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 [2]:

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

Out[2]:

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms
0	53:37.0	60.252000	-152.708100	90.20	1.10	ml	NaN	NaN	NaN	0.2900
1	48:35.5	37.070300	-115.130900	0.00	1.33	ml	4.0	171.43	0.342000	0.0247
2	47:24.0	64.671700	-149.252800	7.10	1.30	ml	NaN	NaN	NaN	1.0000
3	30:54.0	63.188700	-148.957500	96.50	0.80	ml	NaN	NaN	NaN	1.0700
4	30:52.2	32.616833	-115.692500	10.59	1.34	ml	6.0	285.00	0.043210	0.2000
...	...	...	...	...	...	...	...	...	...	...
120103	10:16.0	60.963900	-146.762900	14.80	3.80	ml	NaN	NaN	NaN	0.6900
120104	09:39.0	58.869100	-154.415900	108.40	2.40	ml	NaN	NaN	NaN	0.6700
120105	09:25.3	38.843498	-122.825836	2.37	0.43	md	8.0	107.00	0.008991	0.0300
120106	05:54.0	65.152100	-148.992000	9.50	0.40	ml	NaN	NaN	NaN	0.6900
120107	04:05.0	60.227200	-147.024500	2.50	1.60	ml	NaN	NaN	NaN	0.7300

120108 rows × 15 columns

In [3]:

```
# Create a new column Year
usgs['Year'] = usgs.apply(lambda col: col['updated'][0:4], axis=1)
usgs
```

Out[3]:

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms
0	53:37.0	60.252000	-152.708100	90.20	1.10	ml	NaN	NaN	NaN	0.2900
1	48:35.5	37.070300	-115.130900	0.00	1.33	ml	4.0	171.43	0.342000	0.0247
2	47:24.0	64.671700	-149.252800	7.10	1.30	ml	NaN	NaN	NaN	1.0000
3	30:54.0	63.188700	-148.957500	96.50	0.80	ml	NaN	NaN	NaN	1.0700
4	30:52.2	32.616833	-115.692500	10.59	1.34	ml	6.0	285.00	0.043210	0.2000
...	...	...	...	...	...	...	...	...	...	...
120103	10:16.0	60.963900	-146.762900	14.80	3.80	ml	NaN	NaN	NaN	0.6900
120104	09:39.0	58.869100	-154.415900	108.40	2.40	ml	NaN	NaN	NaN	0.6700
120105	09:25.3	38.843498	-122.825836	2.37	0.43	md	8.0	107.00	0.008991	0.0300
120106	05:54.0	65.152100	-148.992000	9.50	0.40	ml	NaN	NaN	NaN	0.6900
120107	04:05.0	60.227200	-147.024500	2.50	1.60	ml	NaN	NaN	NaN	0.7300

120108 rows × 16 columns

In [4]:

```
# filter, keep only 2014
usgs2014 = usgs.loc[ usgs['Year'] == '2014']
usgs2014
```

Out[4]:

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms
0	53:37.0	60.252000	-152.708100	90.20	1.10	ml	NaN	NaN	NaN	0.2900
1	48:35.5	37.070300	-115.130900	0.00	1.33	ml	4.0	171.43	0.342000	0.0247
2	47:24.0	64.671700	-149.252800	7.10	1.30	ml	NaN	NaN	NaN	1.0000
3	30:54.0	63.188700	-148.957500	96.50	0.80	ml	NaN	NaN	NaN	1.0700
4	30:52.2	32.616833	-115.692500	10.59	1.34	ml	6.0	285.00	0.043210	0.2000
...	...	...	...	...	...	...	...	...	...	...
120102	16:21.0	62.795800	-150.387800	79.70	1.20	ml	NaN	NaN	NaN	0.4500
120104	09:39.0	58.869100	-154.415900	108.40	2.40	ml	NaN	NaN	NaN	0.6700
120105	09:25.3	38.843498	-122.825836	2.37	0.43	md	8.0	107.00	0.008991	0.0300
120106	05:54.0	65.152100	-148.992000	9.50	0.40	ml	NaN	NaN	NaN	0.6900
120107	04:05.0	60.227200	-147.024500	2.50	1.60	ml	NaN	NaN	NaN	0.7300

109052 rows × 16 columns

In [5]:

```
# select top 50 earthquake. here 51 is due to that No.52 earthquake is 6.0 mag.
top50 = usgs2014.sort_values('mag', ascending = False).head(51)
usgs2014.sort_values('mag', ascending = False).head(52)
```

Out[5]:

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net
34318	07:23.1	-6.5858	155.0485	60.53	7.1	mww	NaN	21.0	3.729	0.88	us
37367	57:58.8	-19.8927	-70.9455	28.42	6.9	mww	NaN	119.0	0.828	0.93	us
50609	19:15.9	-29.9772	-177.7247	20.00	6.9	mww	NaN	35.0	0.751	0.99	us
61294	54:41.0	-19.8015	-178.4001	615.42	6.9	mww	NaN	15.0	3.934	0.96	us
47934	52:55.2	-55.4703	-28.3669	8.00	6.9	mww	NaN	25.0	4.838	0.76	us
78063	22:03.7	0.8295	146.1688	13.00	6.9	mww	NaN	12.0	6.393	0.93	us
32964	57:01.4	-53.4967	8.7220	11.18	6.8	mww	NaN	27.0	18.877	0.74	us
50587	06:20.7	-29.9414	-177.6073	26.59	6.7	mwc	NaN	43.0	0.748	0.76	us
47776	15:09.3	-14.9831	-175.5096	18.00	6.7	mww	NaN	45.0	6.713	1.39	us
47320	36:35.6	-21.4542	170.3546	106.00	6.6	mww	NaN	10.0	3.340	0.89	us
33523	24:59.7	-11.1284	162.0520	10.00	6.6	mww	NaN	22.0	2.666	0.84	us
46333	15:52.9	-24.6108	179.0856	527.00	6.6	mww	NaN	19.0	5.329	1.05	us
31670	04:03.8	-6.6558	155.0869	29.00	6.6	mww	NaN	11.0	3.803	0.94	us

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net
17606	26:37.8	-32.9076	-177.8806	44.26	6.5	mww	NaN	22.0	3.651	1.28	us
43450	38:36.7	-49.9403	-114.7995	10.47	6.5	mww	NaN	35.0	23.164	1.16	us
34302	16:45.7	-6.7878	154.9502	20.00	6.5	mww	NaN	16.0	3.787	1.37	us
15979	40:13.5	-15.0691	167.3721	122.00	6.5	mww	NaN	20.0	0.409	1.15	us
9062	03:29.0	-13.8633	167.2490	187.00	6.5	mww	NaN	14.0	3.997	0.76	us
53132	10:59.8	-10.1229	91.0921	4.00	6.5	mww	NaN	23.0	5.998	0.85	us
50608	21:46.0	-29.9379	-177.5159	10.00	6.5	mwc	NaN	30.0	0.778	0.97	us
67518	00:27.9	-6.2304	152.8075	20.00	6.5	mww	NaN	15.0	2.124	1.06	us
24605	44:05.8	-60.8623	-19.9775	10.00	6.4	mww	NaN	48.0	10.949	0.82	us
36860	26:15.7	-20.7969	-70.5865	25.00	6.4	mww	NaN	35.0	1.056	0.98	us
21508	41:09.5	7.7448	94.3342	21.54	6.4	mww	NaN	35.0	3.604	1.14	us
26230	56:57.8	-14.7378	169.8234	638.00	6.3	mww	NaN	19.0	7.967	0.89	us
36488	37:50.6	-20.6426	-70.6540	13.71	6.3	mww	NaN	43.0	1.153	1.01	us
42588	16:34.4	6.4264	144.9363	11.00	6.3	mww	NaN	10.0	7.118	1.18	us
67804	50:07.3	-30.4601	-176.4451	35.00	6.3	mww	NaN	13.0	1.768	0.68	us
46331	25:16.0	-25.8072	178.2401	634.21	6.3	mww	NaN	20.0	4.826	1.11	us
45471	52:28.3	-36.1703	-97.0540	16.83	6.3	mww	NaN	26.0	13.799	0.87	us

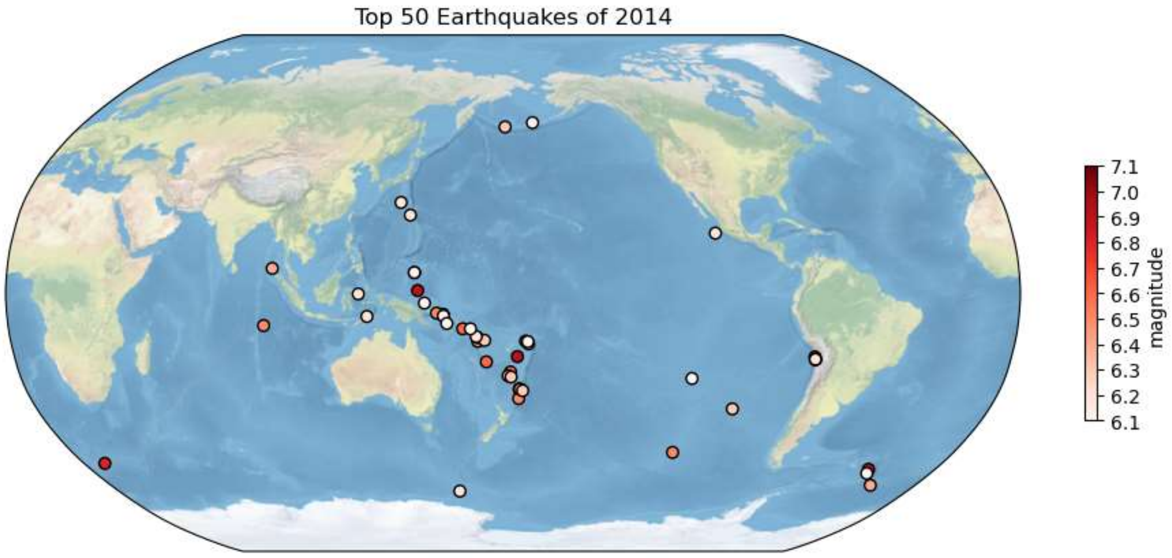
	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net
19989	29:35.7	-26.1692	179.2877	495.00	6.3	mww	NaN	22.0	3.933	0.64	us
50246	15:35.5	52.2045	176.6981	4.00	6.3	mwb	NaN	21.0	1.145	0.69	us
86634	46:06.4	-0.2422	125.1040	35.00	6.2	mww	NaN	28.0	2.463	0.52	us
31273	15:58.1	-7.1646	155.3351	20.00	6.2	mww	NaN	16.0	4.322	0.79	us
47962	56:31.7	24.3877	142.6278	48.00	6.2	mww	NaN	11.0	2.726	0.66	us
47450	55:32.4	28.3404	138.8441	511.00	6.2	mww	NaN	13.0	3.210	0.64	us
51841	17:55.5	-13.5585	166.8278	36.00	6.2	mww	NaN	19.0	1.912	1.12	us
87824	22:59.0	18.7529	-107.0488	17.00	6.2	mww	NaN	85.0	5.572	0.81	us
32168	06:51.5	-62.8705	155.7431	20.56	6.2	mww	NaN	35.0	8.561	1.04	us
34414	01:45.2	-20.6590	-70.6472	13.77	6.2	mww	NaN	44.0	0.489	0.80	us
61911	27:10.0	-15.8239	-174.4517	227.27	6.2	mww	NaN	23.0	3.210	0.65	us
76960	45:22.7	-7.2741	128.0364	10.00	6.2	mww	NaN	16.0	3.309	0.89	us
3308	29:07.0	-15.1443	-174.6812	6.12	6.1	mww	NaN	45.0	3.068	0.89	us
17995	58:44.0	-56.8269	-27.3391	129.97	6.1	mww	NaN	23.0	5.783	0.79	us
31994	13:12.0	-11.1387	164.8139	10.00	6.1	mww	NaN	19.0	5.078	0.78	us
33997	24:23.3	-7.1033	155.2380	20.00	6.1	mww	NaN	24.0	4.210	1.07	us
87982	53:11.8	-26.6478	-114.5000	7.00	6.1	mww	NaN	33.0	4.637	0.57	us

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net
42743	56:13.2	6.4507	144.9238	10.00	6.1	mww	NaN	38.0	7.093	0.81	us
24404	03:09.8	-3.0856	148.5531	7.00	6.1	mww	NaN	17.0	1.574	1.22	us
82055	13:50.0	-9.4618	156.4122	4.00	6.1	mww	NaN	21.0	3.488	1.25	us
9751	13:40.1	53.6047	-171.8210	265.00	6.1	mww	NaN	30.0	1.845	1.15	us
68266	53:29.2	-62.3014	155.1868	10.00	6.0	mww	NaN	36.0	8.070	1.29	us

```
In  [6]:
```

```
# 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(6.0,7.2,0.1),label = 'magnitude')
```

```
Out[6]:

<matplotlib.colorbar.Colorbar at 0x29f3f81c730>
```



But it didn't look like the example map given in assignment.

To recreate the map given as an example.

In [7]:

```
Top50All = usgs.sort_values('mag', ascending = False).head(54)

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

# Drop these points that are missing in example map
Top50All = Top50All.drop(53132)
Top50All = Top50All.drop(64647)
Top50All = Top50All.drop(103919)
Top50All = Top50All.drop(67518)

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

Out[7]:

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	r
37371	46:47.3	-19.6097	-70.7691	25.00	8.2	mww	NaN	23.0	0.60900	0.66	
50562	53:09.7	51.8486	178.7352	109.00	7.9	mww	NaN	22.0	0.13300	0.71	
36918	43:13.1	-20.5709	-70.4931	22.40	7.7	mww	NaN	44.0	1.02900	0.82	
33808	14:39.3	-11.2701	162.1481	22.56	7.6	mww	NaN	13.0	2.82800	0.71	
31496	28:00.8	-6.7547	155.0241	43.37	7.5	mww	NaN	16.0	3.82000	1.25	
33537	36:19.2	-11.4633	162.0511	39.00	7.4	mww	NaN	17.0	2.88500	1.00	
95913	51:34.5	12.5262	-88.1225	40.00	7.3	mww	NaN	18.0	1.07800	0.70	
31850	27:24.9	17.3970	-100.9723	24.00	7.2	mww	NaN	46.0	2.25000	1.20	
34318	07:23.1	-6.5858	155.0485	60.53	7.1	mww	NaN	21.0	3.72900	0.88	
106285	31:41.7	1.8929	126.5217	45.00	7.1	mww	NaN	18.0	1.39700	0.71	
111052	57:22.4	-19.6903	-177.7587	434.00	7.1	mww	NaN	13.0	4.41500	0.84	

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	r
97602	14:31.4	-32.1082	-110.8112	16.54	7.0	mww	NaN	22.0	5.12700	0.43	
47934	52:55.2	-55.4703	-28.3669	8.00	6.9	mww	NaN	25.0	4.83800	0.76	
78063	22:03.7	0.8295	146.1688	13.00	6.9	mww	NaN	12.0	6.39300	0.93	
50609	19:15.9	-29.9772	-177.7247	20.00	6.9	mww	NaN	35.0	0.75100	0.99	
14025	19:49.1	35.9053	82.5864	10.00	6.9	mww	NaN	18.0	7.49600	0.83	
61294	54:41.0	-19.8015	-178.4001	615.42	6.9	mww	NaN	15.0	3.93400	0.96	
37367	57:58.8	-19.8927	-70.9455	28.42	6.9	mww	NaN	119.0	0.82800	0.93	
39915	25:02.4	40.2893	25.3889	6.43	6.9	mww	NaN	25.0	0.40200	0.67	
66278	23:54.8	14.7240	-92.4614	53.00	6.9	mww	NaN	51.0	0.26300	1.38	
24887	18:13.4	40.8287	-125.1338	16.60	6.8	Mw	NaN	230.4	0.65577	0.15	
32964	57:01.4	-53.4967	8.7220	11.18	6.8	mww	NaN	27.0	18.87700	0.74	
101767	33:43.6	1.9604	126.5751	39.00	6.8	mww	NaN	19.0	1.42100	1.19	
71135	21:45.5	-14.5980	-73.5714	101.00	6.8	mww	NaN	18.0	4.10700	0.90	
22968	16:29.6	-19.9807	-70.7022	20.00	6.7	mww	NaN	44.0	1.00800	0.83	
84421	14:45.4	13.7641	144.4294	130.00	6.7	mww	NaN	11.0	0.46000	0.84	
105466	33:20.5	-37.6478	179.6621	22.00	6.7	mww	NaN	25.0	1.07900	0.49	
50587	06:20.7	-29.9414	-177.6073	26.59	6.7	mwc	NaN	43.0	0.74800	0.76	
47776	15:09.3	-14.9831	-175.5096	18.00	6.7	mww	NaN	45.0	6.71300	1.39	

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	r
46333	15:52.9	-24.6108	179.0856	527.00	6.6	mww	NaN	19.0	5.32900	1.05	
33523	24:59.7	-11.1284	162.0520	10.00	6.6	mww	NaN	22.0	2.66600	0.84	
117886	54:52.5	7.9401	-82.6865	20.00	6.6	mww	NaN	43.0	2.48500	1.01	
47320	36:35.6	-21.4542	170.3546	106.00	6.6	mww	NaN	10.0	3.34000	0.89	
119708	11:31.0	6.1572	123.1261	614.00	6.6	mww	NaN	9.0	2.60000	1.32	
97597	32:05.1	-32.0953	-110.8647	10.00	6.6	mww	NaN	32.0	5.12700	0.43	
109305	33:55.3	-5.9873	148.2315	53.19	6.6	mww	NaN	13.0	3.55700	0.91	
118288	22:02.2	-6.5108	154.4603	23.00	6.6	mww	NaN	12.0	3.24600	0.72	
34096	29:13.0	11.6420	-85.8779	135.00	6.6	mww	NaN	20.0	0.76100	1.35	
31670	04:03.8	-6.6558	155.0869	29.00	6.6	mww	NaN	11.0	3.80300	0.94	
15979	40:13.5	-15.0691	167.3721	122.00	6.5	mww	NaN	20.0	0.40900	1.15	
103919	10:19.6	2.2999	127.0562	35.00	6.5	mww	NaN	9.0	1.54900	1.16	
17606	26:37.8	-32.9076	-177.8806	44.26	6.5	mww	NaN	22.0	3.65100	1.28	
26972	11:23.4	27.4312	127.3674	119.00	6.5	mww	NaN	17.0	1.00100	0.97	
36929	58:30.5	-20.3113	-70.5756	24.07	6.5	mww	NaN	82.0	0.82800	0.62	
34302	16:45.7	-6.7878	154.9502	20.00	6.5	mww	NaN	16.0	3.78700	1.37	

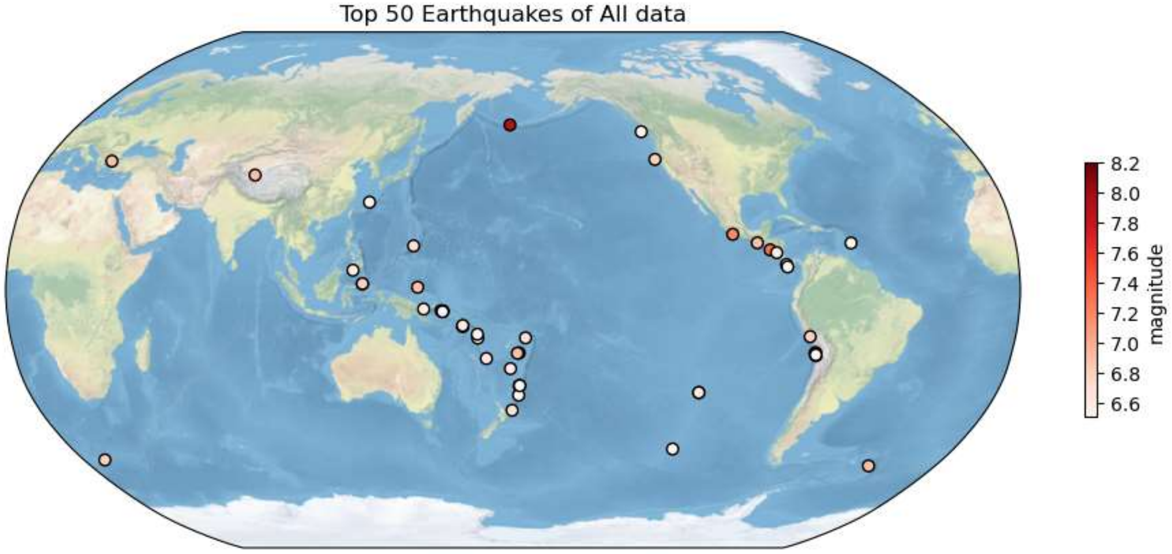
	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	r
50608	21:46.0	-29.9379	-177.5159	10.00	6.5	mwc	NaN	30.0	0.77800	0.97	
64647	22:00.8	37.0052	142.4525	20.00	6.5	mww	NaN	9.0	2.42100	0.75	
12142	27:13.1	14.6682	-58.9272	14.83	6.5	mww	NaN	30.0	1.63500	0.94	
29890	10:10.1	49.6388	-127.7316	10.00	6.5	mww	NaN	NaN	NaN	0.85	
67518	00:27.9	-6.2304	152.8075	20.00	6.5	mww	NaN	15.0	2.12400	1.06	
9062	03:29.0	-13.8633	167.2490	187.00	6.5	mww	NaN	14.0	3.99700	0.76	
43290	35:24.2	7.2096	-82.3045	10.00	6.5	mww	NaN	33.0	3.12100	1.33	
53132	10:59.8	-10.1229	91.0921	4.00	6.5	mww	NaN	23.0	5.99800	0.85	
43450	38:36.7	-49.9403	-114.7995	10.47	6.5	mww	NaN	35.0	23.16400	1.16	
21508	41:09.5	7.7448	94.3342	21.54	6.4	mww	NaN	35.0	3.60400	1.14	

In [8]:

```
# 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 All data')

# Point Plot and style setting
a = ax.scatter('longitude', 'latitude', data=Top50All, 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[8]:

```
<matplotlib.colorbar.Colorbar at 0x29f3f92df40>
```



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 [9]:

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

In [10]:

```
# 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 [11]:

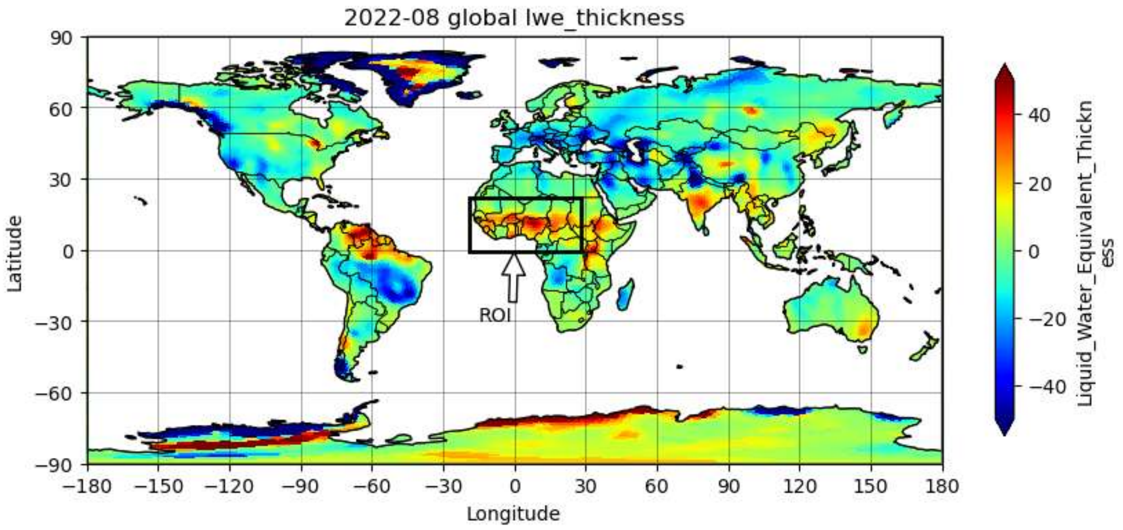
```
# Latest month for land
data = GraceCSR.lwe_thickness.isel(time=-1)
```

In [12]:

```
# 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': 0.7})
        #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[12]:

```
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 [13]:

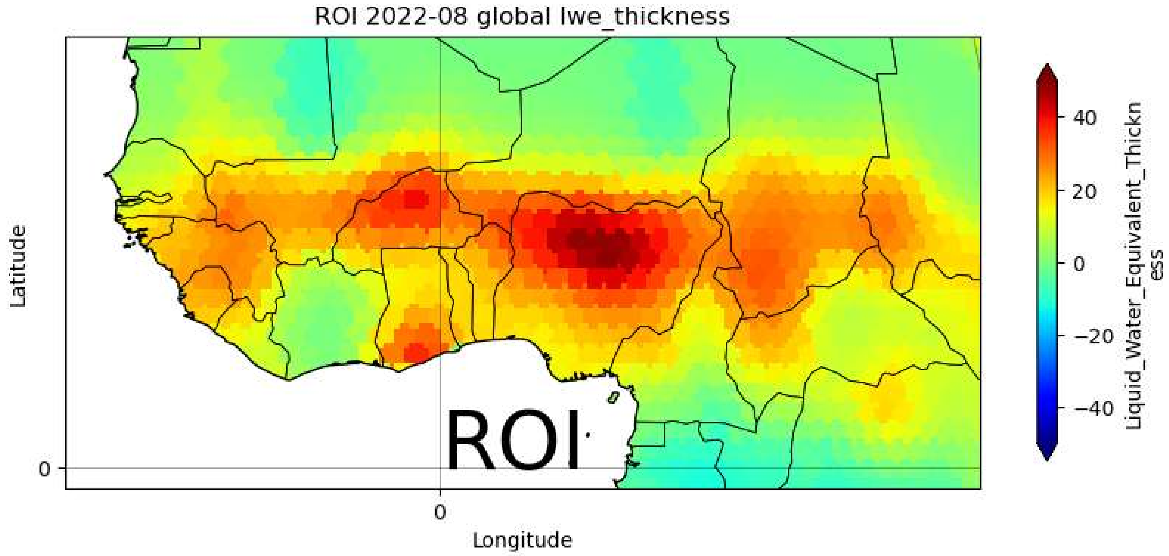
```
# Create and define the size of a figure object
plt.figure(figsize=(10,5), dpi=100)

# 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': 0.7})
        #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[13]:

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

```
D:\ANACONDA\lib\site-packages\cartopy\crs.py:245: ShapelyDeprecationWarning: __le
n__ for multi-part geometries is deprecated and will be removed in Shapely 2.0. C
heck the length of the `geoms` property instead to get the  number of parts of a
multi-part geometry.
  if len(multi_line_string) > 1:
D:\ANACONDA\lib\site-packages\cartopy\crs.py:297: ShapelyDeprecationWarning: Iter
ation over multi-part geometries is deprecated and will be removed in Shapely 2.
0. Use the `geoms` property to access the constituent parts of a multi-part geome
try.
  for line in multi_line_string:
```

```
D:\ANACONDA\lib\site-packages\cartopy\crs.py:364: ShapelyDeprecationWarning: __le
n__ for multi-part geometries is deprecated and will be removed in Shapely 2.0. C
heck the length of the `geoms` property instead to get the  number of parts of a
multi-part geometry.
  if len(p_mline) > 0:
D:\ANACONDA\lib\site-packages\cartopy\crs.py:256: ShapelyDeprecationWarning: __le
n__ for multi-part geometries is deprecated and will be removed in Shapely 2.0. C
heck the length of the `geoms` property instead to get the  number of parts of a
multi-part geometry.
  line_strings = list(multi_line_string)
D:\ANACONDA\lib\site-packages\cartopy\crs.py:256: ShapelyDeprecationWarning: Iter
ation over multi-part geometries is deprecated and will be removed in Shapely 2.
0. Use the `geoms` property to access the constituent parts of a multi-part geome
try.
  line_strings = list(multi_line_string)
D:\ANACONDA\lib\site-packages\cartopy\crs.py:402: ShapelyDeprecationWarning: Iter
ation over multi-part geometries is deprecated and will be removed in Shapely 2.
0. Use the `geoms` property to access the constituent parts of a multi-part geome
try.
  line_strings.extend(multi_line_string)
D:\ANACONDA\lib\site-packages\cartopy\crs.py:402: ShapelyDeprecationWarning: __le
n__ for multi-part geometries is deprecated and will be removed in Shapely 2.0. C
heck the length of the `geoms` property instead to get the  number of parts of a
multi-part geometry.
  line_strings.extend(multi_line_string)
```


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

In [ ]: