ESE5023_Assignments3_

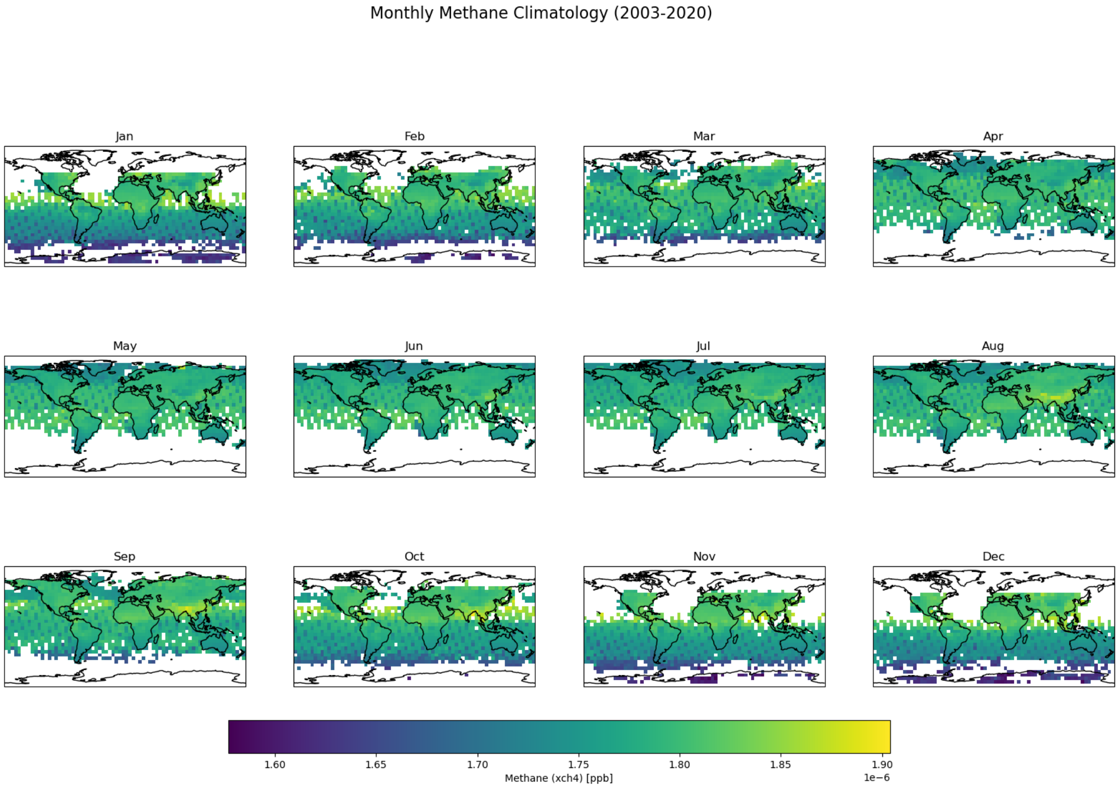
November 26, 2025

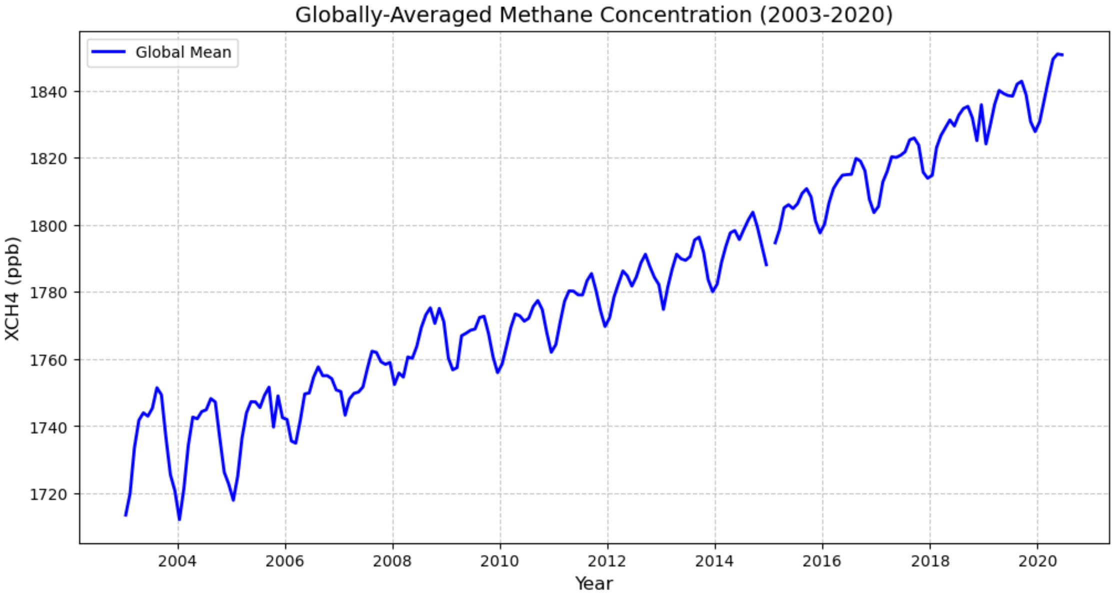
[16]:

```
import xarray as xr
import matplotlib.pyplot as plt
import numpy as np
import cartopy.crs as ccrs

#
ds = xr.open_dataset('200301_202006-C3S-L3_GHG-PRODUCTS-OBS4MIPS-MERGED-v4.3.
 ↪nc')


#         "1 "  "2 "  ...
# groupby('time.month')
climatology = ds['xch4'].groupby('time.month').mean(dim='time', keep_attrs=True)

#   3 4
fig, axes = plt.subplots(nrows=3, ncols=4, figsize=(20, 12),
                         subplot_kw={'projection': ccrs.PlateCarree()})
axes = axes.flatten()

months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
          'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']

#      12    Colorbar
vmin = climatology.min().values
vmax = climatology.max().values

for i, ax in enumerate(axes):
    #    i
    data_month = climatology.isel(month=i)

    #
    im = ax.pcolormesh(data_month.lon, data_month.lat, data_month,
                       transform=ccrs.PlateCarree(),
                       cmap='viridis', vmin=vmin, vmax=vmax)

    ax.coastlines() #
    ax.set_title(f'{months[i]}') #
```

```
#
cbar = fig.colorbar(im, ax=axes, orientation='horizontal', fraction=0.05, pad=0.
 ↪05)
cbar.set_label('Methane (xch4) [ppb]')

plt.suptitle('Monthly Methane Climatology (2003-2020)', fontsize=16)
plt.show()


#  1.2

# 1.
weights = np.cos(np.deg2rad(ds.lat))
weights.name = "weights"

# 2.
ds_weighted = ds['xch4'].weighted(weights)
global_mean = ds_weighted.mean(dim=("lat", "lon"))

