
plt.figure(figsize=(15,7))
plt.plot(bjd, residual, '-r', lw=0.75, label='residual')
plt.plot(bjd, gaussian_filter1d(residual, 1), color='k', label='smoothed')
plt.legend()
plt.ylim(-1000, 1000)

#again, ignore that last dip, from unfit transit
```

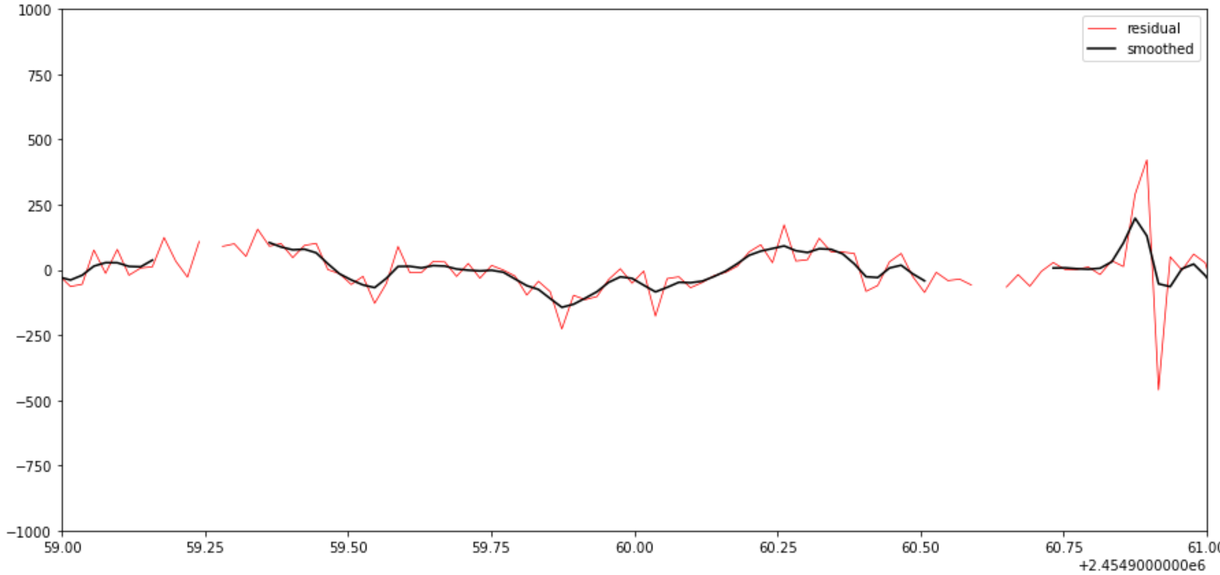
Out[368]: (-1000.0, 1000.0)



In [369…

```
#subtract off the template from the original light curve --> residual
residual = pdcsap_fluxes_contsub - template
plt.figure(figsize=(15,7))
plt.plot(bjd, residual, '-r', lw=0.75, label='residual')
plt.plot(bjd, gaussian_filter1d(residual, 1), color='k', label='smoothed')
plt.legend()
plt.ylim(-1000, 1000)

plt.xlim(59+2.4549e6, 61+2.4549e6)
#again, ignore that last dip, from unfit transit
```

Out[369]: (2454959.0, 2454961.0)



In [271…

Plotting a Less Hot Planet to View Differences in Secondary Eclipses

Kepler 7b

In [103…

```
kep7 = fits.open("Data/kplr-7.fits")
object_name = kep7[0].header['OBJECT']
kep7.info()
print(object_name)
```

```
Filename: Data/kplr-7.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4375R x 20C   [D, E, J, E, E, E, E,
E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (6, 6)   int32
KIC 5780885
```

In [104…

