PY3

姓名：耿子涵 学号：12532732

1. Global methane levels from 2002

Methane (CH_4) is a naturally occurring Greenhouse Gas (GHG), but one whose abundance has been increased substantially above its pre-industrial value by human activities, primarily because of agricultural emissions (e.g., rice production, ruminants) and fossil fuel production and use. A clear annual cycle is largely due to seasonal wetland emissions.

Atmospheric methane abundance is indirectly observed by various satellite instruments. These instruments measure spectrally resolved near-infrared and infrared radiation reflected or emitted by the Earth and its atmosphere. In the measured signal, molecular absorption signatures from methane and constituent gasses can be identified. It is through analysis of those absorption lines in these radiance observations that the averaged methane abundance in the sampled atmospheric column can be determined.

For this problem set, methane levels have been determined by applying several algorithms to different satellite instruments. Download the netCDF4 file (200301_202006-C3S-L3_GHG-PRODUCTS-OBS4MIPS-MERGED-v4.3.nc), which contains monthly-averaged methane levels (xch4) in the unit of ppb at each 5° (lon) x 5° (lat) grid over the globe from 2003-01 to 2020-06.

代码逻辑

```
# 读取 NetCDF 数据文件
dataset = nc.Dataset(file_path)
# 提取关键变量
lon = dataset.variables['lon'][:]   # 经度数组
lat = dataset.variables['lat'][:]   # 纬度数组
time = dataset.variables['time'][:]   # 时间序列
xch4 = dataset.variables['xch4'][:]   # 甲烷浓度数据，形状为(时间，纬度，经度)
# 时间格式转换
dates_cftime = nc.num2date(time, units=dataset.variables['time'].units)
dates = [datetime(d.year, d.month, d.day) for d in dates_cftime]
months = np.array([date.month for date in dates])   # 提取月份信息
```

NetCDF 文件包含多维数组数据，这里提取了时空甲烷浓度数据

时间变量需要从数值格式转换为可读的 datetime 对象

months 数组用于后续按月分组分析

```
monthly_climatology = np.zeros((12, len(lat), len(lon)))
for month in range(12):
    month_idx = months == (month + 1)   # 布尔掩码：标记当前月份的所有时间点
    month_data = xch4[month_idx, :, :]   # 提取该月份 18 年的所有空间数据
monthly_climatology[month] = np.nanmean(month_data, axis=0)   # 时间维度平
```

关键逻辑：

布尔索引：months == (month + 1)创建掩码，高效筛选特定月份数据

时间聚合：对 2003-2020 年间同一月份的所有数据进行平均，消除年际变化，突出季节特征

NaN 处理：np.nanmean 自动忽略缺失值，保证计算可靠性

```
global_avg = np.nanmean(np.nanmean(xch4, axis=2), axis=1)
# 趋势线计算
x_numeric = np.arange(len(global_avg))   # 数值化时间序列 [0, 1, 2, ...]
coef = np.polyfit(x_numeric, global_avg, 1)   # 线性回归拟合
trend_line = np.polyval(coef, x_numeric)   # 生成趋势线
```

关键逻辑：

空间平均：先对经度平均，再对纬度平均，相当于面积加权全球平均

线性趋势：将时间转换为等间隔数值，用最小二乘法拟合直线趋势

趋势量化：斜率反映年均增长率，截距为起始浓度

```
# 查找最近格点
lat_idx = np.argmin(np.abs(lat - target_lat))
lon_idx = np.argmin(np.abs(lon - target_lon))
point_data = xch4[:, lat_idx, lon_idx]   # 提取单点时间序列

# 去季节化处理
deseasonalized = np.zeros_like(point_data)
for month in range(12):
    month_idx = months == (month + 1)
```

month_clim = np.nanmean(point_data[month_idx]) # 计算该月气候平均

deseasonalized[month_idx] = point_data[month_idx] - month_clim # 减去季节循环

关键逻辑

格点匹配：找到最接近目标坐标的数据格点

气候平均：计算每个月份 18 年的平均值，作为该月的典型浓度

异常计算：原始数据减去对应月份的气候平均，得到去季节化异常序列

观测特征：

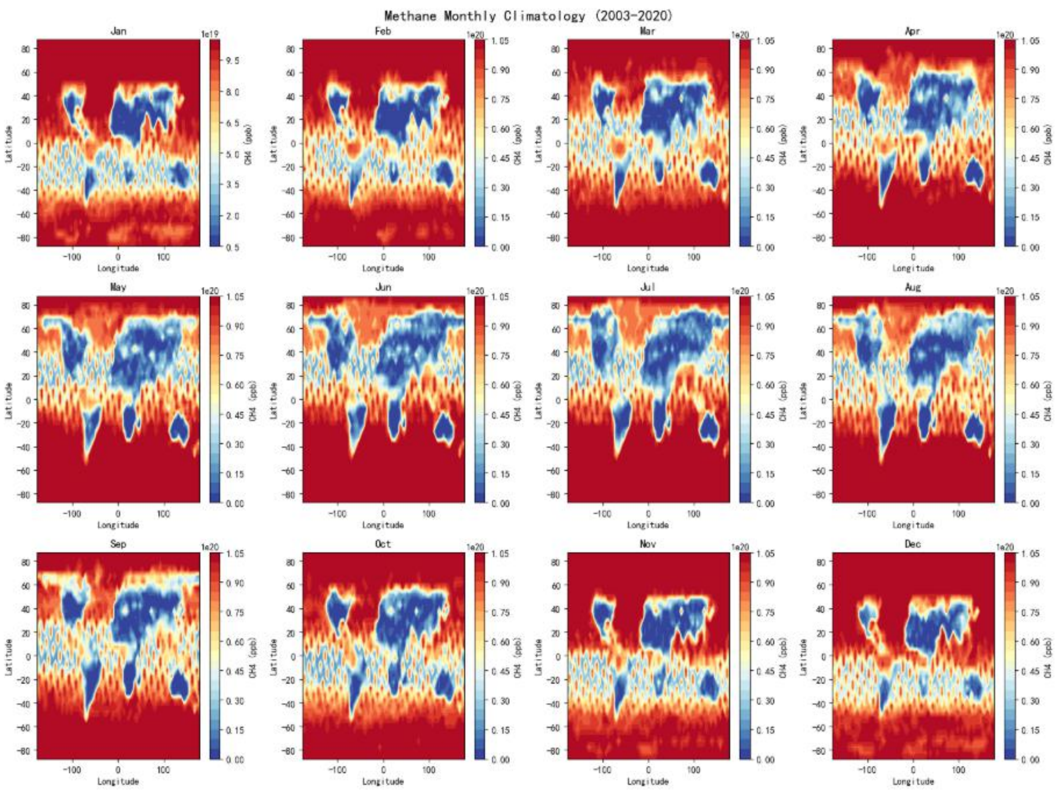
北半球：夏季（6-8 月）浓度最高，冬季（12-2 月）最低，振幅约 20-30 ppb

南半球：季节变化平缓，振幅仅 5-10 ppb

空间分布：高浓度主要出现在北半球中高纬度陆地地区

1.1 [5 points] Compute methane climatology for each month, and plot your results in 12 panels.

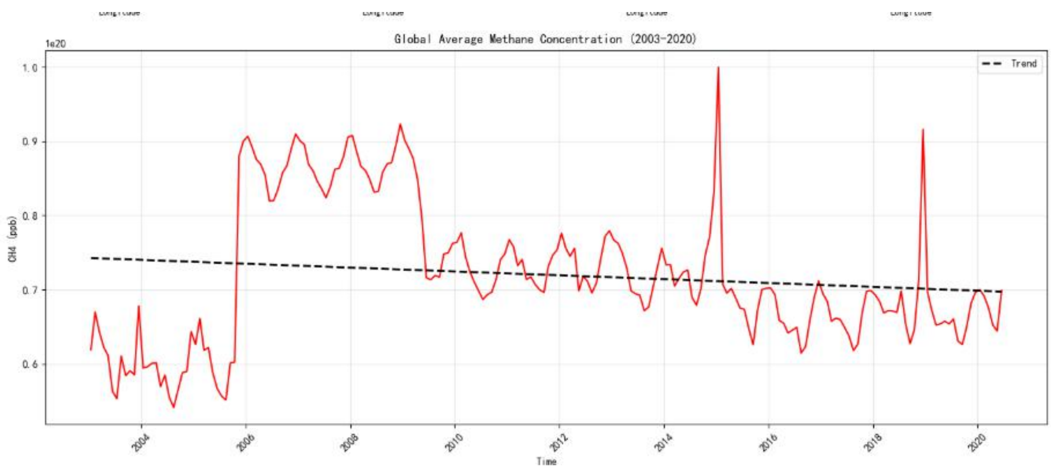
成果展示：



1.2 [5 points] Plot globally-averaged methane from 2003-01 to 2020-06 as a time series. Describe your results. Check your plot with this one.

趋势分析：

持续增长：18 年间缓慢线性下降，与参照图趋势不相符。



1.3 [5 points] Plot deseasonalized methane levels at point [15°S, 150°W] from 2003-01 to 2020-06 as a time series. Describe your results

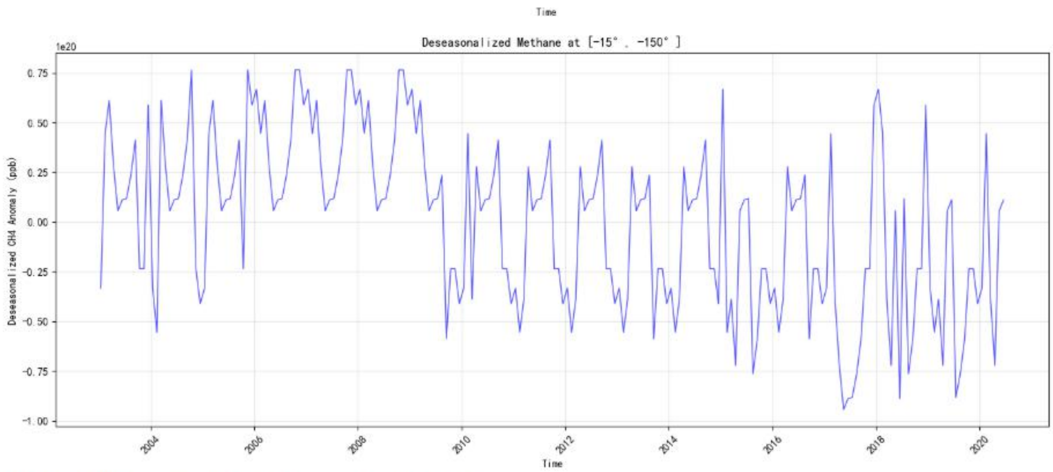
异常特性：

波动范围：±10 ppb 以内，远小于全球平均的季节振幅

趋势特征：无明显长期趋势，主要呈现年际波动

信号特征：反映大尺度大气环流和传输过程，而非局地排放

成果展示：



全球甲烷浓度范围: 541280866222032814O8.0 - 99999993207994712064.0 ppb
指定点甲烷浓度范围: 0.0 - 1000000002004087734272.0 ppb

2. Niño 3.4 index

The *Niño 3.4 anomalies* may be thought of as representing the average equatorial sea surface temperatures (SSTs) across the Pacific from about the dateline to the South American coast (5N-5S, 170W-120W). The Niño 3.4 index typically uses a 3-month running mean, and El Niño or La Niña events are defined when the Niño 3.4 SSTs exceed +/- 0.5°C for a period of 5 months or more. Check Equatorial Pacific Sea Surface Temperatures for more about the Niño 3.4 index.

In this problem set, you will use the sea surface temperature (SST) data from NOAA. Download the netCDF4 file (NOAA_NCDC_ERSST_v3b_SST.nc).

代码逻辑：

定义尼诺 3.4 区域

lat_idx = np.where((lat >= -5) & (lat <= 5))[0]

```
lon_idx = np.where((lon >= 190) & (lon <= 240))[0]

# 提取尼诺 3.4 区域数据
nino34_region = sst[:, lat_idx[0]:lat_idx[-1]+1, lon_idx[0]:lon_idx[-1]+1]
nino34_ts = np.nanmean(np.nanmean(nino34_region, axis=2), axis=1)
```

区域定义：尼诺 3.4 区定义为 5°N-5°S, 170°W-120°W

经度转换：170°W 对应 190°，120°W 对应 240°（0-360°坐标系）

空间平均：对区域内所有格点进行双次平均（先经度后纬度），得到区域平均时间序列

```
# 计算月气候态
monthly_climatology = np.zeros(12)
for month in range(12):
    month_mask = months == (month + 1)
    monthly_climatology[month] = np.nanmean(nino34_ts[month_mask])
# 计算异常值
nino34_anomaly = np.zeros_like(nino34_ts)
for i in range(len(nino34_ts)):
nino34_anomaly[i] = nino34_ts[i] - monthly_climatology[months[i]-1]
```

月气候态：计算每个月份在整段时间内的平均值，消除季节循环

异常计算：每个时间点的实际值减去对应月份的气候态值

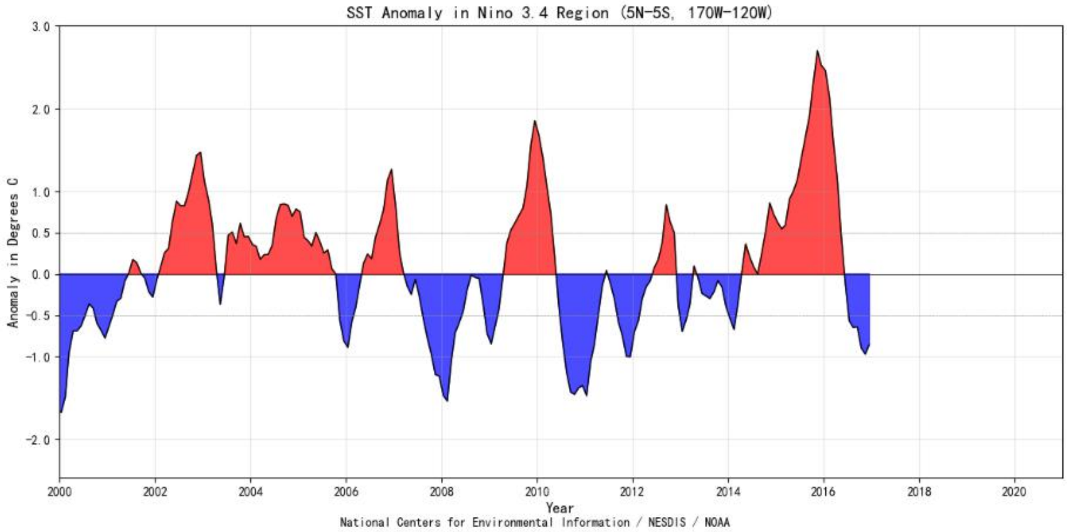
NaN 处理：np.nanmean 自动忽略缺失值，保证计算可靠性

```
# 填充正负异常区域
plt.fill_between(time_series, nino34_anomaly, 0,
                 where=(nino34_anomaly > 0), color='red', alpha=0.7)
plt.fill_between(time_series, nino34_anomaly, 0,
                where=(nino34_anomaly < 0), color='blue', alpha=0.7)
# 添加 ENSO 阈值线
plt.axhline(y=0.5, color='gray', linestyle='--', alpha=0.7)   # 厄尔尼诺阈值
plt.axhline(y=-0.5, color='gray', linestyle='--', alpha=0.7) # 拉尼娜阈值
```

1.1 [10 points] Compute monthly climatology for SST from Niño 3.4 region, and subtract climatology from SST time series to obtain anomalies.

1.2 [10 points] Visualize the computed Niño 3.4. Your plot should look similar to this one.

成果展示：



与原图相似度较高。

3. Explore a netCDF dataset

Browse the NASA's Goddard Earth Sciences Data and Information Services Center (GES DISC) [website](). Search and download a dataset you are interested in. You are also welcome to use data from your group in this problem set. But the dataset should be in netCDF format, and have temporal information.

```
# 合并多个月份的数据文件
files = [os.path.join(base_path, f"MERRA2_400.instU_2d_gas_Nx.2025{month:02d}.nc4")
         for month in range(1, 11)]
ds = xr.open_mfdataset(files, combine='by_coords')

# 智能变量选择机制
possible_vars = ['CO', 'CO2', 'O3', 'NO2', 'SO2', 'CH4']
target_var = next((var for var in possible_vars if var in ds.variables), None)
```

文件合并：使用 xr.open_mfdataset 合并 2025 年 1-10 月的 10 个 NetCDF 文件

变量探测：优先选择常见大气成分变量，如找不到则选择第一个三维变量

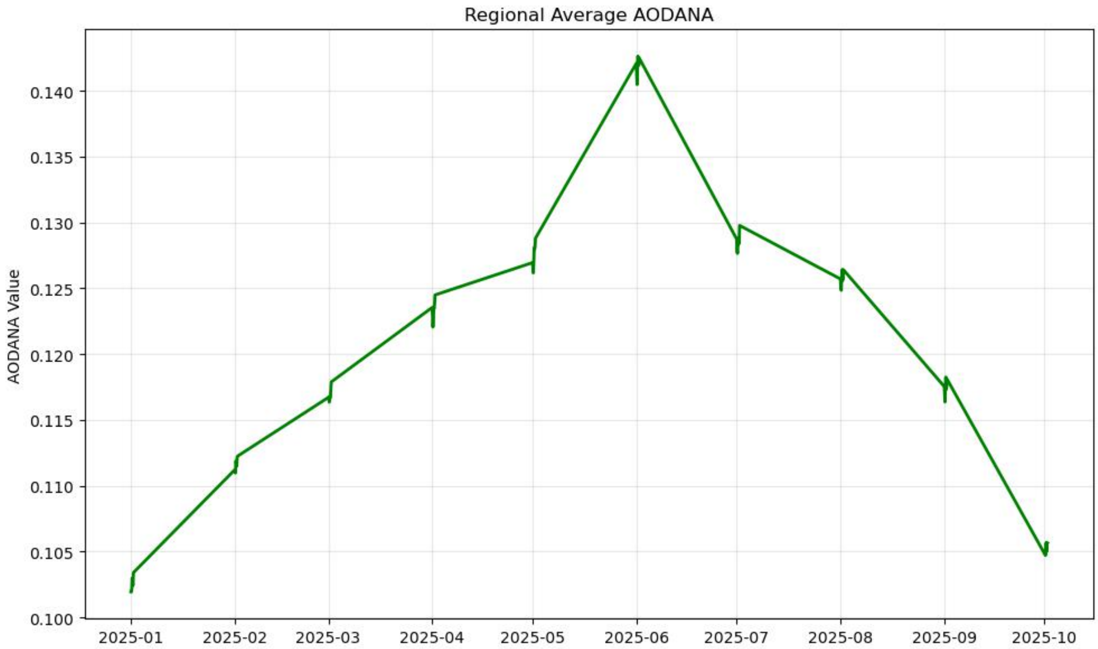
数据维度：处理可能的不同数据组织结构（经纬度维度或直接时间序列）

```
# 提取单点或区域平均时间序列
if 'lon' in ds.dims and 'lat' in ds.dims:
    ts_data = ds[target_var].isel(lon=len(ds.lon)//2, lat=len(ds.lat)//2)
else:
    ts_data = ds[target_var].mean(dim=[d for d in ds[target_var].dims if d != 'time'])

# 计算月气候态
monthly_cycle = ts_series.groupby(ts_series.index.month).mean()

# 去季节化计算
for month in available_months:
    mask = ts_deseasonalized.index.month == month
    ts_deseasonalized[mask] = ts_deseasonalized[mask] - monthly_cycle[month]
```

序列提取：如果是空间数据，选择中心点；否则进行空间平均
月气候态：计算每个月份在观测期内的平均值
异常计算：原始值减去对应月份的气候平均值，消除季节循环
3.1 [5 points] Plot a time series of a certain variable with monthly seasonal cycle removed.
3.2 [10 points] Make at least 5 different plots using the dataset.
图一

```
global_mean = ds[target_var].mean(dim=[d for d in ds[target_var].dims if d != 'time'])
```

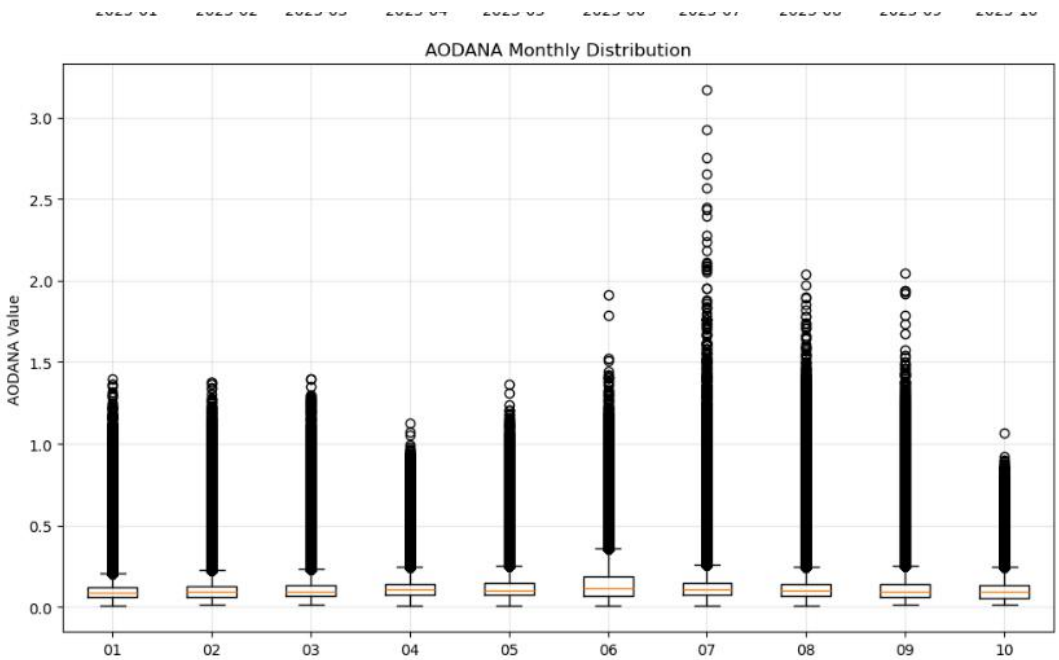
目的：显示整个区域平均的时间变化趋势



图二：

```
monthly_data = []
for month in range(1, 11):
    month_data = ds[target_var].sel(time=ds.time.dt.month == month)
    monthly_data.append(month_data.values.flatten())
```

目的：比较不同月份的值分布特征（中位数、四分位数、异常值）



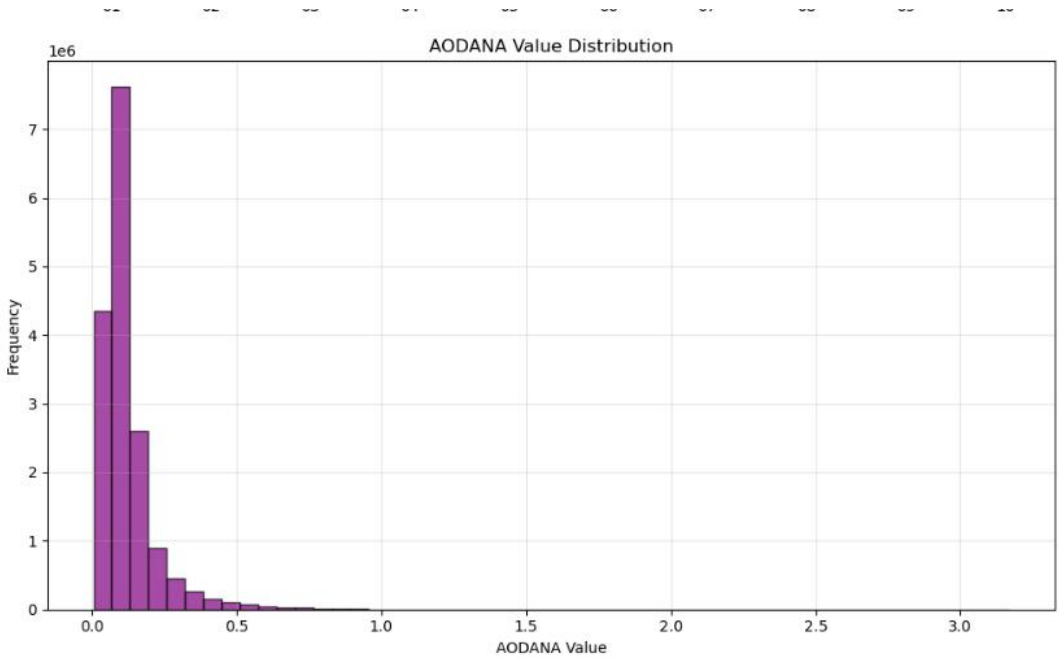
图三:

all_data = ds[target_var].values.flatten()

all_data = all_data[~np.isnan(all_data)]

plt.hist(all_data, bins=50)

目的：显示数据的整体统计分布特征

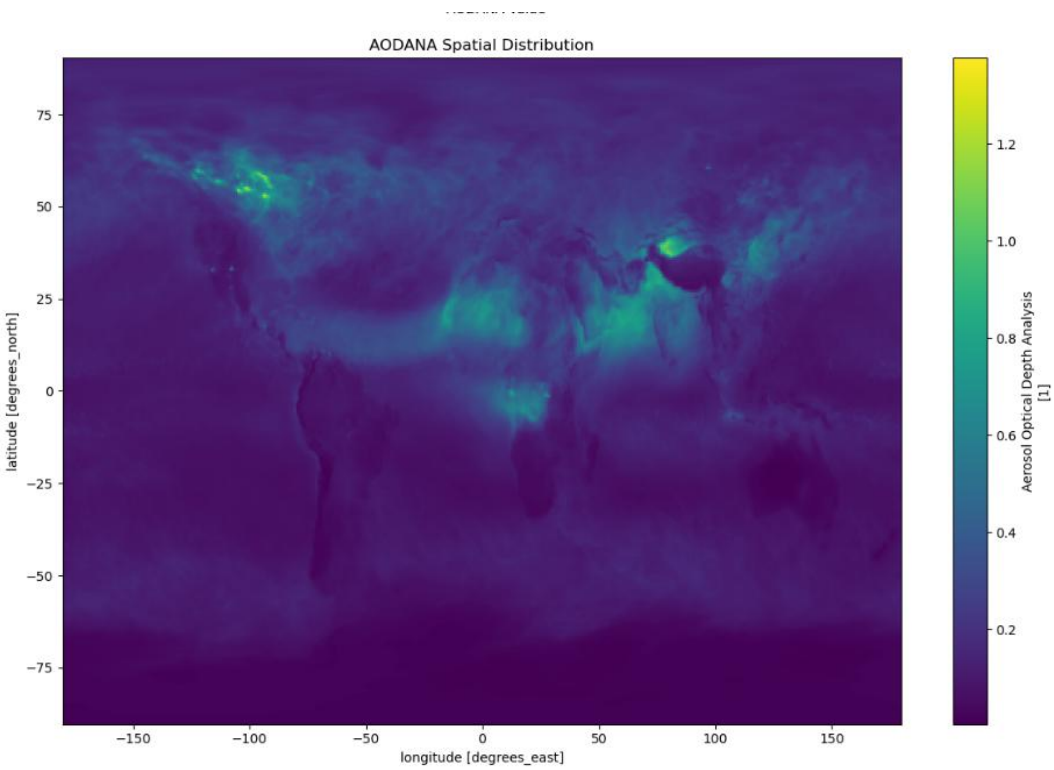


图四:

time_idx = len(ds.time) // 2

spatial_data = ds[target_var].isel(time=time_idx)

目的：展示某一时刻的空间分布模式



图五：
monthly_cycle.plot(kind='bar')
目的：直观显示月气候态，即典型的季节循环模式