```
# Import modules
import numpy as np
import xarray as xr
import pandas as pd
import netCDF4
import matplotlib.pyplot as plt
import matplotlib.ticker as mticker
%matplotlib inline
```

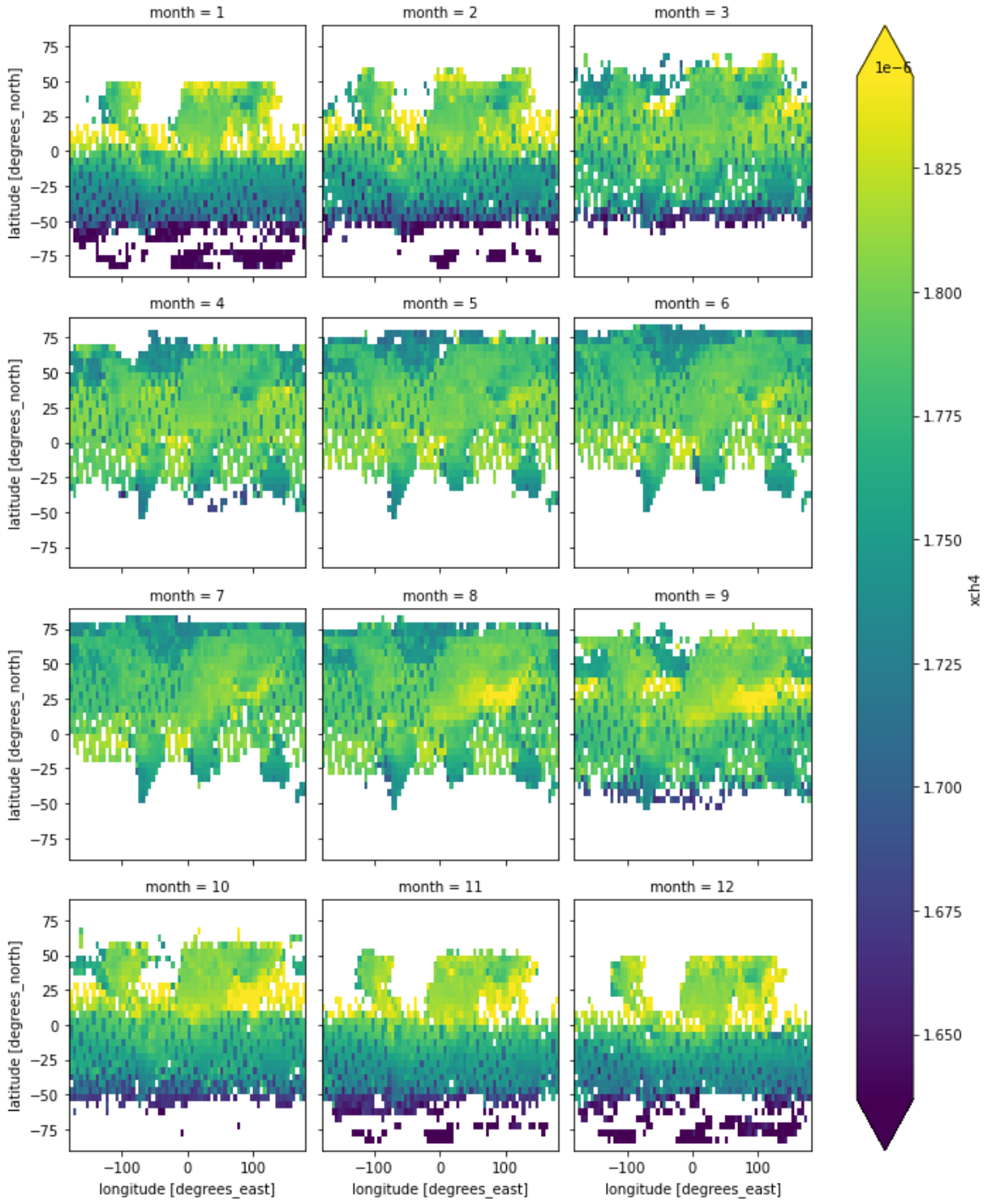
1.Global methane levels

ex1.1

```
methane_data=xr.open_dataset("200301_202006-C3S-L3_GHG-PRODUCTS-OBS4MIPS-MERGED-v4.3.nc",engine="netcdf4")
```

```
methane_data.xch4.sel(time=slice("2003","2020")).groupby("time.month").mean().plot(col="month",col_wrap=3,robust=True)
```

