PS3_1

1.1

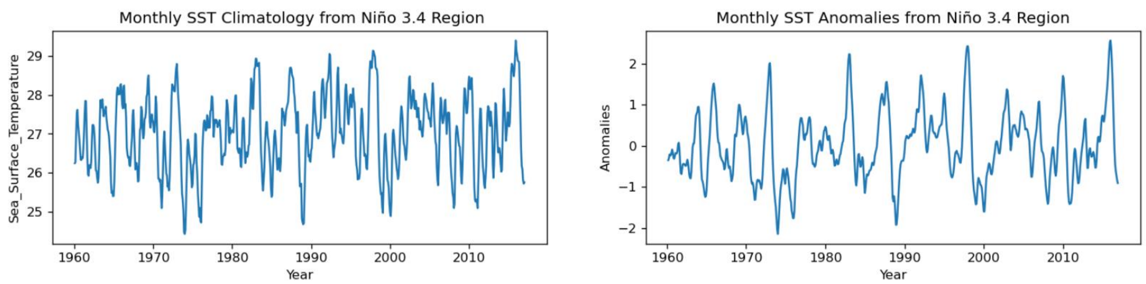
思路：

首先计算根据公式计算 sst 的异常值(Anomalies)，然后在对异常值求三个月的滑动平均值，并对其进行作图。

```
#calculate
sst_clim = ds.sst.sel(lon=slice(190, 240), lat=slice(-5, 5)).groupby('time.month')
sst_anom =sst_clim - sst_clim .mean(dim='time')
sst_anom_rolling = sst_anom .rolling(time=3, center=True).mean()

#plot
#cite;https://blog.csdn.net/m0_46589710/article/details/105383077
time=pd.date_range(start='1960-01',periods=684,freq='m')

fig, (ax1, ax2) = plt.subplots(1,2, figsize=(15,3), sharey=False, dpi=120)
ax1.plot(time,sst_clim.mean(dim=['lat','lon']))
ax2.plot(time,sst_anom_rolling.mean(dim=['lat','lon']))
ax1.set_title(' Monthly SST Climatology from Niño 3.4 Region');
ax2.set_title('Monthly SST Anomalies from Niño 3.4 Region')
ax1.set_xlabel('Year');    ax2.set_xlabel('Year')
ax1.set_ylabel('Sea_Surface_Temperature');    ax2.set_ylabel('Anomalies')
```

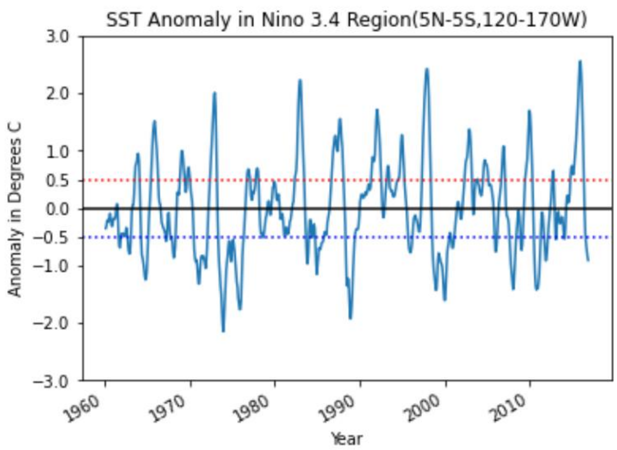


1.2

```
sst_anom_rolling.mean(dim=['lat','lon']).plot()
#cite:https://www.cnblogs.com/shunguo/p/11398148.html
plt.axhline(y=0,ls="-",c="black")
plt.axhline(y=0.5,ls=":",c="red")
plt.axhline(y=-0.5,ls=":",c="blue")

plt.title('SST Anomaly in Nino 3.4 Region(5N-5S,120-170W)')
plt.xlabel('Year')
plt.ylabel('Anomaly in Degrees C')

#cite:https://blog.csdn.net/weixin_38725737/article/details/82667461
plt.yticks([-3,-2,-1,-0.5,0,0.5,1,2,3])
```



