

实验 1 L2 全球海表温度变化分析

编写人：陈玥舟（报告撰写与代码编写）、蒋浩宇（内容设计与完善）

一、实验目的

1. 了解如何获取全球海表面温度数据；
2. 初步掌握用 python 对网格化数据进行时空分析的思路；
3. 复习用 python 数据可视化的基本操作。

二、实验任务

- 1、在 <https://psl.noaa.gov/data/gridded/tables/sst.html> 下载 OISST V2 1990 年至今的 SST 月分辨率数据（sst.wkmean.1990-present.nc）
- 2、用 python 进行数据的信息查询和读取。
- 3、画出 1990 年 1 月 SST 的全球分布图，标题、网格、坐标轴、colorbar 的要素都要有。
- 4、画出 1990 年-2019 年 SST 平均值的全球分布图。
- 5、选择一个位置，画出该点 1990-2019 年海表面温度的时间序列，并利用线性拟合计算出该点海表面温度上升的趋势。
- 6、画出 1990 年-2019 年 SST 变化趋势的全球分布图。

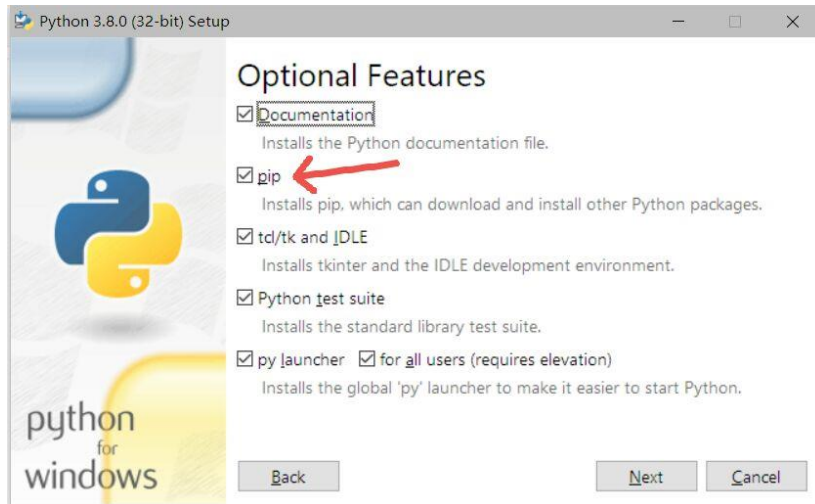
三、实验步骤

1. Python 环境搭建

搭建 Python 环境的常见模式是安装一个 Python 的 IDE（集成开发环境），再使用 pip 或 conda 等命令安装特别扩展的第三方包，常见的 Python IDE/平台有 Python 自带的 IDE（IDLE）、Anaconda、Pycharm、VScode 和 Jupyter Notebook 等。其中，Jupyter Notebook 是一个基于 Web 网页形式的交互式笔记本应用，能可视化展示并保存运行结果。本报告将以 Jupyter Notebook 作为基本编程工具进行实验。

（1）在 Python 官方网址 www.python.org/downloads/ 下载相应的 Python3.x 版本（建议下载 3.6 以上版本）（Python 2.7.9 + 或 Python 3.4+ 以上版本，如下图可在安装时勾选 pip 选项完成 pip 配置。Python3.x 版本可以通过 cmd 下输入命令 `pip3 --version` 来判断是否已安装。如未安装，则可打

开网址 <https://bootstrap.pypa.io/get-pip.py> -o get-pip.py 下载安装相对脚本，接着 cmd 下输入命令 python3 get-pip.py 运行安装脚本）；



(2) 打开 cmd，输入命令：pip install jupyter notebook 安装 Jupyter Notebook；

(3) 安装一个 Python 的专门处理 nc 文件的库：netCDF4。这个库可以通过打开 cmd，直接输入命令 pip install netCDF4 来安装。（除了 netCDF4 这个库以外，还会用到 numpy、matplotlib 库，同样可以通过在 cmd 输入命令 pip install “需要安装的库名” 来进行安装）

(4) 安装 Basemap 工具包。这是 python 可视化库 Matplotlib 下的一个工具包，主要功能是绘制二维地图，对空间数据的可视化非常重要。打开网址 “<http://www.lfd.uci.edu/~gohlke/pythonlibs/>”，根据自己安装的 python 版本手动下载相应的.whl 文件（可在页面上按 Ctrl+F，输入 basemap 快速定位），如 “basemap-1.2.2-cp36-cp36m-win_amd64.whl”；把文件下载到一个路径下，如 “C:\Users\famil\basemap-1.2.2-cp36-cp36m-win_amd64.whl”；在 cmd 下定位到文件所在文件夹，然后输入命令 “pip install ‘.whl 文件名’” 完成安装，如 “pip install basemap-1.2.2-cp36-cp36m-win_amd64.whl”。

2. 下载数据

1、登录 <https://psl.noaa.gov/> 网站，浏览网站信息。

2、找到数据获取页面，按照数据目录找到需要的数据，下载 sst.wkmean.1990-present.nc。

1

2 Climate Datasets: By Category

- All
- Sub-daily
- Daily
- Monthly
- Surface
- Temperature
- SST**
- Precipitation

3

Temperature (NOAA GlobalTemp)

NOAA ERSST V4 dataset.

