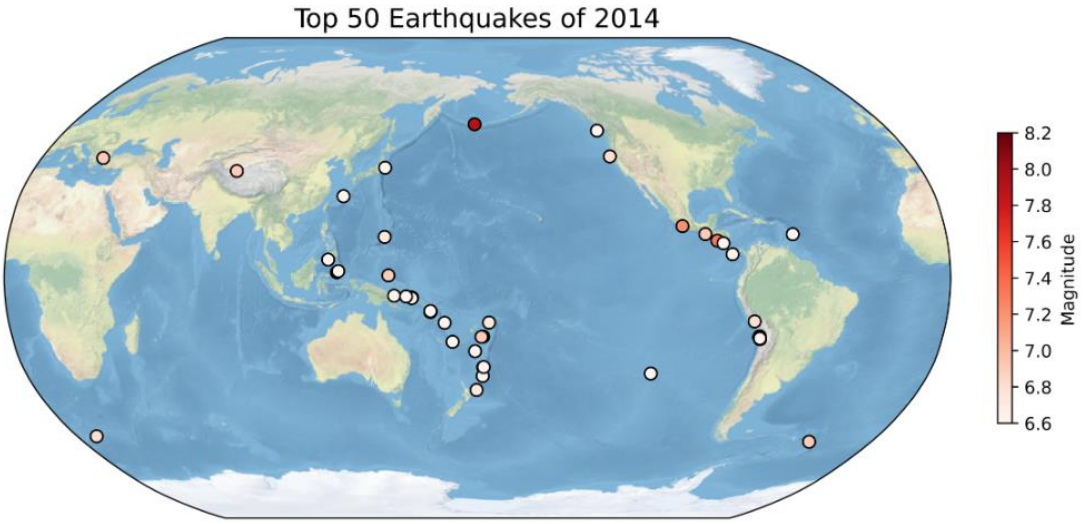
PS4

```
import netCDF4
import numpy as np
import pandas as pd
import xarray as xr
from matplotlib import pyplot as plt
%matplotlib inline
import cartopy.crs as ccrs
import cartopy.feature as cfeature
```

导入所需要的库

```
#Question 1

earthquakes = pd.read_csv('usgs_earthquakes.csv')

# 按震级降序排列 DataFrame，并取前 50 行
top_50_earthquakes = earthquakes.sort_values('mag', ascending=False).head(50)

# 创建一个使用罗宾逊投影的图表
fig = plt.figure(figsize=(12, 12), dpi=1000)
projection = ccrs.Robinson(central_longitude=180, globe=None)
ax = plt.axes(projection=projection)
ax.set_global()
ax.stock_img()

# 提取经度和纬度
lon = top_50_earthquakes.longitude
lat = top_50_earthquakes.latitude

#坐标转换方式（'transform=ccrs.PlateCarree()'，表示使用 PlateCarree 投影）
scatter = ax.scatter(lon, lat, c=top_50_earthquakes.mag, s=50, marker='o', cmap='Reds',
vmin=6.6, vmax=8.2, edgecolors='k', transform=ccrs.PlateCarree())

# 添加颜色条
colorbar = plt.colorbar(scatter, ax=ax, shrink=0.25, pad=0.04, label='Magnitude')

plt.title('Top 50 Earthquakes of 2014', fontsize=15)
plt.show()
```

这段代码的目的是通过地图展示 2014 年震级最高的 50 个地震事件的分布情况，颜色映射表示震级的大小，点的大小表示地震事件的相对强度。这样的可视化有助于对地震发生的空间分布进行直观的理解。

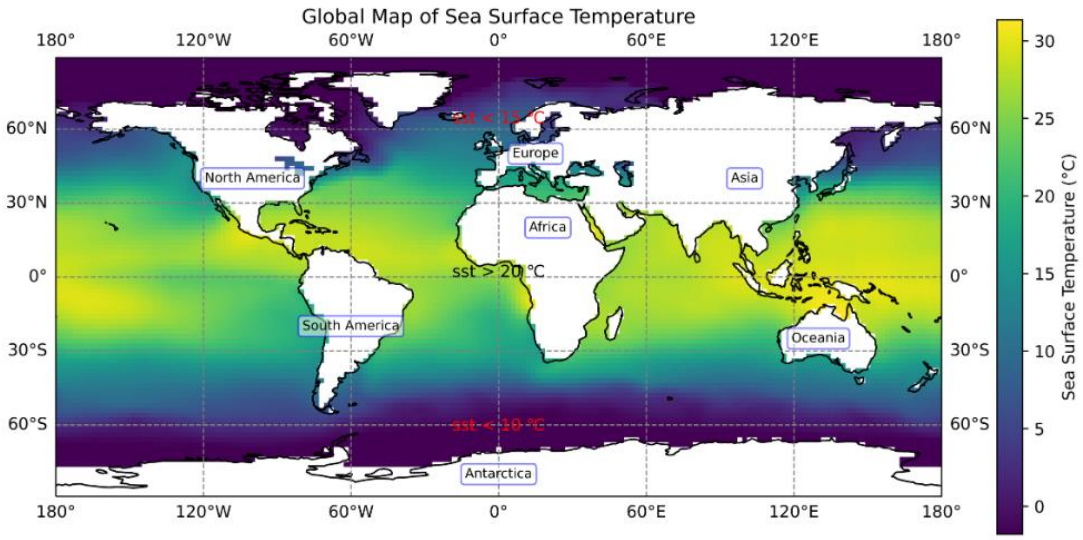


#Question 2.1

```
ds = xr.open_dataset("sst.mnmean.v4.nc", engine="netcdf4")
data = ds['sst']
#https://zhuanlan.zhihu.com/p/660009602
# 使用 PlateCarree 投影
fig, ax = plt.subplots(figsize=(12, 8), dpi=1000, subplot_kw={'projection': ccrs.PlateCarree()})
#使用 pcolormesh 方法绘制海表温度的颜色图
c = ax.pcolormesh(data['lon'], data['lat'], data.isel(time=10), cmap='viridis',
transform=ccrs.PlateCarree())

# 设置标签
ax.set_xlabel('Longitude')
ax.set_ylabel('Latitude')
ax.set_title('Global Map of Sea Surface Temperature')

# 加入背景网格线
ax.gridlines(draw_labels=True, linestyle='--', color='gray')

# 加入颜色条垂直右侧
cb = plt.colorbar(c, label='Sea Surface Temperature (°C)', orientation='vertical', shrink=0.7)

#加上大陆板框的海岸线，加入国家线的时候需要下载地图，解压之后一直运用不了
ax.add_feature(cfeature.COASTLINE, edgecolor='black')

# 在地图不同温度区域加上注释
ax.annotate('sst < 15 ℃', xy=(0.5, 0.85), xycoords='axes fraction', fontsize=10,
```

```
ha='center',color='red')
ax.annotate('sst > 20 ℃', xy=(0.5, 0.5), xycoords='axes fraction', fontsize=10, ha='center')
ax.annotate('sst < 10 ℃', xy=(0.5, 0.15), xycoords='axes fraction', fontsize=10,
ha='center',color='red')

# 创建一个包含所有大陆的字典
continent_names = {'NA': 'North America', 'SA': 'South America', 'EU': 'Europe', 'AF': 'Africa',
'AS': 'Asia', 'OC': 'Oceania', 'AN': 'Antarctica'}

# 大陆的位置根据经纬度调整
coordinates = {'NA': (40, -100), 'SA': (-20, -60), 'EU': (50, 15), 'AF': (20, 20), 'AS': (40, 100), 'OC':
(-25, 130), 'AN': (-80, 0)}

# 建立一个循环来遍历大陆名插入文本框
#code 是大陆的代号，在字典中循环
for code, name in continent_names.items():
    #根据当前大陆的代号获取其中心的纬度
    lat_center, lon_center = coordinates[code]

    text_box_content = name
    #其中包含圆形边框、白色背景、蓝色边缘和透明度为 0.5。
    props = dict(boxstyle='round', facecolor='white', edgecolor='blue', alpha=0.5)
    ax.text(lon_center, lat_center, text_box_content, transform=ccrs.PlateCarree(), fontsize=8,
verticalalignment='center', ha='center', bbox=props)

plt.show()
```

这段代码的目的是通过可视化海表温度的全球分布，以及在地图上标注不同温度区域和大陆，帮助理解和分析海表温度的全球格局

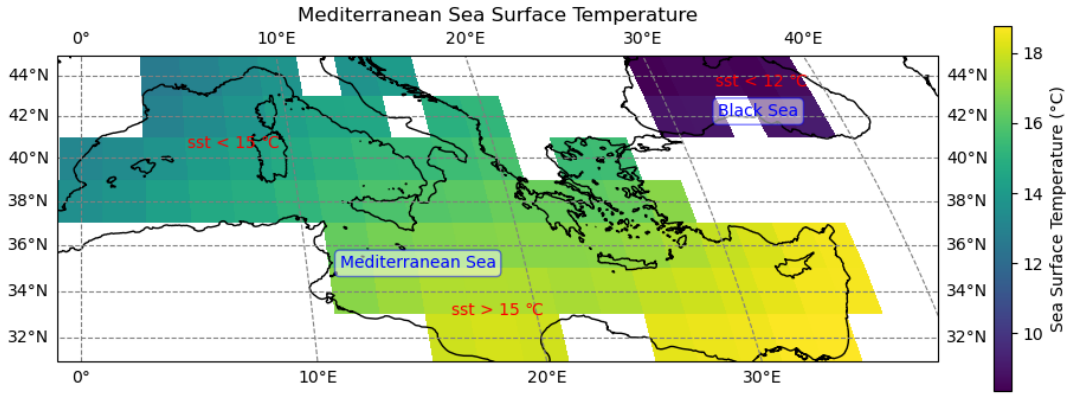


```
data = ds['sst']

# 提取地中海区域的数据,使用 sel 对 data 中 lat 进行索引的时候是空值?
mediterranean_data = data.where((data['lat'] > 30) & (data['lat'] < 45) & (data['lon'] > -20) & (data['lon'] < 40), drop=True)

# 使用 Lambert Conformal 投影
fig, ax = plt.subplots(figsize=(12, 8), dpi=100, subplot_kw={'projection': ccrs.Orthographic()})

# 画出地中海的表面温度，time 取第一个数据
c = ax.pcolormesh(mediterranean_data['lon'], mediterranean_data['lat'], mediterranean_data.isel(time=0),
                  cmap='viridis', transform=ccrs.PlateCarree())

# 建立 x,y 坐标以及范围，不需要添加 x,y 的范围
ax.set_xlabel('Longitude')
#ax.set_xticks([-20, 0, 20, 40])
ax.set_ylabel('Latitude')
#ax.set_yticks([30, 35, 40, 45])

# 画出标题
ax.set_title('Mediterranean Sea Surface Temperature')

# 设置网格线
ax.gridlines(draw_labels=True, linestyle='--', color='gray')

# 添加颜色条垂直右侧
cb = plt.colorbar(c, label='Sea Surface Temperature (°C)', orientation='vertical', shrink=0.5)

# 加入海岸线，加入国家线的时候需要下载地图，解压之后一直运用不了
ax.add_feature(cfeature.COASTLINE, edgecolor='black')

# 在地图不同温度区域加上注释
ax.annotate('sst < 15 ℃', xy=(0.2, 0.7), xycoords='axes fraction', fontsize=10, ha='center',color='red')
ax.annotate('sst < 12 ℃', xy=(0.8, 0.9), xycoords='axes fraction', fontsize=10, ha='center',color='red')
ax.annotate('sst > 15 ℃', xy=(0.5, 0.15), xycoords='axes fraction', fontsize=10, ha='center',color='red')

#加上文本框标注出黑海与地中海
props = dict(boxstyle='round', facecolor='white', edgecolor='blue', alpha=0.5)
ax.text(35, 42, 'Black Sea', transform=ccrs.PlateCarree(), fontsize=10, ha='center', color='blue', bbox=props)
```

ax.text(15, 35, 'Mediterranean Sea', transform=ccrs.PlateCarree(), fontsize=10, ha='center', color='blue', bbox=props)

plt.show()

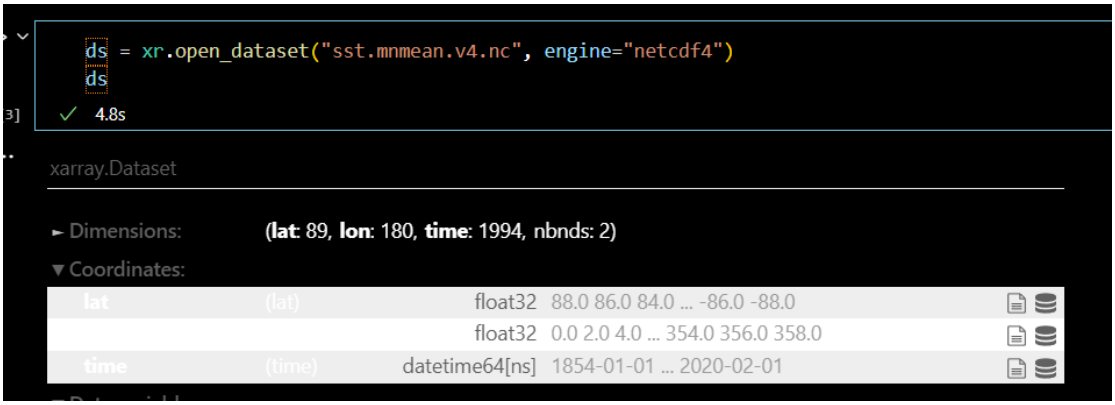
这段代码通过地图 Orthographic()投影可视化方式，直观地展示了地中海区域的海表温度分布



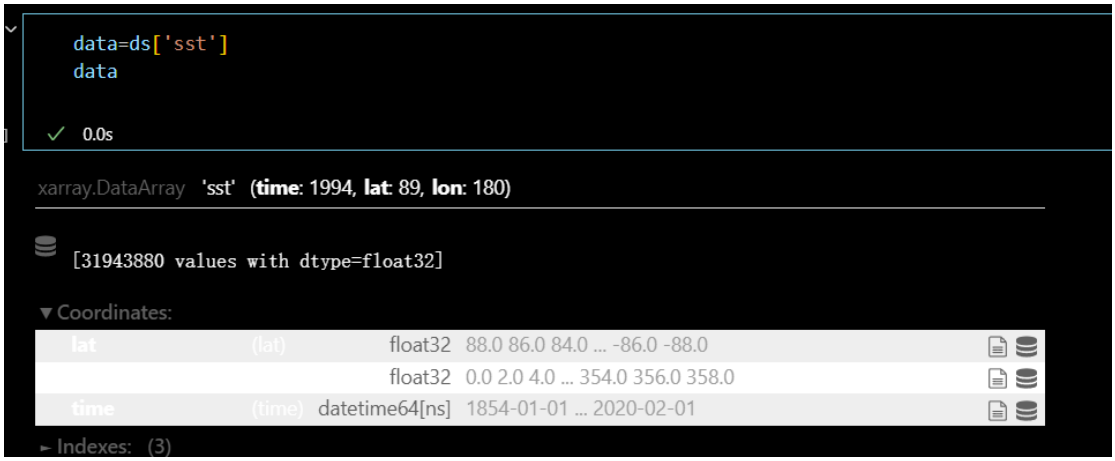
ERROR：

使用 data.sel 去索引的时候 lat 全是空值

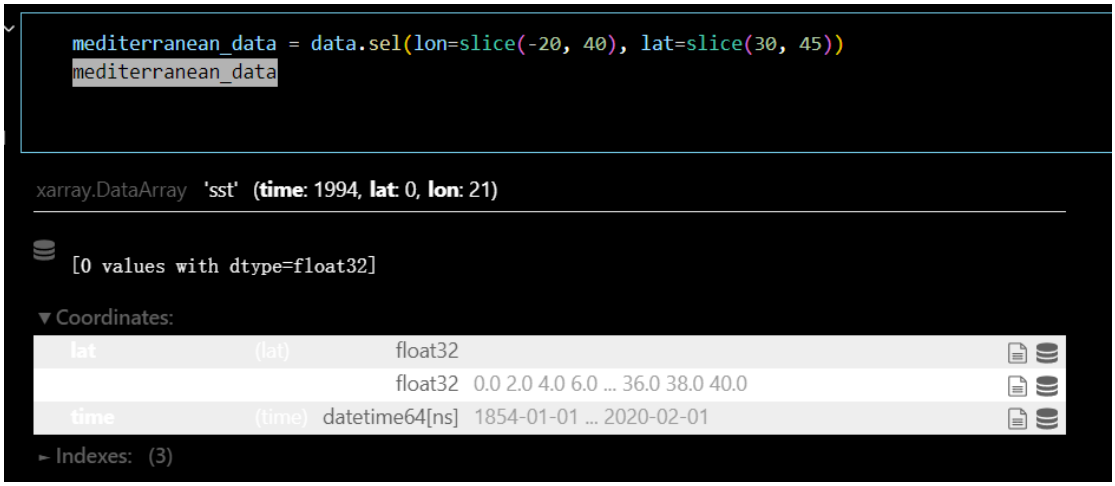
ds



data



mediterranean_data



lat 就变成空的了

```
mediterranean_data = data.where((data['lat'] > 30) & (data['lat'] < 45)
& (data['lon'] > -20) & (data['lon'] < 40), drop=True)
```

当我使用 where 的时候发现数据没错