```
datatext = kep7[1].data
data = Table(datatext)
data[:5]
```

Out[104]: *Table length=5*

TIME	TIMECORR	CADENCENO	SAP_FLUX	SAP_FLUX_ERR	SAP_BKG	SAP_
float64	float32	int32	float32	float32	float32	
630.1750905090666	0.002585089	25509	99404.48	8.732295	1402.6284	0.
630.1955235005662	0.0025843806	25510	99406.94	8.732274	1400.8407	0
630.2159563918249	0.0025836718	25511	99421.414	8.732852	1401.8455	(
630.2363893826259	0.0025829626	25512	99425.9	8.732966	1401.2278	
630.2568221734255	0.0025822534	25513	99439.91	8.733522	1402.0792	0.

In [105…

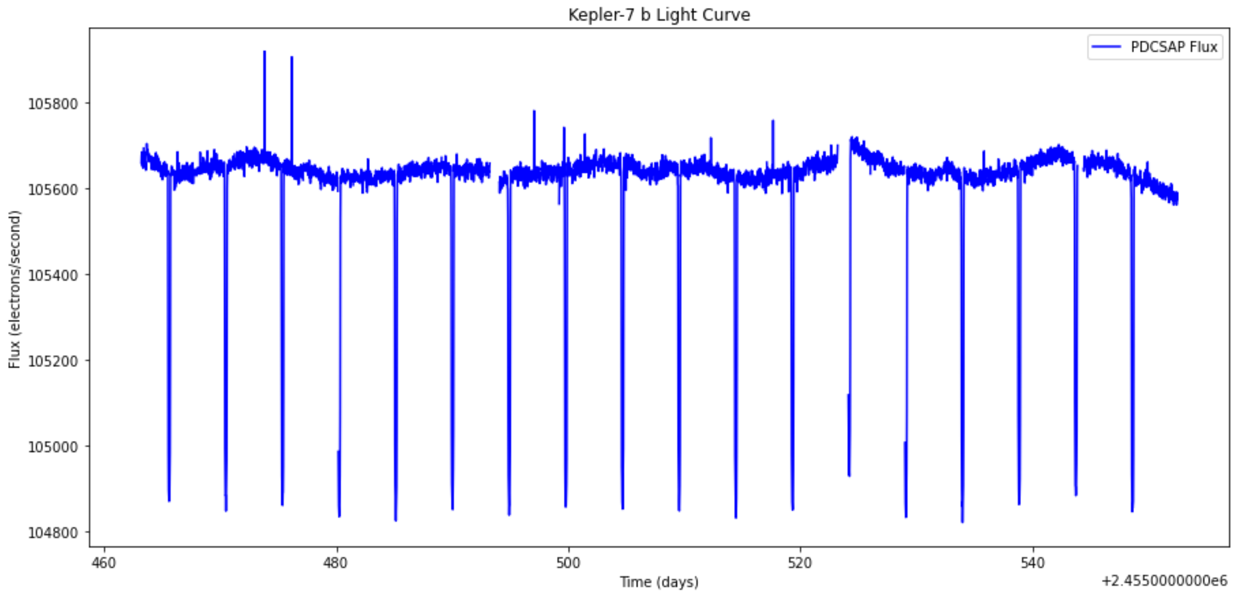
```
i = kep7[1].header['BJDREFI']
ff = kep7[1].header['BJDREFF']

times = kep7[1].data['time']
sap_fluxes = kep7[1].data['SAP_FLUX'] #Simple Aperture Photometry (SAP)
pdcsap_fluxes = kep7[1].data['PDCSAP_FLUX'] #Pre-search Data Conditioning SAP
```

In [106…

```
fig = go.Figure()
original = fig.add_trace(go.Scatter(x = bjd, y = pdcsap_fluxes, name="Original
fig.update_layout(title = 'Kepler Light Curve',
                xaxis_title="Time (Days)",
                yaxis_title="Flux (Electrons/Second)",
                legend_title="Light Curves",)
fig.show()
```

In [107…

```
#extracting the times
bjd = times + fi + ff

# original plot
plt.figure(figsize=(15,7))
original = plt.plot(bjd, (pdcsap_fluxes), '-b', label='PDCSAP Flux') #correcte
plt.title('Kepler-7 b Light Curve')
plt.xlabel('Time (days)')
plt.ylabel('Flux (electrons/second)')
#plt.xlim(480+2.455e6, 520+2.455e6)
plt.legend()