NOAA Optimum Interpolation (OI) SST V2

Produced weekly on a 1° grid. The analysis uses in situ and satellite SST's plus SST's simulated by sea-ice cover. Before the analysis is computed, the satellite data is adjusted for biases. Monthly and monthly long-term means are also available.

NODC (Levitus) World Annual, seasonal and monthly long term means for temperature and

Download/Plot Data:

4

Variable	Statistic	Level	Download File	Create Plot/Subset
Sea Surface Temperature	Mean	Surface	sst.wkmean.1981-1989.nc sst.wkmean.1990-present.nc	
Ice Concentration	Mean	Surface	icec.wkmean.1981-1989.nc icec.wkmean.1990-present.nc	
Normalized Error Variance	Mean	Surface	err.wkmean.1981-1989.nc err.wkmean.1990-present.nc	
Sea Surface Temperature	Mean	Surface	sst.mnmean.nc	
Ice Concentration	Mean	Surface	icec.mnmean.nc	
Sea Surface Temperature	Long Term Mean	Surface	sst.ltm.1961-1990.nc sst.ltm.1971-2012.nc	
Land Sea Mask	Mask	Surface	lsmask.nc	

3. 读取数据信息

在 cmd 下输入命令：jupyter notebook，打开 Jupyter Notebook 编辑器。
Jupyter 窗口输入以下内容：

```
import netCDF4 as nc #导入库
file=nc.Dataset('sst.wkmean.1990-present.nc')
print(file)#读取文件信息
```

可以看到数据的位置、类型、全局属性、变量以及文件夹等信息，如下图所示：

```
<class 'netCDF4._netCDF4.Dataset'>
root group (NETCDF3_CLASSIC data model, file format NETCDF3):
  title: NOAA Optimum Interpolation (OI) SST V2
  Conventions: CF-1.0
  history: Created 10/2002 by RHS
  comments: Data described in Reynolds, R.W., M.A. Rayner, T.M.
Smith, D.C. Stokes, and W. Wang, 2002: An Improved In Situ and Satellite
SST Analysis for Climate, J. Climate
  platform: Model
  source: NCEP Climate Modeling Branch
  institution: National Centers for Environmental Prediction
  References: https://www.psl.noaa.gov/data/gridded/data.noaa.oisst.v2.html
  NC0: 4.0.0
  dataset_title: NOAA Optimum Interpolation (OI) SST V2
  source_url: http://www.emc.ncep.noaa.gov/research/cmb/sst_analysis/
  dimensions(sizes): lat(180), lon(360), time(1631), nbnds(2)
  variables(dimensions): float32 lat(lat), float32 lon(lon), int16 sst(time, lat, lon), float64 time(time), float64 time_bnds(time, nbnds)
```

4. 数据读取

结合数据信息，将经纬度、时间和海表面温度信息读入数组中。

```
lon = file.variables['lon'][:] #经度
lat = file.variables['lat'][:] #纬度
sst = file.variables['sst'][:] #海表温度
time = file.variables['time'][:] #时间
time_bnds = file.variables['time_bnds'][:] #时间段
```

5. 画出 1998 年和 1990 年-2019 年 SST 平均值的全球分布图

在 python 进行地图绘制方法众多，basemap 包是一个很常用的插件。

首先筛选规定时间内的数据，以下代码筛选出 1998 年 1 年的 sst 数据，并求其平均值。

```
#导入库
import numpy as np
import datetime
#筛选 1998 年的 sst 数据
stdate = [1800,1,1] #days since 1800-1-1 00:00:00, 起始时间
time1 = datetime.date(1998,1,1) - datetime.date(1800,1,1) #1998.1.1 距起始时间的天数
time2 = datetime.date(1998,12,31) - datetime.date(1800,1,1) #1998.12.31 距起始时间的天数
trange = np.where((time >= time1.days)&(time <= time2.days),1,0) #找出所有一月的数据
sst1998 = np.mean(sst[trange==1,:,:),0) #平均
```

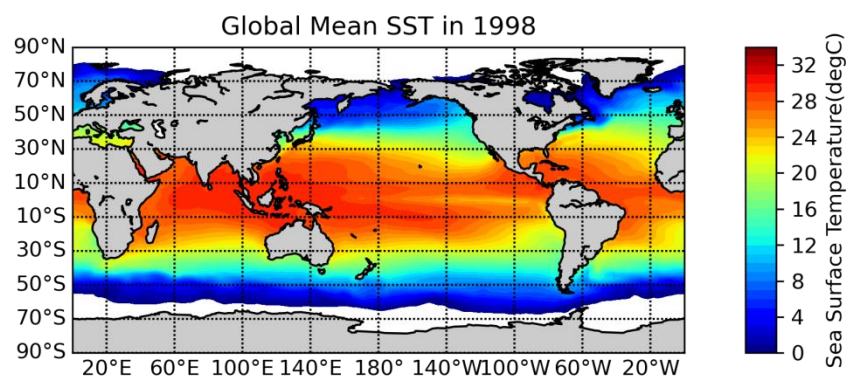
在绘制地图之前，先对经度、纬度进行网格化，经纬度网格与 sst 保持一致。数据准备完成后始做图，以下是绘图代码：

```

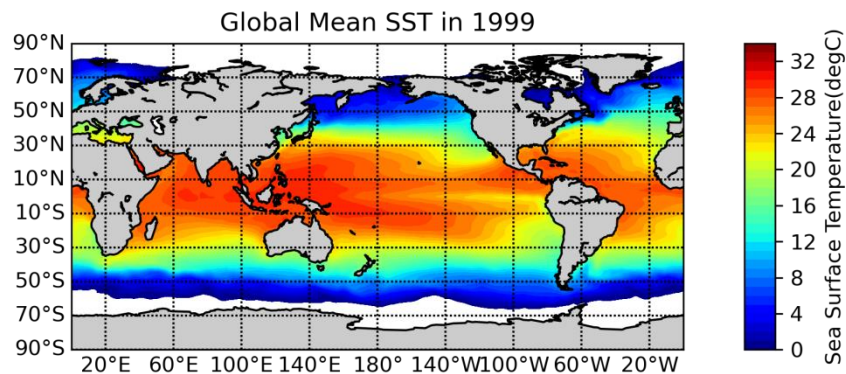
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
#获取地图中心经纬度坐标 lat0 和 lon0
lon0 = lon.mean()
lat0 = lat.mean()
#对经纬度网格化
lons, lats = np.meshgrid(lon, lat)
#画图
m= Basemap(lat_0=lat0, lon_0=lon0)
#为地图添加上海岸线和经纬网格
m.drawparallels(np.arange(-90.,91.,20.),labels=[1,0,0,0],fontsize=10)
m.drawmeridians(np.arange(-180.,181.,40.), labels=[0,0,0,1], fontsize=10)
m.drawcoastlines()
m.fillcontinents()
cs=m.contourf(lons,lats,sst1998,range(0,35,1),cmap='jet')
cbar = m.colorbar(cs,pad="10%",label='Sea Surface Temperature(degC)') #显示颜色图例
plt.title('Global Mean SST in 1998') #添加标题
plt.show()

```

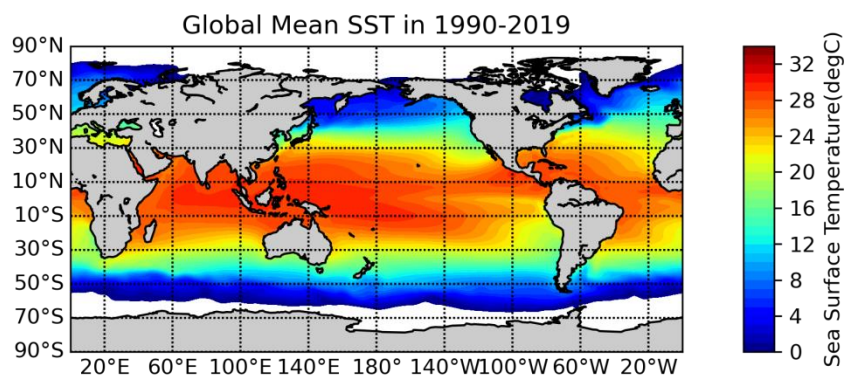
结果如下：



利用类似的代码画出 1999 年的平均 SST



利用类似的代码画出 1990 年-2019 年多年的 SST 平均值。



进一步分别尝试画出 1998 和 1999 年的 SST Anomaly（距平），即某年的均值与多年均值的差值。能否发现 1998 年的数据和 1999 年的数据有什么区别？（见附录）

6. 利用线性拟合计算海表面温度上升的趋势

使用函数 `polyfit` 进行线性拟合，拟合结果能写为： $y=kx+c$ ， y 代表海表面温度， x 代表时间， k 可以反应温度变化的速率， c 为常数项。

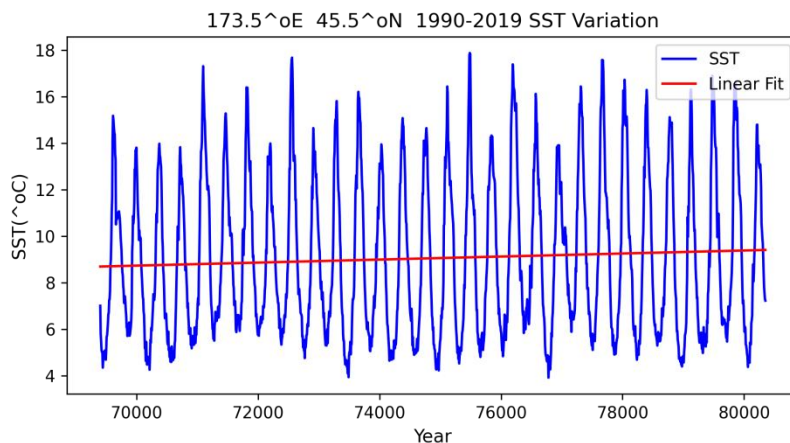
为计算一点的 1990 年-2019 年 SST 变化趋势，先筛选出 1990 至 2019 年的数据，这里选取东经 173.5 度北纬 45.5 度进行分析，结合这一点的海表面温度的时间序列，画出拟合结果，代码如下：

```

#筛选 1990-2019 年的 sst 数据
startdate = [1800,1,1]      #days since 1800-1-1 00:00:00, 起始时间
#1990.1.1 距起始时间的天数
time1 = datetime.date(1990,1,1) - datetime.date(1800,1,1)
#2019.12.31 距起始时间的天数
time2 = datetime.date(2019,12,31) - datetime.date(1800,1,1)
trange2 = np.where((time >= time1.days)&(time <= time2.days),1,0)
time9019 = time[trange2==1]
sst1 = np.squeeze(sst[trange2==1,45,174])  #根据 lon、lat 定位目标位置
#返回斜率, x=0 时 y 的值
reg = np.polyfit(time9019, sst1,1)
#拟合结果, 等效于 yi2 = p(1)* time9019 +p(2)
ry = np.polyval(reg, time9019)
plt.figure(figsize=(8, 4))
plt.plot(time9019, sst1,'b')#蓝色
# 拟合绘制
plt.plot(time9019, ry, 'r')#红色
plt.xlabel('Year',fontsize=11)
plt.ylabel('SST(^oC)',fontsize=11)
plt.title('173.5^oE 45.5^oN 1990-2019 SST Variation')
plt.legend(['SST','Linear Fit'],loc='upper right')
plt.show()

```

结果如下:

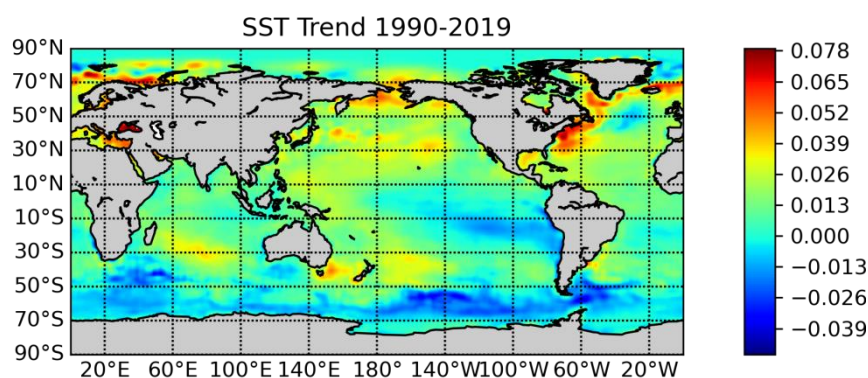


全球海表温度变化的趋势,就是把一点的趋势拟合扩展到全球,所有数据点都拟合一次,取拟合结果的斜率乘以 29 年的时间,作为温度变化趋势的指标,代码如下:


```
#先定义一个矩阵存储趋势
trendsst = np.zeros((180,360))
for n in range(0,180):
    for m in range(0,360):
        sst1=np.squeeze(sst[trange2==1,n,m])
        p = np.polyfit(time9019, sst1,1)
        trendsst[n,m] = p[0]*365.25 #拟合直线的斜率即趋势,度/天
```

得到变化趋势的数据后，重复第 5 步全球 SST 分布的画法，绘制全球 SST 变化趋势全球分布。

结果如下：



四、课堂/课后作业：

1.基本内容

- 1、画出 1990 年-2019 年多年的 SST 平均值。
- 2、进一步分别尝试画出 1998 和 1999 年的 SST Anomaly（距平），即某年的均值与多年均值的差值。能否发现 1998 年的数据和 1999 年的数据有什么区别？为什么？
- 3、绘制 1990 年-2019 年全球 SST 变化趋势全球分布。
- 4、绘制 1990 年-2019 年全球不同位置 SST 最大和最小值，以及最大最小值差值的分布，并描述分布的特征。
- 5、绘制 1990 年-2019 年全球不同位置 SST 的标准差，并描述分布特征。
- 6、选取 180° 经线上 5 个不同纬度（0°，20° N，40° N，60° N，80° N）的点，在同一张图上画出他们 1990 年-2000 年温度变化的曲线，能够发现什么规律？

7、另选取 180° 经线上 5 个不同纬度 (60° S, 30° S, 0° , 30° N, 60° N) 的点, 在同一张图上画出他们 1990 年-2000 年温度变化的曲线, 能够发现什么规律?

报告提交内容: (1) 数据、(2) 所有的 .m/.py 的完整程序源代码、
(3) 输出的图片(动画不用保存)、(4) 报告正文, 打包放在一个文件夹中。
要求运行代码即可完成结果的

附录：

#1998 和 1999 年的 SST Anomaly (距平)

```
import numpy as np
import datetime
import netCDF4 as nc
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt

#读取文件
file=nc.Dataset('sst.wkmean.1990-present.nc')
#读取文件里的变量
lon = file.variables['lon'][:]    #经度
lat = file.variables['lat'][:]    #纬度
sst = file.variables['sst'][:]    #海表温度
time = file.variables['time'][:]  #时间
time_bnds = file.variables['time_bnds'][:] #时间段

#多年均值
stdte = [1800,1,1]                #days since 1800-1-1 00:00:00, 起始时间
time1 = datetime.date(1990,1,1) - datetime.date(1800,1,1) #1999.1.1 距起始时间的天数
time2 = datetime.date(2019,12,31) - datetime.date(1800,1,1)#2019.12.31 距起始时间的天数
trange = np.where((time >= time1.days)&(time <= time2.days),1,0)
sstall = np.mean(sst[:, :, 0]) #平均

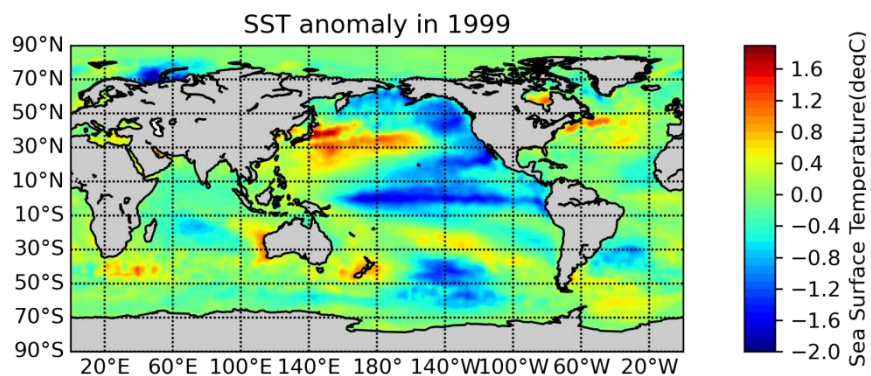
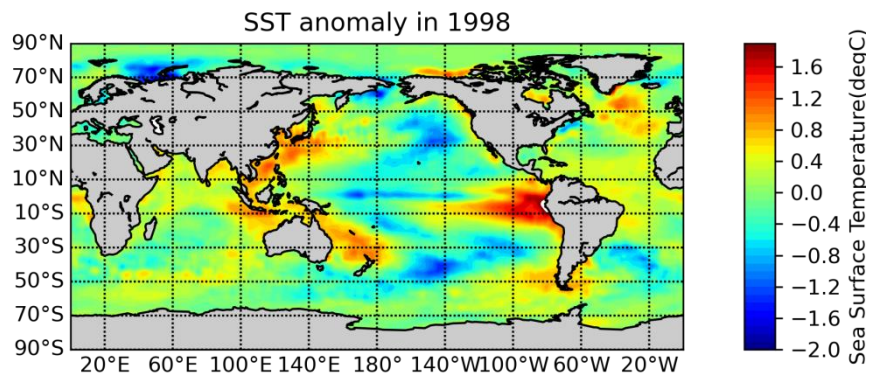
#1998 年均值
time1 = datetime.date(1998,1,1) - datetime.date(1800,1,1)
time2 = datetime.date(1998,12,31) - datetime.date(1800,1,1)
trange = np.where((time >= time1.days)&(time <= time2.days),1,0)
sst1998 = np.mean(sst[trange==1, :, 0]) #平均

#1998 年距平
ssta1998 = sst1998-sstall

#获取地图中心经纬度坐标
lon0 = lon.mean()
lat0=lat.mean()
lons, lats = np.meshgrid(lon, lat)

#画图
m= Basemap(lat_0=lat0, lon_0=lon0)
#为地图添加上海岸线和经纬网格
m.drawparallels(np.arange(-90.,91.,20.),labels=[1,0,0,0],fontsize=10)
m.drawmeridians(np.arange(-180.,181.,40.), labels=[0,0,0,1], fontsize=10)
m.drawcoastlines()
m.fillcontinents()
cs=m.contourf(lons,lats,ssta1998,np.arange(-2,2,0.1),cmap='jet')
cbar = m.colorbar(cs,pad="10%",label='Sea Surface Temperature(degC)') #显示颜色图例
```

```
plt.title('SST anomaly in 1998')    #添加标题
plt.savefig('C:\\Users\\DELL\\Desktop\\图片名.png',dpi=300) #保存图片
plt.show()
#1999 的距平与 1998 的距平代码相似
```



课后题

4.

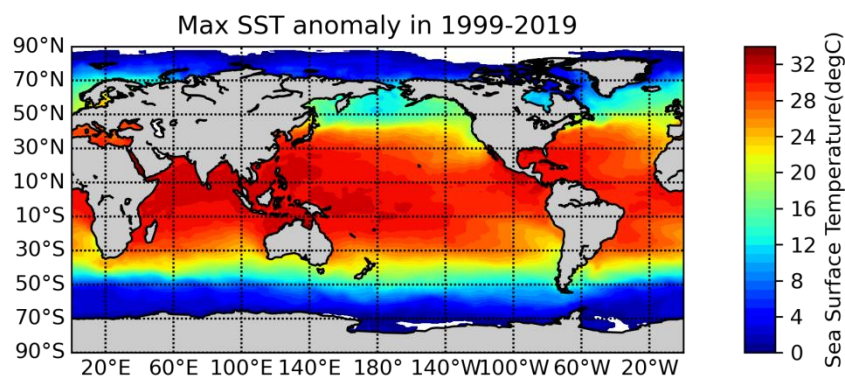
1990 年-2019SST 最大值

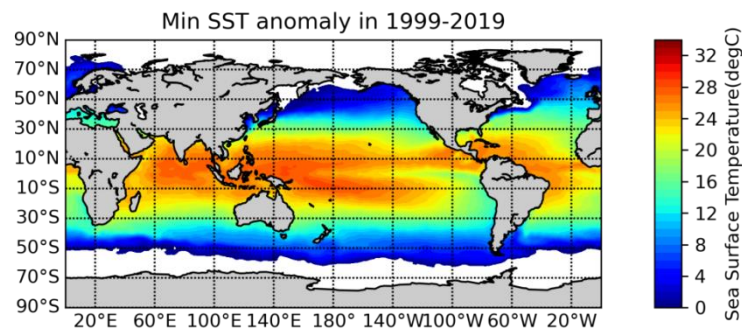
```
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
import numpy as np
import datetime
import netCDF4 as nc
file=nc.Dataset('sst.wkmean.1990-present.nc')
#读取文件里的变量
```

```

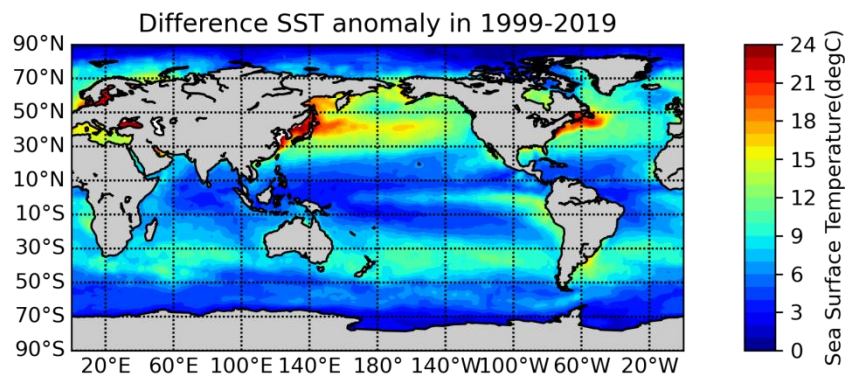
lon = file.variables['lon'][:] #经度
lat = file.variables['lat'][:] #纬度
sst = file.variables['sst'][:] #海表温度
time = file.variables['time'][:] #时间
time_bnds = file.variables['time_bnds'][:] #时间段
startdate = [1800,1,1] #days since 1800-1-1 00:00:00, 起始时间
time1 = datetime.date(1990,1,1) - datetime.date(1800,1,1) #1999.1.1 距起始时间的天数
time2 = datetime.date(2019,12,31) - datetime.date(1800,1,1)#2019.12.31 距起始时间的天数
trange = np.where((time >= time1.days)&(time <= time2.days),1,0)
sstmax = np.max(sst[trange==1,:],0) #最大值
#画图
lon0 = lon.mean()
lat0=lat.mean() #获取地图中心经纬度坐标
lons, lats = np.meshgrid(lon, lat)
m= Basemap(lat_0=lat0, lon_0=lon0)
#为地图添加上海岸线和经纬网格
m.drawparallels(np.arange(-90.,91.,20.),labels=[1,0,0,0],fontsize=10)
m.drawmeridians(np.arange(-180.,181.,40.), labels=[0,0,0,1], fontsize=10)
m.drawcoastlines()
m.fillcontinents()
cs=m.contourf(lons,lats,sstmax,range(0,35,1),cmap='jet')
cbar = m.colorbar(cs,pad="10%",label='Sea Surface Temperature(degC)') #显示颜色图例
plt.title('Max SST anomaly in 1999-2019') #添加标题
plt.show()
#最小值与最大值相似, sstmin = np.min(sst[trange==1,:],0)

```





#最大最小值差值的分布，只需用最大值减去最小值，重复画图步骤即可
`sstdiff = sstmax-sstmin`



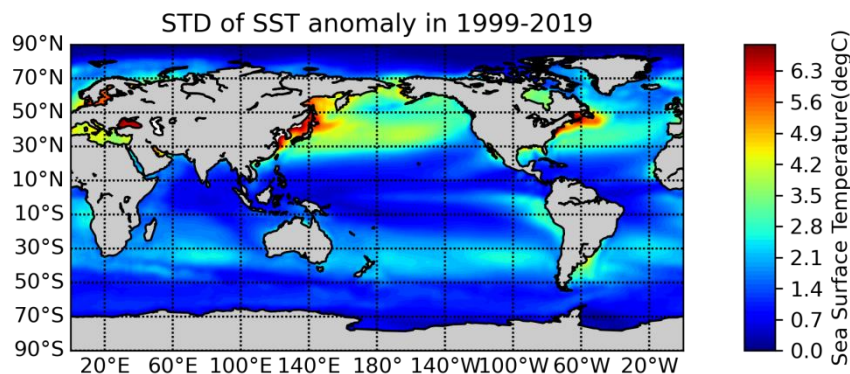
5.1990 年-2019 年全球不同位置 SST 的标准差

```
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
import numpy as np
import datetime
import netCDF4 as nc #导入库
file=nc.Dataset('sst.wkmean.1990-present.nc')
#读取文件里的变量
lon = file.variables['lon'][:] #经度
lat = file.variables['lat'][:] #纬度
sst = file.variables['sst'][:] #海表温度
time = file.variables['time'][:] #时间
```

```

stddate = [1800,1,1]          #days since 1800-1-1 00:00:00, 起始时间
time1 = datetime.date(1990,1,1) - datetime.date(1800,1,1) #1990.1.1 距起始时间的天数
time2 = datetime.date(2019,12,31) - datetime.date(1800,1,1)#2019.12.31 距起始时间的天数
trange = np.where((time >= time1.days)&(time <= time2.days),1,0)
sststd= np.std(sst[trange==1,:],0) #标准差
#画图
lon0 = lon.mean()
lat0=lat.mean() #获取地图中心经纬度坐标
lons, lats = np.meshgrid(lon, lat)
m= Basemap(lat_0=lat0, lon_0=lon0)
#为地图添加上海岸线和经纬网格
m.drawparallels(np.arange(-90.,91.,20.),labels=[1,0,0,0],fontsize=10)
m.drawmeridians(np.arange(-180.,181.,40.), labels=[0,0,0,1], fontsize=10)
m.drawcoastlines()
m.fillcontinents()
cs=m.contourf(lons,lats,sststd,np.arange(0,7,0.1),cmap='jet')
cbar = m.colorbar(cs,pad="10%",label='Sea Surface Temperature(degC)') #显示颜色图例
plt.title('STD of SST anomaly in 1999-2019') #添加标题
plt.show()

```



6.

```

# 180°经线上 5 个不同纬度 (0°, 20°N, 40°N, 60°N, 80°N) 的点, 在同一张图上画出他们
# 1990 年-2000 年温度变化的曲线
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
import numpy as np
import datetime
import netCDF4 as nc #导入库

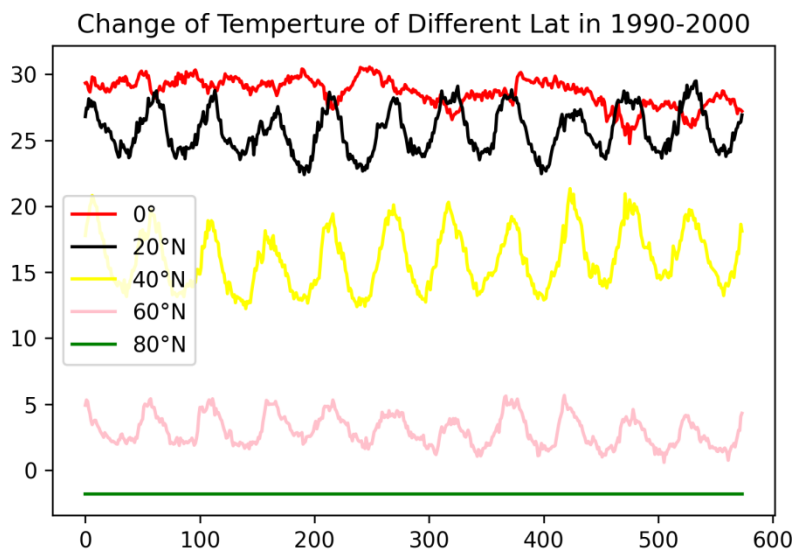
```



```

file=nc.Dataset('sst.wkmean.1990-present.nc')
#读取文件里的变量
lon = file.variables['lon'][:] #经度
lat = file.variables['lat'][:] #纬度
sst = file.variables['sst'][:] #海表温度
time = file.variables['time'][:] #时间
startdate = [1800,1,1] #days since 1800-1-1 00:00:00, 起始时间
time1 = datetime.date(1990,1,1) - datetime.date(1800,1,1) #1990.1.1 距起始时间的天数
time2 = datetime.date(2000,12,31) - datetime.date(1800,1,1) #2000.12.31 距起始时间的天数
trange = np.where((time >= time1.days)&(time <= time2.days),1,0)
sst0 = sst[trange==1,90,180]
sstN20 = sst[trange==1,110,180]
sstN40 = sst[trange==1,130,180]
sstN60 = sst[trange==1,150,180]
sstN80 = sst[trange==1,170,180]
plt.plot(range(0,len(sst0)),sst0,color="red",label = "0°")
plt.plot(range(0,len(sstN20)),sstN20,color="black",label = "20°N")
plt.plot(range(0,len(sstN40)),sstN40,color="yellow",label = "40°N")
plt.plot(range(0,len(sstN60)),sstN60,color="pink",label = "60°N")
plt.plot(range(0,len(sstN80)),sstN80,color="green",label = "80°N")
plt.legend()
plt.title('Change of Temperture of Different Lat in 1990-2000') #添加标题
plt.show()

```



7.

180°经线上 5 个不同纬度 (60°S, 30°S, 0°, 30°N, 60°N) 的点, 在同一张图上画出他们 1990 年-2000 年温度变化的曲线

```

from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
import numpy as np
import datetime
import netCDF4 as nc #导入库
file=nc.Dataset('sst.wkmean.1990-present.nc')
#读取文件里的变量
lon = file.variables['lon'][:] #经度
lat = file.variables['lat'][:] #纬度
sst = file.variables['sst'][:] #海表温度
time = file.variables['time'][:] #时间
stdate = [1800,1,1] #days since 1800-1-1 00:00:00, 起始时间
time1 = datetime.date(1990,1,1) - datetime.date(1800,1,1) #1990.1.1 距起始时间的天数
time2 = datetime.date(2000,12,31) - datetime.date(1800,1,1)#2000.12.31 距起始时间的天数
trange = np.where((time >= time1.days)&(time <= time2.days),1,0)
sstS60 = sst[trange==1,30,180]
sstS30 = sst[trange==1,60,180]
sst0 = sst[trange==1,90,180]
sstN30 = sst[trange==1,120,180]
sstN60 = sst[trange==1,150,180]
plt.plot(range(0,len(sstS60)),sstS60,color="red",label = "60°S")
plt.plot(range(0,len(sstS30)),sstS30,color="black",label = "30°S")
plt.plot(range(0,len(sst0)),sst0,color="yellow",label = "0°")
plt.plot(range(0,len(sstN30)),sstN30,color="pink",label = "30°N")
plt.plot(range(0,len(sstN60)),sstN60,color="green",label = "60°N")
plt.legend()
plt.title('Change of Temperture of Different Lat in 1990-2000') #添加标题
plt.show()

```