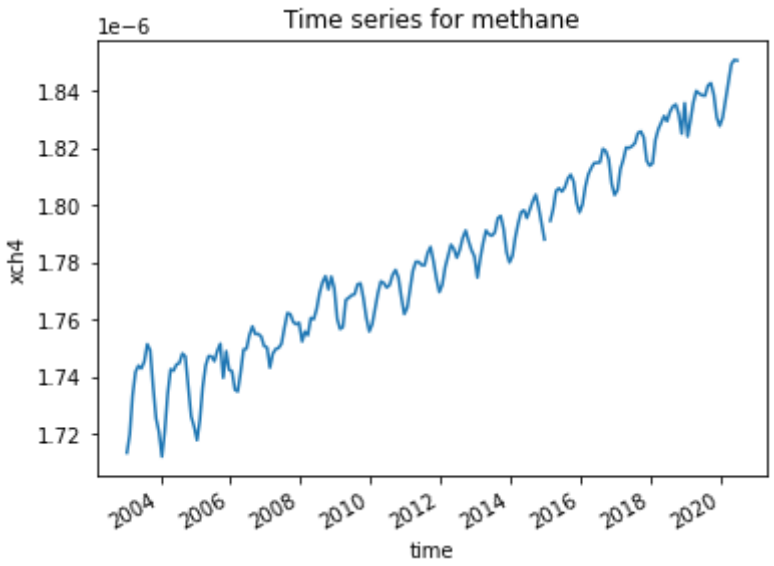
```
<xarray.plot.facetgrid.FacetGrid at 0x2600bc86b20>
```



ex1.2

```
# adding weights factor for the data
weights=np.cos(np.deg2rad(methane_data.lat))
xch4_weighted=methane_data.xch4.weighted(weights)
# plot the
xch4_weighted.mean(dim=("lat","lon")).sel(time=slice("2003-01","2020-
06")).plot()
plt.title("Time series for methane")
```

```
Text(0.5, 1.0, 'Time series for methane')
```



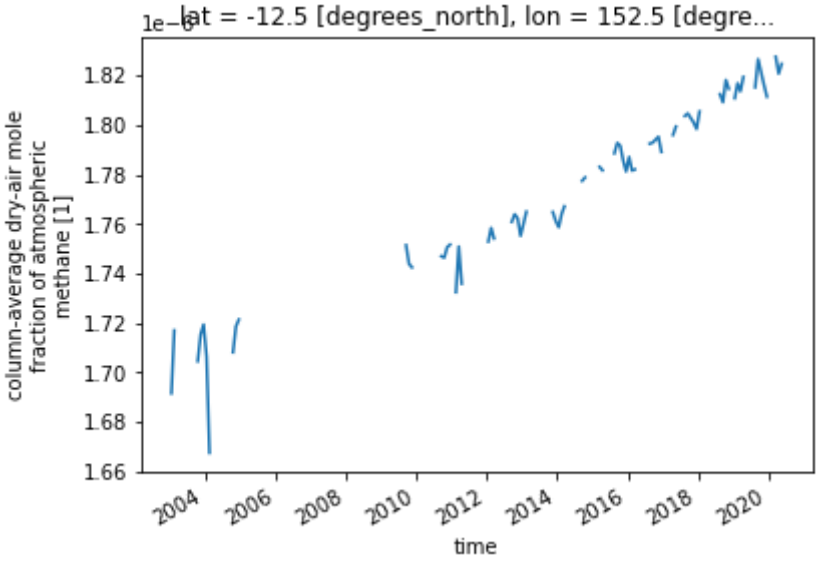
The plot shows the concentration of methane is increasing years by years

ex1.3

```
methane_data.xch4.sel(lat=-15,lon=150,method="nearest").plot()
```

```
D:\Anaconda3\lib\site-packages\xarray\core\indexes.py:234: FutureWarning:
Passing method to Float64Index.get_loc is deprecated and will raise in a future
version. Use index.get_indexer([item], method=...) instead.
  indexer = self.index.get_loc(
D:\Anaconda3\lib\site-packages\xarray\core\indexes.py:234: FutureWarning:
Passing method to Float64Index.get_loc is deprecated and will raise in a future
version. Use index.get_indexer([item], method=...) instead.
  indexer = self.index.get_loc(
```

```
[<matplotlib.lines.Line2D at 0x25f832705b0>]
```



in this point, methane still increases on the whole, but many missing datas makes the curve incomplete.

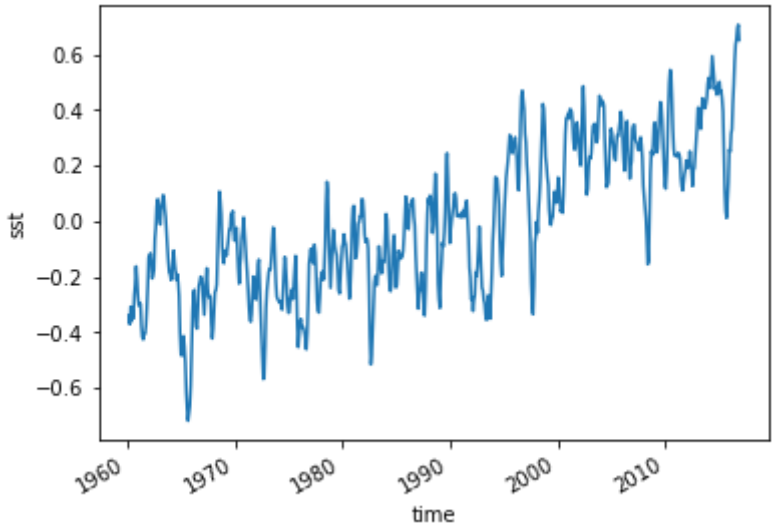
2. nino 3.4 index

2.1 get anomalies

```
Nino=xr.open_dataset("NOAA_NCDC_ERSST_v3b_SST.nc",engine="netcdf4")
```

```
# Group the data by month
group_data=Nino.sst.sel(lon=slice(120,170),lat=slice(-5,5)).groupby("time.month"
)
sst_anom=group_data-group_data.mean(dim="time")
sst_anom_rolling=sst_anom.rolling(time=3,center=True).mean()
# plot anomalies
sst_anom_rolling.mean(dim=["lat","lon"]).plot()
```

```
[<matplotlib.lines.Line2D at 0x26011703a90>]
```



2.2 visualize Nino3.4

```
#Use resample() to get a freqency of 3 months and mean() to get values

# get the judgement condition for EIni and Lani
Einino=[False for i in range(len(sst_anom))]
Lanino=[False for i in range(len(sst_anom))]
for i in range(4,len(sst_anom)):
    Einino_el=(sst_anom_rolling[i]>0.5)
    Lanino_el=(sst_anom_rolling[i]<-0.5)
    for j in range(4):
        Einino_el=Einino_el and (sst_rolling[i-j]>0.5)
        Lanino_el=Lanino_el and (sst_rolling[i-j]<-0.5)
    Einino[i]=Einino_el
    Lanino[i]=Lanino_el
```

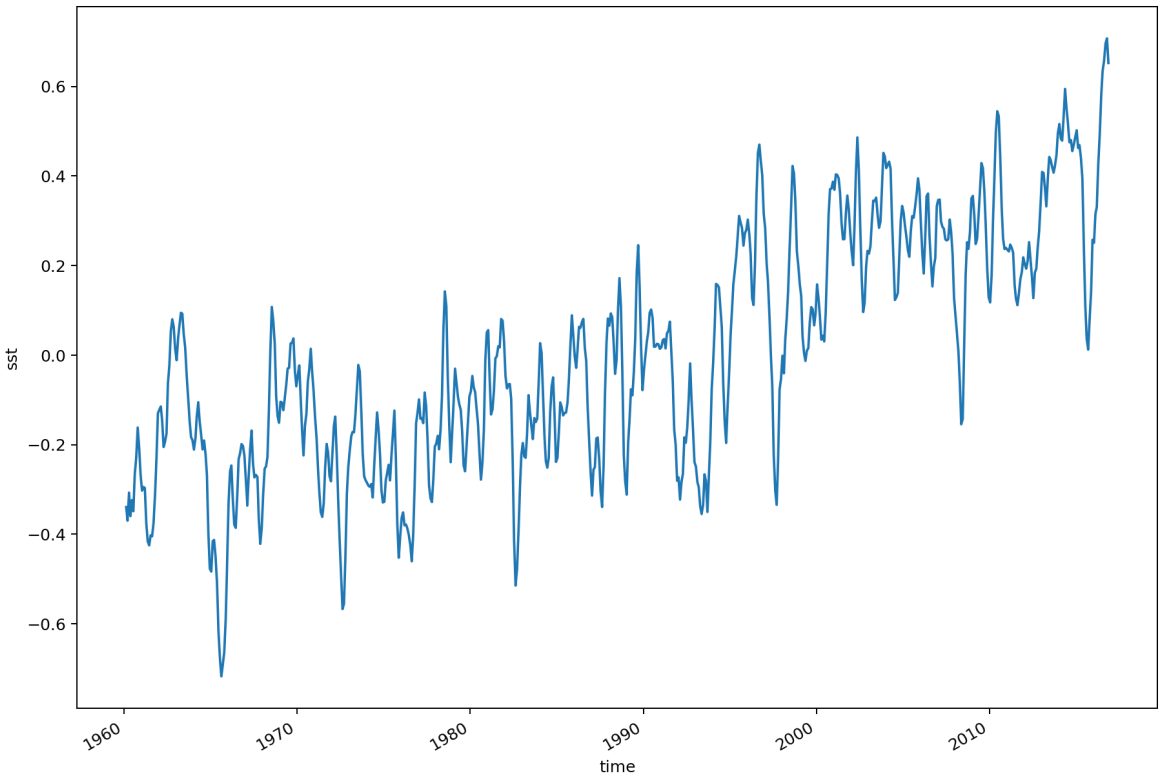
```
---------------------------------------------------------------------------

ValueError                                Traceback (most recent call last)

Input In [168], in <cell line: 6>()
      8 Lanino_el=(sst_anom_rolling[i]<-0.5)
      9 for j in range(4):
---> 10     Einino_el=Einino_el and (sst_rolling[i-j]>0.5)
     11     Lanino_el=Lanino_el and (sst_rolling[i-j]<-0.5)
     12 Einino[i]=Einino_el
```

```
File D:\Anaconda3\lib\site-packages\xarray\core\common.py:136, in
AbstractArray.__bool__(self)
    135 def __bool__(self: Any) -> bool:
--> 136     return bool(self.values)
```

```
ValueError: The truth value of an array with more than one element is ambiguous.
Use a.any() or a.all()
```

```
# Then use the
plt.figure(figsize=(12,8),dpi=180)
sst_anom_rolling.mean(dim=["lat","lon"]).plot()
```

```
[<matplotlib.lines.Line2D at 0x260116cb160>]
```




```
  Input In [178]
    sst_anom_rolling.plot(dim="lon","lat")
                                          ^
SyntaxError: positional argument follows keyword argument
```

```
# plot averaged global soilw at a specific point
soil_data.sel(lon=114.5,lat=22.5,method="nearest").plot(maker="o",size=10)
```

```
D:\anaconda3\lib\site-packages\xarray\core\indexes.py:234: FutureWarning:
Passing method to Float64Index.get_loc is deprecated and will raise in a future
version. Use index.get_indexer([item], method=...) instead.
  indexer = self.index.get_loc(
D:\anaconda3\lib\site-packages\xarray\core\indexes.py:234: FutureWarning:
Passing method to Float64Index.get_loc is deprecated and will raise in a future
version. Use index.get_indexer([item], method=...) instead.
  indexer = self.index.get_loc(
```

```
---------------------------------------------------------------------------

ValueError                                Traceback (most recent call last)

Input In [32], in <cell line: 2>()
      1 # plot averaged global soilw at a specific point
----> 2
soil_data.sel(lon=114.5,lat=22.5,method="nearest").plot(maker="o",size=10)
```

```
File D:\anaconda3\lib\site-packages\xarray\plot\dataset_plot.py:186, in
_Dataset_PlotMethods.__call__(self, *args, **kwargs)
    185 def __call__(self, *args, **kwargs):
--> 186     raise ValueError(
    187         "Dataset.plot cannot be called directly. Use "
    188         "an explicit plot method, e.g. ds.plot.scatter(...)"
    189     )
```

```
ValueError: Dataset.plot cannot be called directly. Use an explicit plot method,
e.g. ds.plot.scatter(...)
```

```
# show the difference of weight and weighed of the influence of LAT
```

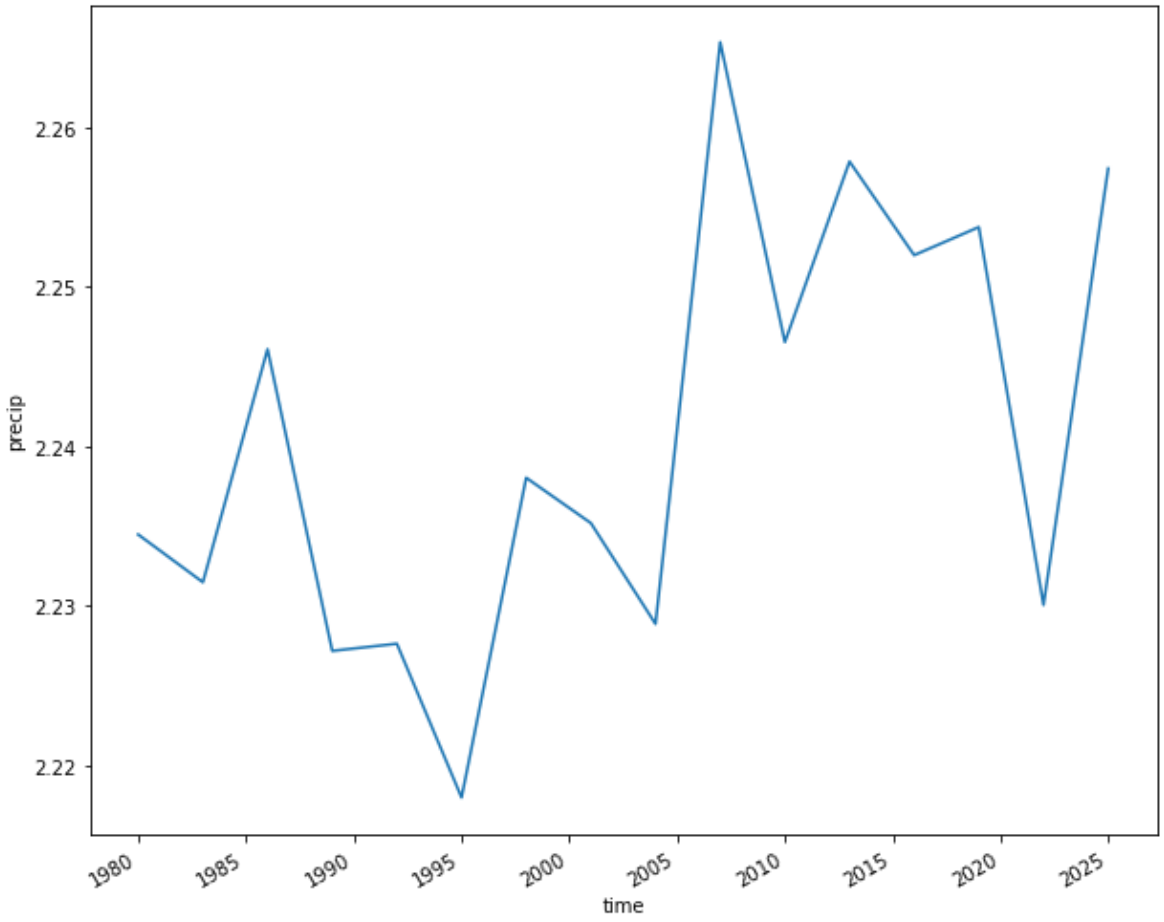
Ex3 Explore a netCDF dataset

3.1

```
gpcp=xr.open_dataset("precip.mon.mean.nc",engine="netcdf4")
```

```
groupData=gpcp.precip.resample(time="3Y")
groupData.mean(dim=["lon","lat","time"]).plot(figsize=(10,8))
```

```
[<matplotlib.lines.Line2D at 0x1dc2010b550>]
```

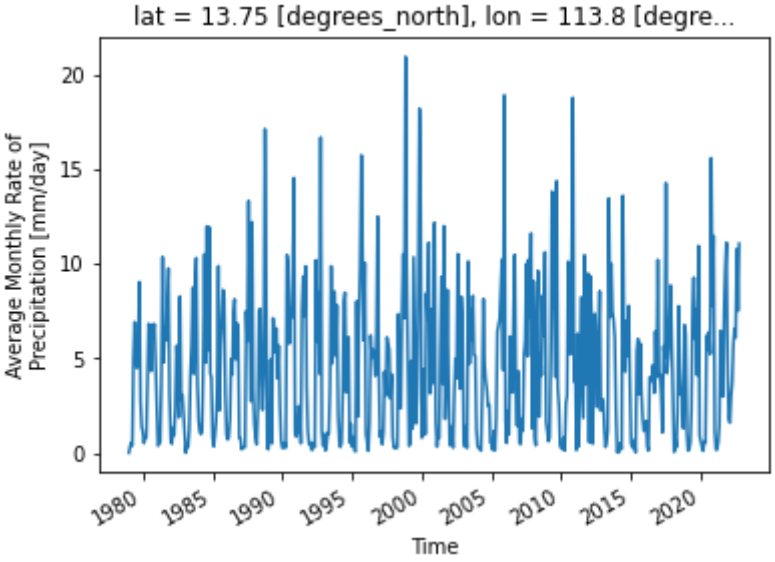


3.2

```
# plot averaged global soilw at a specific point
gpcp.precip.sel(lon=114.5,lat=14,method="nearest").plot()
```

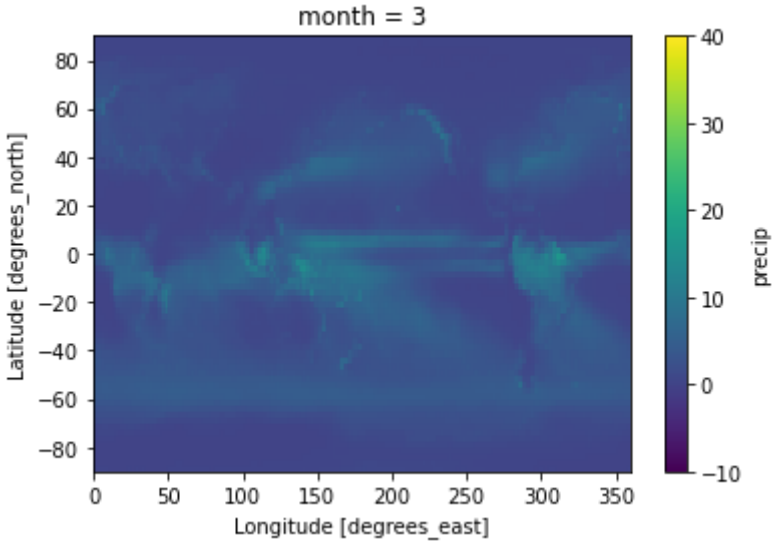
```
D:\anaconda3\lib\site-packages\xarray\core\indexes.py:234: FutureWarning:
Passing method to Float64Index.get_loc is deprecated and will raise in a future
version. Use index.get_indexer([item], method=...) instead.
  indexer = self.index.get_loc(
D:\anaconda3\lib\site-packages\xarray\core\indexes.py:234: FutureWarning:
Passing method to Float64Index.get_loc is deprecated and will raise in a future
version. Use index.get_indexer([item], method=...) instead.
  indexer = self.index.get_loc(
```

```
[<matplotlib.lines.Line2D at 0x1dc269cbf40>]
```



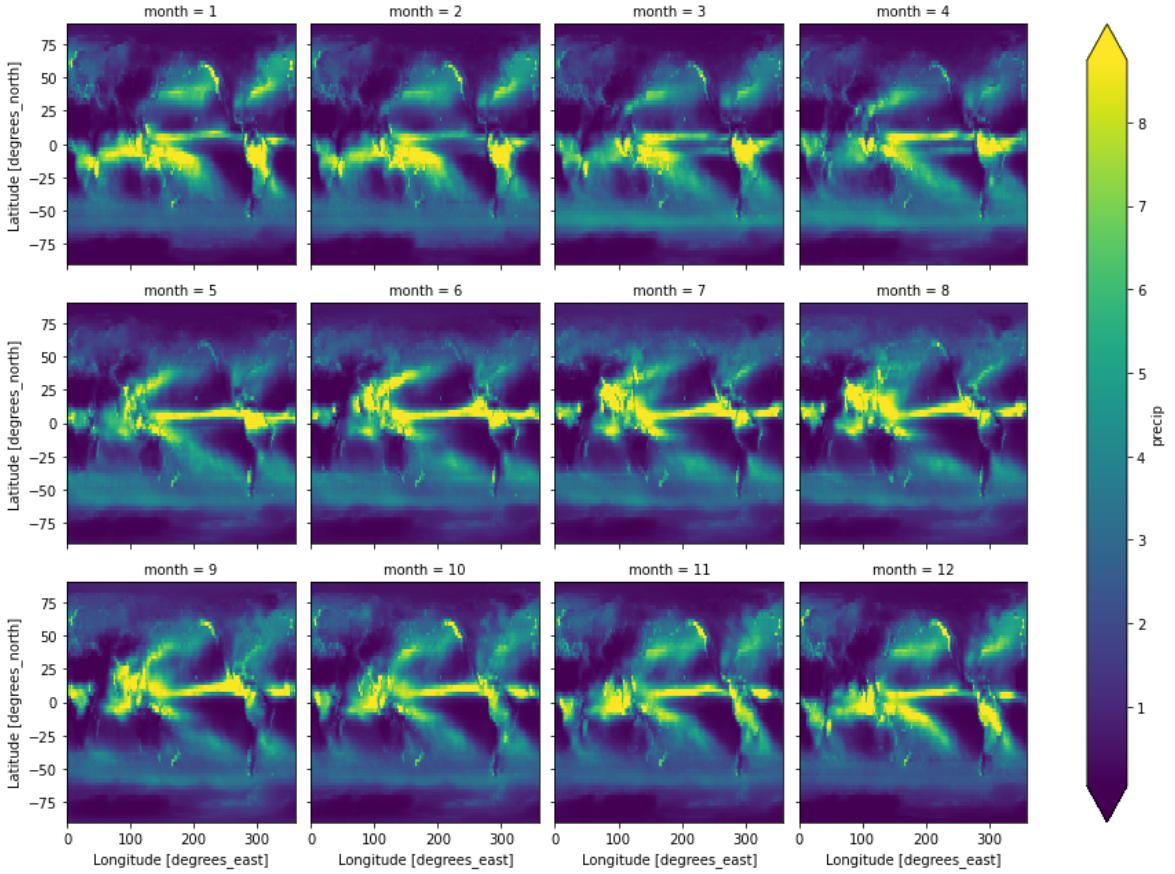

```
# Plot average global precip at March
precip_clim=gpcp.precip.groupby("time.month").mean()
precip_clim[2,:,:].plot(vmin=-10,vmax=40)
```

```
<matplotlib.collections.QuadMesh at 0x1dc26ba0ee0>
```



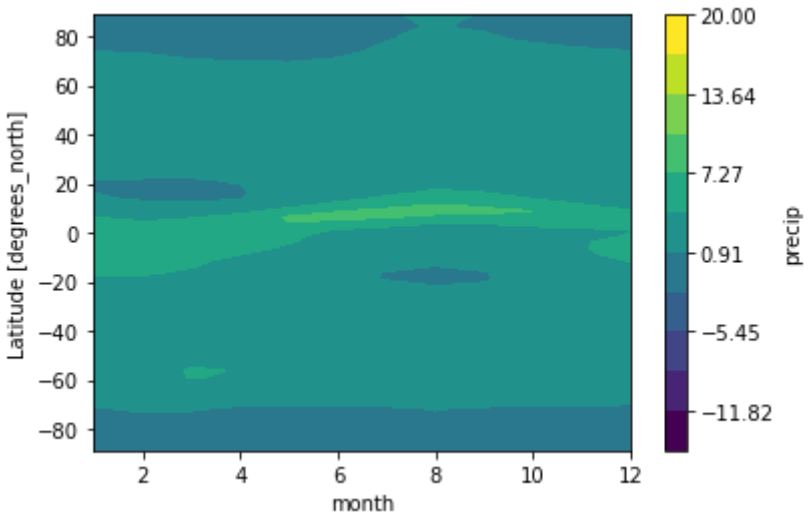

```
# Plot precip  monthly average in a panel
precip_clim.plot(col="month",col_wrap=4,robust=True)
```

```
<xarray.plot.facetgrid.FacetGrid at 0x1dc26c2e9d0>
```




```
# Plot zonal mean
precip_clim.mean(dim="lon").plot.contourf(x="month",levels=12,vmin=-15,vmax=20)
```

```
<matplotlib.contour.QuadContourSet at 0x1dc289ff1c0>
```




```
# Using hv.plot() to show widget
import hvplot.xarray
gpcp.precip.hvplot(groupby="time", clim=(gpcp.precip.min(), gpcp.precip.max()),
cmap='turbo')
```