In [3]:

```
# Import modules
# Make sure you have installed netCDF4, xarray, and nc-time-axis
import numpy as np
import pandas as pd
import xarray as xr
from matplotlib import pyplot as plt
%matplotlib inline
```

In [4]:

```
# Prerequisites

# Load modulesimport numpy as np

import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
```

In [3]:

```
# 打开文件
ds= xr.open_dataset("NOAA_NCDC_ERSST_v3b_SST.nc")
ds
```

Out[3]:

xarray.Dataset

► Dimensions: (**lat**: 89, **lon**: 180, **time**: 684)

▼ Coordinates:

lat	(lat)	float32	-88.0 -86.0 -84.0 ... 86.0 88.0
lon	(lon)	float32	0.0 2.0 4.0 ... 354.0 356.0 35…
time	(time)	datetime64[ns]	1960-01-15 ... 2016-12-15

▼ Data variables:

sst	(time, lat, lon)	float32	...

▼ Attributes:

Conventions :	IRIDL
source :	https://iridl.ldeo.columbia.edu/SOURCES/.NOAA/.NCDC/.ERSST/.version3b/.sst/
history :	extracted and cleaned by Ryan Abernathey for Research Computing in Earth Science

In [4]:

```
#1.1
#选择区域
ds_r = ds.sel(lat=slice(-5,5), lon=slice(190,240))
ds_r
```

Out[4]:

xarray.Dataset

▶ Dimensions: (**lat**: 5, **lon**: 26, **time**: 684)

▼ Coordinates:

lat	(lat)	float32	-4.0 -2.0 0.0 2.0 4.0
lon	(lon)	float32	190.0 192.0 194.0 ... 238.0 2…
time	(time)	datetime64[ns]	1960-01-15 ... 2016-12-15

▼ Data variables:

sst	(time, lat, lon)	float32	...

▼ Attributes:

Conventions :	IRIDL
source :	https://iridl.ldeo.columbia.edu/SOURCES/.NOAA/.NCDC/.ERSST/.version3b/.sst/
history :	extracted and cleaned by Ryan Abernathey for Research Computing in Earth Science

In [ ]:

```

```

In [5]:

```
#计算每月sst的均值
ds_monthly=ds_r.sst.groupby(ds.time.dt.month).mean(dim=['lat', 'lon','time'])
# ds_monthly=ds.sst.groupby(ds.time.dt.year.month).mean()
ds_monthly
```

Out[5]:

xarray.DataArray 'sst' (**month**: 12)

```
array([26.56812 , 26.742603, 27.239906, 27.694029, 27.79552 , 27.598068,
       27.199274, 26.824581, 26.7382  , 26.717516, 26.693666, 26.61345 ],
      dtype=float32)
```

▼ Coordinates:

month	(month)	int64	1 2 3 4 5 6 7 8 9 10 11 12

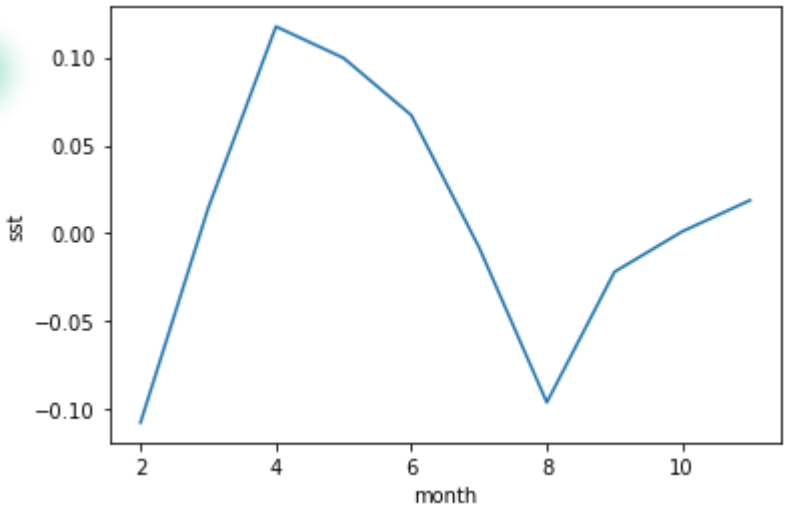
▶ Attributes: (0)

In [6]:

```
#1.1
#借鉴https://www.cnblogs.com/traditional/p/13776180.html
#计算Niño 3.4
ds_rol = ds_monthly.rolling(month=3, center=True).mean()

ds_rol
#计算异常值作图
sst_an=ds_monthly-ds_rol
sst_an
sst_an.plot()

#1.2
#借鉴https://www.cnblogs.com/Gaoqiking/p/11069517.html
# plt.plot(ds_monthly.month,sst_an)
# plt.fill_between(ds_monthly.month.to_numpy(),sst_an,where=(sst_an>0),color='red')
# plt.fill_between(ds_monthly.month.to_numpy(),sst_an,where=(sst_an<0),color='blue')
# #按照月分组后，年不见了不清楚如何将每年的各个月分组进行计算
```

Out[6]:

[<matplotlib.lines.Line2D at 0x1e88876b520>]



In [7]:

```
ds = xr.open_dataset("CERES_EBAF-TOA_200003-201701.nc", engine="netcdf4")
ds
#2.1
#绘制长、短和太阳辐射波的均值2D图
# long=ds.toa_lw_all_mon.mean(dim='time')
# long.plot()
# # short=ds.toa_sw_all_mon.mean(dim='time')
# short.plot()
# solar=ds.solar_mon.mean(dim='time')
# solar.plot()
```

Out[7]:

xarray.Dataset

▶ Dimensions: (**lon**: 360, **time**: 203, **lat**: 180)

▼ Coordinates:

lon	(lon)	float32	0.5 1.5 2.5 ... 357.5 358.5 35…
time	(time)	datetime64[ns]	2000-03-15 ... 2017-01-15
lat	(lat)	float32	-89.5 -88.5 -87.5 ... 88.5 89.5

▼ Data variables:

toa_sw_all_mon	(time, lat, lon)	float32	...
toa_lw_all_mon	(time, lat, lon)	float32	...
toa_net_all_mon	(time, lat, lon)	float32	...
toa_sw_clr_mon	(time, lat, lon)	float32	...
toa_lw_clr_mon	(time, lat, lon)	float32	...
toa_net_clr_mon	(time, lat, lon)	float32	...
toa_cre_sw_mon	(time, lat, lon)	float32	...
toa_cre_lw_mon	(time, lat, lon)	float32	...
toa_cre_net_mon	(time, lat, lon)	float32	...
solar_mon	(time, lat, lon)	float32	...
cldarea_total_d…	(time, lat, lon)	float32	...
cldpress_total_…	(time, lat, lon)	float32	...
cldtemp_total_d…	(time, lat, lon)	float32	...
cldtau_total_da…	(time, lat, lon)	float32	...

▼ Attributes:

title :	CERES EBAF (Energy Balanced and Filled) TOA Fluxes. Monthly Averages and 07/2005 to 06/2015 Climatology.
institution :	NASA/LaRC (Langley Research Center) Hampton, Va
Conventions :	CF-1.4
comment :	Data is from East to West and South to North.
Version :	Edition 4.0; Release Date March 7, 2017
Fill_Value :	Fill Value is -999.0
DOI :	10.5067/TERRA+AQUA/CERES/EBAF-TOA_L3B.004.0
Production_Files :	List of files used in creating the present Master netCDF file: /homedir/nloeb/ebaf/monthly_means/adj_fluxes/deliverable/sw*.gz /homedir/nloeb/ebaf/monthly_means/adj_fluxes/deliverable/lw*.gz /homedir/nloeb/ebaf/monthly_means/adj_fluxes/deliverable/net*.gz

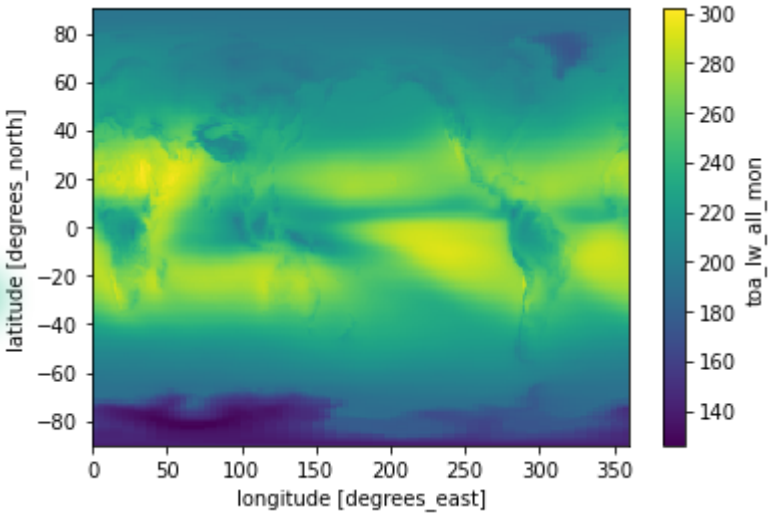
/homedir/nloeb/ebaf/monthly_means/adj_fluxes/deliverable/solflx*.gz
/homedir/nloeb/ebaf/monthly_means/out_glob.dat

In [8]:

```
#2.1
#一开始不太理解，向谢栋学姐请教相关问题
#绘制长、短和太阳辐射波的均值2D图
long=ds.toa_lw_all_mon.mean(dim='time')
long.plot()
# short=ds.toa_sw_all_mon.mean(dim='time')
# short.plot()
# solar=ds.solar_mon.mean(dim='time')
# solar.plot()
```

Out[8]:

<matplotlib.collections.QuadMesh at 0x1e888f3efd0>

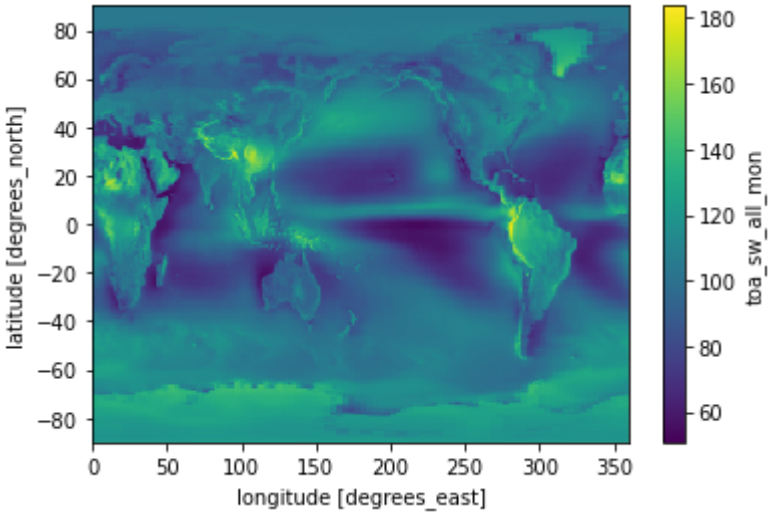


In [37]:

```
short=ds.toa_sw_all_mon.mean(dim='time')
short.plot()
```

Out[37]:

<matplotlib.collections.QuadMesh at 0x2ca3dde8c40>

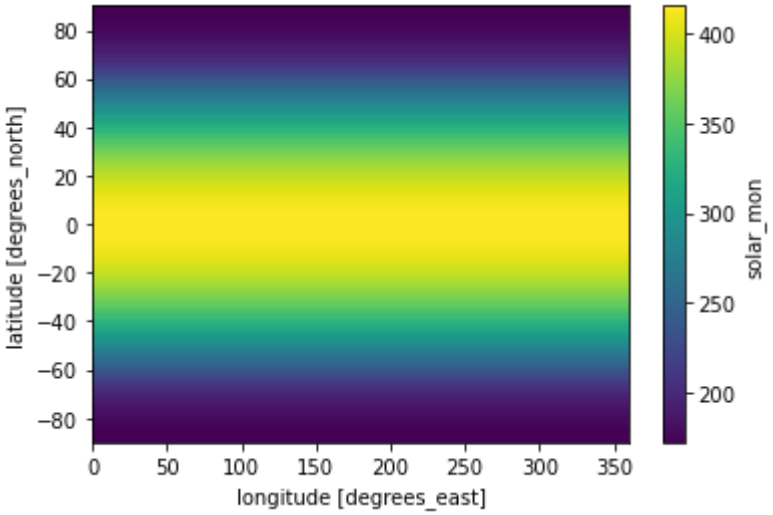


In [60]:

```
solar=ds.solar_mon.mean(dim='time')
solar.plot()
```

Out[60]:

```
<matplotlib.collections.QuadMesh at 0x188a1b6b850>
```

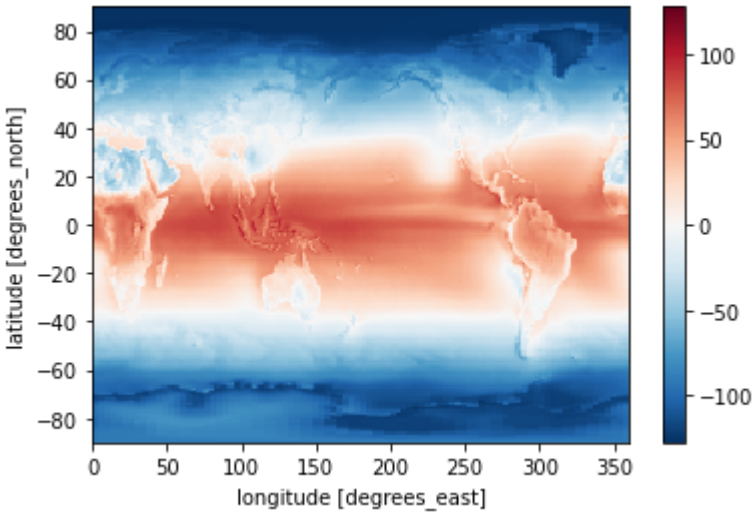


In [69]:

```
# ds1=ds.solar_mon.mean(dim="time")-ds.toa_lw_all_mon.mean(dim="time")-ds.toa_sw_all_mon.mean(dim="t
#TOA net=solar-long-short
add=solar-long-short
add.plot()

# ds1.plot()
```

Out[69]:

```
<matplotlib.collections.QuadMesh at 0x188a54dd910>
```

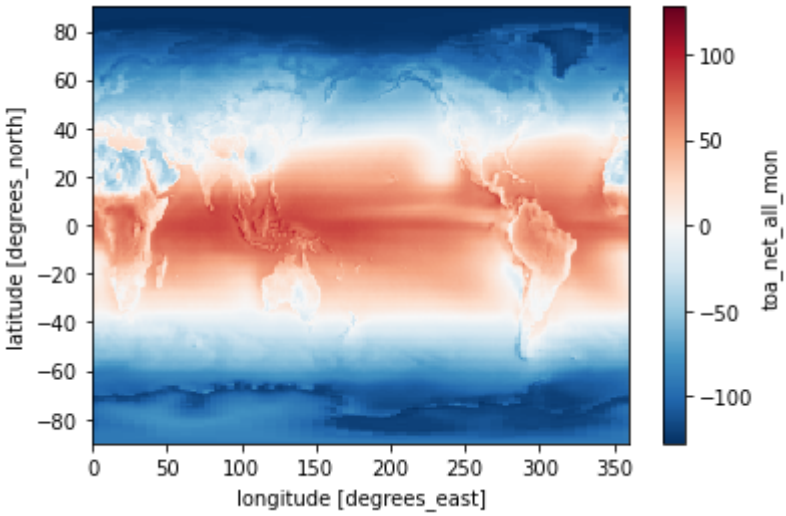


In [9]:

```
#绘制TOA net的图
toa_net=ds.toa_net_all_mon.mean(dim='time')
toa_net.plot()
#TOA的图与三种波长相叠加的图相同
```

Out[9]:

<matplotlib.collections.QuadMesh at 0x1e889339d30>



In [10]:

```
#2.2
#太阳光发射的
launch=ds.solar_mon.mean(dim='time')
#创建加权
weights=np.cos(np.deg2rad(ds.lat))
# launch
w_solar=solar.weighted(weights)
w_long=long.weighted(weights)
w_short=short.weighted(weights)

# 计算总的均值
w_solar_mean=w_solar.mean()
w_long_mean=w_long.mean()
w_short_mean=w_short.mean()
print(w_solar_mean,w_long_mean,w_short_mean)
```

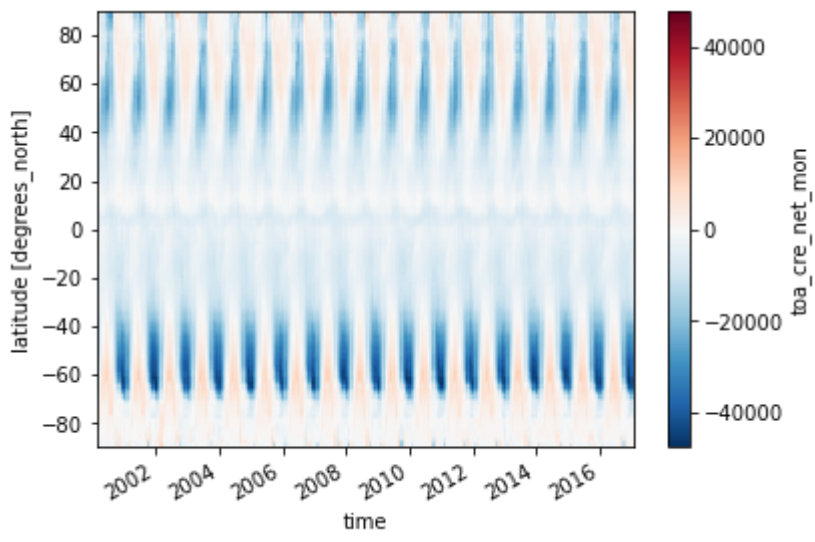
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-10-882cbe10a16e> in <module>
      5 weights=np.cos(np.deg2rad(ds.lat))
      6 # launch
----> 7 w_solar=solar.weighted(weights)
      8 w_long=long.weighted(weights)
      9 w_short=short.weighted(weights)

NameError: name 'solar' is not defined
```

In [87]:

```
#2.3
#以经度为基准进行求和并绘图
ra_sum=ds.toa_cre_net_mon.sum(dim='lon')
ra_sum.transpose().plot()
```

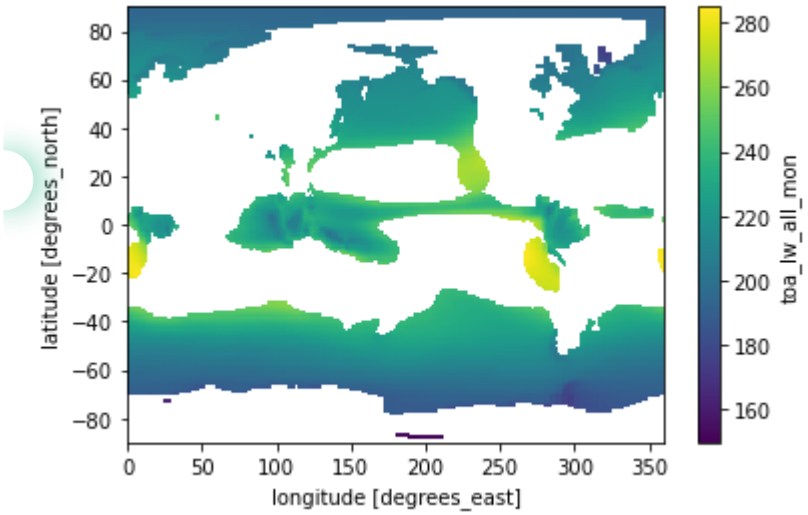
Out[87]:

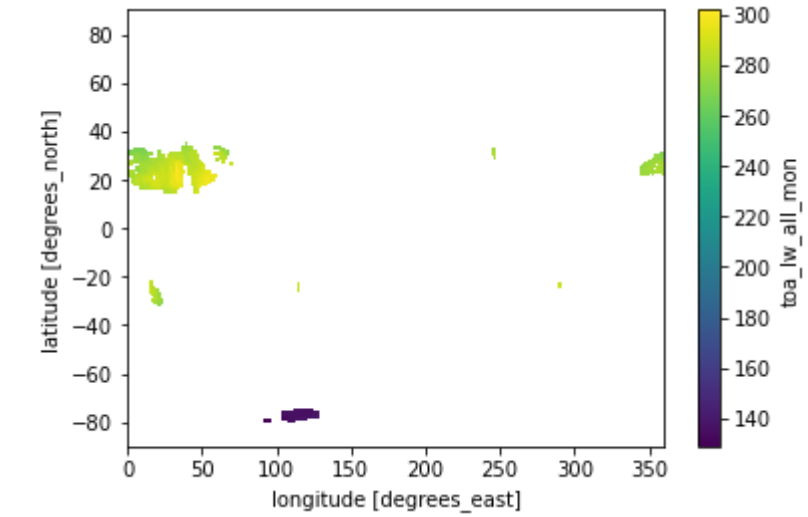
```
<matplotlib.collections.QuadMesh at 0x188a8921d30>
```

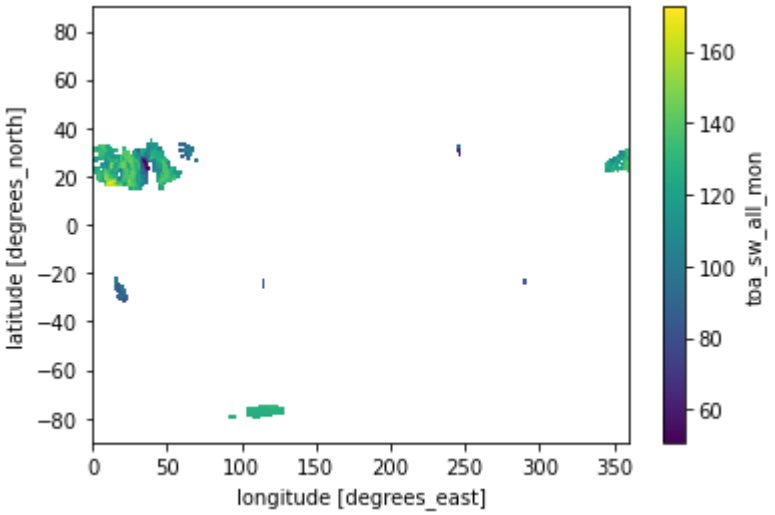


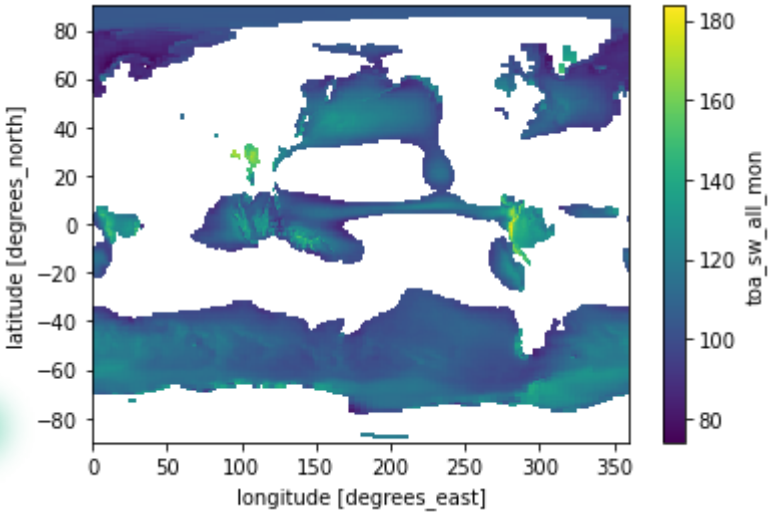
In [11]:

```
#2.4
#计算平均云面积
cldarea = ds.cldarea_total_daynight_mon.mean(dim='time')
high_area = (cldarea>=75)
low_area = (cldarea<=25)

#计算长波和短波的平均辐射值
long_v=ds.toa_lw_all_mon.mean(dim='time')
short_v=ds.toa_sw_all_mon.mean(dim='time')

#绘图
long_v.where(high_area).plot()
plt.show()
long_v.where(low_area).plot()
plt.show()
short_v.where(low_area).plot()
plt.show()
short_v.where(high_area).plot()
plt.show()
```



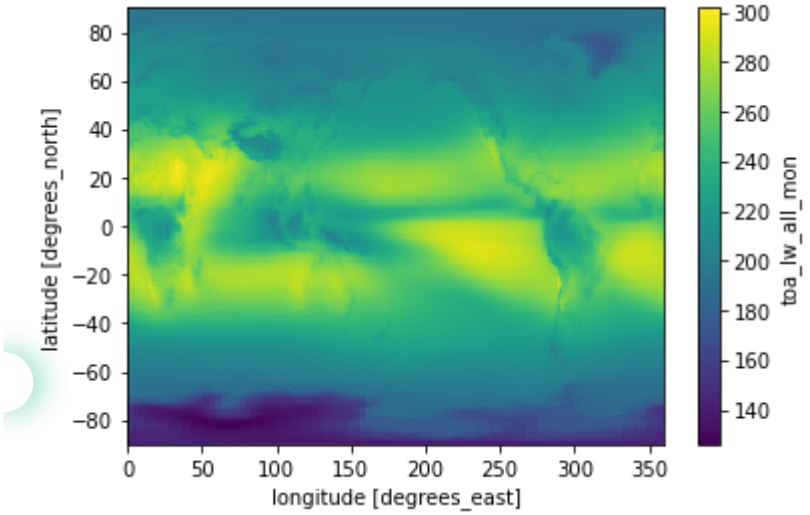


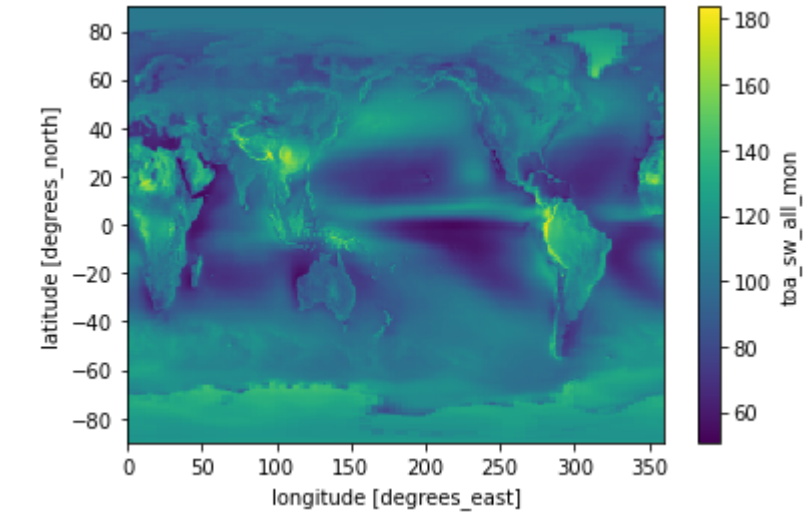

latitude [degrees_north]
longitude [degrees_east]
toa_sw_all_mon


latitude [degrees_north]
longitude [degrees_east]
toa_sw_all_mon

In [12]:

```
#2.5
#计算短波和长波全球的平均值
long_t=ds.toa_lw_all_mon.mean(dim='time')
short_t=ds.toa_sw_all_mon.mean(dim='time')

#计算云面积的平均值
cldarea = ds.cldarea_total_daynight_mon.mean(dim='time')

#绘图
long_v.where(cldarea).plot()
plt.show()

short_v.where(cldarea).plot()
plt.show()
```





In [13]:

```
# 打开文件
ds = xr.open_dataset("sst.mnmean.nc")
ds
```

Out[13]:

xarray.Dataset

▶ Dimensions: (**lat**: 89, **lon**: 180, **time**: 2014, nbnds: 2)

▼ Coordinates:

lat	(lat)	float32	88.0 86.0 84.0 ... -86.0 -88.0
lon	(lon)	float32	0.0 2.0 4.0 ... 354.0 356.0 …
time	(time)	datetime64[ns]	1854-01-01 ... 2021-10-01

▼ Data variables:

time_bnds	(time, nbnds)	float64	...
sst	(time, lat, lon)	float32	...

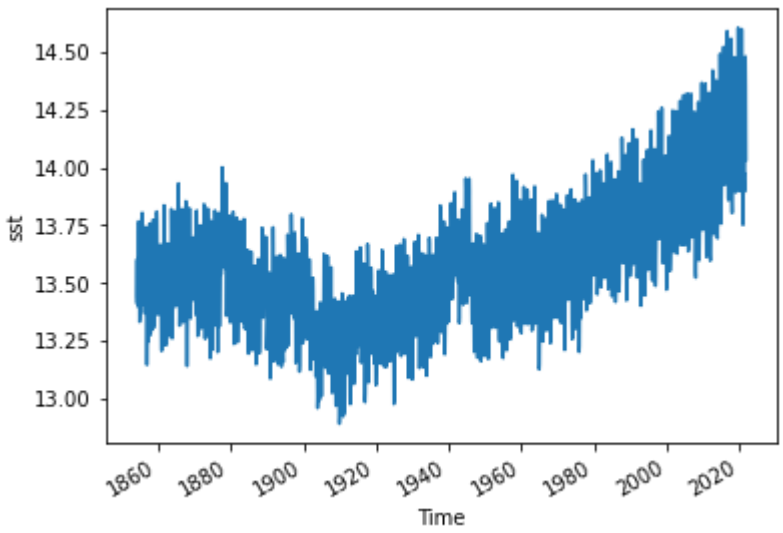
▶ Attributes: (37)

In [14]:

```
#没有找到正确的方法下载带有连续时间的数据文件，所以应用了之前老师上传的一个数据文件
#3.1
ds.sst.mean(dim=['lat','lon']).transpose().plot()
```

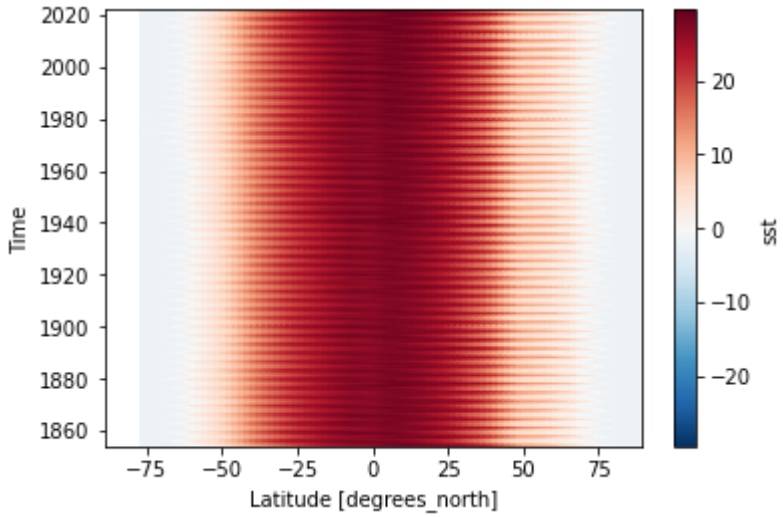
Out[14]:

```
[<matplotlib.lines.Line2D at 0x1e88ac56ee0>]
```



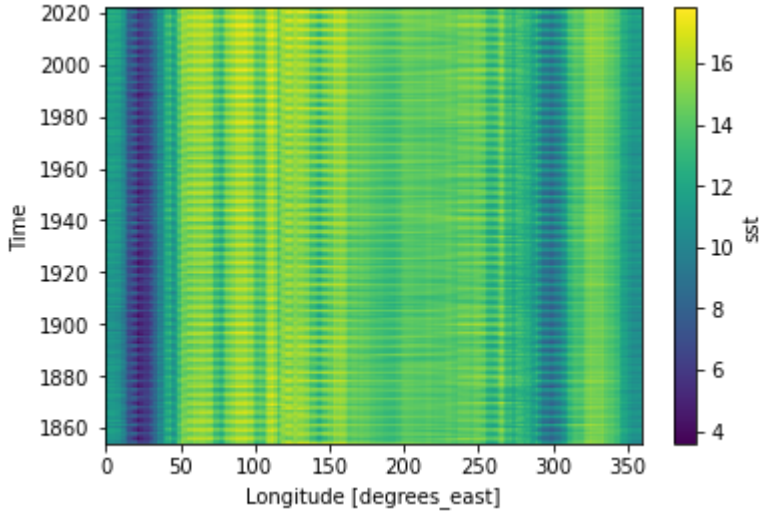
In [15]:

```
#3.2
ds.sst.mean(dim='lon').plot()
plt.show()
```



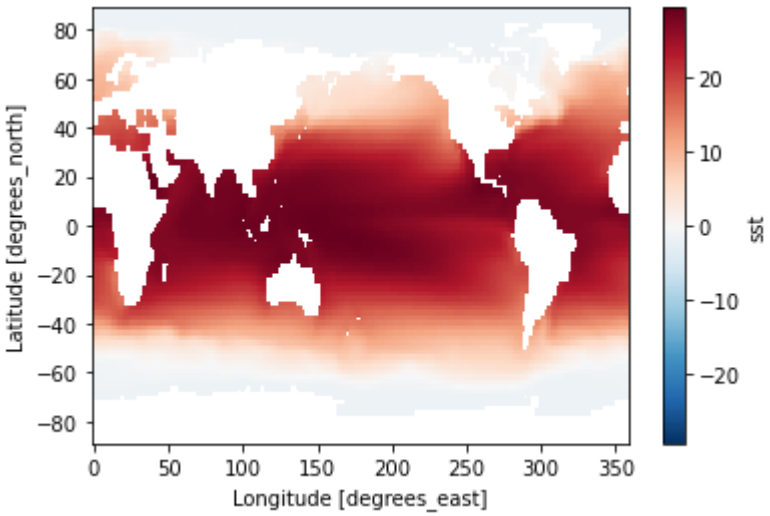
In [16]:

```
#3.2
ds.sst.mean(dim='lat').plot()
plt.show()
```



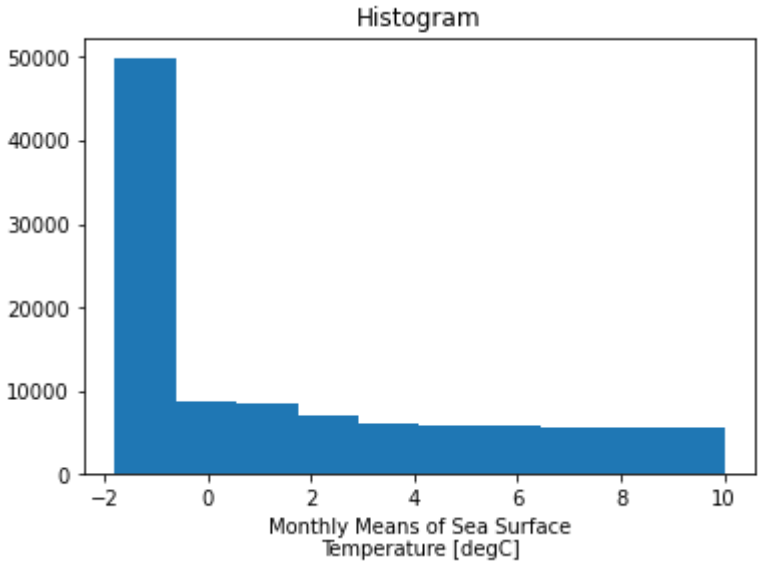
In [17]:

```
#3.2
ds.sst.mean(dim='time').plot()
plt.show()
```



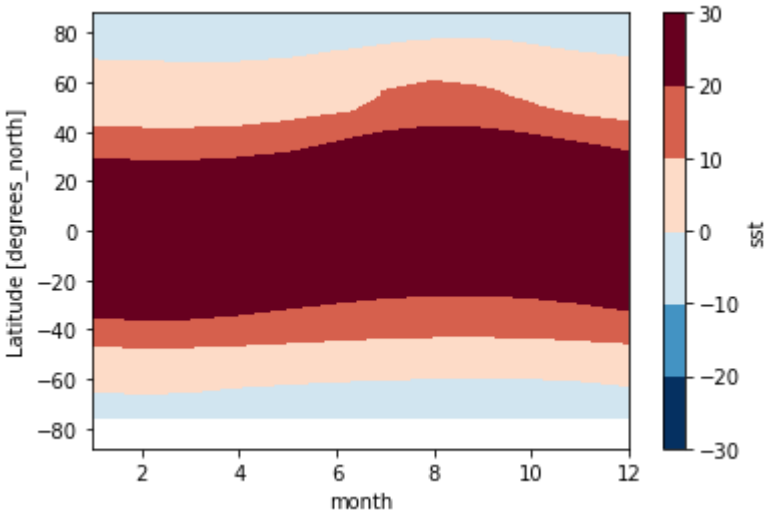
In [18]:

```
#3.2
data = ds.sst.sel(time=slice('2013-01-01', '2014-12-31'))
data1 = data.where(data < 10)
data1.plot()
plt.show()
```



In [19]:

```
# ds.sst.groupby('time.month').mean().mean(dim='lon').transpose().plot.contourf()
# plt.show()
```



In [5]:

```
#3.2
#从师姐得到一个分享的数据包
ds = xr.open_dataset("slp.mon.mean.nc")
ds
```

Out[5]:

xarray.Dataset

▶ Dimensions: (**lat**: 73, **lon**: 144, **time**: 886)

▼ Coordinates:

lat	(lat)	float32	90.0 87.5 85.0 ... -87.5 -90.0	
lon	(lon)	float32	0.0 2.5 5.0 ... 352.5 355.0 35…	
time	(time)	datetime64[ns]	1948-01-01 ... 2021-10-01	

▼ Data variables:

slp	(time, lat, lon)	float32	...	

▼ Attributes:

description :	Data is from NMC initialized reanalysis (4x/day). These are the 0.9950 sigma level values.
platform :	Model
Conventions :	COARDS
NCO :	20121012
history :	Thu May 4 18:12:35 2000: ncrcat -d time,0,622 /Datasets/ncep.reanalysis.derived/surface/slp.mon.mean.nc ./surface/slp.mon.mean.nc Mon Jul 5 23:22:35 1999: ncrcat slp.mon.mean.nc /Datasets/ncep.reanalysis.derived/surface/slp.mon.mean.nc /dm/dmwork/nmc.rean.ingest/combinedMMs/slp.mon.mean.nc /home/hoop/crdc/cpreanjuke2farm/cpreanjuke2farm Thu Oct 26 23:42:16 1995 from pre.sig995.85.nc created 95/02/06 by Hoop (netCDF2.3) Converted to chunked, deflated non-packed NetCDF4 2014/09
title :	monthly mean slp from the NCEP Reanalysis
dataset_title :	NCEP-NCAR Reanalysis 1
References :	http://www.psl.noaa.gov/data/gridded/data.ncep.reanalysis.derived.html

In [19]:

```
#3.2的图
ds.slp.mean(dim='lon').plot()
plt.show()

ds.slp.mean(dim='time').plot()
plt.show()
ds.slp.groupby('time.month').mean().mean(dim='lon').transpose().plot.contourf()
plt.show()

group_data = ds.slp.groupby('time.month')
sst_anom = group_data - group_data.mean(dim='time')
sst_anom.sel(lon=114.55+180, lat=22.5, method='nearest').plot()
plt.show()

ds.slp.mean(dim=['lat','lon']).transpose().plot()
plt.show()
data = ds.slp.sel(time=slice('2013-01-01', '2014-12-31'))
data1 = data.where(data < 10)
data1.plot()
plt.show()

# sst_anom
```