1. Global Earthquakes

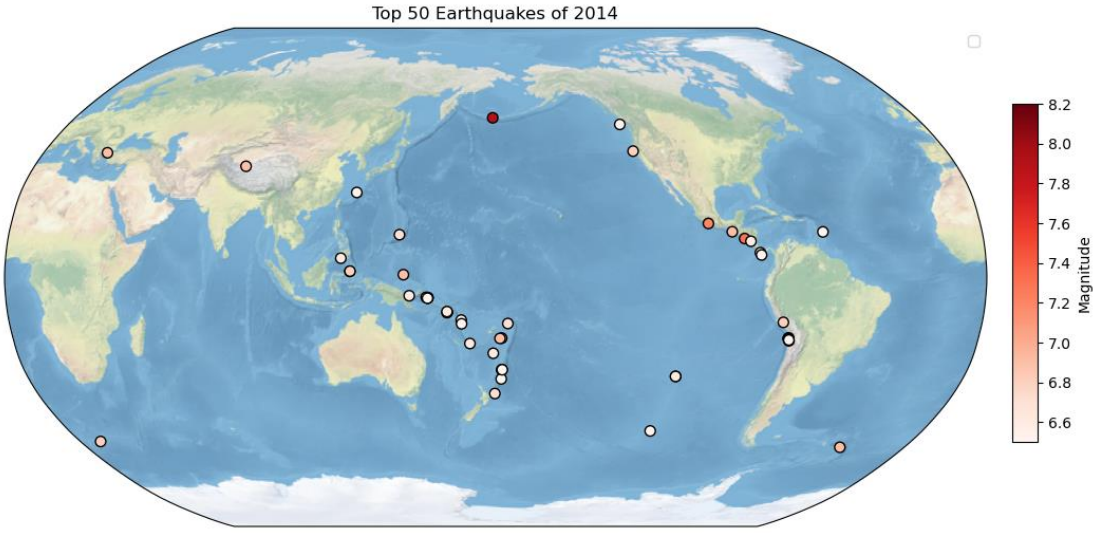
In this problem set, we will use this file from the USGS Earthquakes Database. The dataset is similar to the one you use in Assignment 02. Use the file provided (`usgs_earthquakes.csv`) to recreate the following map. Use the `mag` column for magnitude. **[10 points]**

Code

```
#Q1.Global Earthquakes
import pandas as pd
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
from cartopy.feature import LAND, OCEAN, COASTLINE, BORDERS
from matplotlib.cm import Reds
from matplotlib.colorbar import ColorbarBase
from matplotlib.colors import Normalize

# Load earthquake data
Sig_Eqs = pd.read_csv('usgs_earthquakes.csv')
Sig_Eqs['year'] = Sig_Eqs['time'].str.split('-').str[0]

# Filter earthquakes from 2014
df_2014 = Sig_Eqs[Sig_Eqs['year'] == '2014']

# Select top 50 earthquakes by magnitude
top_50_eqs = df_2014.nlargest(50, 'mag')

# Create a scatter plot on a map with a rotated Robinson projection
fig, ax = plt.subplots(figsize=(12, 8), subplot_kw={'projection': ccrs.Robinson(central_longitude=1
80)})
# Adjust the central_longitude and figure size as needed
plt.title('Top 50 Earthquakes of 2014')
ax.set_global()
ax.stock_img()
# Scatter plot with color-mapped magnitudes
scatter = ax.scatter(top_50_eqs['longitude'], top_50_eqs['latitude'], transform=ccrs.PlateCarree(), c=
top_50_eqs['mag'], cmap=Reds, s=50, marker='o', edgecolor='k')

# Add color bar
cbar = ColorbarBase(ax=fig.add_axes([0.92, 0.3, 0.02, 0.4]), cmap=Reds, norm=Normalize(vmin=
top_50_eqs['mag'].min(), vmax=top_50_eqs['mag'].max()), orientation='vertical', label='Magnitude')


# Show legend and title
```

```
ax.legend()

# Show the plot
plt.show()
```

The Result



Problem-solving ideas

The code aims to create a scatter plot on a map with a rotated Robinson projection, displaying the top 50 earthquakes of 2014 based on magnitude, here are some problem-solving ideas:

1.Improve Legend: If you intend to include a legend, you need to specify the labels for each part of the legend. For example, you might want to label the colorbar or the scatter plot.

2.Handle Missing Data: Ensure that your dataset doesn't contain missing or NaN values in the longitude, latitude, or magnitude columns. If there are missing values, consider cleaning the data before plotting.

3.Axis Labels and Title: Add axis labels (x-axis and y-axis) and a title to provide more context and information about what the plot is showing.

4.Enhance Colorbar: You might want to add more ticks to the colorbar to improve readability. You can use the ticks parameter in the ColorbarBase constructor.

5.Plot Earthquake Locations Only: The code currently uses ax.stock_img() to add a stock image of the Earth, but you might consider plotting only the earthquake locations on a blank background for better clarity.

2. Explore a `netCDF` dataset

Browse the NASA's Goddard Earth Sciences Data and Information Services Center (GES DISC) website. Search and download a dataset you are interested in. You are also welcome to use data from your group in this problem set. But the dataset should be in `netCDF` format. For this problem set, you are welcome to use the same dataset you used in Assignment 03.

2.1 [10 points] Make a global map of a certain variable. Your figure should contain: a project, x label and ticks, y label and ticks, title, gridlines, legend, colorbar, masks or features, annotations, and text box (**1 point each**).

2.2 [10 points] Make a regional map of the same variable. Your figure should contain: a different project, x label and ticks, y label and ticks, title, gridlines, legend, colorbar, masks or features, annotations, and text box (**1 point each**).

Code

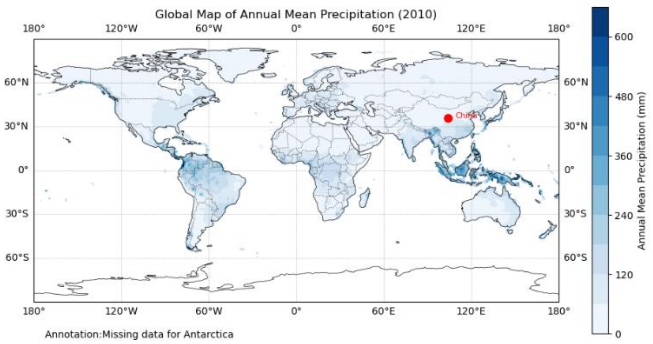
```
#Q2.Explore a netCDF dataset
#Q2.1
import numpy as np
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import xarray as xr

# Load netCDF dataset
ds = xr.open_dataset('precip.monitor.mon.total.1x1.v2020.nc')

# Select data for the year 2010
precipitation_2010 = ds['precip'].sel(time='2010')

# Calculate the annual mean
annual_mean_precipitation = precipitation_2010.mean(dim='time')

# Extract latitude and longitude
lat = ds['lat']
lon = ds['lon']

# Create a global map
fig, ax = plt.subplots(figsize=(12, 6), subplot_kw={'projection': ccrs.PlateCarree()})
ax.set_global()

# Plot annual mean precipitation on the map
c = ax.contourf(lon, lat, annual_mean_precipitation, transform=ccrs.PlateCarree(), cmap='Blues', l
evels=10)

# Add map features
ax.add_feature(cfeature.COASTLINE, linewidth=0.5, edgecolor='black')
ax.add_feature(cfeature.BORDERS, linestyle=':', linewidth=0.5, edgecolor='black')

# Add gridlines
ax.gridlines(draw_labels=True, linewidth=0.5, color='gray', alpha=0.5, linestyle='--')

# Add labels and title
ax.set_xlabel('Longitude')
ax.set_ylabel('Latitude')
```

```
ax.set_title('Global Map of Annual Mean Precipitation (2010)')

# Add colorbar
cbar = plt.colorbar(c, ax=ax, orientation='vertical', pad=0.05, aspect=20)
cbar.set_label('Annual Mean Precipitation (mm)')

# Add annotation below the plot
fig.text(0.25, 0.1, 'Annotation:Missing data for Antarctica', ha='center', fontsize=10)

# Add a point and label for China
china_lat, china_lon = 35.8617, 104.1954  # Coordinates for China
ax.plot(china_lon, china_lat, 'ro', markersize=8, transform=ccrs.PlateCarree(), label='China')
ax.text(china_lon + 5, china_lat, 'China', transform=ccrs.PlateCarree(), fontsize=8, color='red')

# Save or show the plot
plt.savefig('global_annual_mean_precipitation_2010.png', dpi=300, bbox_inches='tight')
plt.show()
#Q2.2
import numpy as np
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import xarray as xr

# Load netCDF dataset
ds = xr.open_dataset('precip.monitor.mon.total.1x1.v2020.nc')

# Select data for the year 2010
precipitation_2010 = ds['precip'].sel(time='2010')

# Calculate the annual mean
annual_mean_precipitation = precipitation_2010.mean(dim='time')

# Extract latitude and longitude
lat = ds['lat']
lon = ds['lon']

# Define the latitude and longitude range for China
china_lon_range = [70, 140]
china_lat_range = [15, 50]

# Create a map with China's extent
fig, ax = plt.subplots(figsize=(12, 6), subplot_kw={'projection': ccrs.PlateCarree()})
ax.set_extent([china_lon_range[0], china_lon_range[1], china_lat_range[0], china_lat_range[1]])
```

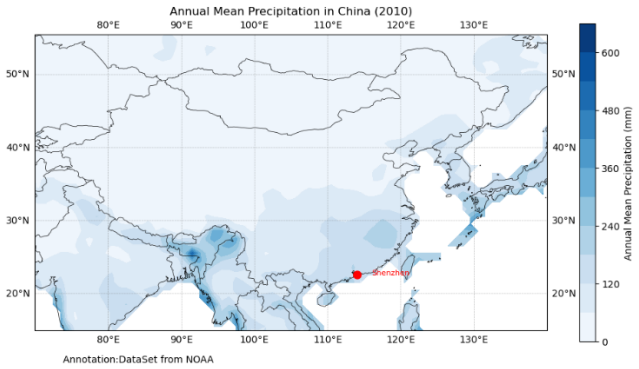
```
# Plot annual mean precipitation on the map
c = ax.contourf(lon, lat, annual_mean_precipitation, transform=ccrs.PlateCarree(), cmap='Blues', levels=10)

# Add map features
ax.add_feature(cfeature.COASTLINE, linewidth=0.5, edgecolor='black')
ax.add_feature(cfeature.BORDERS, linestyle='-', linewidth=0.5, edgecolor='black')

# Add gridlines
ax.gridlines(draw_labels=True, linewidth=0.5, color='gray', alpha=0.5, linestyle='--')

# Add labels and title
ax.set_xlabel('Longitude')
ax.set_ylabel('Latitude')
ax.set_title('Annual Mean Precipitation in China (2010)')

# Add colorbar
cbar = plt.colorbar(c, ax=ax, orientation='vertical', pad=0.05, aspect=20)
cbar.set_label('Annual Mean Precipitation (mm)')

# Add annotation below the plot
fig.text(0.25, 0.06, 'Annotation:DataSet from NOAA', ha='center', fontsize=10)

# Add a point and label for Shenzhen
shenzhen_lat, shenzhen_lon = 22.5431, 114.0579  # Coordinates for Shenzhen
ax.plot(shenzhen_lon, shenzhen_lat, 'ro', markersize=8, transform=ccrs.PlateCarree(), label='Shenzhen')
ax.text(shenzhen_lon + 2, shenzhen_lat, 'Shenzhen', transform=ccrs.PlateCarree(), fontsize=8, color='red')

# Save or show the plot
plt.savefig('annual_mean_precipitation_china.png', dpi=300, bbox_inches='tight')
plt.show()
```

The Result

Q2.1



Q2.2



Problem-solving ideas

The code explores a netCDF dataset containing precipitation data, creating global and China-sp ecific maps for the year 2010. Here are some ideas for problem-solving and improvement:

Q2.1 - Global Map of Annual Mean Precipitation (2010)

1.Annotation Placement:

The annotation about missing data for Antarctica is placed below the plot. Consider placing it in a way that it doesn't overlap with the colorbar or other important elements of the plot. Adj ust the vertical position as needed.

2.Improve Code Readability:

Add comments to explain the purpose of each section of the code. This can be helpful for so meone else (or yourself in the future) trying to understand or modify the code.

Q2.2 - Annual Mean Precipitation in China (2010)

1.Annotation Placement:

Similar to the global map, ensure that the annotation is placed appropriately and does not inter fere with other elements of the plot.

2.Improve Code Readability:

Add comments to explain the purpose of each section of the code. This can enhance the unde rstanding of the code.

3.Adjust Colorbar:

Consider adjusting the colorbar ticks and labels for better readability. You can use the ticks pa rameter in the colorbar function.