plt.show()
```



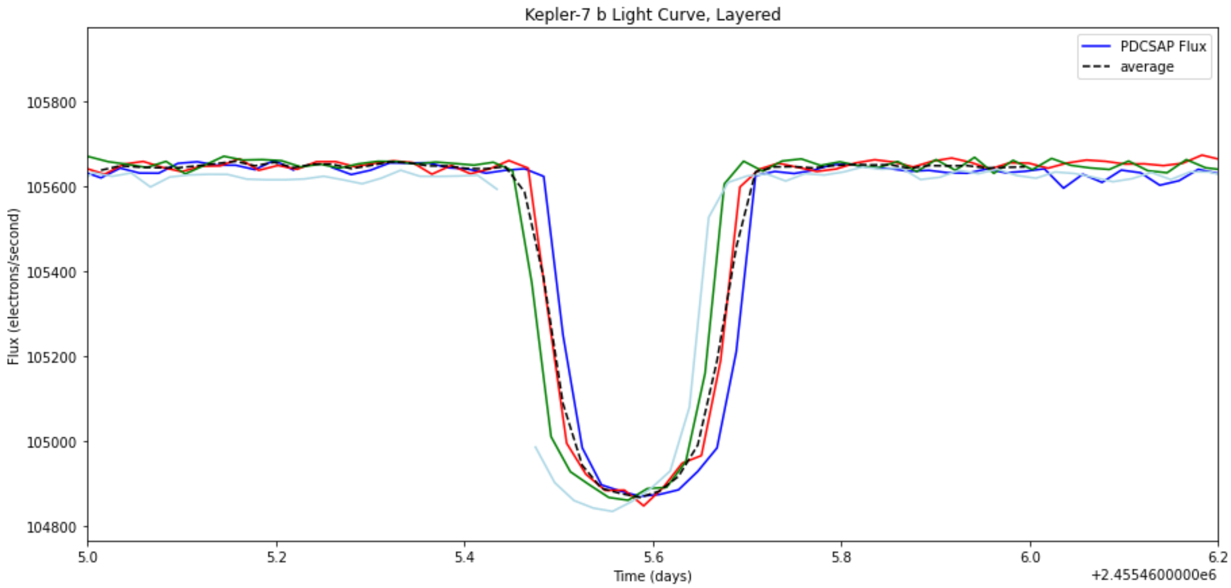
In [108…

```
#times
bjd2 = bjd - 4.9
bjd3 = bjd2 - 4.9
bjd4 = bjd3 - 4.9



curve_x = [t for t in bjd if 65.0+2.4554e6 < t < 66.0+2.4554e6]
#are these numbers hard coded, analyze how i can do this computationally
f_y1 = interpolate.interp1d(bjd, pdcsap_fluxes)
f_y2 = interpolate.interp1d(bjd2, pdcsap_fluxes)
f_y3 = interpolate.interp1d(bjd3, pdcsap_fluxes)
#f_y4 = interpolate.interp1d(bjd4, pdcsap_fluxes)
curve_y1 = f_y1(curve_x)
curve_y2 = f_y2(curve_x)
curve_y3 = f_y3(curve_x)
#curve_y4 = f_y4(curve_x)

all_curves = np.vstack((curve_y1, curve_y2, curve_y3))#, curve_y4))
mean_curve_y = np.nanmean(all_curves, axis=0)
```

In [109…

```
plt.figure(figsize=(15,7))

plt.plot(bjd, pdcsap_fluxes, '-b', label='PDCSAP Flux') #corrected flux
plt.plot(bjd2, pdcsap_fluxes, '-r')
plt.plot(bjd3, pdcsap_fluxes, '-g')
plt.plot(bjd4, pdcsap_fluxes, 'lightblue')
plt.plot(curve_x, mean_curve_y, '-k', ls='--', label='average')


plt.xlim((465+2.455e6), (466.2+2.455e6))
#plt.xlim(460+2.4550000000e6, 480+2.4550000000e6)
plt.title('Kepler-7 b Light Curve, Layered')
plt.xlabel('Time (days)')
plt.ylabel('Flux (electrons/second)')
plt.legend()
plt.show()
```

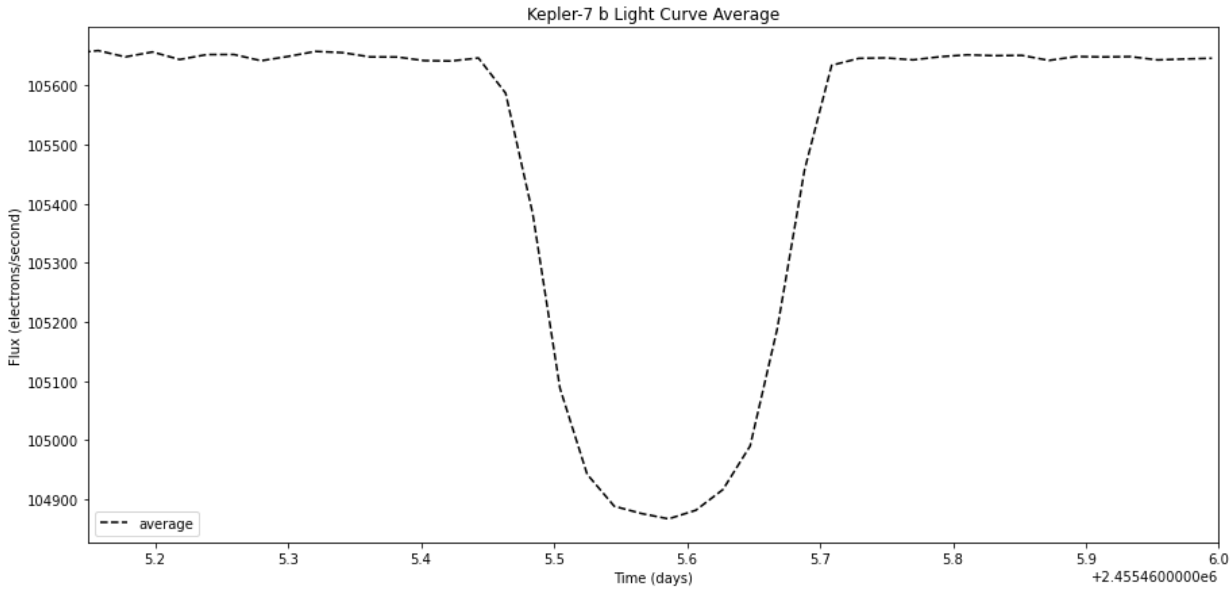
```
/var/folders/m2/1g4yxt715m54blcw4245j4ch0000gn/T/ipykernel_96087/1488917385.p
y:7: UserWarning:

linestyle is redundantly defined by the 'linestyle' keyword argument and the f
mt string "-k" (-> linestyle='-'). The keyword argument will take precedence.
```



In [110…

```
plt.figure(figsize=(15,7))
plt.plot(curve_x, mean_curve_y, '-k', ls='--', label='average')


plt.xlim((465.15+2.455e6), (466+2.455e6))
#plt.xlim(460+2.4550000000e6, 480+2.4550000000e6)
plt.title('Kepler-7 b Light Curve Average')
plt.xlabel('Time (days)')
plt.ylabel('Flux (electrons/second)')
plt.legend()
plt.show()
```

```
/var/folders/m2/1g4yxt715m54blcw4245j4ch0000gn/T/ipykernel_96087/3922574468.p
y:2: UserWarning:

linestyle is redundantly defined by the 'linestyle' keyword argument and the f
mt string "-k" (-> linestyle='-'). The keyword argument will take precedence.
```



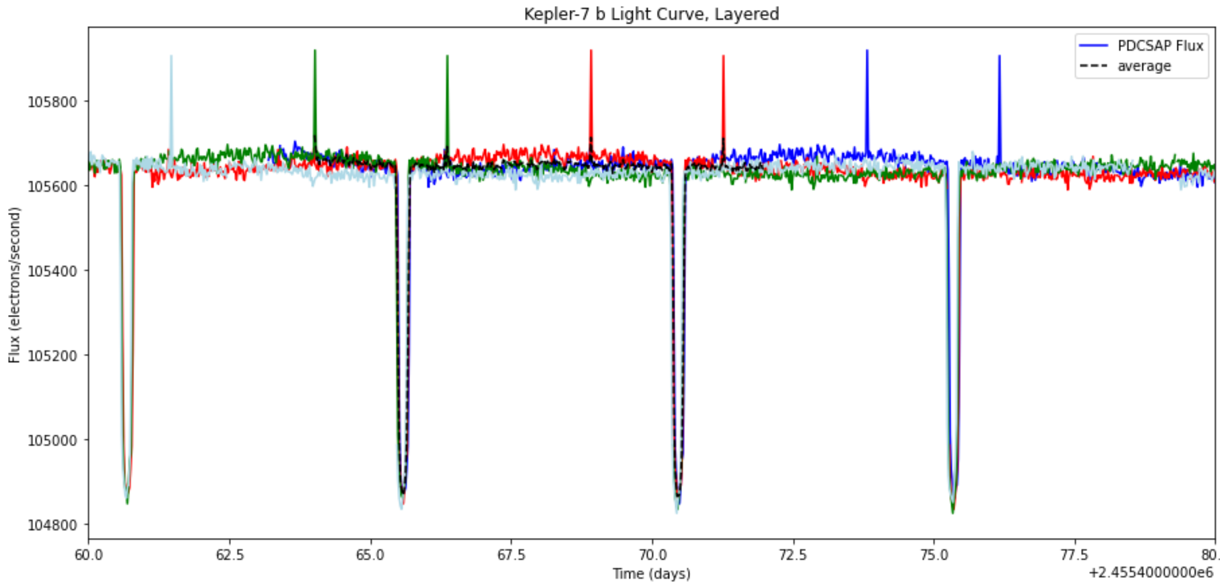
In [111…

```
#replicate the average
```

In [112…

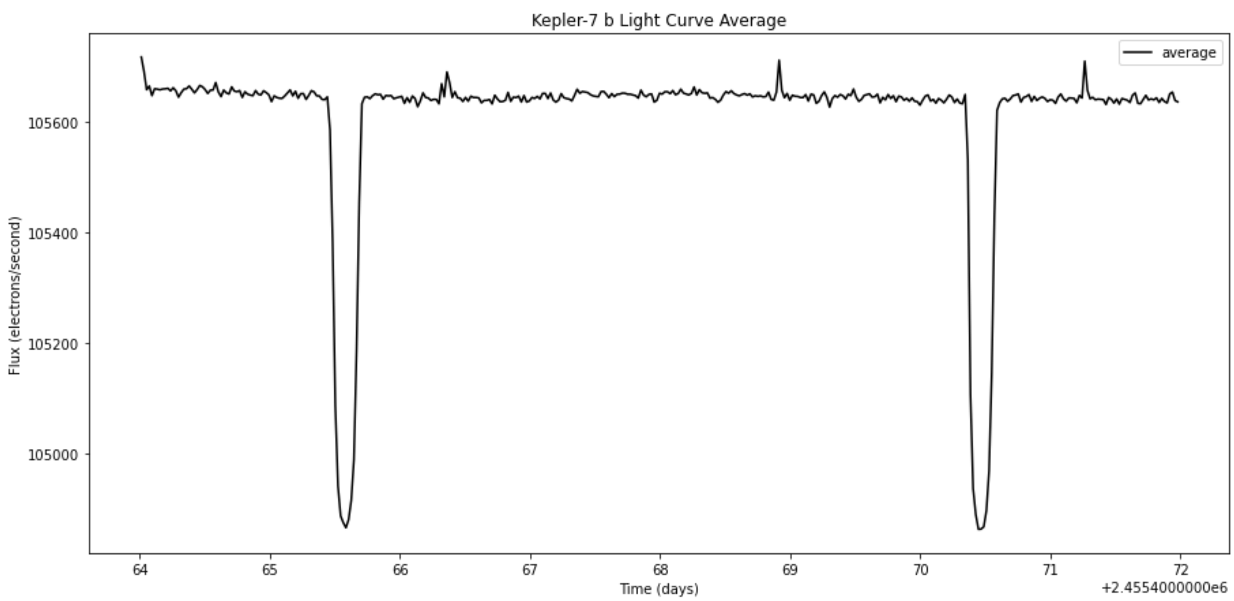
```
#times
bjd2 = bjd - 4.9
bjd3 = bjd2 - 4.9
bjd4 = bjd3 - 4.9



curve_x = [t for t in bjd if 64.0+2.4554e6 < t < 72.0+2.4554e6]
#are these numbers hard coded, analyze how i can do this computationally
f_y1 = interpolate.interp1d(bjd, pdcsap_fluxes)
f_y2 = interpolate.interp1d(bjd2, pdcsap_fluxes)
f_y3 = interpolate.interp1d(bjd3, pdcsap_fluxes)
#f_y4 = interpolate.interp1d(bjd4, pdcsap_fluxes)
curve_y1 = f_y1(curve_x)
curve_y2 = f_y2(curve_x)
curve_y3 = f_y3(curve_x)
#curve_y4 = f_y4(curve_x)

all_curves = np.vstack((curve_y1, curve_y2, curve_y3))#, curve_y4))
mean_curve_y = np.nanmean(all_curves, axis=0)

plt.figure(figsize=(15,7))

plt.plot(bjd, pdcsap_fluxes, '-b', label='PDCSAP Flux') #corrected flux
plt.plot(bjd2, pdcsap_fluxes, '-r')
plt.plot(bjd3, pdcsap_fluxes, '-g')
plt.plot(bjd4, pdcsap_fluxes, 'lightblue')
plt.plot(curve_x, mean_curve_y, '-k', ls='--', label='average')


plt.xlim((460+2.455e6), (480+2.455e6))
#plt.xlim(460+2.4550000000e6, 480+2.4550000000e6)
plt.title('Kepler-7 b Light Curve, Layered')
plt.xlabel('Time (days)')
plt.ylabel('Flux (electrons/second)')
plt.legend()
plt.show()
```

```
/var/folders/m2/1g4yxt715m54blcw4245j4ch0000gn/T/ipykernel_96087/2601852933.p
y:28: UserWarning:

linestyle is redundantly defined by the 'linestyle' keyword argument and the f
mt string "-k" (-> linestyle='-'). The keyword argument will take precedence.
```



In [113…

```
plt.figure(figsize=(15,7))
plt.plot(curve_x, mean_curve_y, '-k', label='average')


#plt.xlim((465.15+2.455e6), (466+2.455e6))
#plt.xlim(460+2.4550000000e6, 480+2.4550000000e6)
plt.title('Kepler-7 b Light Curve Average')
plt.xlabel('Time (days)')
plt.ylabel('Flux (electrons/second)')
plt.legend()
plt.show()
```



In [114…

```
fig = go.Figure()
original = fig.add_trace(go.Scatter(x = curve_x, y = mean_curve_y, name="Origi
fig.update_layout(title = 'Kepler Light Curve',
```

```
                xaxis_title="Time (Days)",
                yaxis_title="Flux (Electrons/Second",
                legend_title="Light Curves",)
fig.show()
```

Data Analysis

In [115…

```
#creating a duplicate dataframe that contains the reduced light curve
reduced = pd.DataFrame((zip(bjd, pdcsap_fluxes)))
reduced.columns =['Time', 'Flux'] #renaming columns
reduced.dropna() #removing nan values
reduced.head()
```

Out[115]:

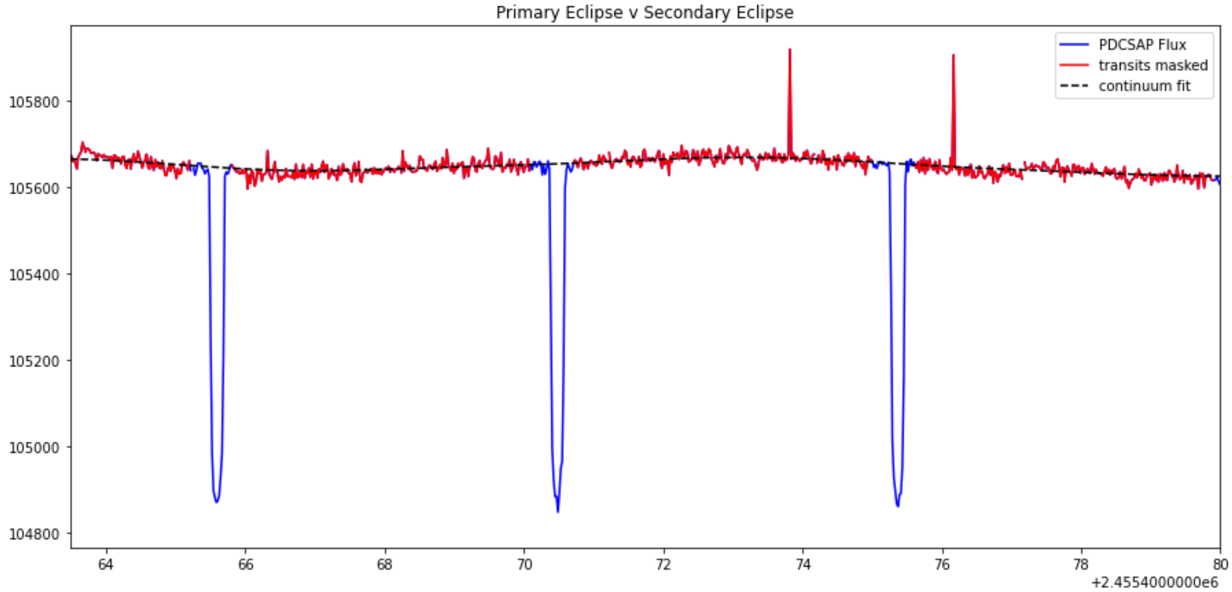
	Time	Flux
0	2.455463e+06	NaN
1	2.455463e+06	105658.898438
2	2.455463e+06	105676.984375
3	2.455463e+06	105677.101562
4	2.455463e+06	105684.304688

In [116…

```
#LHW:
#get and fit a continuum level to the data
#transits 2.205 apart (from above...keep in mind this is estimate)
#first dip appears at x~2454954.365, mask out +/- 0.300
#mask out each dip (4)
dif = 4.9
masked_ydata = []
for x, y in zip(bjd, pdcsap_fluxes):
    if 2455465.2 < x < 2455465.8:
        masked_ydata.append(np.nan)
        continue
    if 2455465.2+dif < x < 2455465.8+dif:
        masked_ydata.append(np.nan)
        continue
    if 2455465.2+(dif*2) < x < 2455465.8+(dif*2):
        masked_ydata.append(np.nan)
        continue
    if 2455465.2+(dif*3) < x < 2455465.8+(dif*3):
        masked_ydata.append(np.nan)
        continue
    if 2455465.2+(dif*4) < x < 2455465.8+(dif*4):
        masked_ydata.append(np.nan)
        continue
    masked_ydata.append(y)


masked_ydata = np.asarray(masked_ydata)
masked_xdata2 = [b for b, myd in zip(bjd, masked_ydata) if np.isnan(myd)==False
masked_ydata2 = [myd for b, myd in zip(bjd, masked_ydata) if np.isnan(myd)==Fa
spl = UnivariateSpline(masked_xdata2, masked_ydata2, s=1,k=1)
spline = spl(bjd)
continuum = np.asarray(gaussian_filter1d(spline, 50))

plt.figure(figsize=(15,7))
plt.plot(bjd, pdcsap_fluxes, '-b', label='PDCSAP Flux')
plt.plot(bjd, masked_ydata, '-r', label='transits masked')
plt.plot(bjd, continuum, 'k', ls='--', label='continuum fit')

plt.title("Primary Eclipse v Secondary Eclipse")
plt.xlim((463.5+2.455e6), (480+2.455e6))
plt.legend()
```

Out[116]:

```
<matplotlib.legend.Legend at 0x7fab91fb7910>
```

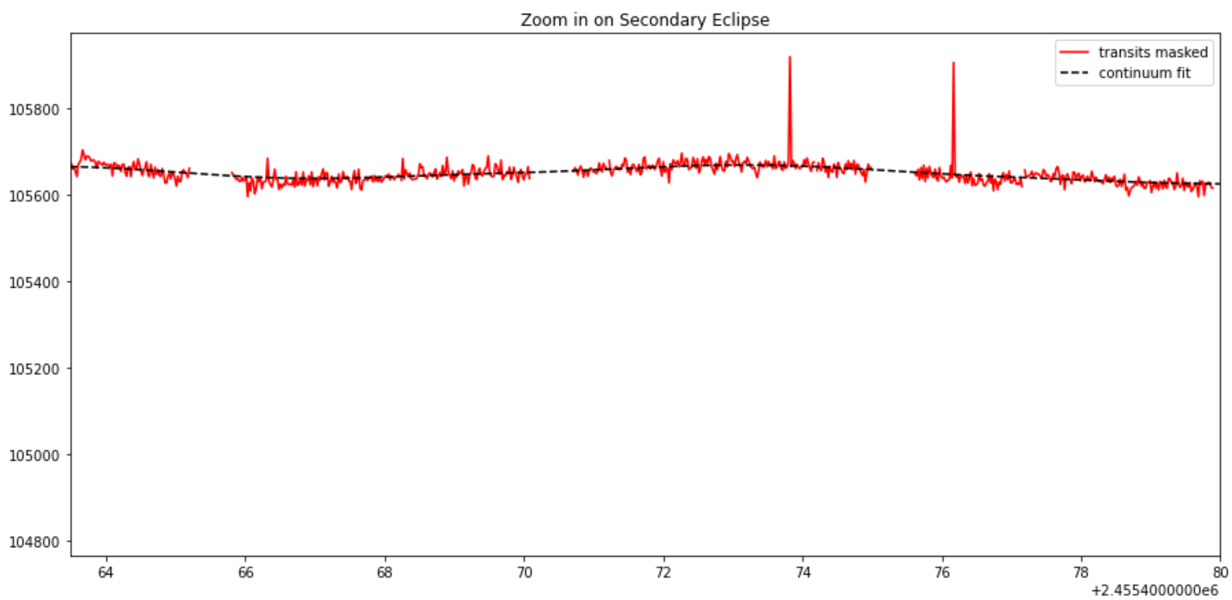


In [117…

```
plt.figure(figsize=(15,7))
plt.xlim((463.5+2.455e6), (480+2.455e6))
plt.title("Zoom in on Secondary Eclipse")

plt.plot(bjd, masked_ydata, '-r', label='transits masked')
plt.plot(bjd, continuum, 'k', ls='--', label='continuum fit')
plt.legend()
```

Out[117]:

```
<matplotlib.legend.Legend at 0x7fab91f85dc0>
```

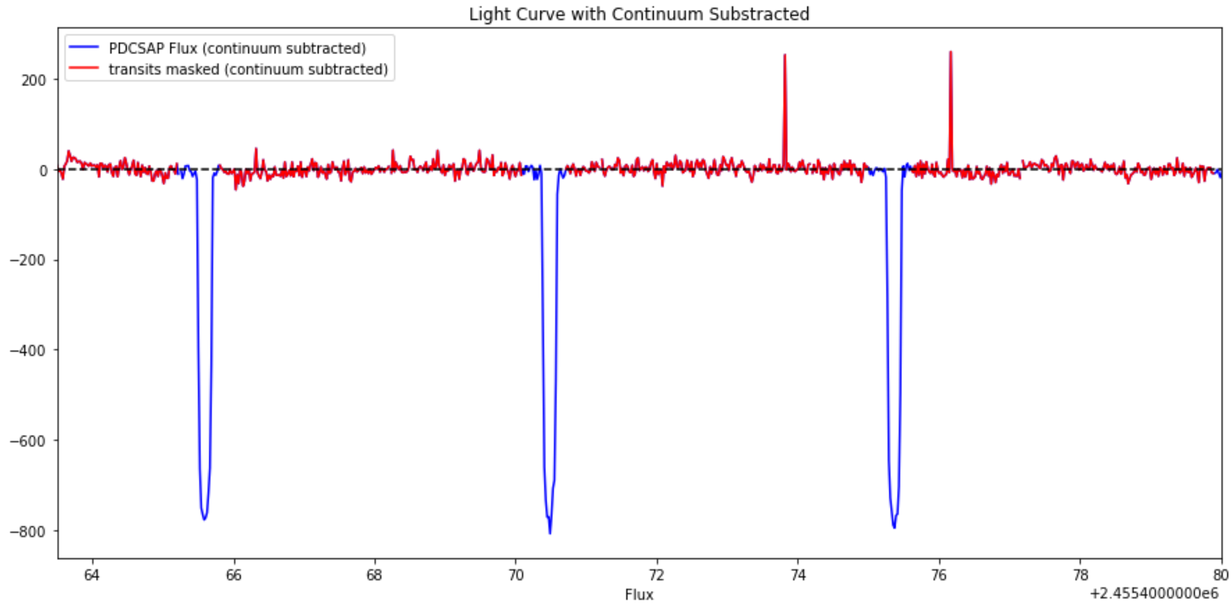


In [118…

```
#LHW:
#subtract off continuum, fit not too bad. will work for now



pdcsap_fluxes_contsub = pdcsap_fluxes - continuum
masked_ydata_contsub = masked_ydata - continuum
plt.figure(figsize=(15,7))
plt.plot(bjd, pdcsap_fluxes_contsub, '-b', label='PDCSAP Flux (continuum subtra
plt.plot(bjd, masked_ydata_contsub, '-r', label='transits masked (continuum sul
plt.axhline(y=0, color='k', ls='--')
```

```
plt.legend()
plt.xlim((463.5+2.455e6), (480+2.455e6))
plt.title("Light Curve with Continuum Substracted")
plt.xlabel("Flux")
```

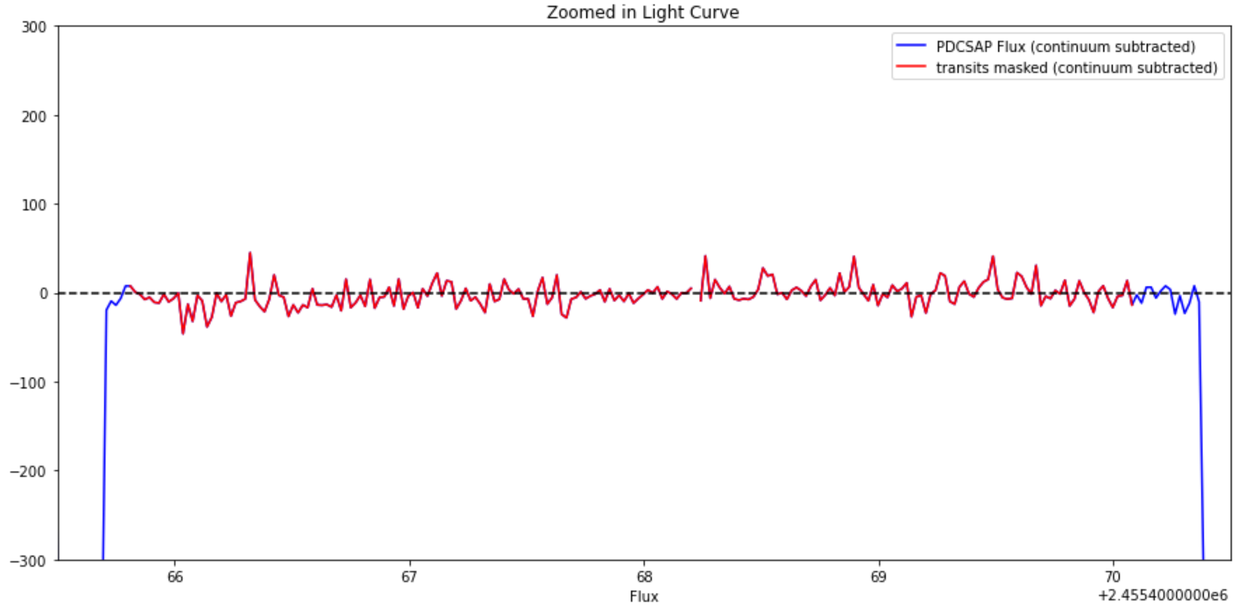
Out[118]: Text(0.5, 0, 'Flux')



In [119…

```
pdcsap_fluxes_contsub = pdcsap_fluxes - continuum
masked_ydata_contsub = masked_ydata - continuum
plt.figure(figsize=(15,7))
plt.plot(bjd, pdcsap_fluxes_contsub, '-b', label='PDCSAP Flux (continuum subtra
plt.plot(bjd, masked_ydata_contsub, '-r', label='transits masked (continuum sul
plt.axhline(y=0, color='k', ls='--')
plt.legend()
plt.xlim((465.5+2.455e6), (470.5+2.455e6))
plt.xlabel("Flux")


plt.ylim(-300, 300)


plt.title("Zoomed in Light Curve")
plt.xlabel("Flux")
```

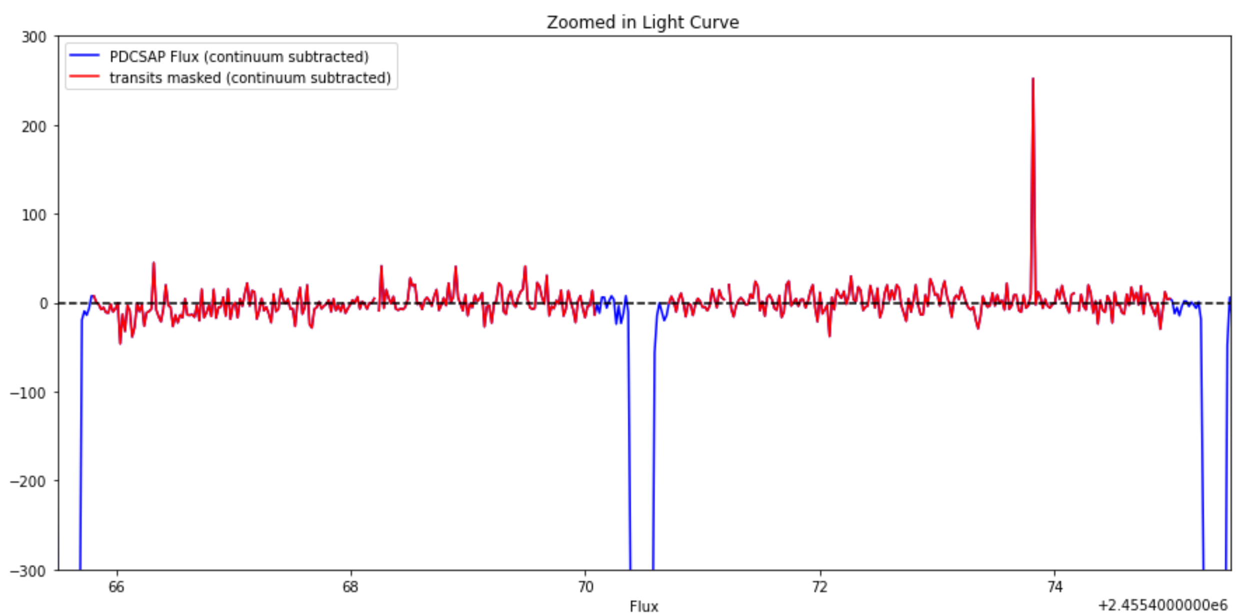
Out[119]: Text(0.5, 0, 'Flux')



In [120…

```
pdcsap_fluxes_contsub = pdcsap_fluxes - continuum
masked_ydata_contsub = masked_ydata - continuum
plt.figure(figsize=(15,7))
plt.plot(bjd, pdcsap_fluxes_contsub, '-b', label='PDCSAP Flux (continuum subtra
plt.plot(bjd, masked_ydata_contsub, '-r', label='transits masked (continuum sul
plt.axhline(y=0, color='k', ls='--')
plt.legend()
plt.xlabel("Flux")

plt.xlim((465.5+2.455e6), (475.5+2.455e6))
plt.ylim(-300, 300)

plt.title("Zoomed in Light Curve")
plt.xlabel("Flux")
```

Out[120]: Text(0.5, 0, 'Flux')



In [ ]: