

1.1 Compute monthly climatology for SST from Niño 3.4 region

使用 `numpy` 库中的 `cos` 函数和 `deg2rad` 函数计算纬度对应的权重。

选取特定经纬度范围内的 SST 数据，并使用权重进行加权平均。

对加权后的 SST 数据求月平均值。

计算每月 SST 的异常值（即当前值减去月平均值）。

对 SST 异常值进行移动平均处理，得到连续三个月的移动平均异常值。

创建一个数据框，包含每月的三个月移动平均异常值、月份和日期等信息。

添加厄尔尼诺判断条件：如果连续五个月的三个月移动平均异常值超过阈值，则视为一次厄尔尼诺现象。

1.2 Visualize the computed Niño 3.4

以 "date" 作为 x 轴，"mean_3_anomalies" 作为 y 轴，绘制折线图。

添加三条水平直线，分别表示阈值（0.5°C）、零线（0°C）和负阈值（-0.5°C）。

设置 x 轴标签为 'year'，表示年份；设置 y 轴标签为 'sst_anomalies[°C]'，表示海洋表面温度异常（单位：°C）。

设置图例中的字体旋转角度为 45 度，以便在狭小的空间内展示更多的信息。

设置标题为 'Niño 3.4 (5° N-5° S, 120° W-170° W)'，表示绘制的区域和范围。

最后 `show()` 显示图像

2.1

用 `xarray` 库打开一个 NetCDF 文件。

创建一个 `figure`，设置其大小为 10x6，并设置每英寸 120 像素的分辨率。

定义一个 5 行 9 列网格布局

分别绘制 5 个子图，并设置标题和图例，以便在图中清晰显示各变量的含义。

使用 `tight_layout` 函数调整图形的布局，使其更加紧凑。

2.2

首先，清空列表 `temp`，用于存储计算结果。

计算太阳辐射通量、短波辐射通量和长波辐射通量之差，并将结果存储在列表 `x` 中。

计算顶层大气净辐射通量，并将结果存储在列表 `y` 中。

使用 `matplotlib` 的 `scatter` 函数绘制散点图，其中 x 轴表示计算得到的差值，y 轴表示顶层大气净辐射通量。

设置 x 轴和 y 轴的标签，分别为 'calculate_toa' 和 'toa_net'。

使用 `matplotlib` 的 `sum` 函数计算 x 和 y 轴数据的和。

2.3

使用 `numpy` 库中的 `cos` 函数和 `deg2rad` 函数计算纬度对应的权重。

对 `d.toa_net_all_mon` 进行加权，权重为计算得到的纬度权重（`weights`）。

对加权后的顶层大气净辐射通量求月平均值，并使用 `plot` 函数绘制结果。

设置 x 轴和 y 轴的标签，分别为 'lat/' 和 'toa_net_all_mon/W'。

设置标题为 'Total amount of net radiation'

`Plt.Show()` 显示图像

2.4

创建一个 `figure`，设置其大小为 10x6，并设置每英寸 120 像素的分辨率。

定义一个网格布局，包含 4 行 4 列。

分别绘制四个子图，每个子图表示一种辐射类型，SW 和 LW 在 `cloud area <= 25%` 和 `cloud area >= 75%` 的月平均值。

在每个子图的标题中，分别说明所绘制的辐射类型和云覆盖区域。

使用 `tight_layout` 函数调整图形的布局，使其更加紧凑。

2.5

创建一个 `figure`，设置其大小为 `18x6`，并设置每英寸 600 像素的分辨率。

定义一个网格布局，包含 1 行 2 列。

在两个子图中，分别绘制 `TOA_SW_ALL_MON` 和 `TOA_LW_ALL_MON` 在全球、`clarea_total_daynight_mon >= 75` 和 `clarea_total_daynight_mon <= 25` 的月平均值，并在图例中添加标签以区分各区域。

为每个子图设置 x 轴和 y 轴的标签，分别为 `'time'` 和 `'toa_sw_all_mon'`。

设置图例的位置为 `'best'`，以便在图中清晰显示。

设置 x 轴和 y 轴的刻度标签字体大小为 18 像素。

使用 `tight_layout` 函数调整图形的布局，使其更加紧凑。

3.1

使用 `xarray` 库打开一个 NetCDF 文件。

计算地表温度（T2M）的月平均值，`dimensions` 为 `('latitude', 'longitude')`。

计算权重，权重为纬度对应的余弦值。

对 T2M 数据进行加权处理，并计算加权后的月平均值。

计算全球地表温度的日平均值，存储在数组 `t_day` 中。

将 `t_day` 数组转换为 `pandas DataFrame` 格式，并添加日期和年度、月份列。

对月度数据进行分组，计算每年的月平均温度。

计算每月地表温度与年平均地表温度之间的差值。

将差值数据绘制出来，并设置 y 轴标签为 `'temperature[°C]'`。

这幅图展示了地表温度与年平均地表温度之间的差异。

3.2

1 画均值时间序列图。

2 画一个正态分布图。

导入所需的库。

设置随机数种子，以便每次运行时生成的数据相同。

定义正态分布的均值（`mu`）和标准差（`sigma`）。

生成一个随机数列表（`x`），符合正态分布。

计算 `histogram` 的参数，包括 `bin` 数量（`num_bins`）和密度（`density=1`）。

使用 `matplotlib` 绘制 `histogram`。

设置图形的标题、x 轴和 y 轴标签。

使用 `tight_layout` 调整图形的布局，以避免 y 轴标签被截断。

显示图形。

3 画一个箱线图。

将数据 `a` 赋值给变量 `data`。

使用 `matplotlib` 中的 `boxplot` 函数绘制箱线图。

4 绘制一个地理空间分布图。

使用了 `xarray` 和 `matplotlib` 库来绘制一个关于地表温度（T2M）的地图。

首先对一个名为 `T_weighted` 的数据结构进行操作。使用 `.mean()` 来计算 T2M 在经度和纬度方向上的平均值。`dim=('longitude', 'latitude')` 是一个参数，用于指定在哪个维度上进行求均值操作。

设置地图的 y 轴标签。`t2m[K]` 表示地表温度的单位，这里用（K）表示。

5 绘制一个月度地表温度（T2M）的时间序列图。

首先对一个名为 `monthly_T` 的数据结构进行操作。使用 `.plot()` 来绘制时间序列图。设置时间序列图的 `y` 轴标签。`t2m[K]` 表示地表温度的单位，这里用（K）表示。