HW 4

In [2]:

```
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. [10 points]

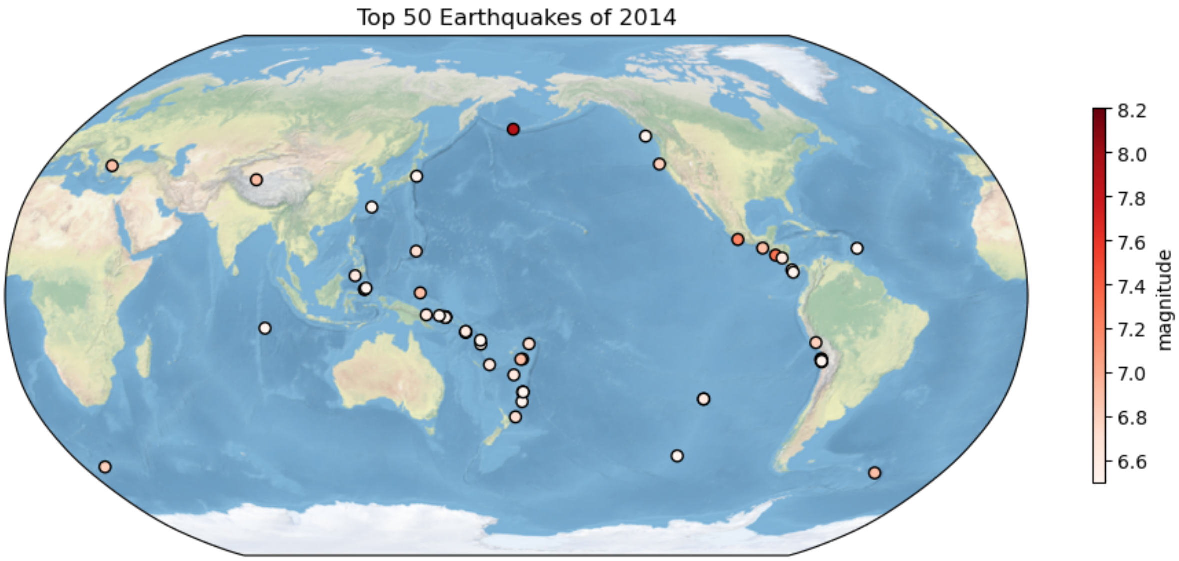
In [2]:

```
ds1 =pd.read_csv('usgs_earthquakes.csv')
# find the 50th mag
mag_50 =ds1.where(ds1.type =='earthquake').sort_values('mag',ascending=False).head(
# select data which is larger than 50th mag
ds1_top50 =ds1.loc[ds1.mag >= mag_50].where(ds1.type =='earthquake').sort_values('m
print(ds1_top50['mag'])
```

```
0     8.2
1     7.9
2     7.7
3     7.6
4     7.5
5     7.4
6     7.3
7     7.2
8     7.1
9     7.1
10    7.1
11    7.0
12    6.9
13    6.9
14    6.9
15    6.9
16    6.9
17    6.9
18    6.9
19    6.9
20    6.8
21    6.8
22    6.8
23    6.8
24    6.7
25    6.7
26    6.7
27    6.7
28    6.7
29    6.6
30    6.6
31    6.6
32    6.6
33    6.6
34    6.6
35    6.6
36    6.6
37    6.6
38    6.6
39    6.5
40    6.5
41    6.5
42    6.5
43    6.5
44    6.5
45    6.5
46    6.5
47    6.5
48    6.5
49    6.5
50    6.5
51    6.5
52    6.5
53    6.5
Name: mag, dtype: float64
```

In [3]:

```
plt.figure(figsize=(12,12), dpi=100)

# Create an axes with an Robinson projection style
proj = ccrs.Robinson(180)
ax = plt.axes(projection=proj)
ax.stock_img()

# adding data
ny_lon, ny_lat = -75, 43
delhi_lon, delhi_lat = 77.23, 28.61

plt.scatter('longitude', 'latitude',data= ds1_top50,
            transform=ccrs.PlateCarree(),
            c= 'mag',
            cmap='Reds',
            edgecolors='black'
         )

plt.title('Top 50 Earthquakes of 2014',fontsize =12)
plt.colorbar(shrink=0.3,label ='magnitude',format ="%1.1f",aspect =30)
plt.show()
```



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.

In [3]:

```
ds2 =xr.open_dataset('MERRA2_400.tavg1_2d_ocn_Nx.20220923.nc4')
ds2.info()
# open water skin temperature
ds2_tem =ds2.TSKINWTR.mean(dim='time')
# 10m wind speed
ds2_v10m =ds2.V10M.mean(dim ='time')
ds2_u10m =ds2.U10M.mean(dim ='time')
```

```
xarray.Dataset {
dimensions:
        lon = 576 ;
        lat = 361 ;
        time = 24 ;

variables:
        float64 lon(lon) ;
                lon:long_name = longitude ;
                lon:units = degrees_east ;
                lon:vmax = 999999986991104.0 ;
                lon:vmin = -999999986991104.0 ;
                lon:valid_range = [-1.e+15  1.e+15] ;
        float64 lat(lat) ;
                lat:long_name = latitude ;
                lat:units = degrees_north ;
                lat:vmax = 999999986991104.0 ;
                lat:vmin = -999999986991104.0 ;
                lat:valid_range = [-1.e+15  1.e+15] ;
        datetime64[ns] time(time) ;
                time:long_name = time ;
                time:time_increment = 10000 ;
                time:begin_date = 20220923 ;
                time:begin_time = 3000 ;
                time:vmax = 999999986991104.0 ;
                time:vmin = -999999986991104.0 ;
                time:valid_range = [-1.e+15  1.e+15] ;
        float32 EFLUXICE(time, lat, lon) ;
                EFLUXICE:long_name = sea_ice_latent_energy_flux ;
                EFLUXICE:units = W m-2 ;
                EFLUXICE:fmissing_value = 999999986991104.0 ;
                EFLUXICE:standard_name = sea_ice_latent_energy_flux ;
                EFLUXICE:vmax = 999999986991104.0 ;
                EFLUXICE:vmin = -999999986991104.0 ;
                EFLUXICE:valid_range = [-1.e+15  1.e+15] ;
        float32 EFLUXWTR(time, lat, lon) ;
                EFLUXWTR:long_name = open_water_latent_energy_flux ;
                EFLUXWTR:units = W m-2 ;
                EFLUXWTR:fmissing_value = 999999986991104.0 ;
                EFLUXWTR:standard_name = open_water_latent_energy_flux ;
                EFLUXWTR:vmax = 999999986991104.0 ;
                EFLUXWTR:vmin = -999999986991104.0 ;
                EFLUXWTR:valid_range = [-1.e+15  1.e+15] ;
        float32 FRSEAICE(time, lat, lon) ;
                FRSEAICE:long_name = ice_covered_fraction_of_tile ;
                FRSEAICE:units = 1 ;
                FRSEAICE:fmissing_value = 999999986991104.0 ;
                FRSEAICE:standard_name = ice_covered_fraction_of_tile ;
                FRSEAICE:vmax = 999999986991104.0 ;
                FRSEAICE:vmin = -999999986991104.0 ;
                FRSEAICE:valid_range = [-1.e+15  1.e+15] ;
        float32 HFLUXICE(time, lat, lon) ;
                HFLUXICE:long_name = sea_ice_upward_sensible_heat_flux ;
                HFLUXICE:units = W m-2 ;
                HFLUXICE:fmissing_value = 999999986991104.0 ;
                HFLUXICE:standard_name = sea_ice_upward_sensible_heat_flux ;
                HFLUXICE:vmax = 999999986991104.0 ;
                HFLUXICE:vmin = -999999986991104.0 ;
                HFLUXICE:valid_range = [-1.e+15  1.e+15] ;
        float32 HFLUXWTR(time, lat, lon) ;
                HFLUXWTR:long_name = open_water_upward_sensible_heat_flux ;
                HFLUXWTR:units = W m-2 ;
                HFLUXWTR:fmissing_value = 999999986991104.0 ;
                HFLUXWTR:standard_name = open_water_upward_sensible_heat_flux ;
                HFLUXWTR:vmax = 999999986991104.0 ;
                HFLUXWTR:vmin = -999999986991104.0 ;
                HFLUXWTR:valid_range = [-1.e+15  1.e+15] ;
        float32 LWGNTICE(time, lat, lon) ;
                LWGNTICE:long_name = sea_ice_net_downward_longwave_flux ;
                LWGNTICE:units = W m-2 ;
                LWGNTICE:fmissing_value = 999999986991104.0 ;
                LWGNTICE:standard_name = sea_ice_net_downward_longwave_flux ;
                LWGNTICE:vmax = 999999986991104.0 ;
                LWGNTICE:vmin = -999999986991104.0 ;
                LWGNTICE:valid_range = [-1.e+15  1.e+15] ;
        float32 LWGNTWTR(time, lat, lon) ;
                LWGNTWTR:long_name = open_water_net_downward_longwave_flux ;
                LWGNTWTR:units = W m-2 ;
                LWGNTWTR:fmissing_value = 999999986991104.0 ;
                LWGNTWTR:standard_name = open_water_net_downward_longwave_flux ;
                LWGNTWTR:vmax = 999999986991104.0 ;
                LWGNTWTR:vmin = -999999986991104.0 ;
                LWGNTWTR:valid_range = [-1.e+15  1.e+15] ;
        float32 PRECSNOOCN(time, lat, lon) ;
                PRECSNOOCN:long_name = ocean_snowfall ;
                PRECSNOOCN:units = kg m-2 s-1 ;
                PRECSNOOCN:fmissing_value = 999999986991104.0 ;
                PRECSNOOCN:standard_name = ocean_snowfall ;
                PRECSNOOCN:vmax = 999999986991104.0 ;
                PRECSNOOCN:vmin = -999999986991104.0 ;
                PRECSNOOCN:valid_range = [-1.e+15  1.e+15] ;
        float32 QV10M(time, lat, lon) ;
                QV10M:long_name = 10-meter_specific_humidity ;
                QV10M:units = kg kg-1 ;
                QV10M:fmissing_value = 999999986991104.0 ;
                QV10M:standard_name = 10-meter_specific_humidity ;
                QV10M:vmax = 999999986991104.0 ;
                QV10M:vmin = -999999986991104.0 ;
                QV10M:valid_range = [-1.e+15  1.e+15] ;
        float32 RAINOCN(time, lat, lon) ;
                RAINOCN:long_name = ocean_rainfall ;
                RAINOCN:units = kg m-2 s-1 ;
                RAINOCN:fmissing_value = 999999986991104.0 ;
                RAINOCN:standard_name = ocean_rainfall ;
                RAINOCN:vmax = 999999986991104.0 ;
                RAINOCN:vmin = -999999986991104.0 ;
                RAINOCN:valid_range = [-1.e+15  1.e+15] ;
        float32 SWGNTICE(time, lat, lon) ;
                SWGNTICE:long_name = sea_ice_net_downward_shortwave_flux ;
                SWGNTICE:units = W m-2 ;
                SWGNTICE:fmissing_value = 999999986991104.0 ;
                SWGNTICE:standard_name = sea_ice_net_downward_shortwave_flux ;
```

```
            SWGNTICE:vmax = 999999986991104.0 ;
            SWGNTICE:vmin = -999999986991104.0 ;
            SWGNTICE:valid_range = [-1.e+15  1.e+15] ;
    float32 SWGNTWTR(time, lat, lon) ;
            SWGNTWTR:long_name = open_water_net_downward_shortwave_flux ;
            SWGNTWTR:units = W m-2 ;
            SWGNTWTR:fmissing_value = 999999986991104.0 ;
            SWGNTWTR:standard_name = open_water_net_downward_shortwave_flux ;
            SWGNTWTR:vmax = 999999986991104.0 ;
            SWGNTWTR:vmin = -999999986991104.0 ;
            SWGNTWTR:valid_range = [-1.e+15  1.e+15] ;
    float32 T10M(time, lat, lon) ;
            T10M:long_name = 10-meter_air_temperature ;
            T10M:units = K ;
            T10M:fmissing_value = 999999986991104.0 ;
            T10M:standard_name = 10-meter_air_temperature ;
            T10M:vmax = 999999986991104.0 ;
            T10M:vmin = -999999986991104.0 ;
            T10M:valid_range = [-1.e+15  1.e+15] ;
    float32 TAUXICE(time, lat, lon) ;
            TAUXICE:long_name = eastward_stress_over_ice ;
            TAUXICE:units = N m-2 ;
            TAUXICE:fmissing_value = 999999986991104.0 ;
            TAUXICE:standard_name = eastward_stress_over_ice ;
            TAUXICE:vmax = 999999986991104.0 ;
            TAUXICE:vmin = -999999986991104.0 ;
            TAUXICE:valid_range = [-1.e+15  1.e+15] ;
    float32 TAUXWTR(time, lat, lon) ;
            TAUXWTR:long_name = eastward_stress_over_water ;
            TAUXWTR:units = N m-2 ;
            TAUXWTR:fmissing_value = 999999986991104.0 ;
            TAUXWTR:standard_name = eastward_stress_over_water ;
            TAUXWTR:vmax = 999999986991104.0 ;
            TAUXWTR:vmin = -999999986991104.0 ;
            TAUXWTR:valid_range = [-1.e+15  1.e+15] ;
    float32 TAUYICE(time, lat, lon) ;
            TAUYICE:long_name = northward_stress_over_ice ;
            TAUYICE:units = N m-2 ;
            TAUYICE:fmissing_value = 999999986991104.0 ;
            TAUYICE:standard_name = northward_stress_over_ice ;
            TAUYICE:vmax = 999999986991104.0 ;
            TAUYICE:vmin = -999999986991104.0 ;
            TAUYICE:valid_range = [-1.e+15  1.e+15] ;
    float32 TAUYWTR(time, lat, lon) ;
            TAUYWTR:long_name = northward_stress_over_water ;
            TAUYWTR:units = N m-2 ;
            TAUYWTR:fmissing_value = 999999986991104.0 ;
            TAUYWTR:standard_name = northward_stress_over_water ;
            TAUYWTR:vmax = 999999986991104.0 ;
            TAUYWTR:vmin = -999999986991104.0 ;
            TAUYWTR:valid_range = [-1.e+15  1.e+15] ;
    float32 TSKINICE(time, lat, lon) ;
            TSKINICE:long_name = sea_ice_skin_temperature ;
            TSKINICE:units = K ;
            TSKINICE:fmissing_value = 999999986991104.0 ;
            TSKINICE:standard_name = sea_ice_skin_temperature ;
```

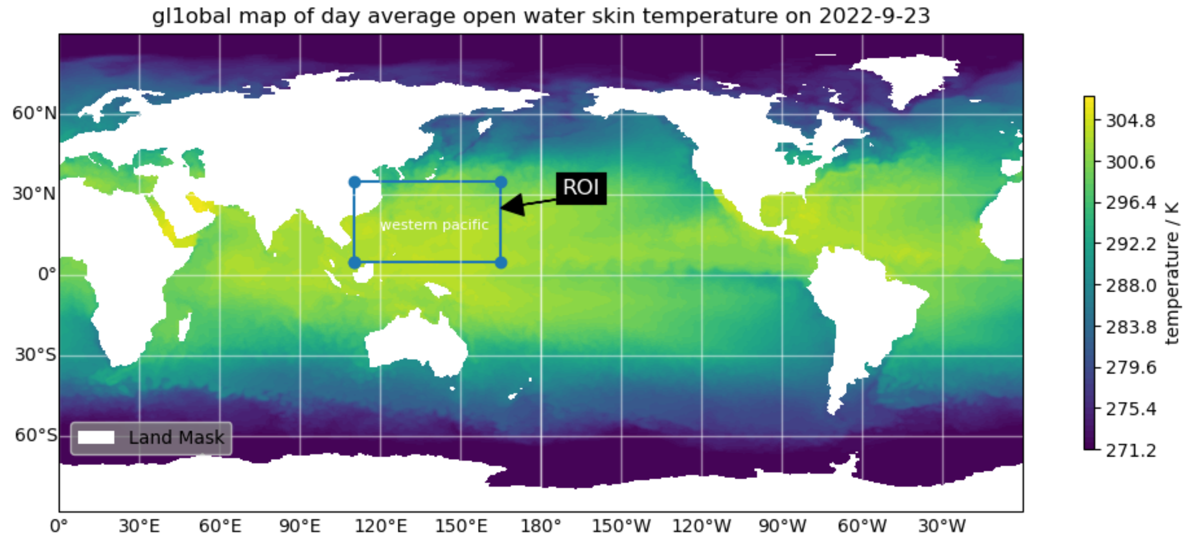
```
                TSKINICE:vmax = 999999986991104.0 ;
                TSKINICE:vmin = -999999986991104.0 ;
                TSKINICE:valid_range = [-1.e+15  1.e+15] ;
        float32 TSKINWTR(time, lat, lon) ;
                TSKINWTR:long_name = open_water_skin_temperature ;
                TSKINWTR:units = K ;
                TSKINWTR:fmissing_value = 999999986991104.0 ;
                TSKINWTR:standard_name = open_water_skin_temperature ;
                TSKINWTR:vmax = 999999986991104.0 ;
                TSKINWTR:vmin = -999999986991104.0 ;
                TSKINWTR:valid_range = [-1.e+15  1.e+15] ;
        float32 U10M(time, lat, lon) ;
                U10M:long_name = 10-meter_eastward_wind ;
                U10M:units = m s-1 ;
                U10M:fmissing_value = 999999986991104.0 ;
                U10M:standard_name = 10-meter_eastward_wind ;
                U10M:vmax = 999999986991104.0 ;
                U10M:vmin = -999999986991104.0 ;
                U10M:valid_range = [-1.e+15  1.e+15] ;
        float32 V10M(time, lat, lon) ;
                V10M:long_name = 10-meter_northward_wind ;
                V10M:units = m s-1 ;
                V10M:fmissing_value = 999999986991104.0 ;
                V10M:standard_name = 10-meter_northward_wind ;
                V10M:vmax = 999999986991104.0 ;
                V10M:vmin = -999999986991104.0 ;
                V10M:valid_range = [-1.e+15  1.e+15] ;

// global attributes:
        :History = Original file generated: Mon Oct  3 15:25:36 2022 GMT ;
        :Comment = GMAO filename: d5124_m2_jan10.tavg1_2d_ocn_Nx.20220923.nc4 ;
        :Filename = MERRA2_400.tavg1_2d_ocn_Nx.20220923.nc4 ;
        :Conventions = CF-1 ;
        :Institution = NASA Global Modeling and Assimilation Office ;
        :References = http://gmao.gsfc.nasa.gov ;
        :Format = NetCDF-4/HDF-5 ;
        :SpatialCoverage = global ;
        :VersionID = 5.12.4 ;
        :TemporalRange = 1980-01-01 -> 2016-12-31 ;
        :identifier_product_doi_authority = http://dx.doi.org/ ;
        :ShortName = M2T1NXOCN ;
        :GranuleID = MERRA2_400.tavg1_2d_ocn_Nx.20220923.nc4 ;
        :ProductionDateTime = Original file generated: Mon Oct  3 15:25:36 2022 GM
T ;
        :LongName = MERRA2 tavg1_2d_ocn_Nx: 2d,1-Hourly,Time-Averaged,Single-Leve
l,Assimilation,Ocean Surface Diagnostics ;
        :Title = MERRA2 tavg1_2d_ocn_Nx: 2d,1-Hourly,Time-Averaged,Single-Level,As
similation,Ocean Surface Diagnostics ;
        :SouthernmostLatitude = -90.0 ;
        :NorthernmostLatitude = 90.0 ;
        :WesternmostLongitude = -180.0 ;
        :EasternmostLongitude = 179.375 ;
        :LatitudeResolution = 0.5 ;
        :LongitudeResolution = 0.625 ;
        :DataResolution = 0.5 x 0.625 ;
        :Source = CVS tag: GEOSadas-5_12_4_p35_SLES12_M2-OPS ;
```

```
            :Contact = http://gmao.gsfc.nasa.gov ;
            :identifier_product_doi = 10.5067/Y67YQ1L3ZZ4R ;
            :RangeBeginningDate = 2022-09-23 ;
            :RangeBeginningTime = 00:00:00.000000 ;
            :RangeEndingDate = 2022-09-23 ;
            :RangeEndingTime = 23:59:59.000000 ;
}
```

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).

In [9]:

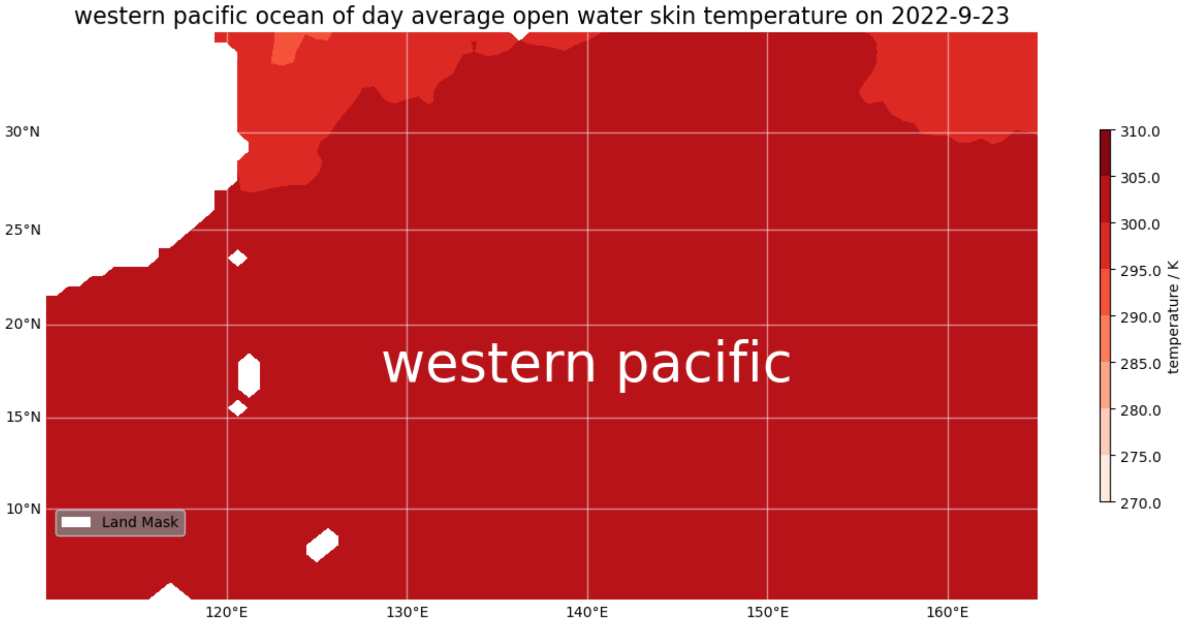
```
plt.figure(figsize=(12,12), dpi=100)

# Create an axes with an PlateCarree style
proj1 =ccrs.PlateCarree(180)
ax1 = plt.axes(projection=proj1)
ax1.add_feature(cfeature.LAND, edgecolor='white', facecolor='white', zorder=0)

# adding data
plt.contourf(ds2_tem.lon, ds2_tem.lat, ds2_tem.values, 60,
             transform=ccrs.PlateCarree())
plt.colorbar(shrink=0.3,label ='temperature / K',format ="%1.1f",aspect =35)

# Add lat/lon gridlines, draw gridlines
gl1 = ax1.gridlines(linewidth=1, color='white', alpha=0.5, draw_labels=True)
# Manipulate latitude and longitude gridline numbers and spacing
gl1.ylocator = mticker.FixedLocator(np.arange(-90,91,30))
gl1.xlocator = mticker.FixedLocator(np.arange(-180, 181, 30))
gl1.top_labels = False
gl1.right_labels = False

# title
ax1.set_title('gl1obal map of day average open water skin temperature on 2022-9-23'

# annotations and text
box_x, box_y =[110,110,165,165,110],[5,35,35,5,5]
ax1.plot(box_x, box_y, marker='o', transform=ccrs.PlateCarree())
plt.annotate('ROI', xy=(-15, 25), xytext=(15, 30),
             bbox=dict(boxstyle='square', fc='black', linewidth=0.1),
             arrowprops=dict(facecolor='black', shrink=0.001, width=0.1),
             fontsize=12, color='white', horizontalalignment='center')
plt.text(140, 17, "western pacific", size =8,
             transform=ccrs.PlateCarree(), horizontalalignment='center', color='whi

# legend
direct_hit = mpatches.Rectangle((0, 0), 1, 1, facecolor="white")
labels = ['Land Mask']
ax1.legend([direct_hit], labels,
            loc='lower left', bbox_to_anchor=(0.0025,0.1), fancybox=True, facecolor
plt.show()
```



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).

In [16]:

```
fig =plt.figure(figsize=(12,12), dpi=100)

# Create an axes with an PlateCarree style
proj2 =ccrs.Orthographic(130)
ax2 =fig.add_axes([0, 0, 1, 1], projection=ccrs.Miller(),
                      frameon=False)
ax2.add_feature(cfeature.LAND, edgecolor='white', facecolor='white', zorder=0)

# adding data
plt.contourf(ds2_tem.lon, ds2_tem.lat, ds2_tem.values,cmap ='Reds',
             transform=ccrs.PlateCarree(),zorder=1)

# colorbar
plt.colorbar(shrink=0.3,label ='temperature / K',format ="%1.1f",aspect =35)

# region
extent = [110,165, 5,35]
ax2.set_extent(extent, ccrs.PlateCarree())

# title
ax2.set_title('western pacific ocean of day average open water skin temperature on
ax2.set_xlabel('lat')

# grid
gl2 = ax2.gridlines(linewidth=1, color='white', alpha=0.5, draw_labels=True)
gl2.top_labels = False
gl2.right_labels = False

# annotations and text
plt.annotate('CHINA', xy=(-2000000,2500000), xytext=(-1000000,1300000),
             bbox=dict(boxstyle='square', fc='black', linewidth=0.1),
             arrowprops=dict(facecolor='black', shrink=0.001, width=0.1),
             fontsize=12, color='white')
plt.text(140, 17, "western pacific", size =38,
             transform=ccrs.PlateCarree(), horizontalalignment='center', color='whi

# legend
ax2.legend([mpatches.Rectangle((0, 0), 1, 1, facecolor="white")], ['Land Mask'],
            loc='lower left', bbox_to_anchor=(0.0025,0.1), fancybox=True, facecolor
plt.show()
```


western pacific ocean of day average open water skin temperature on 2022-9-23
western pacific
Land Mask
30°N
25°N
20°N
15°N
10°N
120°E
130°E
140°E
150°E
160°E
310.0
305.0
300.0
295.0
290.0
285.0
280.0
275.0
270.0
temperature / K