

# 第一题：

## 第一问：

`ds.sst.groupby('time.month')`: 这是一个使用 Python 的数据处理库（如 xarray）的代码。`ds` 是一个数据集，而 `sst` 是这个数据集中的变量，代表海面温度（Sea Surface Temperature）。`.groupby('time.month')` 的作用是将数据按照时间中的月份进行分组。

`tas_clim = ds.sst.groupby('time.month').mean()`: 这行代码计算每个月份的海面温度平均值。`mean()` 函数计算每个月份的平均海面温度，结果被存储在 `tas_clim` 变量中。这里 `tas_clim` 可以被理解为“月平均气温”。

`tas_clim.mean(dim='lon').transpose().plot.contourf(levels=12, robust=True, cmap='turbo')`: 这行代码生成一个温度分布的等高线图。

`.mean(dim='lon')` 计算所有经度（longitude）上的平均值，可能是为了得到一个纬度（latitude）上的平均温度分布。

`.transpose()` 函数改变数据的排列顺序，这通常是为了图形绘制的需要。

`.plot.contourf()` 是绘制等高线填充图的函数，这里用于可视化温度分布。

`levels=12` 表示等高线的数量。

`robust=True` 是为了更好地处理和显示异常值。

`cmap='turbo'` 指定了使用的颜色映射方案，'turbo' 是一种常用的、颜色丰富的方案。

## 第二问：

`group_data = ds.sst.groupby('time.month')`: 将数据集 `ds` 中的海表温度（sst）按照月份进行分组。

`tas_anom = group_data - group_data.mean(dim='time')`: 计算温度异常。这里从每个月的温度数据中减去该月的长期平均温度，从而得到温度异常值。

`tas_anom2 = tas_anom.sel(lon=slice(190, 240), lat=slice(-5, 5))`: 从温度异常数据中选择特定的经纬度范围（120°到 170°西经，5°南到 5°北），这通常被称为厄尔尼诺区域。

`global_mean_anomaly = tas_anom2.mean(dim=['lat', 'lon'])`: 计算上述选定区域内的全球平均温度异常。

接下来的几行代码使用 Matplotlib 库来绘制图表：

`global_mean_anomaly.plot()`: 绘制平均温度异常随时间的变化曲线。

`plt.xlabel('Year')`, `plt.ylabel('Anomaly Degrees C')`, `plt.title(...)`: 设置图表的 x 轴标签、y 轴标签和标题。

`plt.fill_between(...)`: 在温度异常值高于或低于 0 的区域填充颜色，红色表示温度异常高于 0（暖异常，通常与 El Niño 现象相关），蓝色表示温度异常低于 0（冷异常，通常与 La Niña 现象相关）。

`mpatches.Patch(...)`: 创建用于图例的色块。

`plt.legend(...)`: 添加图例。

`plt.text(...)`: 在图表下方添加文本，可能是数据来源或版权信息。

`plt.show()`: 显示最终的图表。

## 第二题:

### 第一问:

`mean_data=ds.mean(dim='time')`: 计算数据集 `ds` 在时间维度上的平均值。这通常用于获取长期平均的气候数据。

`lon = mean_data['lon']` 和 `lat = mean_data['lat']`: 从平均数据中提取经度 (`lon`) 和纬度 (`lat`) 信息。

提取三个变量的平均数据:

`variable_a = mean_data['toa_lw_all_mon']`: 提取长波辐射数据的时间平均值。

`variable_b = mean_data['toa_sw_all_mon']`: 提取短波辐射数据的时间平均值。

`variable_c = mean_data['solar_mon']`: 提取太阳辐射数据的时间平均值。

`plt.figure(figsize=(10, 6))`: 创建一个尺寸为 10x6 英寸的图形。

使用 `plt.subplot(1, 3, 1)`, `plt.subplot(1, 3, 2)`, `plt.subplot(1, 3, 3)` 来创建三个子图。这些代码表示图形将有 1 行 3 列的子图, 并分别激活第 1、2、3 个子图进行绘制。

`plt.contourf(lon, lat, variable_x, cmap='viridis')`: 对每个变量使用等高线填充图 (`contourf`) 进行绘制。这里使用了 'viridis' 颜色映射。

`plt.colorbar()`: 为每个子图添加颜色条, 显示颜色映射的对应值。

`plt.title(...)`: 为每个子图设置标题, 说明所显示的变量和它们是时间平均值。

`plt.tight_layout()`: 调整子图布局, 使之不会重叠。

`plt.show()`: 显示最终的图形。

然后计算三者之和是否相等即可

### 第二问:

`toa_solar_in = ds['solar_mon']`: 从数据集 `ds` 中提取名为 `solar_mon` 的变量, 这代表每月的太阳辐射入射量。

`toa_lw_out = ds['toa_lw_all_mon']`: 提取 `toa_lw_all_mon` 变量, 代表大气顶部全天空条件下的长波辐射出射量。

`toa_sw_out = ds['toa_sw_all_mon']`: 提取 `toa_sw_all_mon` 变量, 代表大气顶部全天空条件下的短波辐射出射量。

`earth_radius = 6371 * 1000`: 定义地球半径的变量, 以米为单位。

`lat = ds['lat']`: 提取数据集中的纬度值。

`lon = ds['lon']`: 提取数据集中的经度值。

`lat_rad = np.deg2rad(lat.values)`: 将纬度值从度转换为弧度。

`grid_area`: 计算格网面积。这一步使用了地球的半径和纬度值来计算每个格网的表面积。

`grid_area_xr`: 将计算出来的格网面积转换为 Xarray 数据数组, 以便与其他数据进行操作。

avg\_toa\_solar\_in: 计算加权后的太阳辐射入射量的平均值。

avg\_toa\_lw\_out: 计算加权后的长波辐射出射量的平均值。

avg\_toa\_sw\_out: 计算加权后的短波辐射出射量的平均值。

**第三问:** start\_lat = ds['lat'].min().values; end\_lat = ds['lat'].max().values: 这两行代码计算

数据集 ds 中纬度 (lat) 的最小值和最大值。这些值用于确定数据选择的纬度范围。

ds\_subset = ds.sel(lat=slice(start\_lat, end\_lat)): 选择数据集 ds 中介于最小纬度和最大纬度之间的所有数据。这样做可以缩小数据分析的范围, 专注于特定的纬度区域。

net\_radiation = ds\_subset['toa\_net\_all\_mon'].sum(dim='lon'): 计算每个纬度上的总辐射 (toa\_net\_all\_mon) 沿经度方向的总和。这一步骤实际上是将每个纬度线上的总辐射值加总起来。

plt.plot(ds\_subset['lat'], net\_radiation.mean(dim='time')): 绘制平均总辐射随纬度的变化曲线。这里, mean(dim='time') 表示计算时间平均值, 这样就可以得到每个纬度在整个时间段内的平均总辐射值。

plt.xlabel('Latitude (degrees)'), plt.ylabel('Net Radiation (units)'), plt.title('Net Radiation vs Latitude'): 设置图表的 x 轴标签、y 轴标签和标题, 分别为纬度、净辐射量以及图表标题。

plt.show(): 显示绘制的图表。

**第四问:**

low\_cloud\_area = ds['cldarea\_total\_daynight\_mon'] <= 25 和 high\_cloud\_area = ds['cldarea\_total\_daynight\_mon'] >= 75: 这两行代码定义了低云和高云区域。低云区域的云层覆盖度小于或等于 25%, 而高云区域的云层覆盖度大于或等于 75%

计算各个条件下的短波和长波辐射的时间平均值:

toa\_sw\_low\_cloud 和 toa\_sw\_high\_cloud 分别计算在低云和高云条件下的短波辐射 (toa\_sw\_all\_mon) 的时间平均值。

toa\_lw\_low\_cloud 和 toa\_lw\_high\_cloud 分别计算在低云和高云条件下的长波辐射 (toa\_lw\_all\_mon) 的时间平均值。

使用 Matplotlib 创建一个 2 行 2 列的子图布局 (plt.figure(figsize=(12, 8)) 和 plt.subplot(2, 2, x)), 并绘制以下四个图:

toa\_sw\_low\_cloud.plot(...): 绘制低云条件下短波辐射的分布图。

toa\_sw\_high\_cloud.plot(...): 绘制高云条件下短波辐射的分布图。

toa\_lw\_low\_cloud.plot(...): 绘制低云条件下长波辐射的分布图。

toa\_lw\_high\_cloud.plot(...): 绘制高云条件下长波辐射的分布图。

每个图都使用了特定的颜色映射 (cmap) 和值范围 (vmin 和 vmax)。

plt.title(...): 为每个子图设置标题, 说明是低云还是高云条件下的短波或长波辐射。

plt.tight\_layout(): 优化子图的布局。

plt.show(): 显示绘制的图表。

**第五问:**

定义云层覆盖度范围:

low\_cloud\_area = ds['cldarea\_total\_daynight\_mon'] <= 25: 定义低云区域为云层覆盖度小于或

等于 25%。

`high_cloud_area = ds['cldarea_total_daynight_mon'] >= 75`: 定义高云区域为云层覆盖度大于或等于 75%。

计算不同云层条件下的全球平均辐射值:

`global_mean_sw_low_cloud` 和 `global_mean_sw_high_cloud` 分别计算低云和高云条件下的全球平均短波辐射 (`toa_sw_all_mon`)。

`global_mean_lw_low_cloud` 和 `global_mean_lw_high_cloud` 分别计算低云和高云条件下的全球平均长波辐射 (`toa_lw_all_mon`)。

这些计算使用 `.where()` 函数来选择符合特定云层条件的数据, 并通过 `mean(dim=('lat', 'lon', 'time'))` 在纬度、经度和时间上取平均值。

计算整体的全球平均辐射值:

`global_mean_sw` 和 `global_mean_lw` 分别计算整体的全球平均短波和长波辐射值。

输出结果:

代码使用 `print` 语句输出全球平均的短波和长波辐射值, 以及这些值在低云和高云区域的对应值。

## 第三题:

### 第一问:

`variable_name = 'app'`: 定义要分析的变量名。

`variable_data = data[variable_name]`: 从数据集 `data` 中提取名为 `'app'` 的变量数据。

计算月度平均值:

`monthly_mean = variable_data.resample(time='1M').mean()`: 将变量数据按月重新采样 (`resample`), 并计算每月的平均值。

计算月度气候学平均值:

`monthly_climatology = variable_data.groupby('time.month').mean(dim='time')`: 将数据按月份分组并计算每个月份的长期平均值 (气候学平均值)。

计算异常值:

`anomaly = variable_data.groupby('time.month') - monthly_climatology`: 计算每个月的异常值。

这里的异常值是指每个月的实际值与其长期气候学平均值的差。

绘制异常值图表:

使用 `plt.figure(figsize=(10, 6))` 创建一个大小为 10x6 英寸的图表。

`anomaly.mean(dim=('latitude', 'longitude')).plot(label='Seasonal Anomaly')`: 计算在所有经度和纬度上的异常值的平均值, 并绘制为一条曲线。这里显示的是季节性异常值。

`plt.title(...)`, `plt.xlabel(...)`, `plt.ylabel(variable_data.units)`: 设置图表的标题、x 轴和 y 轴标签。y 轴标签使用变量的单位。

`plt.legend()`: 添加图例。

`plt.grid(True)`: 开启网格线, 以便于观察数据点。

`plt.show()`: 显示绘制的图表。

## 第二问：

分别绘制全球平均时间序列图、经度时间图、纬度时间图、经纬度标准差热图、经纬度热图