In [1]:

```
# Import modules
import numpy as np
import xarray as xr
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.ticker as mticker
import matplotlib.patches as mpatches
import cartopy.crs as ccrs
import cartopy.feature as cfeature
%matplotlib inline
```

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.

In [8]:

```
#  read data
usgs = pd.read_csv('usgs_earthquakes.csv')
usgs
```

Out[8]:

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net	
0	53:37.0	60.252000	-152.708100	90.20	1.10	ml	NaN	NaN	NaN	0.2900	ak	ak111
1	48:35.5	37.070300	-115.130900	0.00	1.33	ml	4.0	171.43	0.342000	0.0247	nn	nn004
2	47:24.0	64.671700	-149.252800	7.10	1.30	ml	NaN	NaN	NaN	1.0000	ak	ak111

In [9]:

```
# No.40 to No.54 all are 6.5 mag. Select all 54 earthquakes.
Top50 = usgs.sort_values('mag', ascending = False).head(54)

#usgs.iloc[[53132,64647,103919,67518]]

# Drop these points that are missing in example map.
# Points Selected by observations.
Top50 = Top50.drop(53132)
Top50 = Top50.drop(64647)
Top50 = Top50.drop(103919)
Top50 = Top50.drop(67518)

# No.55 is 6.4 mag. choose 54 and drop 4 points
usgs.sort_values('mag', ascending = False).head(55)
```

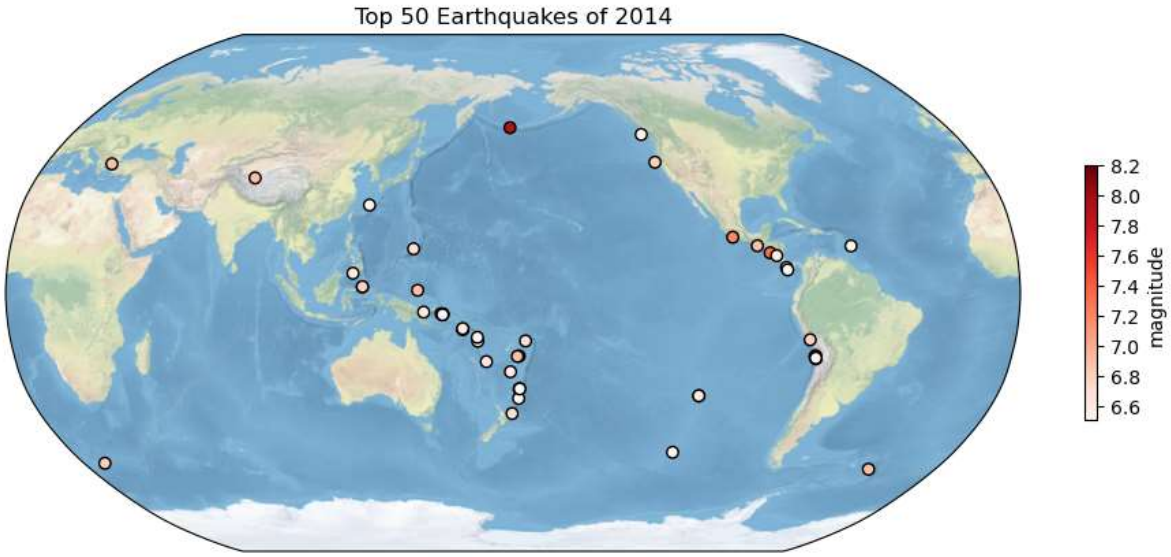
9062	03:29.0	-13.8633	167.2490	187.00	6.5	mww	NaN	14.0	3.99700	0.76	us	usc000lvb5
43290	35:24.2	7.2096	-82.3045	10.00	6.5	mww	NaN	33.0	3.12100	1.33	us	usb000qk64
53132	10:59.8	-10.1229	91.0921	4.00	6.5	mww	NaN	23.0	5.99800	0.85	us	usc000rfh2
43450	38:36.7	-49.9403	-114.7995	10.47	6.5	mww	NaN	35.0	23.16400	1.16	us	usb000qjhh
21508	41:09.5	7.7448	94.3342	21.54	6.4	mww	NaN	35.0	3.60400	1.14	us	usc000njrq

In [11]:

```
# Create and define the size of a figure object
plt.figure(figsize=(12,11))

# Create an ax with Robinson projection style with 180 in the middle.
ax = plt.axes(projection=ccrs.Robinson(180))

# Add background
ax.stock_img()

# Set title
ax.set_title('Top 50 Earthquakes of 2014')

# Point Plot and style setting
a = ax.scatter('longitude', 'latitude', data=Top50, transform=ccrs.PlateCarree(),
            s=37, c='mag', cmap='Reds',
            edgecolors='black', linewidths=1.0)

# Colorbar setting
plt.colorbar(a,shrink = 0.22,ticks = np.arange(6.6,8.3,0.2),label = 'magnitude')
```

Out[11]:

<matplotlib.colorbar.Colorbar at 0x2a4399565b0>



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 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.

In [12]:

```
# Open CSR Grace Data
GraceCSR = xr.open_dataset("CSR_GRACE_GRACE-FO_RL06_Mascons_all-corrections_v02.nc", engine="netcdf4
```

In [13]:

```
# Change the time dimention.

# Day.txt viewed by Arcmap and summarized by myself manually
day = pd.read_csv("day.txt",header = None)
time = pd.to_datetime(day[0]).to_numpy()

# time dimention changed
GraceCSR.coords['time'] = ('time',time)
```

In [14]:

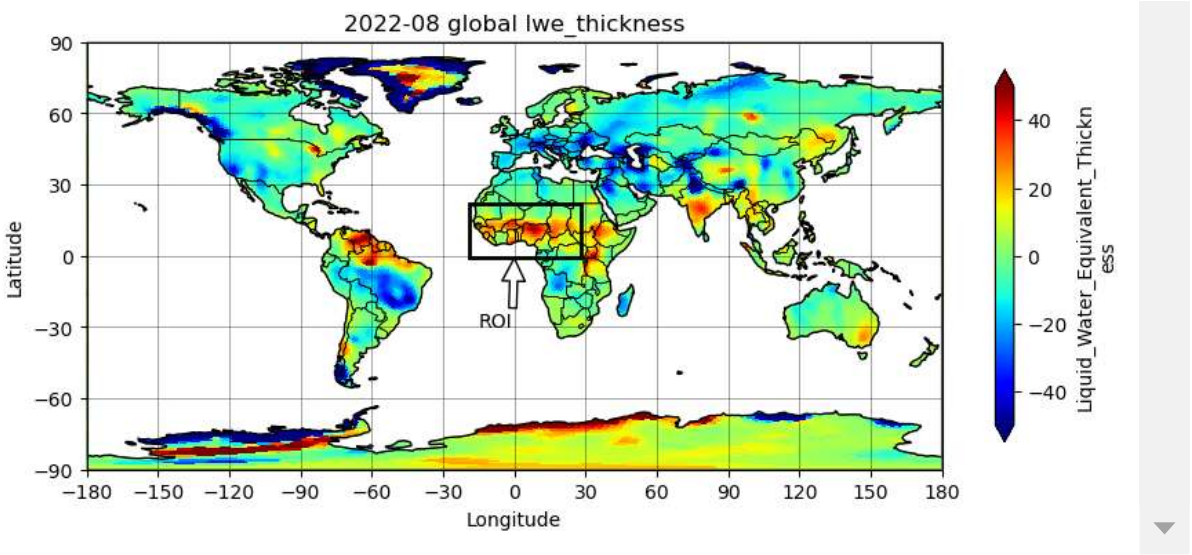
```
# Latest month for land
data = GraceCSR.lwe_thickness.isel(time=-1)
```

In [15]:

```
# Create and define the size of a figure object
plt.figure(figsize=(10,5), dpi=100)

# Create an axes with PlateCarree projection style
proj = ccrs.PlateCarree()
ax = plt.axes(projection=proj)

# Plot
data.plot(ax=ax, transform=ccrs.PlateCarree(),robust = True, cmap = 'jet',cbar_kwargs={'shrink': 0.
         #vmin=250, vmax=300, cbar_kwargs={'shrink': 0.4})

# Add border lines over countries
ax.add_feature(cfeature.NaturalEarthFeature(category='cultural',
                                            name='admin_0_countries',
                                            scale='110m',
                                            facecolor='none',
                                            edgecolor='black',
                                            linewidth=0.5))

# Add lat/lon gridlines, draw gridlines
gl = ax.gridlines(crs=ccrs.PlateCarree(), linewidth=0.5, color='black', alpha=0.5)

# Manipulate latitude and longitude gridline numbers and spacing
gl.ylocator = mticker.FixedLocator(np.arange(-90,90,30))
gl.xlocator = mticker.FixedLocator(np.arange(-180, 180, 30))

#axis setting
ax.set_xticks(np.arange(-180,181,30))
ax.set_yticks(np.arange(-90,91,30))
ax.set_xlabel('Longitude')
ax.set_ylabel('Latitude')

#Ocean
ax.add_feature(cfeature.OCEAN, facecolor='white',edgecolor='black', zorder=1)


# annotate
ax.annotate('ROI',xy = (0,-1),xytext = (-15,-30), color = 'black',
                    arrowprops = dict(facecolor = 'white', shrink = 1))

# box
ax.add_patch(mpatches.Rectangle( (-19,-1),47,23,facecolor = 'none',
                                edgecolor = 'black',linewidth=2) )

# title
ax.set_title('2022-08 global lwe_thickness')
```

Out[15]:

```
Text(0.5, 1.0, '2022-08 global lwe_thickness')
```



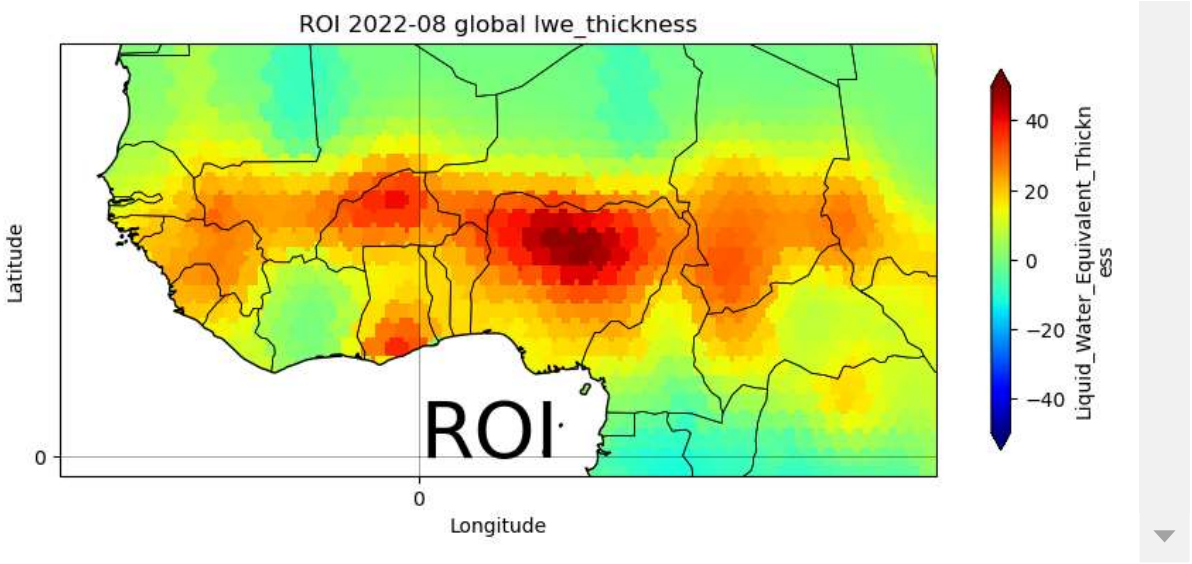
2.2 Make a regional map of the same variable. Your figure should contain: a different project, x label and ticks, y label and ticks, title, gridlines, legend, colorbar, masks or features, annotations, and text box.

In [17]:

```
import warnings
warnings.filterwarnings('ignore')

# Create and define the size of a figure object
plt.figure(figsize=(10,5), dpi=100)

# Create an axes with Orthographic projection style
proj = ccrs.Orthographic()
ax = plt.axes(projection=proj)

# Plot
data.plot(ax=ax, transform=ccrs.PlateCarree(),robust = True, cmap = 'jet',cbar_kwargs={'shrink': 0.
         #vmin=250, vmax=300, cbar_kwargs={'shrink': 0.4})

# Add border lines over countries
ax.add_feature(cfeature.NaturalEarthFeature(category='cultural',
                                            name='admin_0_countries',
                                            scale='110m',
                                            facecolor='none',
                                            edgecolor='black',
                                            linewidth=0.5))

# Add lat/lon gridlines, draw gridlines
gl = ax.gridlines(crs=ccrs.PlateCarree(), linewidth=0.5, color='black', alpha=0.5)

# Manipulate latitude and longitude gridline numbers and spacing
gl.ylocator = mticker.FixedLocator(np.arange(-90,90,30))
gl.xlocator = mticker.FixedLocator(np.arange(-180, 180, 30))

#axis setting
ax.set_xticks([0])
ax.set_yticks([0])
ax.set_xlabel('Longitude')
ax.set_ylabel('Latitude')

ax.add_feature(cfeature.OCEAN, facecolor='white',edgecolor='black', zorder=1)

# set region
extent = [-19,28,-1,22]
ax.set_extent(extent)

#text
ax.text(0,0,'ROI',size = 40)

#title
ax.set_title('ROI 2022-08 global lwe_thickness')
```

Out[17]:

```
Text(0.5, 1.0, 'ROI 2022-08 global lwe_thickness')
```


ROI 2022-08 global lwe_thickness
Latitude
Longitude
0
0
ROI
40
20
0
−20
−40
Liquid_Water_Equivalent_Thickness

In [ ]: