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 cartopy.crs as ccrs
import cartopy.feature as cfeature
import cartopy.mpl.ticker as cticker
%matplotlib inline
from matplotlib.ticker import (MultipleLocator, FormatStrFormatter, AutoMinorLoc
```

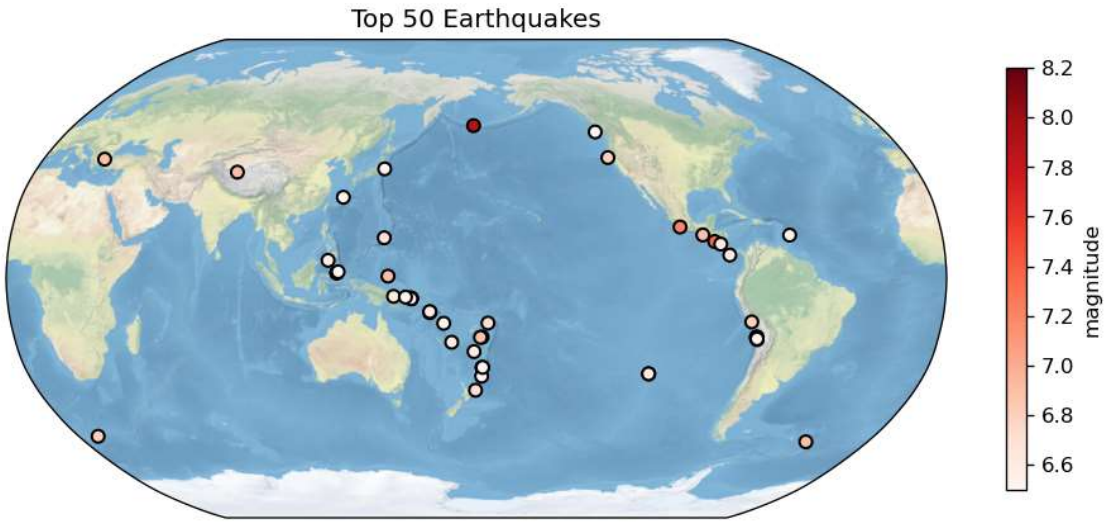
In [2]:
```
#1
#筛选出前50震级
Sig_Eqs=pd.read_csv("usgs_earthquakes.csv")
eq=Sig_Eqs.sort_values("mag",ascending=False).head(50)
eq
```

Out[2]:

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin
37371	2014-04-01 23:46:47.260	-19.6097	-70.7691	25.00	8.2	mww	NaN	23.0	0.60900
50562	2014-06-23 20:53:09.700	51.8486	178.7352	109.00	7.9	mww	NaN	22.0	0.13300
36918	2014-04-03 02:43:13.110	-20.5709	-70.4931	22.40	7.7	mww	NaN	44.0	1.02900
33808	2014-04-12 20:14:39.300	-11.2701	162.1481	22.56	7.6	mww	NaN	13.0	2.82800
31496	2014-04-19 13:28:00.810	-6.7547	155.0241	43.37	7.5	mww	NaN	16.0	3.82000

In [3]:

```
fig=plt.figure(figsize=(10,6),dpi=120)
#制定proj
proj = ccrs.Robinson(central_longitude=180,globe=None)
ax = plt.axes(projection=proj)
#导入背景
ax.set_global()
ax.stock_img()
ax.set_title('Top 50 Earthquakes')
ax0=plt.scatter(eq['longitude'],eq['latitude'],marker='o',c=eq['mag'],cmap='Reds'
plt.colorbar(shrink=0.6,format='%.1f',ticks=[6.6,6.8,7.0,7.2,7.4,7.6,7.8,8.0,8.2]
plt.show()
```



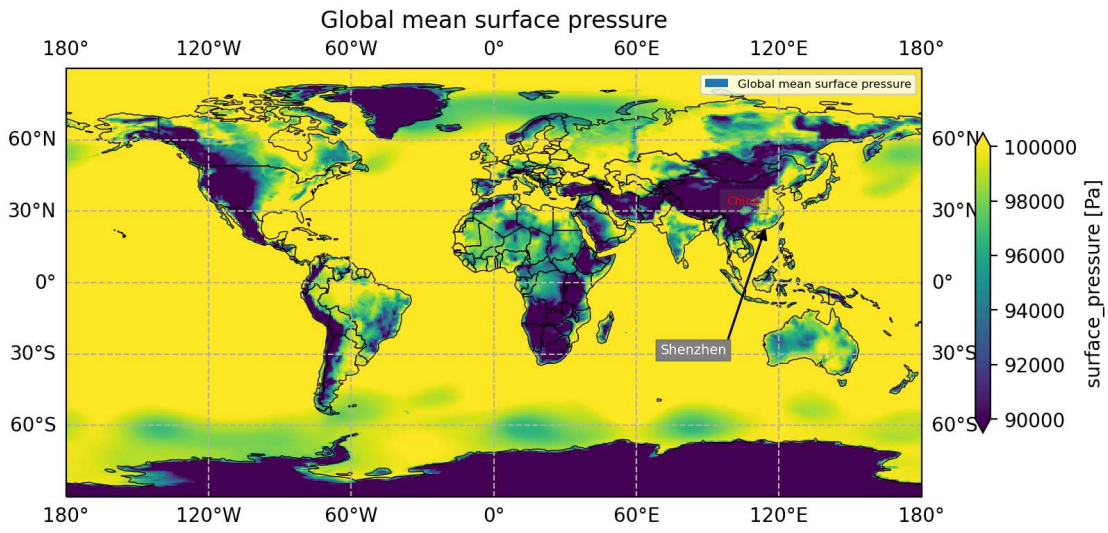
In [2]:

```
#2
ds3 = xr.open_dataset("MERRA2_400.inst3_3d_chm_Nv.20200109.nc4", engine="netcdf4"
ds3
PS = ds3.PS.groupby('time.month').mean(dim=['time'])
```

In [5]:

```
plt.figure(figsize=(10,4), dpi=200)
proj = ccrs.PlateCarree()
ax   = plt.axes(projection=proj)

#masks or features
ax.add_feature(cfeature.NaturalEarthFeature(category='cultural',
                                            name='admin_0_countries',
                                            scale='110m',
                                            facecolor='none',
                                            edgecolor='black',
                                            linewidth=0.5))

#colorbar
PS.plot(ax=ax, transform=ccrs.PlateCarree(),vmin=90000, vmax=100000,cbar_kwargs=

#legend
plt.legend(['Global mean surface pressure'],loc='best',fontsize=6)

#gridlines
gl = ax.gridlines(draw_labels=True, linestyle='--')

#x label and ticks, y label and ticks
gl.xlocator = cticker.LongitudeLocator()
gl.ylocator = cticker.LatitudeLocator()
gl.xformatter = cticker.LongitudeFormatter()
gl.yformatter = cticker.LatitudeFormatter()

# annotations
ax.annotate('Shenzhen',xy=(114.06,22.54),xytext=(70,-30),   # 深圳
            bbox=dict(boxstyle='square', fc='grey',linewidth=0.1),
            arrowprops=dict(facecolor='black', width=0.01,headwidth=5, headlengt
            fontsize=7, color='white', horizontalalignment='left',
            transform=ccrs.PlateCarree())

#text
plt.text(105,33 , 'China', size = 6,
         horizontalalignment='center', color='red',
         bbox=dict(facecolor="grey",alpha=0.2),
         transform=ccrs.PlateCarree())
#title
plt.title('Global mean surface pressure')
```

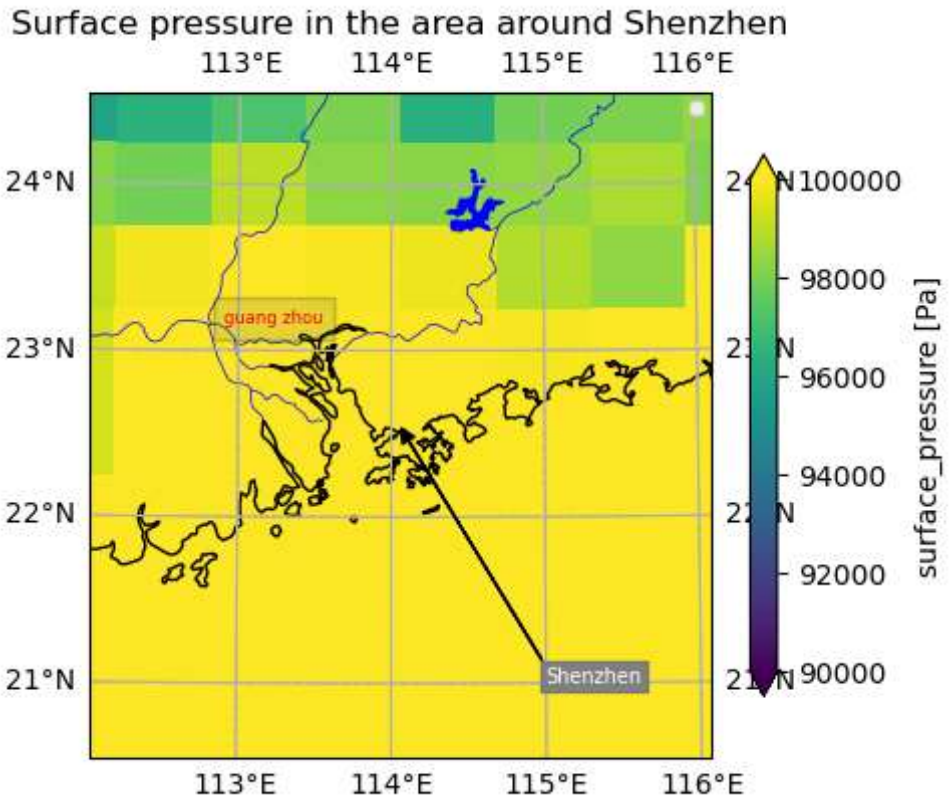
Out[5]: Text(0.5, 1.0, 'Global mean surface pressure')



In [4]:

```
rivers_10m = cfeature.NaturalEarthFeature('physical', 'rivers_lake_centerlines',

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

# Set Orthographic projection style
central_lon, central_lat = 114.06, 22.54 # Shenzhen

#project
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-2, central_lon+2, central_lat-2, central_lat+2]
ax.set_extent(extent)

# Add features to axes using cartopy.feature (cfeature)
ax.add_feature(cfeature.LAKES, edgecolor='blue', facecolor='blue', zorder=2)
ax.add_feature(rivers_10m, facecolor='None', edgecolor='blue', linewidth=0.5)

# masks or features
ax.coastlines(resolution='10m')

#gridlines
ax.gridlines()

#colorbar
PS = ds3.PS.groupby('time.month').mean(dim=['time'])
PS.plot(ax=ax, transform=ccrs.PlateCarree(),vmin=90000, vmax=100000,cbar_kwargs=

#x label and ticks, y label and ticks
gl = ax.gridlines(draw_labels=True, linestyle='--')
gl.xlocator = cticker.LongitudeLocator()
gl.ylocator = cticker.LatitudeLocator()
gl.xformatter = cticker.LongitudeFormatter()
gl.yformatter = cticker.LatitudeFormatter()

#title
plt.title('Surface pressure in the area around Shenzhen')

#annotations
ax.annotate('Shenzhen',xy=(114.06,22.54),xytext=(115,21),   # 深圳
            bbox=dict(boxstyle='square', fc='grey',linewidth=0.1),
            arrowprops=dict(facecolor='black', width=0.01,headwidth=5, headlengt
            fontsize=7, color='white', horizontalalignment='left',
            transform=ccrs.PlateCarree())
#text
plt.text(113.23,23.16 , 'guang zhou', size = 6,
         horizontalalignment='center', color='red',
         bbox=dict(facecolor="grey",alpha=0.2),
         transform=ccrs.PlateCarree())
#legend不知道为什么legend加的很小不显示内容
plt.legend(['surface pressure'],loc='best',fontsize=6)
```

Out[4]: <matplotlib.legend.Legend at 0x1a204d17850>


Surface pressure in the area around Shenzhen
113°E
114°E
115°E
116°E
24°N
23°N
22°N
21°N
guang zhou
Shenzhen
100000
98000
96000
94000
92000
90000
surface_pressure [Pa]