In [2]:

```
#1

# Prerequisites

# Load modules
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
import matplotlib.ticker as mticker
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import xarray as xr

# Show plots in the notebook
%matplotlib inline

# Read earthquakes data
import pandas as pd
earthquakes_data =  pd.read_csv('usgs_earthquakes.csv')
```

In [3]:

```
earthquakes_data
```

Out[3]:

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	
0	2014-01-31 23:53:37.000	60.252000	-152.708100	90.20	1.10	ml	NaN	NaN	NaN	0
1	2014-01-31 23:48:35.452	37.070300	-115.130900	0.00	1.33	ml	4.0	171.43	0.342000	0
2	2014-01-31 23:47:24.000	64.671700	-149.252800	7.10	1.30	ml	NaN	NaN	NaN	1
3	2014-01-31 23:30:54.000	63.188700	-148.957500	96.50	0.80	ml	NaN	NaN	NaN	1
4	2014-01-31 23:30:52.210	32.616833	-115.692500	10.59	1.34	ml	6.0	285.00	0.043210	0
...	...	...	...	...	...	...	...	...	...	
120103	2014-12-01 00:10:16.000	60.963900	-146.762900	14.80	3.80	ml	NaN	NaN	NaN	0
120104	2014-12-01 00:09:39.000	58.869100	-154.415900	108.40	2.40	ml	NaN	NaN	NaN	0
120105	2014-12-01 00:09:25.350	38.843498	-122.825836	2.37	0.43	md	8.0	107.00	0.008991	0
120106	2014-12-01 00:05:54.000	65.152100	-148.992000	9.50	0.40	ml	NaN	NaN	NaN	0
120107	2014-12-01 00:04:05.000	60.227200	-147.024500	2.50	1.60	ml	NaN	NaN	NaN	0

120108 rows × 15 columns

In [4]:

```
#提取出用到的绘图数据
ds=earthquakes_data.loc[:,[’time’,’latitude’,’longitude’,’mag’]]
ds
```

Out[4]:

	time	latitude	longitude	mag
0	2014-01-31 23:53:37.000	60.252000	-152.708100	1.10
1	2014-01-31 23:48:35.452	37.070300	-115.130900	1.33
2	2014-01-31 23:47:24.000	64.671700	-149.252800	1.30
3	2014-01-31 23:30:54.000	63.188700	-148.957500	0.80
4	2014-01-31 23:30:52.210	32.616833	-115.692500	1.34
...	...	...	...	...
120103	2014-12-01 00:10:16.000	60.963900	-146.762900	3.80
120104	2014-12-01 00:09:39.000	58.869100	-154.415900	2.40
120105	2014-12-01 00:09:25.350	38.843498	-122.825836	0.43
120106	2014-12-01 00:05:54.000	65.152100	-148.992000	0.40
120107	2014-12-01 00:04:05.000	60.227200	-147.024500	1.60

120108 rows × 4 columns

In [5]:

```
#按照mag排序后，选择前50的数据并重新排序
ds_top50=ds.sort_values(’mag’,ascending=False).iloc[0:50].reset_index(drop=True)
ds_top50
```

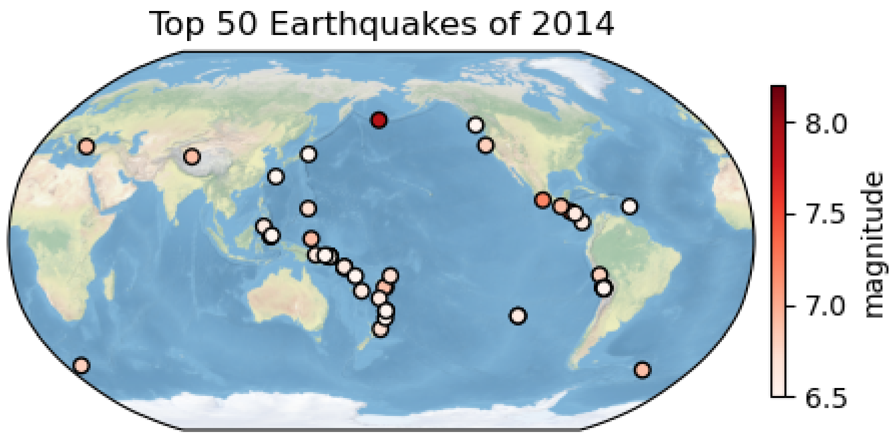
Out[5]:

	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

	time	latitude	longitude	mag
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
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 [18]:

```
#建立经纬度和震级的列表
lons=[]
lats=[]
mags=[]
for i in ds_top50['latitude']:
    lats.append(i)
for i in ds_top50['longitude']:
    lons.append(i)
for i in ds_top50['mag']:
    mags.append(i)

#提取震级的最大值和最小值
max_mag = ds_top50['mag'][0]
min_mag = ds_top50['mag'][49]

#建立绘图面板
# Create and define the size of a figure object
plt.figure(figsize=(5,5), dpi=100)

#确定投影的类型
ax = plt.axes(projection=ccrs.Robinson(central_longitude=180))
ax.stock_img()

#绘图
ax.scatter(lons,lats,c=mags ,transform=ccrs.PlateCarree(),vmin= min_mag, vmax= max_mag , cmap='Reds'
           marker='o',edgecolors='black',s=30, linewidths=1)
#绘制颜色标签
cb=ax.figure.colorbar(ax.collections[0],fraction=0.02, pad=0.02,label = 'magnitude')
#标题
ax.set_title('Top 50 Earthquakes of 2014')
plt.show()

#参照这些网站
#https://blog.csdn.net/weixin_42969619/article/details/99672134
#https://scitools.org.uk/cartopy/docs/latest/gallery/lines_and_polygons/global_map.html#sphx-glr-gal
```



In [ ]:

In [ ]:

```

```

In [ ]:

```

```

In [19]:

```
#2.1


# 打开文件
#采用第三次作业中第一题使用的数据
ds= xr.open_dataset("NOAA_NCDC_ERSST_v3b_SST.nc")
ds
```

Out[19]:

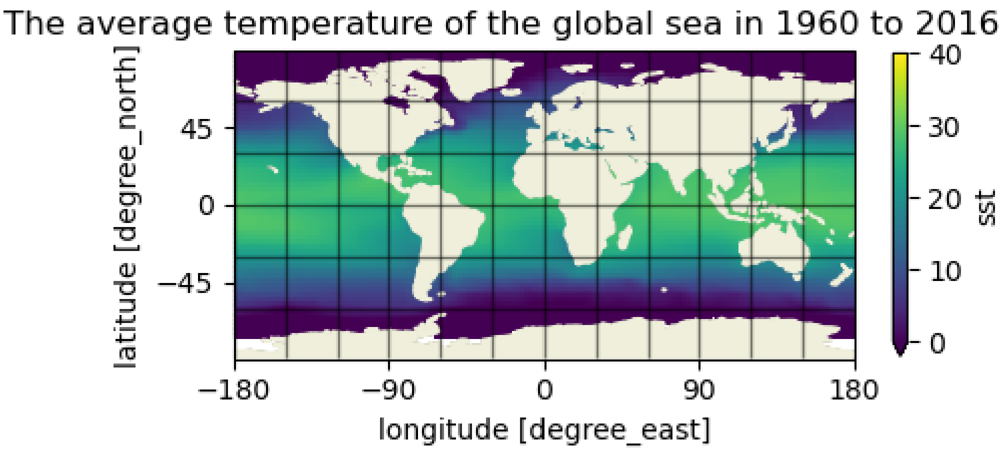
```
xarray.Dataset

▶ Dimensions:       (lat: 89, lon: 180, time: 684)
▼ Coordinates:
    lat             (lat)                 float32  -88.0 -86.0 -84.0 ... 86.0 88.0
    lon             (lon)                 float32  0.0 2.0 4.0 ... 354.0 356.0 35...
    time            (time)          datetime64[ns]  1960-01-15 ... 2016-12-15
▼ Data variables:
    sst             (time, lat, lon)      float32  ...
▼ Attributes:
    Conventions :   IRIDL
    source :        https://iridl.ldeo.columbia.edu/SOURCES/.NOAA/.NCDC/.ERSST/.ver
                    sion3b/.sst/
    history :       extracted and cleaned by Ryan Abernathey for Research Computing i
                    n Earth Science
```

In [20]:

```
#建立绘图面板
# Create and define the size of a figure object
plt.figure(figsize=(5,5), dpi=100)

#确定投影的类型
ax = plt.axes(projection=ccrs.PlateCarree())
ax.tick_params(labelsize=10)
ax.set_xticks(np.linspace(-180, 180, 5), crs=ccrs.PlateCarree())
ax.set_yticks(np.linspace(-90, 90, 5), crs=ccrs.PlateCarree())
ax.add_feature(cfeature.NaturalEarthFeature(category='cultural',
                                            name='admin_0_countries',
                                            scale='110m',
                                            facecolor='none',
                                            edgecolor='black',
                                            linewidth=0.5))
# ax = plt.axes(projection=ccrs.Robinson(central_longitude=180))
# ax.stock_img()
# Draw gridlines
gl = ax.gridlines(crs=ccrs.PlateCarree(), linewidth=1, 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))

#计算每个坐标的平均温度
surface_T = ds.sst.mean(dim='time')
surface_T

# Plot the surface temperature
surface_T.plot(ax=ax, transform=ccrs.PlateCarree(),
         vmin=0, vmax=40, cbar_kwargs={'shrink': 0.4})

ax.add_feature(cfeature.LAND, zorder=1)
plt.title("The average temperature of the global sea in 1960 to 2016 year")
```

Out[20]:

```
Text(0.5, 1.0, 'The average temperature of the global sea in 1960 to 2016 year')
```

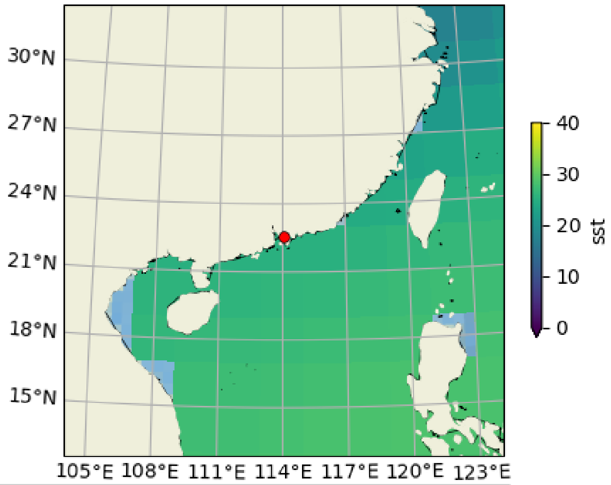


In [21]:

```
#2.2

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

# 选择深圳地域位置
central_lon, central_lat = 114.06, 22.54 # Shenzhen
proj = ccrs.Orthographic(central_lon, central_lat)

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

# Set a region and plot
extent = [central_lon-10, central_lon+10, central_lat-10, central_lat+10]
ax.set_extent(extent)



#绘制轮廓
ax.add_feature(cfeature.NaturalEarthFeature(category='cultural',
                                           name='admin_0_countries',
                                           scale='110m',
                                           facecolor='none',
                                           edgecolor='black',
                                           linewidth=0.5))
ax.coastlines(resolution='10m', linewidth=0.5)
ax.stock_img()


#标记出深圳的位置
ax.scatter(116.3,39.9,s=30,c='r',marker='o',zorder=5,edgecolors='k', linewidths=0.5)

#绘制网格和xy刻度
gl = ax.gridlines(draw_labels=True)
gl.top_labels = False
gl.right_labels = False
#绘制海洋表面温度变化
surface_T.plot(ax=ax, transform=ccrs.PlateCarree(),
         vmin=0, vmax=40, cbar_kwargs={'shrink': 0.4})

#覆盖陆地
ax.add_feature(cfeature.LAND, zorder=1)

#标题
ax.text(0.995,
        -0.13,
        "The average of sea temperature around the Shenzhen in 1960 to 2016 year",
        horizontalalignment='right',
        transform=ax.transAxes,
        fontsize=14,
        bbox=dict(boxstyle='square, pad=0.25',
                  facecolor='white',
                  edgecolor='black'))

#参照网址
#https://geocat-examples.readthedocs.io/en/latest/gallery/Polygons/NCL_polyg_4.html#sphx-glr-gallery
```

Out[21]:

```
Text(0.995, -0.13, 'The average of sea temperature around the Shenzhen in 1960 to 2016 year')
```



The average of sea temperature around the Shenzhen in 1960 to 2016 year