PS3_2

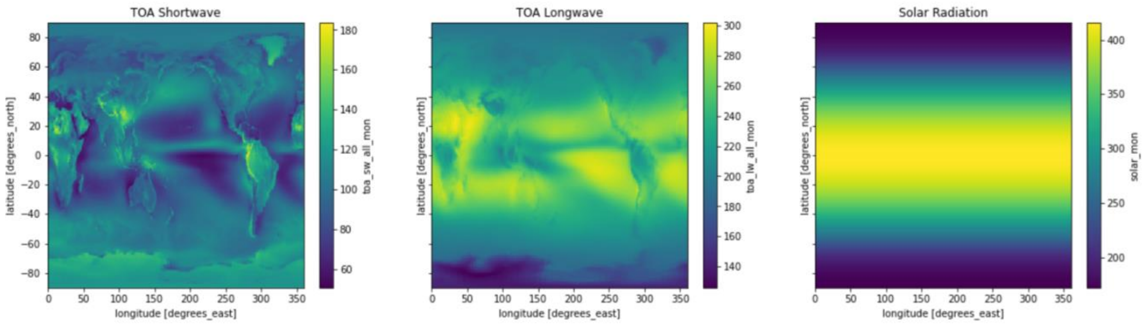
2.1

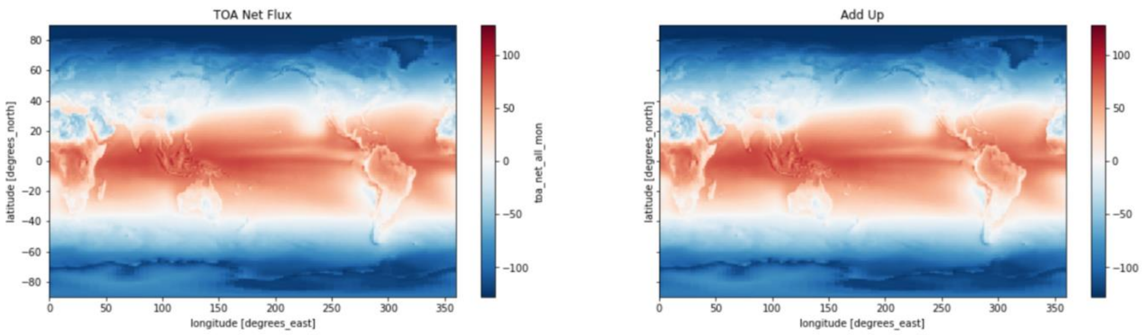
思路：

先分别将长波，短波与太阳辐射的二维图画出，然后再画出通量的图。
之后对长波，短波与太阳辐射按照关系计算，所得到的数据再进行画图，可以得到与通量相同的图，这就证明，通量与三个变量计算的值相等。

```
#plot
fig,(ax1,ax2,ax3) = plt.subplots(1,3, figsize=(20,5),sharey=True)
ds.toa_sw_all_mon.mean(dim=['time']).plot(ax=ax1)
ds.toa_lw_all_mon.mean(dim=['time']).plot(ax=ax2)
ds.solar_mon.mean(dim=['time']).plot(ax=ax3)
ax1.set_title('TOA Shortwave'); ax2.set_title('TOA Longwave'); ax3.set_title('Solar
Radiation')

#Add up calculate
net_cal=-ds.toa_sw_all_mon.mean(dim=['time'])-
ds.toa_lw_all_mon.mean(dim=['time'])+ds.solar_mon
#Plot
fig,(ax4,ax5) = plt.subplots(1,2, figsize=(20,5),sharey=True)
ds.toa_net_all_mon.mean(dim=['time']).plot(ax=ax4)
net_cal.mean(dim=['time']).plot(ax=ax5)
ax4.set_title('TOA Net Flux'); ax5.set_title('Add Up');
```





2.2
思路：
先计算权重，然后对太阳辐射，短波，长波数据分别进行加权，再求时间均值。

```
weights = np.cos(np.deg2rad(ds.lat))
weights

solar_mon_weighted= ds.solar_mon.weighted(weights)
incoming_solar=solar_mon_weighted.mean(dim=('lon', 'lat')).sum()/203

toa_sw_all_mon_weighted= ds.toa_sw_all_mon.weighted(weights)
outgoing_longwave=toa_sw_all_mon_weighted.mean(dim=('lon', 'lat')).sum()/203

toa_lw_all_mon_weighted= ds.toa_lw_all_mon.weighted(weights)
outgoing_shortwave=toa_lw_all_mon_weighted.mean(dim=('lon', 'lat')).sum()/203

incoming_solar,outgoing_longwave,outgoing_shortwave
```

```
(<xarray.DataArray 'solar_mon' ()>
 array(340.28348214),
 <xarray.DataArray 'toa_sw_all_mon' ()>
 array(99.13856604),
 <xarray.DataArray 'toa_lw_all_mon' ()>
 array(240.26681804))
```

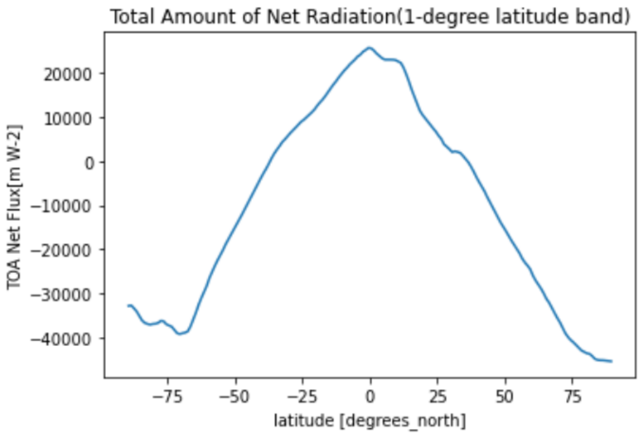
Calculate: 340.28; 99.14;240.27
Cartoon: 340.4; 99.9; 239.9

2.3

思路：

先对通量数据求时间均值，再对每个维度带的数据求总和，再绘图。

```
ds.toa_net_all_mon.mean(dim=['time']).sum(dim=['lon']).plot()
plt.title('Total Amount of Net Radiation(1-degree latitude band)')
plt.ylabel('TOA Net Flux[m W-2]')
```



2.4

思路：

先计算 cldarea_total_daynight_mon 的时间均值用于对区域的划分；再分别在低云与高云的地区对长波与短波进行绘图

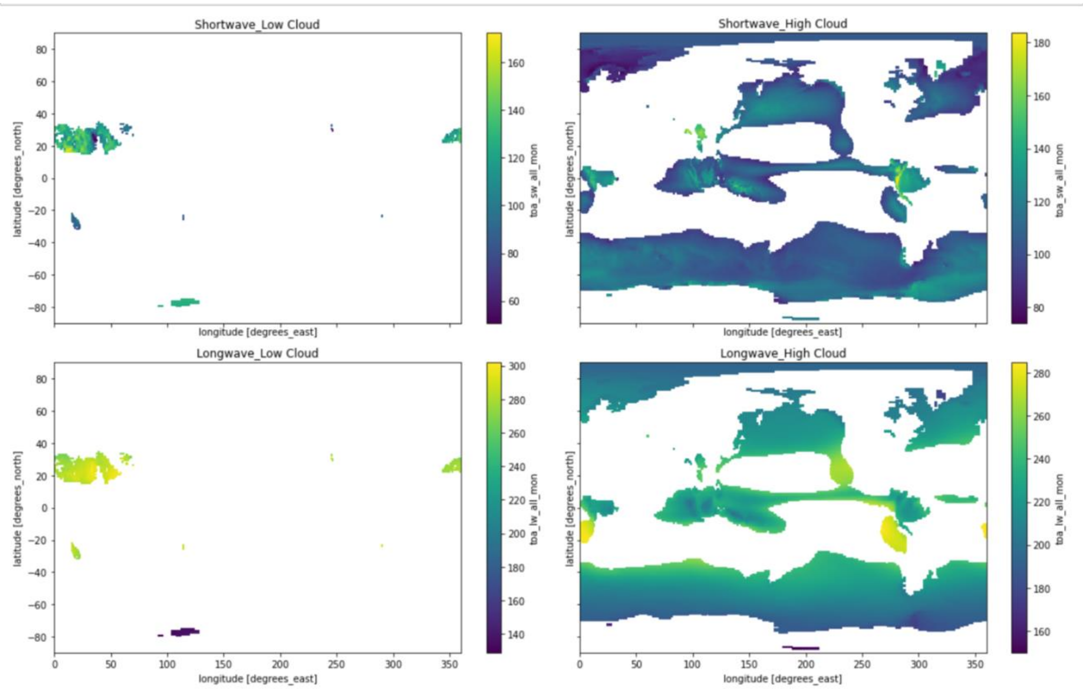
```
arr=ds.cldarea_total_daynight_mon.mean(dim='time')

fig,((ax1,ax2),(ax3,ax4)) = plt.subplots(2,2,
figsize=(16,10),sharex=True,sharey=True)

ds.toa_sw_all_mon.mean(dim='time').where(~np.isnan(arr.where(arr<=25).values)).pl
ot(ax=ax1)
ds.toa_sw_all_mon.mean(dim='time').where(~np.isnan(arr.where(arr>=75).values)).pl
ot(ax=ax2)
ds.toa_lw_all_mon.mean(dim='time').where(~np.isnan(arr.where(arr<=25).values)).pl
ot(ax=ax3)
ds.toa_lw_all_mon.mean(dim='time').where(~np.isnan(arr.where(arr>=75).values)).pl
ot(ax=ax4)

ax1.set_title('Shortwave_Low Cloud',); ax2.set_title('Shortwave_High Cloud');
ax3.set_title('Longwave_Low Cloud');ax4.set_title('Longwave_High Cloud');

plt.tight_layout()
```