#         (  1.8e-6)      1e9    ppb
global_mean_ppb = global_mean * 1e9

# 3.
plt.figure(figsize=(12, 6))

#
plt.plot(global_mean_ppb.time, global_mean_ppb, color='b', linewidth=2,␣
 ↪label='Global Mean')

plt.title('Globally-Averaged Methane Concentration (2003-2020)', fontsize=14)
plt.ylabel('XCH4 (ppb)', fontsize=12) #      ppb
plt.xlabel('Year', fontsize=12)

plt.grid(True, which='both', linestyle='--', alpha=0.7)
plt.legend()
plt.show()
print("          1720-1850+ ppb    2003-2006                  ")

# 1.3:       ---

# 1.
point_ts = ds['xch4'].sel(lat=-15, lon=-150, method='nearest')

#      ppb
point_ts_ppb = point_ts * 1e9
```

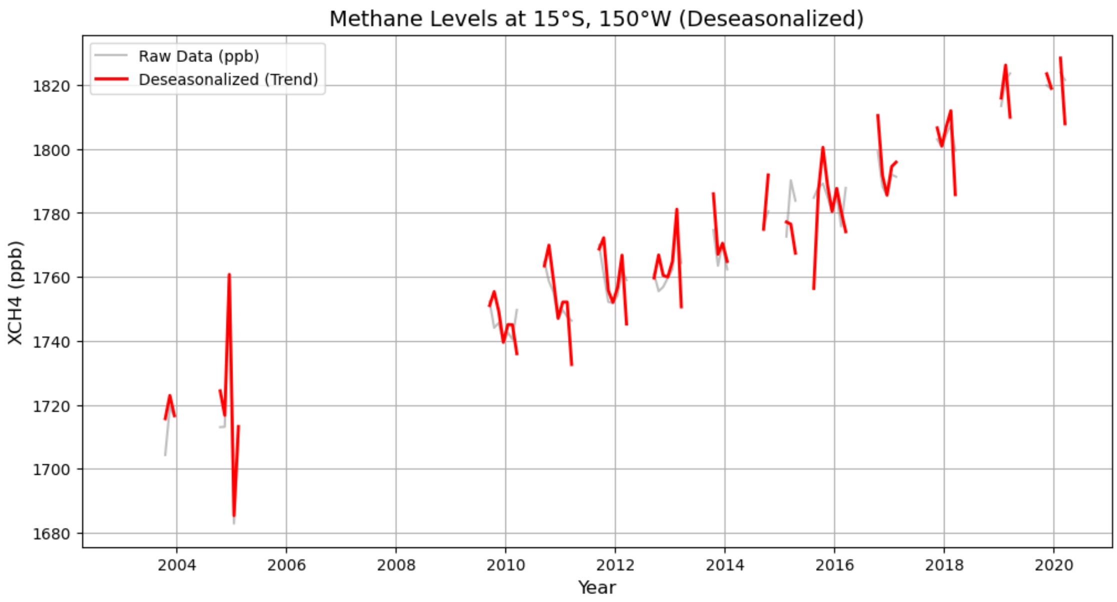
```
#     (    ppb  )
point_clim = point_ts_ppb.groupby('time.month').mean('time')
overall_mean = point_ts_ppb.mean()

#   (ppb) -   (ppb) +   (ppb)
deseasonalized_ppb = point_ts_ppb.groupby('time.month') - point_clim +␣
 ↪overall_mean

#3.
plt.figure(figsize=(12, 6))

#
plt.plot(point_ts_ppb.time, point_ts_ppb, color='gray', alpha=0.5, label='Raw␣
 ↪Data (ppb)')

#
plt.plot(deseasonalized_ppb.time, deseasonalized_ppb, color='red', linewidth=2,␣
 ↪label='Deseasonalized (Trend)')

plt.title('Methane Levels at 15°S, 150°W (Deseasonalized)', fontsize=14)
plt.ylabel('XCH4 (ppb)', fontsize=12) #   ppb
plt.xlabel('Year', fontsize=12)
plt.legend()
plt.grid(True)
plt.show()
print("      15°S,␣
 ↪150°W      \n                                        \n         2003    1720␣
 ↪ppb     2020    1820 ppb                   ")
```





1720−1850+

ppb 2003−2006



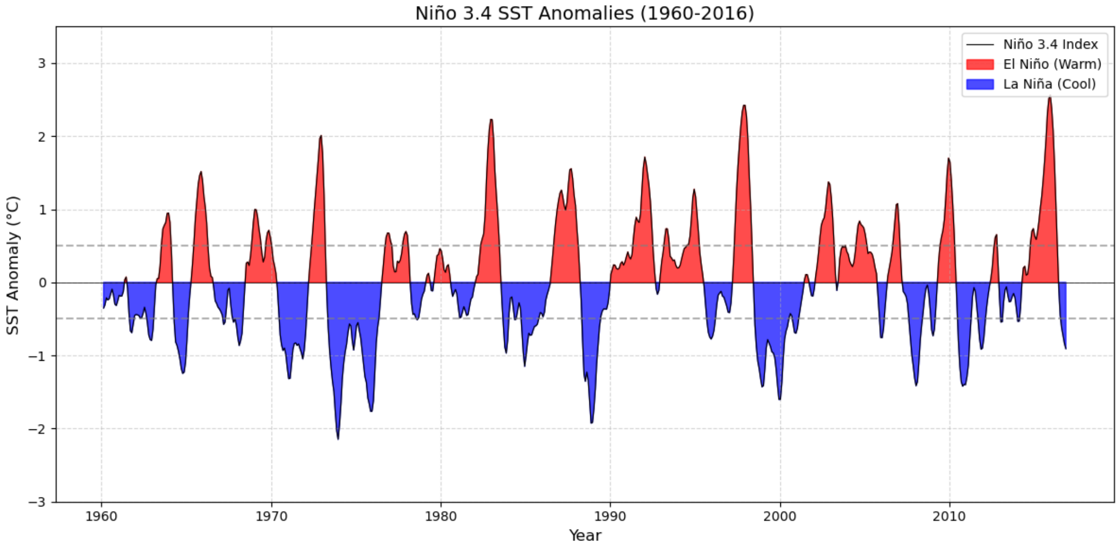
15°S, 150°W

2003 1720 ppb 2020 1820 ppb

[22]:
```
import xarray as xr
import matplotlib.pyplot as plt
import numpy as np

#1.1   Niño3.4



#
file_path = 'NOAA_NCDC_ERSST_v3b_SST.nc'
ds = xr.open_dataset(file_path)

#  Niño 3.4
#  : 5N - 5S (   -88   88   -5   5)
#  : 170W - 120W (   0-360   190   240)
# 170W = 360 - 170 = 190    120W = 360 - 120 = 240

#
subset = ds['sst'].sel(lat=slice(-5, 5), lon=slice(190, 240))

#     (Area-Weighted Average)   # cos( )
weights = np.cos(np.deg2rad(subset.lat))
weights.name = "weights"
```

```
#   lat   lon         time
sst_weighted = subset.weighted(weights)
nino34_raw = sst_weighted.mean(dim=("lat", "lon"))

#    (Climatology)    (Anomalies)       (  1960-2016  1    2   ...)
climatology = nino34_raw.groupby('time.month').mean('time')

#
# groupby
nino34_anom = nino34_raw.groupby('time.month') - climatology



#1.2:   Niño 3.4
#  3     (3-Month Running Mean)
# center=True          (    +    +    ) / 3
nino34_index = nino34_anom.rolling(time=3, center=True).mean()

#
plt.figure(figsize=(12, 6))

#
times = nino34_index.time
values = nino34_index.values

#     (  )
plt.plot(times, values, color='k', linewidth=0.8, label='Niño 3.4 Index')

#     (El Niño    / La Niña   )
plt.fill_between(times, values, 0, where=(values >= 0),
                 color='red', alpha=0.7, interpolate=True, label='El Niño␣
 ↪(Warm)')
plt.fill_between(times, values, 0, where=(values <= 0),
                 color='blue', alpha=0.7, interpolate=True, label='La Niña␣
 ↪(Cool)')

#      (+/- 0.5°C    /     )
plt.axhline(0.5, color='gray', linestyle='--', alpha=0.6)
plt.axhline(-0.5, color='gray', linestyle='--', alpha=0.6)
plt.axhline(0, color='black', linewidth=0.5)

#
plt.title('Niño 3.4 SST Anomalies (1960-2016)', fontsize=14)
plt.ylabel('SST Anomaly (°C)', fontsize=12)
plt.xlabel('Year', fontsize=12)
```

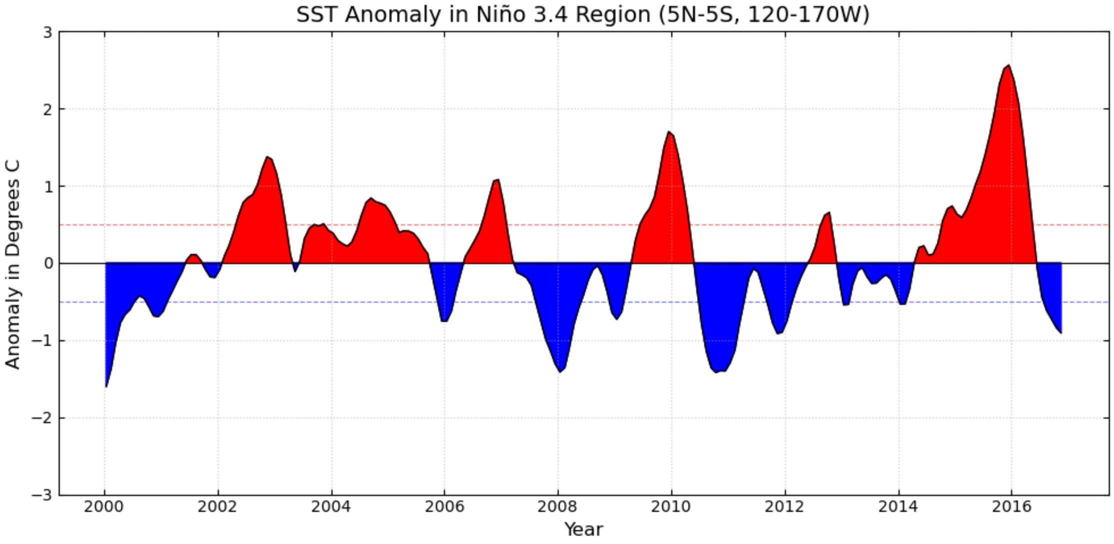
```
# Y  (  -3  3  )
plt.ylim(-3.0, 3.5)

#
plt.legend(loc='upper right')
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()

plt.show()
```



[26]:
```
#

import xarray as xr
import matplotlib.pyplot as plt
import numpy as np



#
file_path = 'NOAA_NCDC_ERSST_v3b_SST.nc'
ds = xr.open_dataset(file_path)

#    : 5N-5S, 170W-120W (   190-240)
subset = ds['sst'].sel(lat=slice(-5, 5), lon=slice(190, 240))

#
weights = np.cos(np.deg2rad(subset.lat))
sst_weighted = subset.weighted(weights)
nino34_raw = sst_weighted.mean(dim=("lat", "lon"))
```

```
#     (Anomalies)
climatology = nino34_raw.groupby('time.month').mean('time')
nino34_anom = nino34_raw.groupby('time.month') - climatology

#   3
nino34_index = nino34_anom.rolling(time=3, center=True).mean()


#1.2   (    2000 )

#  2000-01-01     ***
nino34_recent = nino34_index.sel(time=slice('2000-01-01', '2016-12-31'))

#
plt.figure(figsize=(10, 5)) #

times = nino34_recent.time
values = nino34_recent.values

#
plt.axhline(0, color='black', linewidth=0.8) # 0
plt.axhline(0.5, color='red', linestyle='--', linewidth=0.8, alpha=0.5)  #
plt.axhline(-0.5, color='blue', linestyle='--', linewidth=0.8, alpha=0.5) #

#    (3   )
plt.plot(times, values, color='k', linewidth=1, label='3mth running mean')

#
# 0
plt.fill_between(times, values, 0, where=(values >= 0),
                 color='red', alpha=1.0, interpolate=True)
# 0
plt.fill_between(times, values, 0, where=(values <= 0),
                 color='blue', alpha=1.0, interpolate=True)

#
plt.title('SST Anomaly in Niño 3.4 Region (5N-5S, 120-170W)', fontsize=14)
plt.ylabel('Anomaly in Degrees C', fontsize=12)
plt.xlabel('Year', fontsize=12)

# Y   (-3   3)
plt.ylim(-3.0, 3.0)

#     (      )
plt.tick_params(direction='in', top=True, right=True)
```

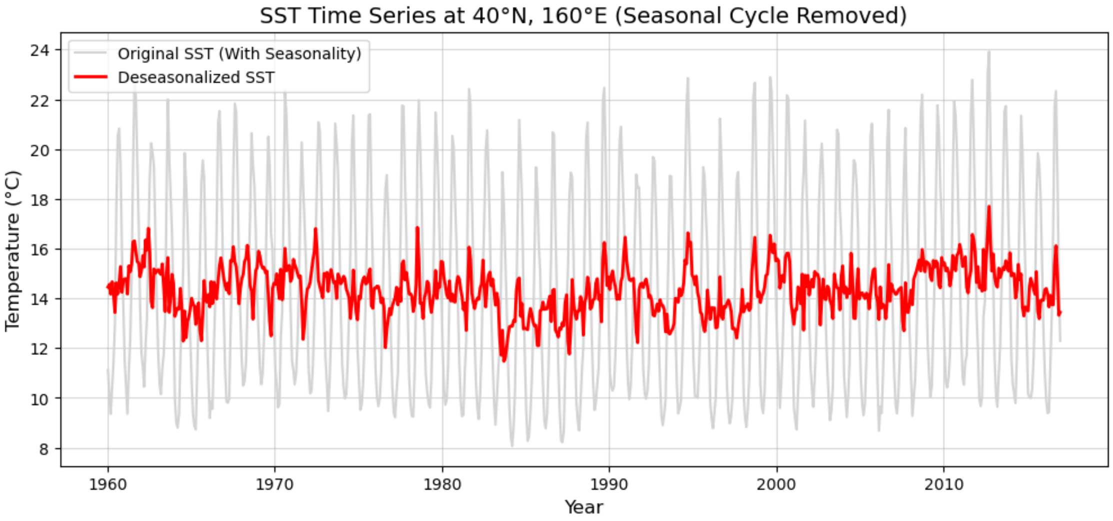
```
#    ( )
plt.grid(True, linestyle=':', alpha=0.6)

plt.tight_layout()
plt.show()
```



[38]:

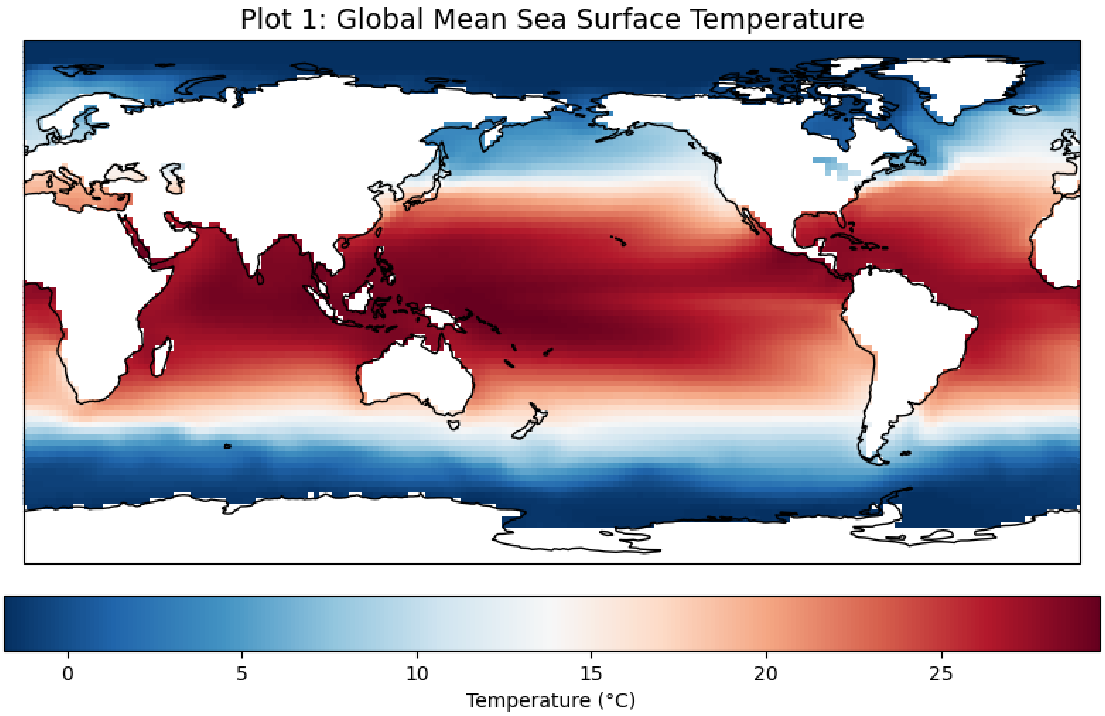
```
import xarray as xr
import matplotlib.pyplot as plt
import numpy as np

# 1.
file_path = 'NOAA_NCDC_ERSST_v3b_SST.nc'
ds = xr.open_dataset(file_path)

#      (    :  40,   160)
point_data = ds['sst'].sel(lat=40, lon=160, method='nearest')

#
climatology = point_data.groupby('time.month').mean('time')
# B.     -    +
deseasonalized = point_data.groupby('time.month') - climatology + point_data.
  ↪mean()

#
plt.figure(figsize=(12, 5))

#
```

```
plt.plot(point_data.time, point_data, color='lightgray', label='Original SST␣
↪(With Seasonality)')
#
plt.plot(deseasonalized.time, deseasonalized, color='red', linewidth=2,␣
↪label='Deseasonalized SST')

plt.title('SST Time Series at 40°N, 160°E (Seasonal Cycle Removed)',␣
↪fontsize=14)
plt.ylabel('Temperature (°C)', fontsize=12)
plt.xlabel('Year', fontsize=12)
plt.legend()
plt.grid(True, alpha=0.5)
plt.show()
```

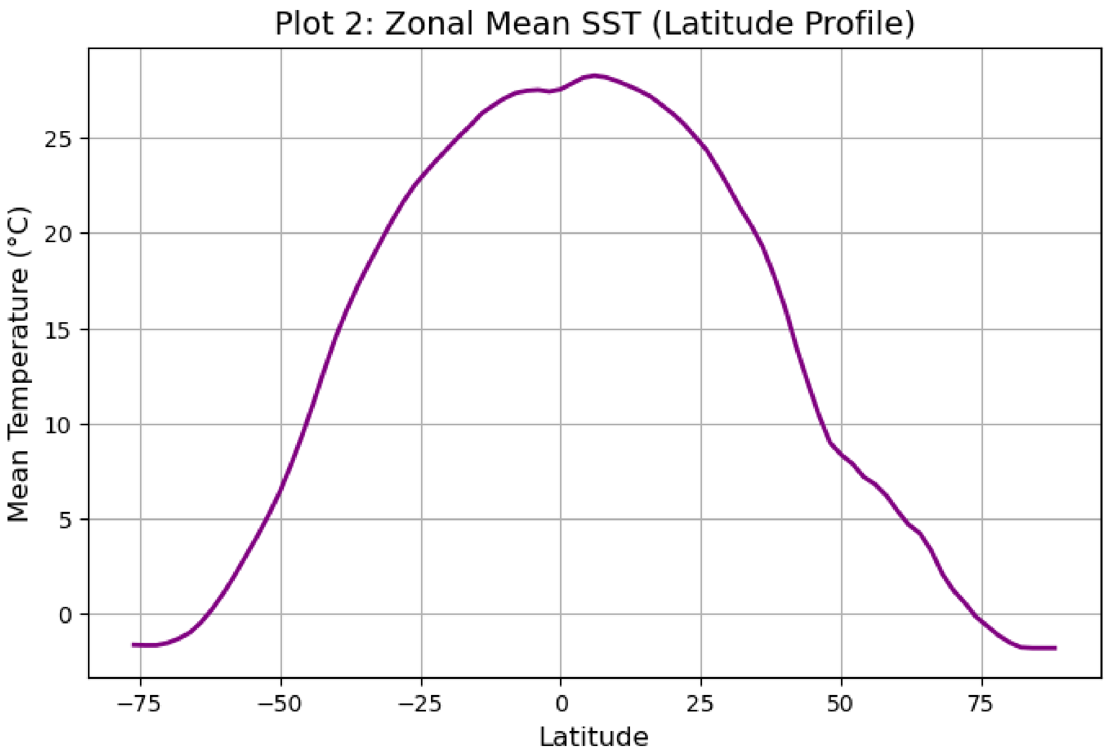


[32]:

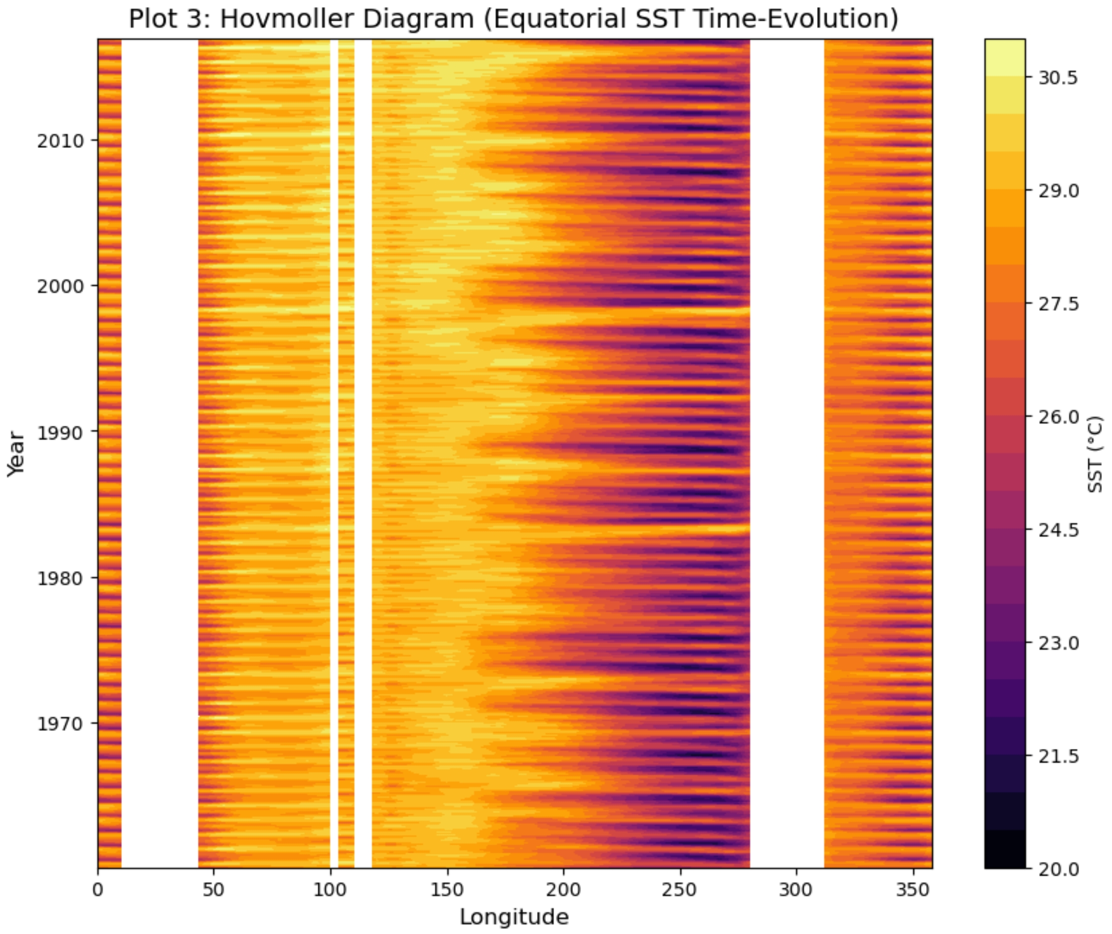
```
import cartopy.crs as ccrs

#
sst_mean = ds['sst'].mean(dim='time')

plt.figure(figsize=(10, 6))
ax = plt.axes(projection=ccrs.PlateCarree(central_longitude=180))
im = ax.pcolormesh(sst_mean.lon, sst_mean.lat, sst_mean,
                   transform=ccrs.PlateCarree(), cmap='RdBu_r')
ax.coastlines()
plt.colorbar(im, label='Temperature (°C)', orientation='horizontal', pad=0.05)
plt.title('Plot 1: Global Mean Sea Surface Temperature', fontsize=14)
plt.show()
```



```
[34]: #   (lon)  (time)
zonal_mean = ds['sst'].mean(dim=('time', 'lon'))

plt.figure(figsize=(8, 5))
plt.plot(zonal_mean.lat, zonal_mean, color='purple', linewidth=2)
plt.title('Plot 2: Zonal Mean SST (Latitude Profile)', fontsize=14)
plt.xlabel('Latitude', fontsize=12)
plt.ylabel('Mean Temperature (°C)', fontsize=12)
plt.grid(True)
plt.show()
```

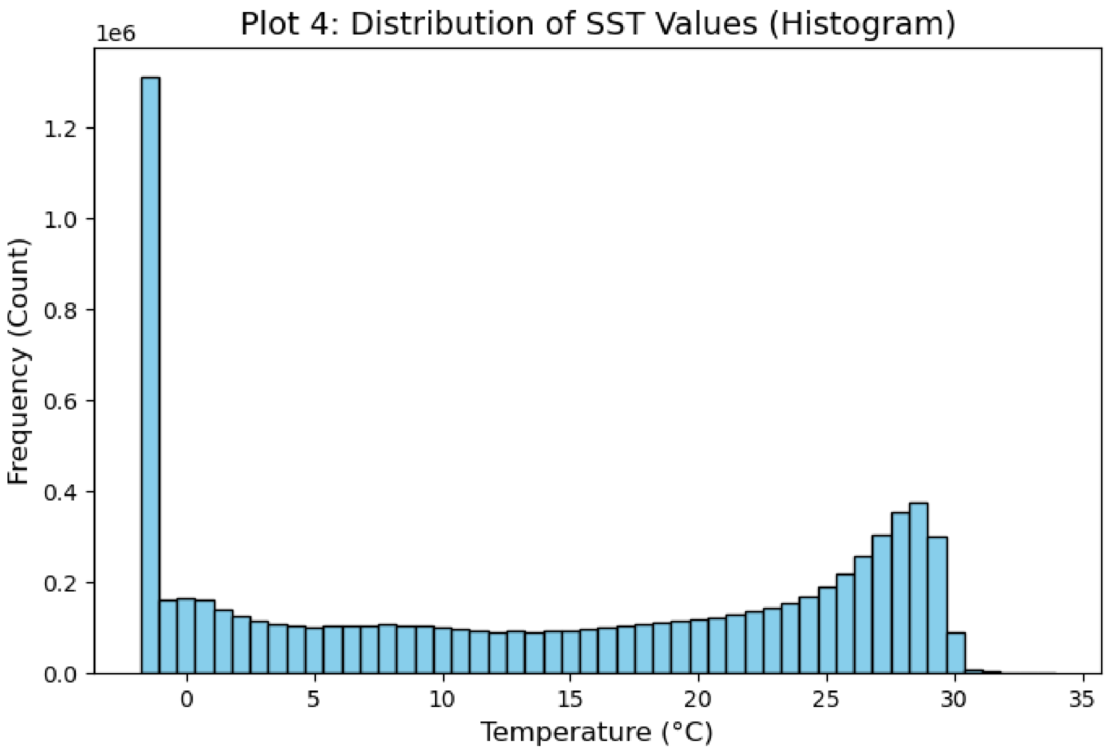


[36]:
```
#
equatorial_sst = ds['sst'].sel(lat=0, method='nearest')

plt.figure(figsize=(10, 8))
#   contourf
plt.contourf(equatorial_sst.lon, equatorial_sst.time, equatorial_sst,
             levels=20, cmap='inferno')
plt.title('Plot 3: Hovmoller Diagram (Equatorial SST Time-Evolution)',␣
 ↪fontsize=14)
plt.xlabel('Longitude', fontsize=12)
plt.ylabel('Year', fontsize=12)
plt.colorbar(label='SST (°C)')
plt.show()
```



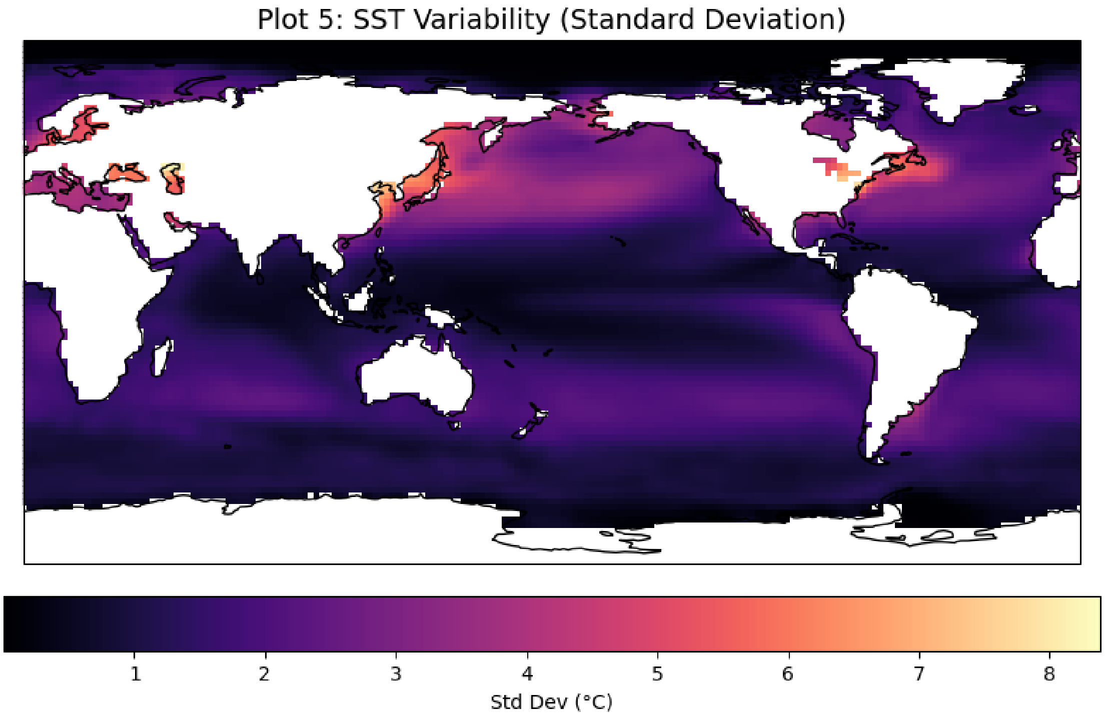
[40]:

```python
plt.figure(figsize=(8, 5))
# flatten()
plt.hist(ds['sst'].values.flatten(), bins=50, color='skyblue', edgecolor='black')
plt.title('Plot 4: Distribution of SST Values (Histogram)', fontsize=14)
plt.xlabel('Temperature (°C)', fontsize=12)
plt.ylabel('Frequency (Count)', fontsize=12)
plt.show()
```



[42]:

```
#     (Standard Deviation)
sst_std = ds['sst'].std(dim='time')

plt.figure(figsize=(10, 6))
ax = plt.axes(projection=ccrs.PlateCarree(central_longitude=180))
im = ax.pcolormesh(sst_std.lon, sst_std.lat, sst_std,
                   transform=ccrs.PlateCarree(), cmap='magma')
ax.coastlines()
plt.colorbar(im, label='Std Dev (°C)', orientation='horizontal', pad=0.05)
plt.title('Plot 5: SST Variability (Standard Deviation)', fontsize=14)
plt.show()
```


Plot 5: SST Variability (Standard Deviation)
1
2
3
4
5
6
7
8
Std Dev (°C)

[ ]: