In [1]:

```
import numpy as np
import pandas as pd
import xarray as xr
from matplotlib import pyplot as plt
%matplotlib inline
```

```
C:\Users\CKZ\Anaconda3\lib\site-packages\dask\config.py:131: YAMLLoadWarning: callin
g yaml.load() without Loader=... is deprecated, as the default Loader is unsafe. Ple
ase read https://msg.pyyaml.org/load (https://msg.pyyaml.org/load) for full details.
  data = yaml.load(f.read()) or {}
C:\Users\CKZ\Anaconda3\lib\site-packages\distributed\config.py:20: YAMLLoadWarning:
 calling yaml.load() without Loader=... is deprecated, as the default Loader is unsa
fe. Please read https://msg.pyyaml.org/load (https://msg.pyyaml.org/load) for full d
etails.
  defaults = yaml.load(f)
```

In [2]:

```
ds = xr.open_dataset('NOAA_NCDC_ERSST_v3b_SST.nc', engine = 'netcdf4')
ds_region = ds.sel(lon = slice(190, 240), lat = slice(-5, 5))

group_data = ds_region.sst.groupby('time.month')
sst_anom = group_data - group_data.mean(dim = 'time')
sst_anom
```

Out[2]:

xarray.DataArray 'sst' (**time**: 684, **lat**: 5, **lon**: 26)

```
array([[[-0.43157768, -0.41846275, -0.39795303, ..., -0.2116642 ,
         -0.23776245, -0.24401474],
        [-0.41259003, -0.4067192 , -0.3875141 , ..., -0.52064896,
         -0.5346451 , -0.51997185],
        [-0.40932274, -0.39743805, -0.36237717, ..., -0.6373882 ,
         -0.6171951 , -0.583725  ],
        [-0.4140854 , -0.37909317, -0.3215618 , ..., -0.43292618,
         -0.38404274, -0.3352623 ],
        [-0.5043678 , -0.43894005, -0.3710251 , ..., -0.17453575,
         -0.11044502, -0.06918144]],

       [[-0.5374584 , -0.52739716, -0.50823593, ..., -0.40254593,
         -0.44382668, -0.45287704],
        [-0.55093956, -0.539135  , -0.51673317, ..., -0.6660595 ,
         -0.7127285 , -0.710968  ],
        [-0.61242104, -0.5959244 , -0.5572338 , ..., -0.7235069 ,
         -0.7326374 , -0.73106194],
        [-0.6798363 , -0.6483364 , -0.5889931 , ..., -0.5397434 ,
         -0.50793266, -0.49977684],
        [-0.7830448 , -0.7286701 , -0.6683655 , ..., -0.33967972,
...
         -0.2555828 , -0.13972664],
        [-0.989378  , -1.0497723 , -1.0954857 , ..., -0.86087227,
         -0.7690697 , -0.65498734],
        [-1.1887245 , -1.252285  , -1.3029232 , ..., -1.0460625 ,
         -0.9661274 , -0.8785801 ],
        [-1.002367  , -1.0756893 , -1.1325111 , ..., -0.7207298 ,
         -0.6597252 , -0.5900669 ],
        [-0.5770798 , -0.65514374, -0.72174263, ..., -0.4353485 ,
         -0.36265945, -0.28103828]],

       [[-0.3578701 , -0.41542053, -0.47110367, ..., -0.2400589 ,
         -0.1464405 , -0.03788376],
        [-0.7678585 , -0.83501625, -0.9024124 , ..., -0.727829  ,
         -0.61603355, -0.48027992],
        [-0.96187973, -1.0445309 , -1.1224213 , ..., -0.9327831 ,
         -0.81235695, -0.6655674 ],
        [-0.82112694, -0.9206734 , -1.0085506 , ..., -0.6531601 ,
```

```
       -0.5626869 ,  -0.4374504 ],
      [-0.4864292 ,  -0.5823746 ,  -0.6702862 ,  ...,  -0.36221695,
       -0.30041504,  -0.1987915 ]]], dtype=float32)
```

▼Coordinates:

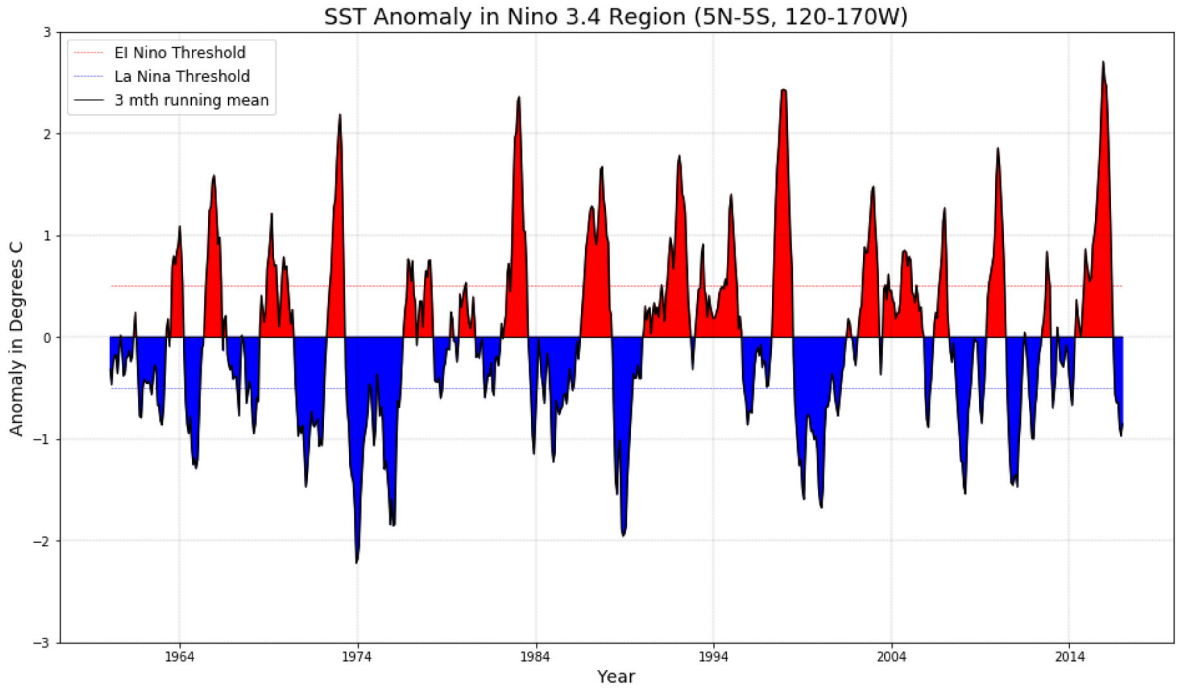
lat	(lat)	float32	-4.0 -2.0 0.0 2.0 4.0	
lon	(lon)	float32	190.0 192.0 194.0 ... 238.0 240.0	
time	(time)	datetime64[ns]	1960-01-15 ... 2016-12-15	
month	(time)	int64	1 2 3 4 5 6 7 ... 6 7 8 9 10 11 12	

► Attributes: (0)

In [4]:

```
sst_anom = np.nanmean(sst_anom, axis=(1,2))
time = pd.date_range(start = '1960-01', periods = 684, freq = 'm')

plt.figure(figsize = (16, 9))
plt.plot(time, sst_anom, color = 'k')
plt.title('SST Anomaly in Nino 3.4 Region (5N-5S, 120-170W)',fontsize = 18)
plt.xlabel('Year', fontsize = 14)
plt.ylabel('Anomaly in Degrees C', fontsize = 14)
plt.ylim(-3, 3)

plt.grid(linestyle = '--', linewidth = 0.3, alpha = 0.5, color = 'k')

plt.hlines(y = 0.5, xmin = time[0], xmax = time[-1], color = 'r', linestyles = '--', lw = 0.5, labe
plt.hlines(y = -0.5, xmin = time[0], xmax = time[-1], color = 'b', linestyles = '--', lw = 0.5, lab
plt.hlines(y = 0, xmin = time[0], xmax = time[-1], color = 'k', linestyles = 'solid', lw = 1, label

plt.legend(loc = 'best', fontsize = 12)

plt.fill_between(time, 0, sst_anom, where = (sst_anom > 0), color = 'r')
plt.fill_between(time, 0, sst_anom, where = (sst_anom < 0), color = 'b')

plt.show()
```

```
C:\Users\CKZ\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py:2064: FutureWa
rning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated an
d will be removed in a future version.  Convert to a numpy array before indexing ins
tead.
  x[:, None]
C:\Users\CKZ\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:248: FutureWarnin
g: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and wi
ll be removed in a future version.  Convert to a numpy array before indexing instea
d.
  x = x[:, np.newaxis]
```



In [ ]:

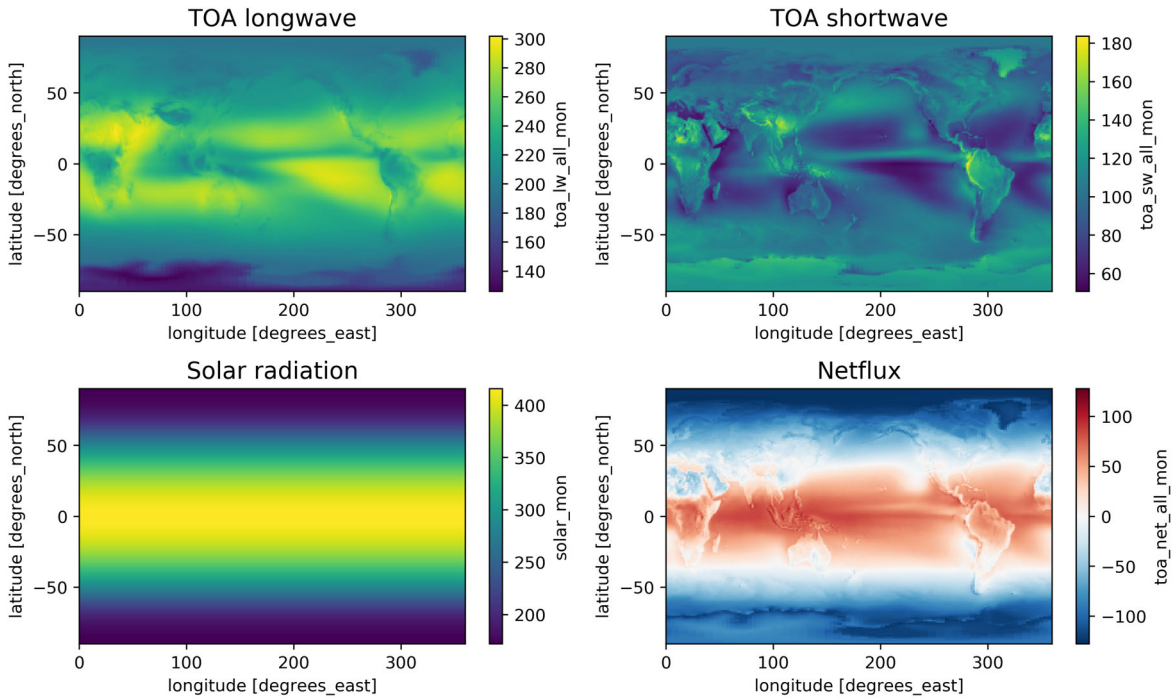
In [2]:

```
import numpy as np
import pandas as pd
import xarray as xr
from matplotlib import pyplot as plt
import nc_time_axis
%matplotlib inline
```

```
C:\Users\CKZ\Anaconda3\lib\site-packages\dask\config.py:131: YAMLLoadWarning: callin
g yaml.load() without Loader=... is deprecated, as the default Loader is unsafe. Ple
ase read https://msg.pyyaml.org/load (https://msg.pyyaml.org/load) for full details.
  data = yaml.load(f.read()) or {}
C:\Users\CKZ\Anaconda3\lib\site-packages\distributed\config.py:20: YAMLLoadWarning:
 calling yaml.load() without Loader=... is deprecated, as the default Loader is unsa
fe. Please read https://msg.pyyaml.org/load (https://msg.pyyaml.org/load) for full d
etails.
  defaults = yaml.load(f)
```

In [8]:

```
ds = xr.open_dataset("CERES_EBAF-TOA_200003-201701.nc", engine = 'netcdf4')

fig, axes = plt.subplots(2, 2, figsize = (10, 6), sharex = False, sharey = False, dpi = 400)
ds.toa_lw_all_mon.mean('time').plot(ax = axes[0, 0])
ds.toa_sw_all_mon.mean('time').plot(ax = axes[0, 1])
ds.solar_mon.mean('time').plot(ax = axes[1, 0])
ds.toa_net_all_mon.mean('time').plot(ax = axes[1, 1])

axes[0,0].set_title('TOA longwave', fontsize = 14)
axes[0,1].set_title('TOA shortwave', fontsize = 14)
axes[1,0].set_title('Solar radiation', fontsize = 14)
axes[1,1].set_title('Netflux', fontsize = 14)

plt.tight_layout()
```

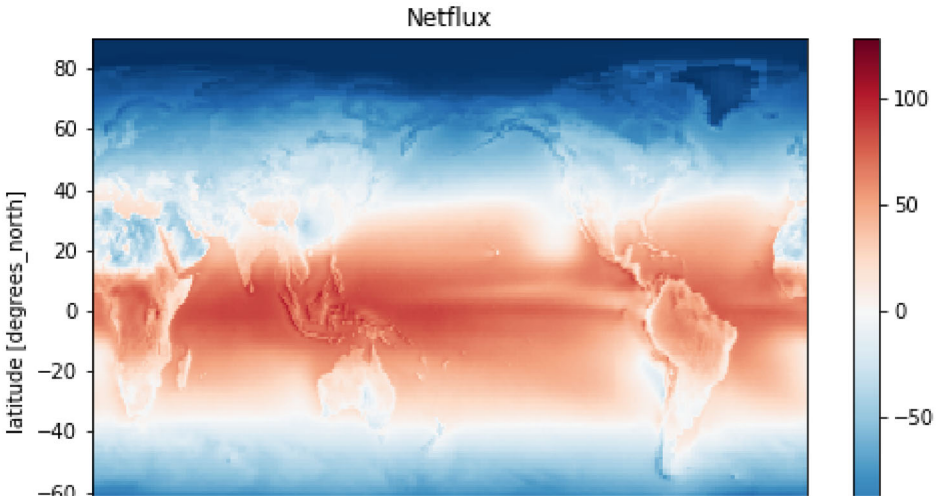


In [14]:

```
da_netflux = ds.solar_mon - ds.toa_lw_all_mon - ds.toa_sw_all_mon
da_netflux.mean('time').plot(figsize = (8, 5))
plt.title('Netflux')
```

Out[14]:

Text(0.5,1,'Netflux')



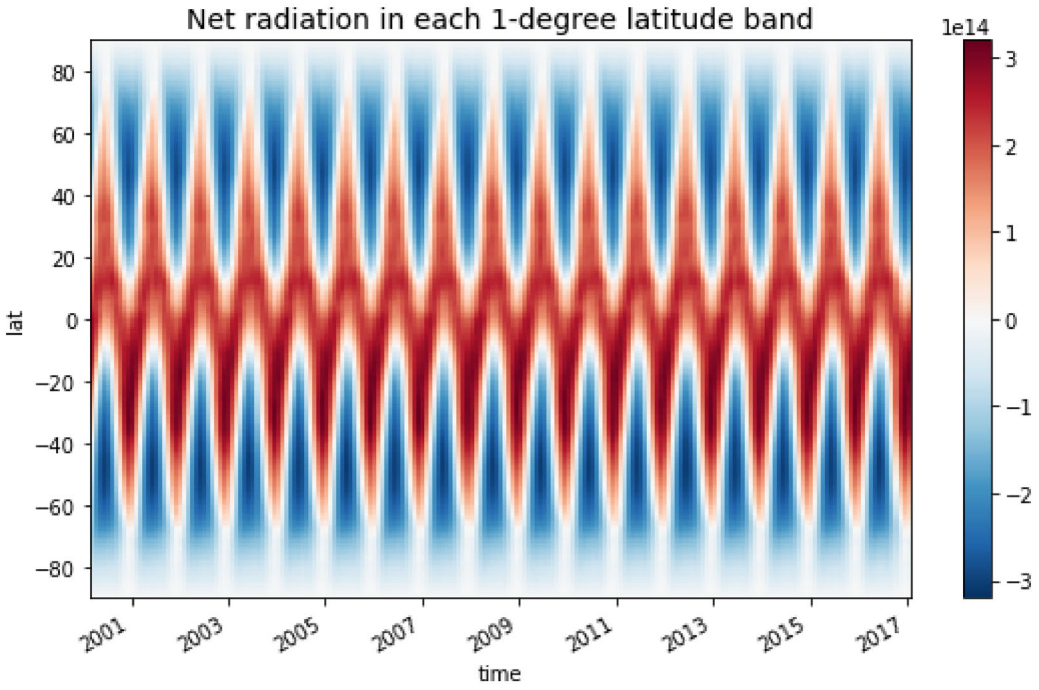
In [23]:

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

weighted_solar = ds.solar_mon.weighted(weights)
weighted_lw = ds.toa_lw_all_mon.weighted(weights)
weighted_sw = ds.toa_sw_all_mon.weighted(weights)

print(weighted_solar.mean(dim = ('lon', 'lat','time')).values)
print(weighted_lw.mean(dim = ('lon', 'lat','time')).values)
print(weighted_sw.mean(dim = ('lon', 'lat','time')).values)
```

340.28326598091286
240.26691846331997
99.13806041149408

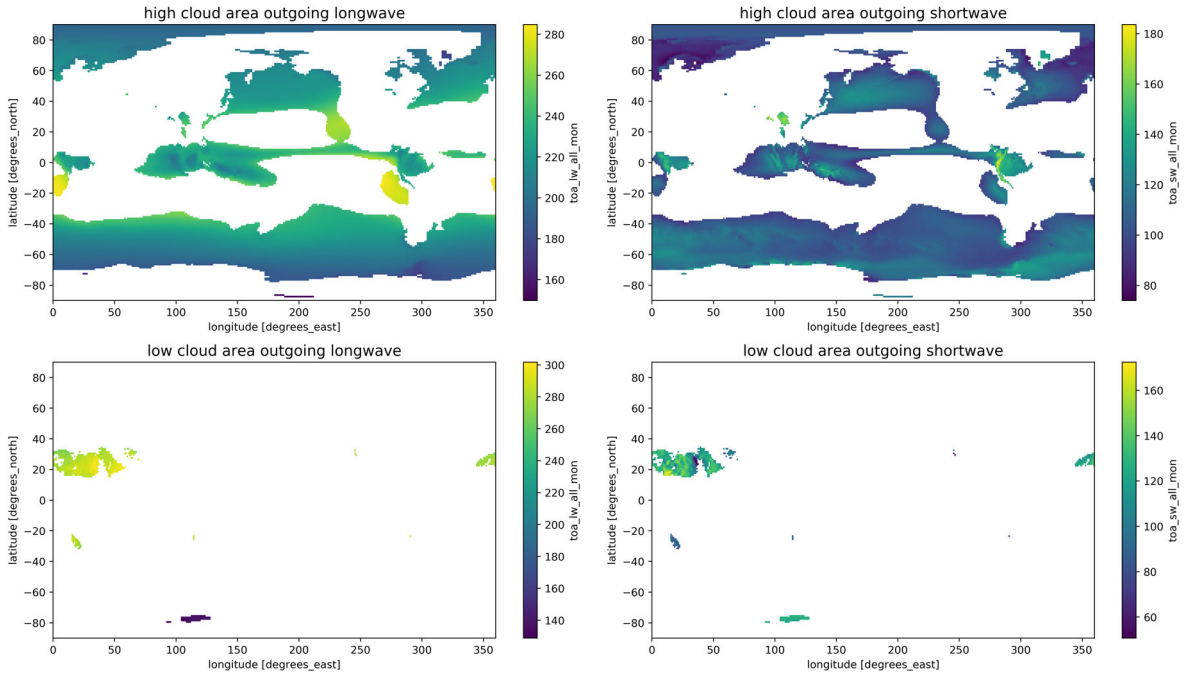
In [25]:

```
weighted_net = ds.toa_net_all_mon.weighted(weights)

R = 6371393
pi = np.pi
weighted_lat = 4*pi*R**2*weights/180
total_amount = weighted_net.mean(dim='lon') * weighted_lat
total_amount.transpose().plot(figsize=(8, 5))
plt.title('Net radiation in each 1-degree latitude band', fontsize = 14)
plt.tight_layout()
```



In [27]:

```
cldarea = ds.cldarea_total_daynight_mon.mean(dim='time').values

fig, axes = plt.subplots(2, 2, figsize=(16, 9), dpi = 400)

ds.toa_lw_all_mon.mean(dim='time').where(cldarea >= 75).plot(ax=axes[0, 0])
ds.toa_sw_all_mon.mean(dim='time').where(cldarea >= 75).plot(ax=axes[0, 1])
ds.toa_lw_all_mon.mean(dim='time').where(cldarea <= 25).plot(ax=axes[1, 0])
ds.toa_sw_all_mon.mean(dim='time').where(cldarea <= 25).plot(ax=axes[1, 1])

axes[0, 0].set_title('high cloud area outgoing longwave', fontsize = 14)
axes[0, 1].set_title('high cloud area outgoing shortwave', fontsize = 14)
axes[1, 0].set_title('low cloud area outgoing longwave', fontsize = 14)
axes[1, 1].set_title('low cloud area outgoing shortwave', fontsize = 14)

plt.tight_layout()
```



In [28]:

```
print('high cloud longwave:',np.mean(ds.toa_lw_all_mon.mean(dim='time')))
print('high cloud shortwave:',np.mean(ds.toa_sw_all_mon.mean(dim='time')))
print('low cloud longwave:',np.mean(ds.toa_lw_all_mon.mean(dim='time')))
print('low cloud shortwave:',np.mean(ds.toa_sw_all_mon.mean(dim='time')))
```

```
high cloud longwave: <xarray.DataArray 'toa_lw_all_mon' ()>
array(224.75517, dtype=float32)
high cloud shortwave: <xarray.DataArray 'toa_sw_all_mon' ()>
array(102.30433, dtype=float32)
low cloud longwave: <xarray.DataArray 'toa_lw_all_mon' ()>
array(224.75517, dtype=float32)
low cloud shortwave: <xarray.DataArray 'toa_sw_all_mon' ()>
array(102.30433, dtype=float32)
```

In [5]:

```
import numpy as np
import pandas as pd
import xarray as xr
from matplotlib import pyplot as plt
import nc_time_axis
%matplotlib inline
```

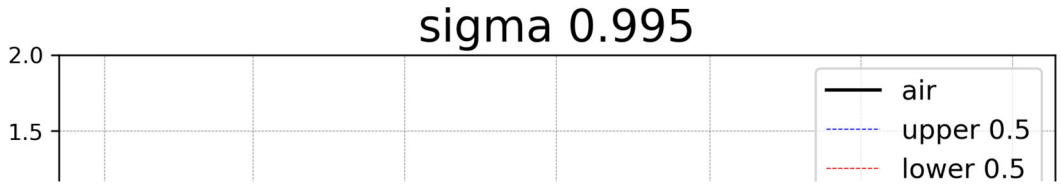
In [15]:

```
ds = xr.open_dataset('air temperature anomaly 2012.nc', engine = 'netcdf4')

ds_air = ds.air.sel(lon = slice(120, 170), lat = slice(-30, 30))
group_data = ds.air.groupby('time.month')
air_anom = group_data - group_data.mean(dim = 'time')
time = pd.date_range(start = '2012-01-01', periods = 366, freq = 'd')

fig,ax = plt.subplots(1, 1, figsize = (8, 5), dpi = 300)
ax.plot(time, new_air_anom, color = 'k')

ax.set_ylabel('Anomaly Degrees C', color = 'k', fontsize = 15)
ax.set_xlabel('Time', color = 'k', fontsize = 15)
ax.set_title("sigma 0.995", fontsize = 20)

ax.grid(linestyle = '--',linewidth = 0.3, alpha = 0.5, color = 'k')

ax.hlines(y = 0.5, xmin = time[0], xmax = time[-1], color = 'b',linestyles = '--',lw = 0.5,label =
ax.hlines(y = -0.5, xmin = time[0], xmax = time[-1], color = 'r',linestyles = '--',lw = 0.5,label =
ax.hlines(y = 0, xmin = time[0], xmax = time[-1], color = 'k', linestyles = 'solid', lw = 1, label

ax.set_ylim(-2, 2)
ax.legend(loc = 'best', fontsize = 12)

ax.fill_between(time, 0, new_air_anom, where = (new_air_anom > 0), color = 'b')
ax.fill_between(time, 0, new_air_anom, where = (new_air_anom < 0), color = 'r')
```

```
  x = x[:, np.newaxis]
```

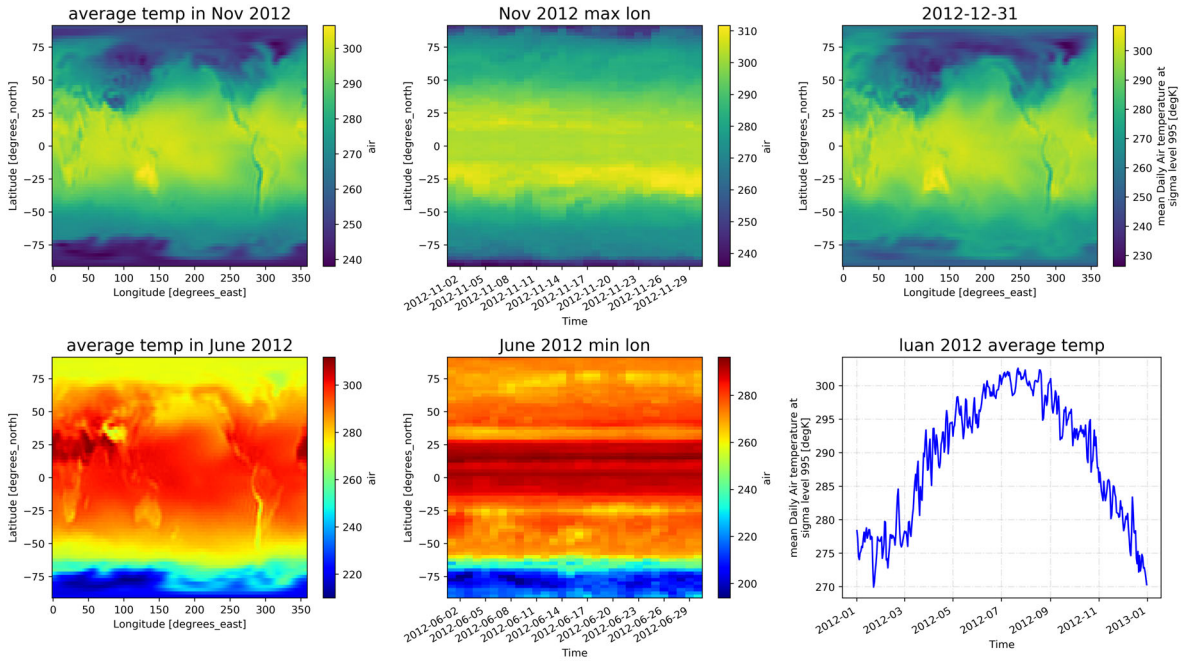
Out[15]:

<matplotlib.collections.PolyCollection at 0x13e27868780>



In [12]:

```
fig, axes = plt.subplots(2, 3, figsize = (16, 9), sharex = False, sharey = False, dpi = 400)

ds_air_Nov = ds.air.sel(time = slice('2012-11-01', '2012-11-30'))
ds_air_June = ds.air.sel(time = slice('2012-06-01', '2012-06-30'))
ds_air_luan = ds.air.sel(lon = '116', lat = '31', method = 'nearest')
ds_air_day = ds.air.isel(time = -1)

ds_air_Nov.mean('time').plot(ax = axes[0,0])
ds_air_Nov.max('lon').transpose().plot(ax = axes[0,1])
ds_air_day.plot(ax = axes[0, 2])
ds_air_June.mean('time').plot(ax = axes[1,0], cmap = 'jet')
ds_air_June.min('lon').transpose().plot(ax = axes[1,1], cmap = 'jet', alpha = 1)
ds_air_luan.plot(ax = axes[1, 2], c = 'b')

axes[1,2].grid(linestyle = '-.', linewidth = 1, alpha = 0.3)

axes[0,0].set_title('average temp in Nov 2012', fontsize = 16)
axes[0,1].set_title('Nov 2012 max lon', fontsize = 16)
axes[0,2].set_title('2012-12-31', fontsize = 16)
axes[1,0].set_title('average temp in June 2012', fontsize = 16)
axes[1,1].set_title('June 2012 min lon', fontsize = 16)
axes[1,2].set_title('luan 2012 average temp', fontsize = 16)

plt.tight_layout()
```



In [ ]:

In [ ]: