Assignment 04

Name: Yan Qinlin
SID: 12231096
Due: 2022/11/29

In [1]:

```
import random
from math import *
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import datetime
import netCDF4
import xarray as xr
%matplotlib inline
import matplotlib.ticker as mticker
import cartopy.crs as ccrs
import cartopy.feature as cfeature
from cartopy.mpl.gridliner import LONGITUDE_FORMATTER, LATITUDE_FORMATTER
from cartopy.mpl.ticker import LongitudeFormatter,LatitudeFormatter
import cartopy.io.shapereader as shapereader
plt.rcParams['font.sans-serif'] = ['SimHei']
```

Ref: All the programming details were referred to the handout of course ESE5023 by professor Zhu (https://zhu-group.github.io/ese5023 (https://zhu-group.github.io/ese5023)).

1. Global Earthquakes

Data reference: The relief data is downloaded from Natural Earth (https://www.naturalearthdata.com/downloads/ (https://www.naturalearthdata.com/downloads/)).

In [112]:

```
df1 = pd.read_csv("usgs_earthquakes.csv")
df1.head()
```

Out[112]:

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms
0	2014-01-31 23:53:37.000	60.252000	-152.7081	90.20	1.10	ml	NaN	NaN	NaN	0.2900
1	2014-01-31 23:48:35.452	37.070300	-115.1309	0.00	1.33	ml	4.0	171.43	0.34200	0.0247
2	2014-01-31 23:47:24.000	64.671700	-149.2528	7.10	1.30	ml	NaN	NaN	NaN	1.0000
3	2014-01-31 23:30:54.000	63.188700	-148.9575	96.50	0.80	ml	NaN	NaN	NaN	1.0700
4	2014-01-31 23:30:52.210	32.616833	-115.6925	10.59	1.34	ml	6.0	285.00	0.04321	0.2000

In [113]:

```
#读入数据并查看
df1 = pd.read_csv("usgs_earthquakes.csv")
#删除mag列空值所在行
df1.dropna(subset=['mag'],inplace=True)
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 120065 entries, 0 to 120107
Data columns (total 15 columns):
 #   Column     Non-Null Count   Dtype  
---  ------     --------------   -----  
 0   time       120065 non-null  object 
 1   latitude   120065 non-null  float64
 2   longitude  120065 non-null  float64
 3   depth      120065 non-null  float64
 4   mag        120065 non-null  float64
 5   magType    120065 non-null  object 
 6   nst        59653 non-null   float64
 7   gap        94893 non-null   float64
 8   dmin       85641 non-null   float64
 9   rms        119676 non-null  float64
 10  net        120065 non-null  object 
 11  id         120065 non-null  object 
 12  updated    120065 non-null  object 
 13  place      120065 non-null  object 
 14  type       120065 non-null  object 
dtypes: float64(8), object(7)
memory usage: 14.7+ MB
```

In [114]:

```
#按照mag从大到小排序
df_temp = df1.loc[df1['time'].str.contains('2014')].sort_values("mag", ascending=False)

##选出时空信息和mag
df2 = df_temp[['time','latitude','longitude','mag']]

#重新排序，并只要前50个,并打印检查
df2.reset_index(drop=True, inplace=True)
df2 = df2.loc[0:50]
df2.head(50)
```

Out[114]:

	time	latitude	longitude	mag
0	2014-04-01 23:46:47.260	-19.6097	-70.7691	8.2
1	2014-06-23 20:53:09.700	51.8486	178.7352	7.9
2	2014-04-03 02:43:13.110	-20.5709	-70.4931	7.7
3	2014-04-12 20:14:39.300	-11.2701	162.1481	7.6
4	2014-04-19 13:28:00.810	-6.7547	155.0241	7.5
5	2014-04-13 12:36:19.230	-11.4633	162.0511	7.4
6	2014-10-14 03:51:34.460	12.5262	-88.1225	7.3
7	2014-04-18 14:27:24.920	17.3970	-100.9723	7.2
8	2014-04-11 07:07:23.130	-6.5858	155.0485	7.1
9	2014-11-15 02:31:41.720	1.8929	126.5217	7.1
10	2014-11-01 18:57:22.380	-19.6903	-177.7587	7.1
11	2014-10-09 02:14:31.440	-32.1082	-110.8112	7.0
12	2014-06-29 07:52:55.170	-55.4703	-28.3669	6.9
13	2014-08-03 00:22:03.680	0.8295	146.1688	6.9
14	2014-06-23 19:19:15.940	-29.9772	-177.7247	6.9
15	2014-02-12 09:19:49.060	35.9053	82.5864	6.9
16	2014-07-21 14:54:41.000	-19.8015	-178.4001	6.9
17	2014-04-01 23:57:58.790	-19.8927	-70.9455	6.9
18	2014-05-24 09:25:02.440	40.2893	25.3889	6.9
19	2014-07-07 11:23:54.780	14.7240	-92.4614	6.9
20	2014-03-10 05:18:13.400	40.8287	-125.1338	6.8
21	2014-04-15 03:57:01.370	-53.4967	8.7220	6.8
22	2014-11-26 14:33:43.640	1.9604	126.5751	6.8
23	2014-08-24 23:21:45.520	-14.5980	-73.5714	6.8
24	2014-03-16 21:16:29.600	-19.9807	-70.7022	6.7
25	2014-09-17 06:14:45.410	13.7641	144.4294	6.7
26	2014-11-16 22:33:20.450	-37.6478	179.6621	6.7
27	2014-06-23 20:06:20.710	-29.9414	-177.6073	6.7
28	2014-06-29 17:15:09.340	-14.9831	-175.5096	6.7
29	2014-05-04 09:15:52.880	-24.6108	179.0856	6.6
30	2014-04-13 13:24:59.710	-11.1284	162.0520	6.6
31	2014-12-08 08:54:52.520	7.9401	-82.6865	6.6
32	2014-05-01 06:36:35.550	-21.4542	170.3546	6.6
33	2014-12-02 05:11:31.000	6.1572	123.1261	6.6
34	2014-10-09 02:32:05.140	-32.0953	-110.8647	6.6
35	2014-11-07 03:33:55.280	-5.9873	148.2315	6.6
36	2014-12-07 01:22:02.180	-6.5108	154.4603	6.6

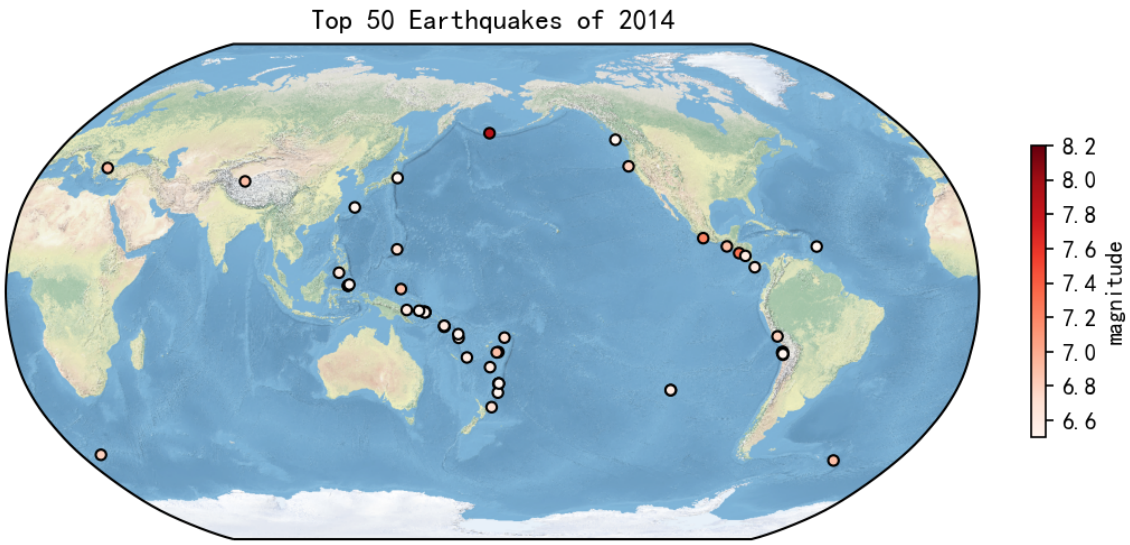
	time	latitude	longitude	mag
37	2014-04-11 20:29:12.970	11.6420	-85.8779	6.6
38	2014-04-19 01:04:03.820	-6.6558	155.0869	6.6
39	2014-02-07 08:40:13.550	-15.0691	167.3721	6.5
40	2014-11-21 10:10:19.630	2.2999	127.0562	6.5
41	2014-02-02 09:26:37.820	-32.9076	-177.8806	6.5
42	2014-03-02 20:11:23.430	27.4312	127.3674	6.5
43	2014-04-03 01:58:30.530	-20.3113	-70.5756	6.5
44	2014-04-11 08:16:45.660	-6.7878	154.9502	6.5
45	2014-06-23 19:21:45.990	-29.9379	-177.5159	6.5
46	2014-07-11 19:22:00.820	37.0052	142.4525	6.5
47	2014-02-18 09:27:13.120	14.6682	-58.9272	6.5
48	2014-04-24 03:10:10.150	49.6388	-127.7316	6.5
49	2014-07-04 15:00:27.860	-6.2304	152.8075	6.5

In [23]:

```
import matplotlib.ticker as ticker

# Create and define the size of a figure object
plt.figure(figsize=(8,8), dpi=150)


# Create an axes with Robinson projection style, 调整中心经度为180
ax = plt.axes(projection=ccrs.Robinson(central_longitude=180))

#画浮雕图
ax.imshow(plt.imread("NE1_50M_SR_W.tif"),origin='upper',transform=ccrs.PlateCarree(),extent=[-180,180,-90,90])

#选择colorbar的style
cm = plt.cm.get_cmap('Reds')

#画散点——2014年最严重（按mag）的50次地震
sc = ax.scatter(df2['longitude'],df2['latitude'], s=20,c=df2['mag'], edgecolors=['black'],cmap=cm,transform=ccrs.PlateCarree())

#定义colorbar的刻度值

#画colorbar,与老师给的图一样
c = plt.colorbar(sc,label='magnitude',fraction=0.014,pad=0.05)
tick_locator = ticker.MaxNLocator(nbins=9)
c.locator = tick_locator
c.update_ticks()

ax.set_title("Top 50 Earthquakes of 2014")
```

Out[23]:

```
Text(0.5, 1.0, 'Top 50 Earthquakes of 2014')
```



Ref: The usage of relief file and the overlap of both the relief and the scatter were referred to the blog of *野生的气象小流星*(https://blog.csdn.net/weixin_42372313/article/details/119885922 (https://blog.csdn.net/weixin_42372313/article/details/119885922)). The usage of ticks resetting of the colorbar was referred to the blogs of *Mr.Jcak* (https://blog.csdn.net/weixin_38314865/article/details/109499030?spm=1001.2101.3001.6650.14&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-14-109499030-blog-117259630.pc_relevant_3mothn_strategy_recovery&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-14-109499030-blog-117259630.pc_relevant_3mothn_strategy_recovery&utm_relevant_index=14 (https://blog.csdn.net/weixin_38314865/article/details/109499030?spm=1001.2101.3001.6650.14&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-14-109499030-blog-117259630.pc_relevant_3mothn_strategy_recovery&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-14-109499030-blog-117259630.pc_relevant_3mothn_strategy_recovery&utm_relevant_index=14)).

2. Explore a netCDF dataset

Data Ref:
Xie, P., and P.A. Arkin, 1997: Global precipitation: A 17-year monthly analysis based on gauge observations, satellite estimates, and numerical model outputs. Bull. Amer. Meteor. Soc., 78, 2539 - 2558.

2.1

In [3]:

```
# Open a netCDF4 file
ds3 = xr.open_dataset("precip.mon.mean.nc", engine="netcdf4")

# Show dataset
ds3

#按月度统计，
precip_clim = ds3.groupby('time.month').mean()
```

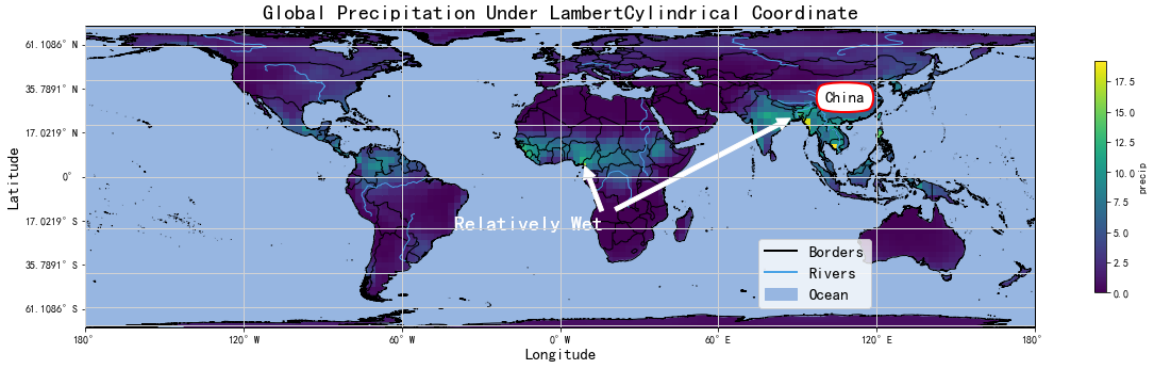
In [82]:

```
import matplotlib as mp
import matplotlib.patches as mpatches
from matplotlib.patches import Patch
from matplotlib.lines import Line2D
#创建画布
fig = plt.figure(figsize=(20,10))

#创建投影，选择cartopy的platecarree投影
proj   = ccrs.LambertCylindrical()
ax = fig.subplots(1, 1, subplot_kw={'projection': proj})

#画数据
precip_clim.precip.sel(month=8).plot(ax=ax, transform=ccrs.PlateCarree(),
         cbar_kwargs={'shrink': 0.4})


# Add lat/lon gridlines, draw gridlines
gl = ax.gridlines(crs=ccrs.PlateCarree(), draw_labels=False,linewidth=1,color='lightgrey',zorde
r=5)

#设置label格式为经纬专用
ax.xaxis.set_major_formatter(LongitudeFormatter(zero_direction_label=True) )
ax.yaxis.set_major_formatter(LatitudeFormatter())

#画tick
ax.set_xticks(np.linspace(-180,180,7), crs=proj)
ax.set_yticks(np.linspace(-50,50,7), crs=proj)

#写label
ax.set_xlabel("Longitude",fontsize=15)
ax.set_ylabel("Latitude",fontsize=15)

# Mask ocean data by adding ocean feature and changing its zorder
ax.add_feature(cfeature.OCEAN, zorder=1)
ax.add_feature(cfeature.BORDERS)
ax.add_feature(cfeature.RIVERS, edgecolor='#4aa3e6', zorder=2)
ax.coastlines(resolution='10m')




#legend
legend_elements = [Line2D([0], [0], color='black', lw=2, label='Borders'),
                   Line2D([0], [0], color='#4aa3e6', lw=2, label='Rivers'),
                   Patch(facecolor='#97b6e1', edgecolor='#97b6e1',label='Ocean')]

# Create the figure
ax.legend(handles=legend_elements, fontsize=15,
    loc='lower left', bbox_to_anchor=(0.7, 0.04), fancybox=True)

#annotation & text box
t = ax.text(108, 30, "China", ha="center", va="center", rotation=0, size=15,
    bbox=dict(boxstyle="round4,pad=0.3", fc="white", ec="r", lw=2))
bb = t.get_bbox_patch()
bb.set_boxstyle("round4", pad=0.6)

#annotation
ax.annotate('Relatively Wet', xy=(91, 24),color='white',fontsize=20,
            xytext=(-40, -20),
```

```
            arrowprops=dict(facecolor='white',edgecolor='white',shrink=0.05))
ax.annotate('Relatively Wet', xy=(9, 5),color='white',fontsize=20,
            xytext=(-40, -20),
            arrowprops=dict(facecolor='white',edgecolor='white',shrink=0.05))
#title
plt.title("Global Precipitation Under LambertCylindrical Coordinate",fontsize=20)
```

Out[82]:

```
Text(0.5, 1.0, 'Global Precipitation Under LambertCylindrical Coordinate')
```



2.2

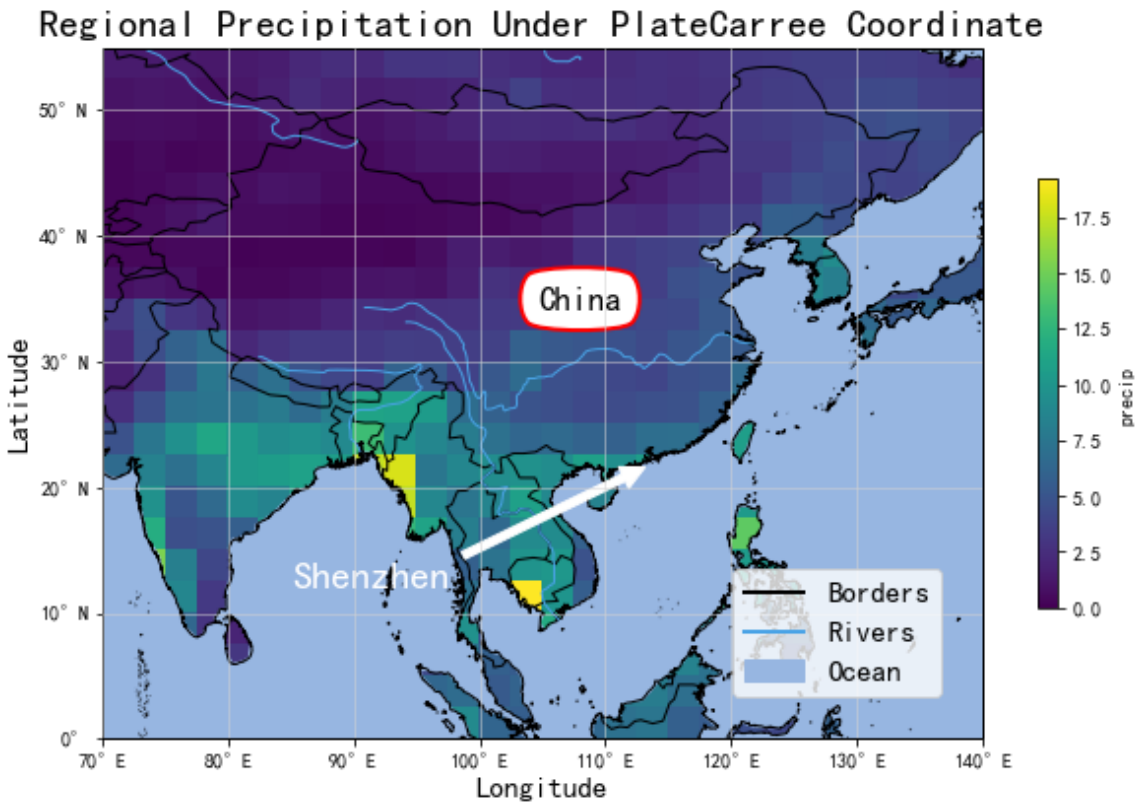
In [70]:

```
import matplotlib as mp
import matplotlib.patches as mpatches
from matplotlib.patches import Patch
from matplotlib.lines import Line2D
#创建画布
fig = plt.figure(figsize=(10,10))

#创建投影，选择cartopy的platecarree投影
proj   = ccrs.PlateCarree()
ax = fig.subplots(1, 1, subplot_kw={'projection': proj})

#画数据
precip_clim.precip.sel(month=8).plot(ax=ax, transform=ccrs.PlateCarree(),
         cbar_kwargs={'shrink': 0.4})

#local 局部画
ax.set_extent([70,140,0,55], crs=proj)

# Add lat/lon gridlines, draw gridlines
gl = ax.gridlines(crs=ccrs.PlateCarree(), draw_labels=False,linewidth=0.5,color='lightgrey',zor
der=5)

#设置label格式为经纬专用
ax.xaxis.set_major_formatter(LongitudeFormatter(zero_direction_label=True) )
ax.yaxis.set_major_formatter(LatitudeFormatter())

#画tick
ax.set_xticks(np.linspace(70,140,8), crs=proj)
ax.set_yticks(np.linspace(0,50,6), crs=proj)

#写label
ax.set_xlabel("Longitude",fontsize=15)
ax.set_ylabel("Latitude",fontsize=15)

# Mask ocean data by adding ocean feature and changing its zorder
ax.add_feature(cfeature.OCEAN, zorder=1)
ax.add_feature(cfeature.BORDERS)
ax.add_feature(cfeature.RIVERS, edgecolor='#4aa3e6', zorder=2)
ax.coastlines(resolution='10m')




#legend
legend_elements = [Line2D([0], [0], color='black', lw=2, label='Borders'),
                   Line2D([0], [0], color='#4aa3e6', lw=2, label='Rivers'),
                   Patch(facecolor='#97b6e1', edgecolor='#97b6e1',label='Ocean')]

# Create the figure
ax.legend(handles=legend_elements, fontsize=15,
    loc='lower left', bbox_to_anchor=(0.7, 0.04), fancybox=True)

#annotation & text box
t = ax.text(
    108, 35, "China", ha="center", va="center", rotation=0, size=17,
    bbox=dict(boxstyle="round4,pad=0.3", fc="white", ec="r", lw=2))
bb = t.get_bbox_patch()
bb.set_boxstyle("round4", pad=0.6)
```

```
#annotation
ax.annotate('Shenzhen', xy=(114, 22),color='white',fontsize=20,
            xytext=(85, 12),
            arrowprops=dict(facecolor='white',edgecolor='white',shrink=0.05))
#title
plt.title("Regional Precipitation Under PlateCarree Coordinate",fontsize=20)
```

Out[70]:

```
Text(0.5, 1.0, 'Regional Precipitation Under PlateCarree Coordinate')
```



Ref: The usage of lat/lon format and ticks were referred to the handout on ModelWhale website(https://www.heywhale.com/mw/project/620d03ab7a7c9a0017c95995).The (https://www.heywhale.com/mw/project/620d03ab7a7c9a0017c95995).The) usage of user-defined legend was referred to the official wedsite of matplotlib (https://www.osgeo.cn/matplotlib/gallery/text_labels_and_annotations/custom_legends.html (https://www.osgeo.cn/matplotlib/gallery/text_labels_and_annotations/custom_legends.html)).