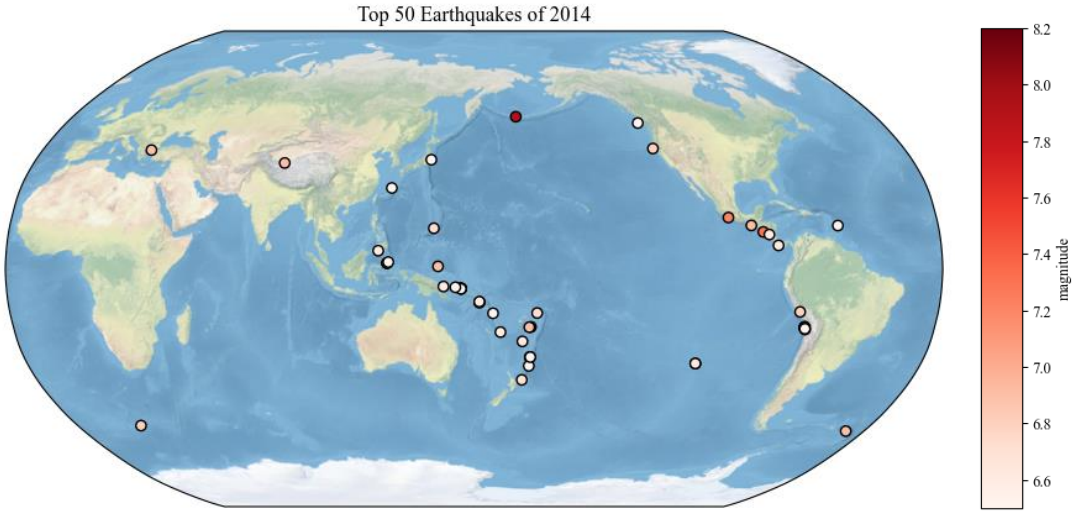
# PS4_1

```
import pandas as pd
eq = pd.read_csv('usgs_earthquakes.csv')
Top50 = eq.sort_values("mag", ascending=False).head(50)

import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import numpy as np
%matplotlib inline

plt.figure(figsize=(10,5), dpi=100)
proj = ccrs.Robinson(central_longitude=160)
ax = plt.axes(projection=proj)
ax.stock_img()

ax.scatter('longitude', 'latitude', c='mag', marker='o', cmap='Reds', alpha=1,
edgecolor='k', data=Top50, transform=ccrs.PlateCarree())

plt.title("Top 50 Earthquakes of 2014", fontsize=12)

import matplotlib as mpl
fig, ax = plt.subplots(figsize=(1,5))
fig.subplots_adjust(right=0.5, top=1)
cmap = 'Reds'
norm = mpl.colors.Normalize(vmin=6.5, vmax=8.2)
fig.colorbar(mpl.cm.ScalarMappable(norm=norm, cmap=cmap), cax=ax,
orientation='vertical', label='magnitude')
```

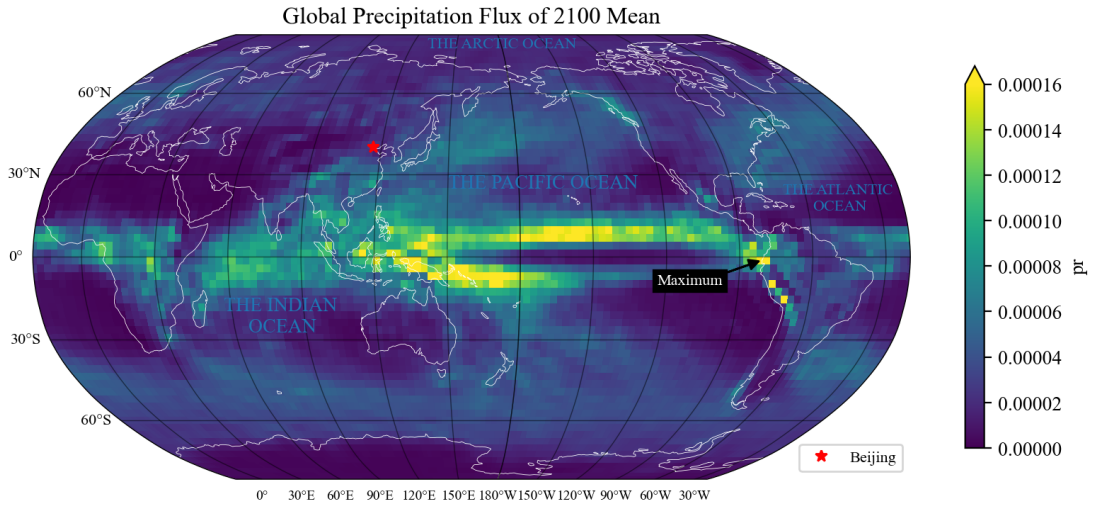


# PS4_2

## 2.1

```
import xarray as xr
import netCDF4

ds = xr.open_dataset("pr_Amon_CanESM5_ssp370-ssp126Lu_r1i1p2f1_gn_201501-
210012.nc", engine="netcdf4")

import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import matplotlib.ticker as mticker

plt.figure(figsize=(10,5), dpi=200)
proj = ccrs.Robinson(central_longitude=160)
ax = plt.axes(projection=proj)

pr = ds.pr.sel(time='2100').mean(dim='time')
pr.plot(ax=ax, transform=ccrs.PlateCarree(),
            vmin=0, vmax=0.00016, cbar_kwargs={'shrink': 0.7})

ax.add_feature(cfeature.NaturalEarthFeature(category='physical',
                                            name='land',
                                            scale='110m',
                                            facecolor='none',
                                            edgecolor='white',
                                            linewidth=0.3))

plt.rcParams['font.sans-serif'] = ['Times New Roman']
plt.rcParams['axes.unicode_minus'] = False

from cartopy.mpl.gridliner import LONGITUDE_FORMATTER,
LATITUDE_FORMATTER
gl = ax.gridlines(crs=ccrs.PlateCarree(), draw_labels=True, linewidth=0.5,
```

```
color='black', alpha=0.5)
gl.top_labels=False
gl.right_labels=False
gl.xformatter = LONGITUDE_FORMATTER
gl.yformatter = LATITUDE_FORMATTER
gl.xlocator = mticker.FixedLocator(np.arange(-180, 181, 30))
gl.ylocator = mticker.FixedLocator(np.arange(-90, 91, 30))
gl.xlabel_style={'size':7}
gl.ylabel_style={'size':8}
ax.spines['geo'].set_linewidth(0.5)

plt.title("Global Precipitation Flux of 2100 Mean", fontsize=12)

Beijing = dict(lon=116, lat=40)
ax.plot(Beijing['lon'], Beijing['lat'], 'r*', transform=ccrs.PlateCarree(), label='Beijing')
plt.legend(loc='lower right', fontsize=8)

plt.annotate('Maximum', xy=(-82, -1.5), xytext=(-110, -10),
transform=ccrs.PlateCarree(),
                    bbox=dict(boxstyle='square', fc='black', linewidth=0.1),
                    arrowprops=dict(facecolor='black', shrink=0.01, width=0.1,
headwidth=5, headlength=5),
                    fontsize=8, color='white', horizontalalignment='center')

plt.text(150, 25, 'THE PACIFIC OCEAN', transform=ccrs.PlateCarree(),
color='#1f77b4', fontsize=10)
plt.text(80, -27, 'THE INDIAN \nOCEAN', ha='center', transform=ccrs.PlateCarree(),
color='#1f77b4', fontsize=10)
plt.text(-47, 17, 'THE ATLANTIC \nOCEAN', ha='center',
transform=ccrs.PlateCarree(), color='#1f77b4', fontsize=8)
plt.text(180, 80, 'THE ARCTIC OCEAN', ha='center', transform=ccrs.PlateCarree(),
color='#1f77b4', fontsize=8)
```




## 2.2

```
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import matplotlib.ticker as mticker

plt.figure(figsize=(8,8), dpi=200)
proj = ccrs.LambertConformal(central_longitude=113.0,
                             central_latitude=22.0,
                             false_easting=0.0,
                             false_northing=0.0,
                             standard_parallels=(0, 44),
                             globe=None)
ax = plt.axes(projection=proj)

central_longitude=113.0
central_latitude=22.0
extent = [central_longitude-28, central_longitude+28, central_latitude-32,
central_latitude+32]
ax.set_extent(extent)

pr = ds.pr.sel(time='2100').mean(dim='time')
pr.plot(ax=ax, transform=ccrs.PlateCarree(),
```

```
                vmin=0, vmax=0.00016, cbar_kwargs={'shrink': 0.7})

ax.coastlines(resolution='10m', linewidth=0.3)

plt.rcParams['font.sans-serif'] = ['Times New Roman']
plt.rcParams['axes.unicode_minus'] = False

from cartopy.mpl.gridliner import LONGITUDE_FORMATTER,
LATITUDE_FORMATTER
gl = ax.gridlines(crs=ccrs.PlateCarree(), draw_labels=True, linewidth=0.2,
x_inline=False, y_inline=False, color='black', alpha=0.5)
gl.top_labels=False
gl.right_labels=False
gl.xformatter = LONGITUDE_FORMATTER
gl.yformatter = LATITUDE_FORMATTER
gl.xlocator = mticker.FixedLocator(np.arange(60, 161, 10))
gl.ylocator = mticker.FixedLocator(np.arange(-10, 61, 10))
gl.xlabel_style={'size':10}
gl.ylabel_style={'size':10}
ax.spines['geo'].set_linewidth(0.2)

plt.title("Precipitation Flux of 2100 Mean in East Asia and Southeast Asia",
fontsize=12)

Beijing = dict(lon=116, lat=40)
Shenzhen = dict(lon=114.06, lat=22.54)
ax.plot(Beijing['lon'], Beijing['lat'], 'r*', transform=ccrs.PlateCarree(), label='Beijing')
ax.plot(Shenzhen['lon'], Shenzhen['lat'], 'y.', transform=ccrs.PlateCarree(),
label='Shenzhen')
plt.legend(loc='upper right', fontsize=8)

plt.text(135, 15, 'THE PACIFIC \nOCEAN', ha='center',
transform=ccrs.PlateCarree(), color='b', fontsize=10, rotation=8)
plt.text(92, -7, 'THE INDIAN \nOCEAN', ha='center', transform=ccrs.PlateCarree(),
color='b', fontsize=10, rotation=-8)
```

```
plt.text(115, 11, 'SOUTH \nCHINA \nSEA', ha='center', transform=ccrs.PlateCarree(),
color='b', fontsize=8, rotation=0)
plt.text(125, 27, 'EAST \nCHINA \nSEA', ha='center', transform=ccrs.PlateCarree(),
color='b', fontsize=8, rotation=0)
plt.text(134, 38, 'SEA OF \nJAPAN', ha='center', transform=ccrs.PlateCarree(),
color='b', fontsize=8, rotation=8)
plt.text(112, -6, 'JAVA SEA', ha='center', transform=ccrs.PlateCarree(), color='b',
fontsize=8, rotation=-8)

plt.annotate('Asia', xy=(95, 42), xytext=(95, 42), transform=ccrs.PlateCarree(),
             bbox=dict(boxstyle='ellipse', fc='black', linewidth=0.1, alpha=0),
             arrowprops=dict(facecolor='black', shrink=0, width=0, headwidth=0,
headlength=0),
             fontsize=12, rotation=-7, color='white', horizontalalignment='center')
```