

PS4_12232253_殷玉领

In [427]:

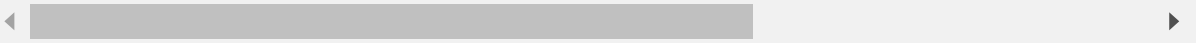
```
1 # Import modules
2 import glob
3 import numpy as np
4 import pandas as pd
5 import xarray as xr
6 import matplotlib as mpl
7 from matplotlib import pyplot as plt
8 from datetime import datetime
9 import cartopy.crs as ccrs
10 %matplotlib inline
11 import cmeps
12 from scipy import stats
13 import cartopy.feature as cfeature
14 import cartopy.io.shapereader as shpreader
15
16 # 导入Cartopy专门提供的经纬度的Formatter
17 from cartopy.mpl.ticker import LongitudeFormatter, LatitudeFormatter
18
19 #隐藏警告
20 import warnings
21 warnings.filterwarnings('ignore')
```

In [428]:

```
1 Eqs = pd.read_csv(r'D:\APP4\Anaconda\YYL_HW\HW4\usgs_earthquakes.csv')
2 Eqs.longitude = Eqs.longitude+180
3 Eqs.head()
```

Out[428]:

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net
0	2014-01-31 23:53:37.000	60.252000	27.2919	90.20	1.10	ml	NaN	NaN	NaN	0.2900	ak
1	2014-01-31 23:48:35.452	37.070300	64.8691	0.00	1.33	ml	4.0	171.43	0.34200	0.0247	nn
2	2014-01-31 23:47:24.000	64.671700	30.7472	7.10	1.30	ml	NaN	NaN	NaN	1.0000	ak
3	2014-01-31 23:30:54.000	63.188700	31.0425	96.50	0.80	ml	NaN	NaN	NaN	1.0700	ak
4	2014-01-31 23:30:52.210	32.616833	64.3075	10.59	1.34	ml	6.0	285.00	0.04321	0.2000	ci



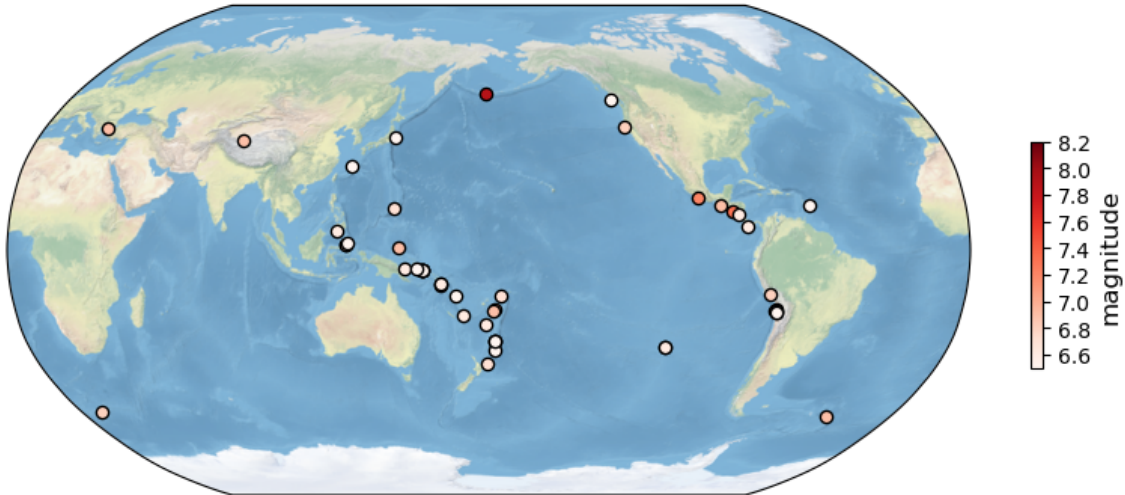
In [430]:

```

1 #取震级最大的前50个地震的数据
2 Eqs_6_5 = Eqs.sort_values(['mag'], ascending = False)[0:50]
3
4 #定义绘图函数
5 def main(Eqs_6_5):
6
7     #创建画布
8     fig = plt.figure(figsize=(8, 6), dpi=100)
9
10    #在画布上添加axes, [0,0,0.8,0.9]依次为左底宽高, 即划定axes在画布上的区域
11    ax=fig.add_axes([0,0,0.8,0.9], projection=ccrs.Robinson(central_longitude=180))
12
13    # make the map global rather than have it zoom in to
14    # the extents of any plotted data
15    ax.set_global()
16    ax.stock_img()
17
18    #在地图上绘制50个地震发生的位置, 填充的颜色越深震级越大
19    for i in range(len(Eqs_6_5)):
20        color = Eqs_6_5.iloc[i].mag
21        cmap = plt.cm.get_cmap('Reds')          #导入色条
22        x_value = Eqs_6_5.iloc[i].longitude+180
23        y_value = Eqs_6_5.iloc[i].latitude
24        ax.scatter(x=x_value, y=y_value,
25                  transform=ccrs.PlateCarree(),
26                  s=35,                          #scale, 尺寸
27                  c=color, ec='k',               #填充颜色和边缘颜色
28                  linewidth=1, vmin=Eqs_6_5.mag.min(),
29                  vmax=Eqs_6_5.mag.max(),
30                  cmap=cmap)
31
32    #根据数据范围得到正则化规则
33    norm=plt.Normalize(Eqs_6_5.mag.min(), Eqs_6_5.mag.max())
34
35    #设置颜色条
36    sm=plt.cm.ScalarMappable(norm=norm, cmap=cmap)
37    plt.title('Top 50 Earthquakes of 2014')
38
39    #颜色条放置的左、底、宽、高
40    cbar_ax=fig.add_axes([0.85, 0.32, 0.01, 0.25])
41
42    #设置颜色条的位置, 颜色, 上下限, 显示刻度, 竖直放置。
43    cbar=mpl.colorbar.ColorbarBase(cbar_ax, cmap='Reds', norm=mpl.colors.Normalize(6.5, 8.2), ticks
44                                   orientation='vertical')
45
46    #添加颜色条标题
47    cbar.ax.set_ylabel('magnitude', fontsize=12)
48    plt.show()
49
50    #调用绘图函数
51    if __name__ == '__main__':
52        main(Eqs_6_5)

```

Top 50 Earthquakes of 2014



In []:

```

1 #2_0 数据说明及预处理
2
3 #0. 数据说明:
4 #1. 所分析的数据为: 全球日最高温度数据集 (CPC .50x.50 全球每日温度)
5 #2. 只含陆地数据, 南极洲数据缺失!!
6 #3. 1991-2020, 格式为nc文件;
7 #4. 数据获取链接: https://psl.noaa.gov/data/gridded/data.cpc.globaltemp.html
8
9 #数据预处理:
10 #由全球日最高&最低温度数据集得到: 全球日温差数据集
11
12 #找到所有文件的路径
13 nc_path = glob.glob(r"Tem_Data2\*.nc")
14 new_nc = []
15 new_nc_time = []
16
17 #读取数据, 暂存
18 for i in range(len(nc_path)):
19     tmax_all=xr.open_dataset(nc_path[i])['tmax']
20     new_nc.append((tmax_all))
21
22
23 #用concat, 按时间合并数据
24 da=xr.concat(new_nc,dim='time')
25
26 #输出合并后的nc文件
27 da.to_netcdf('Tem_Data2\\tmax_1979_2020.nc')
28 #tem_diff = tmax['tmax']-tmin['tmin']
29 #tem_diff

```

In [431]:







```
1 #查看nc文件
2 temp = xr.open_dataset(r'CESM2_200001-201412.nc')
3 x = tmax_1979_2020.tmax
4
5 # Apply mean reduction from coordinates as performed in NCL's dim_rmvmean_n_Wrap(x,0)
6 # Apply this only to x.isel(time=0) because NCL plot plots only for time=0
7
8 tmax_new = x.isel(time=0)
9 tmax_new
```

Out[431]:

xarray.DataArray 'tmax' (lat: 360, lon: 720)

[259200 values with dtype=float32]

▼ Coordinates:

lat	(lat)	float32	89.75 89.25 88.75 ... -89.25 -89.75		
lon	(lon)	float32	0.25 0.75 1.25 ... 359.2 359.8		
time	()	datetime64[ns]	2020-01-01		

► Indexes: (2)

► Attributes: (13)

In [432]:

```

1 #2_1 全球尺度2020.01.01日最高温度等高线图
2
3 # 设置绘图区域.
4 lonmin, lonmax = -180, 180
5 latmin, latmax = -60, 90
6 extents = [lonmin, lonmax, latmin, latmax]
7
8 #北京、深圳、德里经纬度
9 Beijing_lon, Beijing_lat = 116, 40
10 Shenzhen_lon, Shenzhen_lat = 114, 22.5
11 Delhi_lon, Delhi_lat = 77.23, 28.61
12
13 # Generate figure (set its size (width, height) in inches)
14 fig = plt.figure(figsize=(8, 6),dpi=200)
15
16 # Generate axes using Cartopy projection
17 projection = ccrs.PlateCarree()
18 ax = plt.axes(projection=projection)
19
20 # Use global map and draw coastlines
21 ax.set_global()
22 #ax.stock_img() #带颜色的世界地图,控制底图是否填充
23 ax.coastlines(linewidth=0.5, resolution="110m")
24
25 # Add border lines over countries
26 ax.add_feature(cfeature.NaturalEarthFeature(category='cultural',
27                                             name='admin_0_countries',
28                                             scale='110m',
29                                             facecolor='none',
30                                             edgecolor='black',
31                                             linewidth=0.2,
32                                             zorder=2))
33
34 # 导入cmap色条
35 newcmp = cmeps.BlRe
36 newcmp.colors[len(newcmp.colors)
37               //2] = [1, 1, 1] # Set middle value to white to match NCL
38
39 # 绘制填充颜色的等高线图（等温线）
40 p = tmax_new.plot.contourf(
41     ax=ax,
42     vmin=-1,
43     vmax=50,
44     levels=list(np.linspace(-55, 45, 11)),
45     cmap=newcmp,
46     add_colorbar=False,
47     transform=projection,
48     add_labels=False)
49
50 # 添加色条
51 cbar = plt.colorbar(p,
52                     #orientation='horizontal',
53                     shrink=0.4, #缩放系数
54                     extendrect=True,
55                     extendfrac='auto',
56                     pad=0.03, #间距
57                     aspect=20, #条宽
58                     drawedges=False)
59 cbar.ax.tick_params(labelsize=8) #刻度字号

```

```

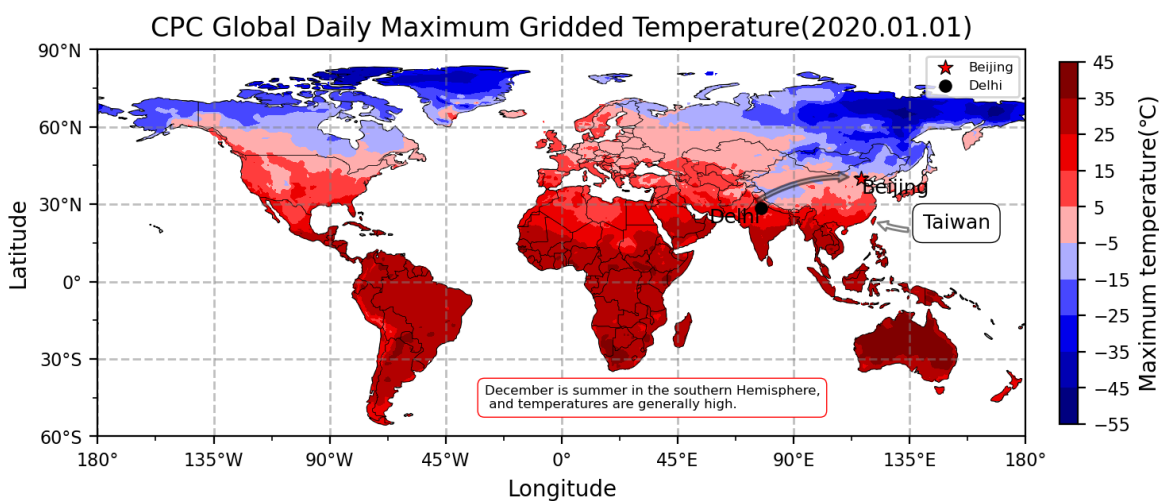
60 cbar.set_ticks(list(np.linspace(-55, 45, 11))) #自定义刻度及间距
61 cbar.ax.set_ylabel('Maximum temperature(°C)', fontsize=10) #为色条添加标题
62
63 # 设置x大刻度和小刻度
64 ax.tick_params(labelsize=8)
65 tick_proj = ccrs.PlateCarree()
66 ax.set_xticks(np.arange(-180, 180 + 45, 45), crs=tick_proj)
67 ax.set_xticks(np.arange(-180, 180 + 22.5, 22.5), minor=True, crs=tick_proj)
68 ax.set_xlabel('Longitude', fontsize=10)
69
70 # 设置y大刻度和小刻度
71 ax.tick_params(labelsize=8)
72 ax.set_yticks(np.arange(-60, 90 + 30, 30), crs=tick_proj)
73 ax.set_yticks(np.arange(-60, 90 + 15, 15), minor=True, crs=tick_proj)
74 ax.set_ylabel('Latitude', fontsize=10)
75
76 # 利用Formatter格式化刻度标签 (可以解决0度经线值不连续的问题)
77 ax.xaxis.set_major_formatter(LongitudeFormatter())
78 ax.yaxis.set_major_formatter(LatitudeFormatter())
79
80 # 在最后调用set_extent, 防止刻度拓宽显示范围.
81 if extents is None:
82     ax.set_global()
83 else:
84     ax.set_extent(extents, crs=proj)
85
86 # 设置标题
87 ax.set_title(r'CPC Global Daily Maximum Gridded Temperature(2020.01.01)', fontsize=12)
88
89 # 设置网格线
90 gl = ax.gridlines(crs=ccrs.PlateCarree(),
91                  xlocs=np.arange(-180, 180 + 45, 45),
92                  ylocs=np.arange(-60, 90 + 30, 30),
93                  #draw_labels=True,
94                  linewidth=1,
95                  color='gray',
96                  alpha=0.5,
97                  linestyle='--')
98
99 # 绘制北京的位置, 显示legend
100 ax.scatter(Beijing_lon, Beijing_lat,
101            s=50,
102            c='red',
103            marker='*',
104            label='Beijing',
105            ec='k',
106            lw=0.5,
107            zorder=3)
108 ax.scatter(Delhi_lon, Delhi_lat,
109            s=30,
110            c='k',
111            marker='o',
112            label='Delhi',
113            ec='k',
114            lw=0.5,
115            zorder=3)
116 ax.legend(loc='upper right', fontsize=6)
117
118 # 设置注释
119 transform = ccrs.PlateCarree().as_mpl_transform(ax) #坐标系转换
120 ax.annotate('Beijing', xy=(116.4, 39.9),

```

```

121         fontsize=9,
122         xycoords=transform,
123         ha='left',
124         va='top')
125 ax.annotate('Delhi', xy=(113, 40.5), xytext=(77.23, 28.61), fontsize=9,
126             #设置箭头, 德里到北京
127             arrowprops=dict(facecolor='gray',
128                             arrowstyle="simple",
129                             connectionstyle="arc3,rad=-0.2",
130                             alpha=0.5),
131             xycoords=transform,
132             ha='right', va='top')
133 ax.annotate('Taiwan', xy=(121, 23.5), xytext=(140, 20.5), fontsize=9,
134            arrowprops=dict(dict(facecolor='white',
135                                arrowstyle="simple",
136                                connectionstyle="arc3,rad=-0.2",
137                                alpha=0.5)),
138            bbox=dict(boxstyle='round,pad=0.5', fc='White', lw=0.5))
139 # 设置文本框
140 ax.text(-30, -50, 'December is summer in the southern Hemisphere,\n and temperatures are genera
141         verticalalignment = 'bottom',
142         horizontalalignment = 'left',
143         transform = transform,
144         color = 'k', fontsize = 6,
145         bbox=dict(boxstyle='round,pad=0.5', fc='White', lw=0.5, ec='r'))
146
147 # 显示图片
148 fig.tight_layout()
149 plt.show()
150
151 # Show the plot
152 plt.show()
153
154 #保存图片
155 fig.savefig('tmax.png', dpi=200, bbox_inches='tight')
156 plt.close(fig)

```



In [435]:

```

1 #2_2 北京附近2020.01.01日最高温度等高线图
2
3 #北京、郑州经纬度
4 Beijing_lon, Beijing_lat = 116, 40
5 Zhengzhou_lon, Zhengzhou_lat = 113.4, 33.5
6
7 # Generate figure (set its size (width, height) in inches)
8 fig = plt.figure(figsize=(8, 6), dpi=100)
9
10 # Generate axes using Cartopy projection
11 projection = ccrs.PlateCarree()
12 ax = plt.axes(projection=projection)
13
14 # Use global map and draw coastlines
15 ax.set_global()
16 #ax.stock_img() #带颜色的世界地图,控制底图颜色
17 ax.coastlines(linewidth=0.5, resolution="110m")
18
19 # Add border lines over countries
20 ax.add_feature(cfeature.NaturalEarthFeature(category='cultural',
21                                             name='admin_0_countries',
22                                             scale='110m',
23                                             facecolor='none',
24                                             edgecolor='black',
25                                             linewidth=0.5,
26                                             zorder=2))
27
28 # 添加河流图层
29 rivers_10m = cfeature.NaturalEarthFeature('physical', 'rivers_lake_centerlines', '10m')
30 ax.add_feature(cfeature.LAKES,
31               edgecolor='blue',
32               facecolor='blue',
33               zorder=2)
34 ax.add_feature(rivers_10m,
35               facecolor='None',
36               edgecolor='blue',
37               linewidth=1,
38               zorder=2)
39
40 # 导入cmap色条
41 newcmp = cmaps.BlRe
42 newcmp.colors[len(newcmp.colors)
43               //2] = [1, 1, 1] # Set middle value to white to match NCL
44
45 # 绘制填充颜色的等高线图（等温线）
46 p = tmax_new.plot.contourf(
47     ax=ax,
48     vmin=-20,
49     vmax=20,
50     levels=list(np.linspace(-20, 20, 11)),
51     cmap=newcmp,
52     add_colorbar=False,
53     transform=projection,
54     add_labels=False)
55
56 # 添加色条
57 cbar = plt.colorbar(p,
58                     #orientation='horizontal',
59                     shrink=0.8,
60                     #缩放系数

```

```

60         extendrect=True,
61         extendfrac='auto',
62         pad=0.04,                                #间距
63         aspect=20,                                #条宽
64         drawedges=False)
65 cbar.ax.tick_params(labelsize=10)                 #刻度字号
66 cbar.set_ticks(list(np.linspace(-20, 20, 11)))    #自定义刻度及间距
67 cbar.ax.set_ylabel('Maximum temperature(°C)', fontsize=12) #为色条添加标题
68
69 # 设置x大刻度和小刻度
70 ax.tick_params(labelsize=10)
71 tick_proj = ccrs.PlateCarree()
72 ax.set_xticks(np.arange(-180, 180 + 5, 5), crs=tick_proj)
73 ax.set_xticks(np.arange(-180, 180 + 2.5, 2.5), minor=True, crs=tick_proj)
74 ax.set_xlabel('Longitude', fontsize=12)
75
76 # 设置y大刻度和小刻度
77 ax.tick_params(labelsize=10)
78 ax.set_yticks(np.arange(-60, 90 + 5, 5), crs=tick_proj)
79 ax.set_yticks(np.arange(-60, 90 + 2.5, 2.5), minor=True, crs=tick_proj)
80 ax.set_ylabel('Latitude', fontsize=12)
81
82 # 利用Formatter格式化刻度标签 （可以解决0度经线值不连续的问题）
83 ax.xaxis.set_major_formatter(LongitudeFormatter())
84 ax.yaxis.set_major_formatter(LatitudeFormatter())
85
86 # 在最后调用set_extent, 防止刻度拓宽显示范围.
87 #if extents is None:
88 #    ax.set_global()
89 #else:
90 #    ax.set_extent(extents, crs=proj)
91
92 # 设置标题
93 ax.set_title(r'CPC Beijing Daily Maximum Temperature(2020.01.01)', fontsize=14)
94
95 # 设置网格线
96 gl = ax.gridlines(crs=ccrs.PlateCarree(),
97                  xlocs=np.arange(-180, 180 + 5, 5),
98                  ylocs=np.arange(-60, 90 + 5, 5),
99                  #draw_labels=True,
100                  linewidth=1.6,
101                  color='gray',
102                  alpha=0.5,
103                  linestyle='--')
104
105 # 绘制北京的位置, 显示legend
106 ax.scatter(Beijing_lon, Beijing_lat,
107            s=50,
108            c='red',
109            marker='*',
110            label='Beijing',
111            ec='k',
112            lw=0.5,
113            zorder=3)
114 ax.scatter(Zhengzhou_lon, Zhengzhou_lat,
115            s=20,
116            c='k',
117            marker='o',
118            label='Zhengzhou',
119            ec='k',
120            lw=0.5,

```

```
121         zorder=3)
122 ax.legend(loc='upper right', fontsize=8)
123
124 # 设置绘图区域.
125 extents = [Beijing_lon-8, Beijing_lon+6, Beijing_lat-8, Beijing_lat+6]
126 ax.set_extent(extents, crs=proj)
127
128 # 设置注释
129 transform = ccrs.PlateCarree().as_mpl_transform(ax) #坐标系转换
130
131 ax.annotate('Beijing', xy=(116.4, 39.9), fontsize=12, xycoords=transform,
132            ha='left', va='top')
133
134 ax.annotate('Zhengzhou', xy=(115.9, 39.9), xytext=(113.4, 33.4), fontsize=12,
135            #设置箭头, 郑州到北京
136            arrowprops=dict(facecolor='gray',
137                            arrowstyle="simple",
138                            connectionstyle="arc3,rad=-0.2",
139                            alpha=0.5),
140            xycoords=transform,
141            ha='center', va='top')
142
143 # 设置文本框
144 ax.text(115.5, 32.5, 'The higher the latitude, the\n lower the maximum temperature.\n But close
145         verticalalignment = 'bottom',
146         horizontalalignment = 'left',
147         transform = transform,
148         color = 'black', fontsize = 9,
149         bbox=dict(boxstyle='round,pad=0.5', fc='White', lw=0.5, ec='k'))
150
151 # 显示图片
152 fig.tight_layout()
153 plt.show()
154
155 # Show the plot
156 plt.show()
157
158 #保存图片
159 fig.savefig('tmax_region.png', dpi=200, bbox_inches='tight')
160 plt.close(fig)
```