2.5

思路：

先对长波与短波数据求时间均值再对其进行区域筛选，选出高云与低云区域的对应数据，再对其进行总体求均值。

```
lc_sw=ds.toa_sw_all_mon.mean(dim='time').where(~np.isnan(arr.where(arr<=25).values)).mean(dim=['lat','lon'])

hc_sw=ds.toa_sw_all_mon.mean(dim='time').where(~np.isnan(arr.where(arr>=75).values)).mean(dim=['lat','lon'])

lc_lw=ds.toa_lw_all_mon.mean(dim='time').where(~np.isnan(arr.where(arr<=25).values)).mean(dim=['lat','lon'])

hc_lw=ds.toa_lw_all_mon.mean(dim='time').where(~np.isnan(arr.where(arr>=75).values)).mean(dim=['lat','lon'])

lc_sw,hc_sw,lc_lw,hc_lw
```

```
Out[56]: (<xarray.DataArray 'toa_sw_all_mon' ()>
          array(122.65546, dtype=float32),
          <xarray.DataArray 'toa_sw_all_mon' ()>
          array(108.09777, dtype=float32),
          <xarray.DataArray 'toa_lw_all_mon' ()>
          array(270.10367, dtype=float32),
          <xarray.DataArray 'toa_lw_all_mon' ()>
          array(216.55675, dtype=float32))
```

低云层短波=122.66
高云层短波=108.10
低云层长波=270.10
高云层长波=216.56

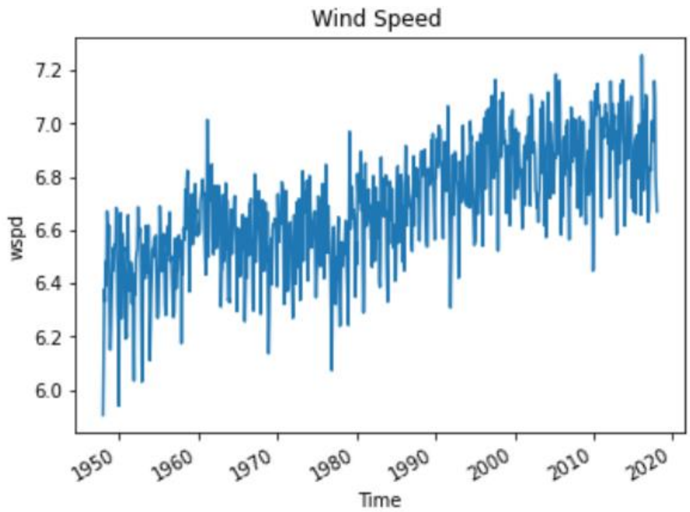
低云区的长短波均高于高云区，证明云层增加辐射的强度。

PS3_3

3.1

```
#monthly seasonal cycle removed
group_data = ds_1000.wspd.groupby('time.month')

#plot
#a time series of global mean wind speed
group_data.mean(dim=('lat','lon')).plot()
```



3.2

```
#global windspeed at June
wspd_june = ds_1000.wspd.groupby('time.month').mean().sel(month=6)

# Time series of  windspeed at Shenzhen
wspd_ShenZhen=ds_1000.wspd.sel(lon=294.1, lat=22.5, method='nearest')

#the anomalies
group_data = ds_1000.wspd.groupby('time.month')
wspd_anom = group_data - group_data.mean(dim='time')

# Sample data where Windspeed is lower than 4
masked_sample1 = ds_1000.wspd.where(ds_1000['wspd'] < 4 )

# Sample data where Windspeed is between 13 and 14
masked_sample2 = ds_1000.wspd.where((ds_1000['wspd'] >
13 )&(ds_1000['wspd'] > 14))

fig,(ax1,ax2,ax3) = plt.subplots(1,3, figsize=(20,5),sharey=False)
wspd_june.plot(ax=ax1)
masked_sample1.mean(dim='time').plot(ax=ax2)
masked_sample2.mean(dim='time').plot(ax=ax3)
ax1.set_title('Global Windspeed at June'); ax2.set_title('Windspeed<4');
ax3.set_title('Windspeed=13~14');


fig,(ax4,ax5) = plt.subplots(1,2, figsize=(20,5),sharey=False)
wspd_ShenZhen.plot(ax=ax4)
wspd_anom.sel(lon=294.1, lat=22.5, method='nearest').plot(ax=ax5)
ax4.set_title('Windspeed at Shenzhen'); ax5.set_title('Anomalies')
```