

Homework #4

Name: 温承彦

SID: 12332279

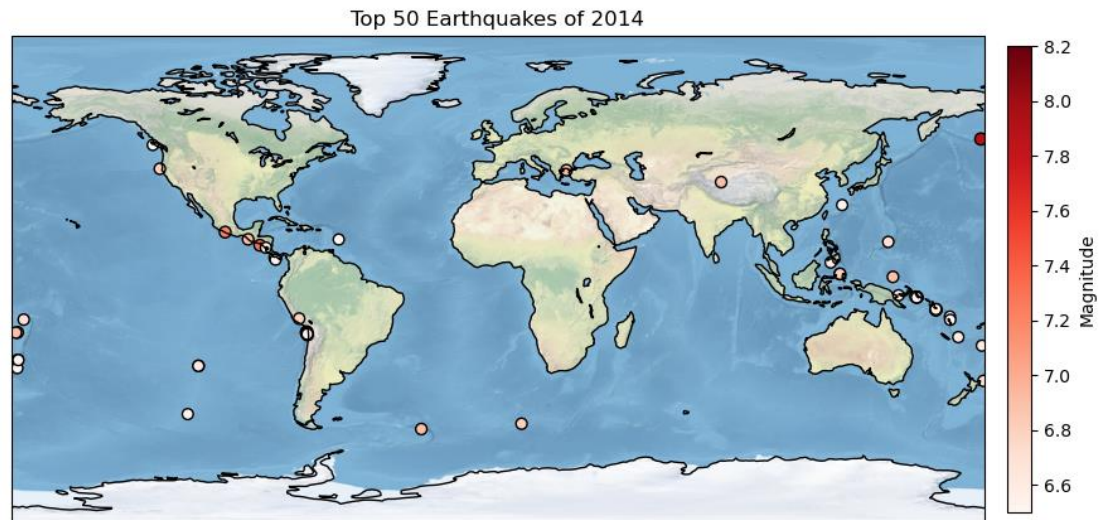
Problem 1: In this problem set, we will use this file from the USGS Earthquakes Database. The dataset is similar to the one you use in Assignment 02. Use the file provided (usgs_earthquakes.csv) to recreate the following map. Use the mag column for magnitude. [10 points]

Answer: First, import the necessary libraries, including **pandas**, **xarray**, **matplotlib**, and **cartopy**. Then, use the `pd.read_csv` function to read the earthquake dataset named "**usgs_earthquakes.csv**" and store it in a DataFrame called **Sig_Eqs**. Utilize the `pd.to_datetime` function to convert the "time" column in **Sig_Eqs** to a datetime format. Apply a time condition to filter earthquake data from the dataset for the year 2014, creating a DataFrame named **Sig_Eqs_2014**. Sort **Sig_Eqs_2014** based on earthquake magnitude using the `nlargest` function, and select the top 50 earthquake events to create a DataFrame named **top_50_eqs**.

Next, create a plot using the `plt.subplots` function to generate a canvas and coordinate axes. Set the projection of the coordinate axes to `PlateCarree`, add a world map background using the `ax.stock_img` function, and incorporate map features such as coastlines, land, and lakes using the `ax.add_feature` function.

Plot earthquake points on the map using the `ax.scatter` function, providing longitude, latitude, and magnitude as parameters. Set attributes

such as color, marker, size, and transparency. Finally, add additional elements to the plot, including a color bar, set the plot title, and display the graph to complete the drawing. The final graph is as follows:



Problem 2: 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.

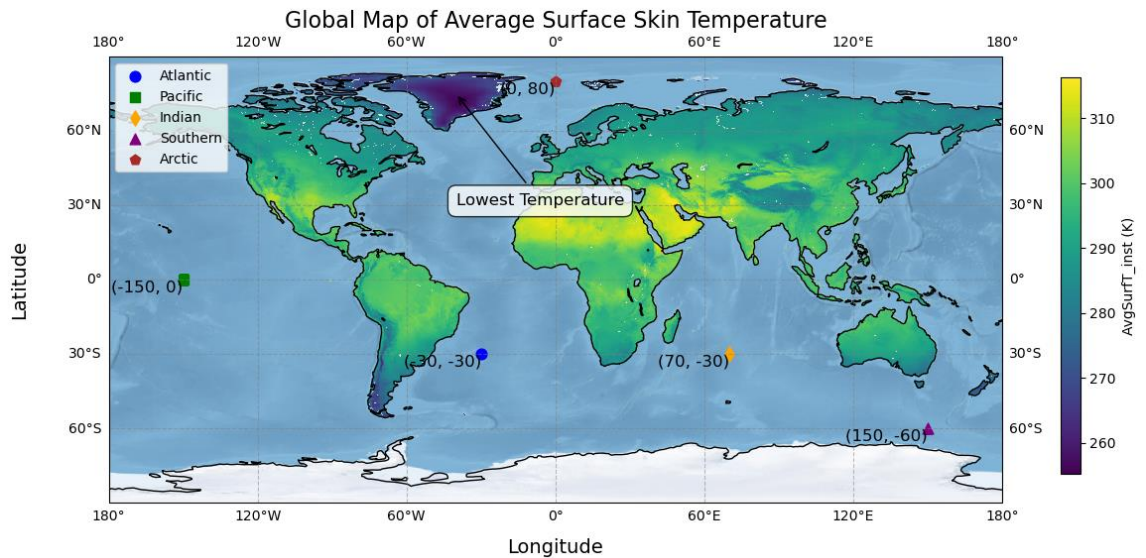
2.1 [10 points] Make a global map of a certain variable. Your figure should contain: a project, x label and ticks, y label and ticks, title, gridlines, legend, colorbar, masks or features, annotations, and text box (**1 point each**).

Answer: Firstly, we import the necessary libraries, including **xarray**, **matplotlib**, and **cartopy**. Next, we use the `xr.open_dataset()` function to read the netCDF file named "**GLDAS_NOAH025_M.A202308.021.nc**"

and store the dataset in the variable **ds**. Then, we extract the surface mean temperature variable using the **ds[variable_name].squeeze()** function and store the result in the variable 'variable'. Afterward, we create a canvas and axes using the **plt.subplots()** function, set the projection of the axes to PlateCarree, and specify the size of the figure. We add a world map background using the **ax.stock_img()** function.

Moving on, we plot the global surface mean temperature on the map using the **ax.pcolormesh()** function, passing longitude, latitude, and surface mean temperature as parameters, and setting the color map. We add map features, including coastlines, land, and lakes, using the **ax.add_feature()** function. Next, we label the five oceans and add a legend. We use a loop to iterate through the oceans in the dictionary 'oceans,' including their corresponding longitude, latitude, color, marker, and size. We use the **ax.scatter()** and **ax.text()** functions for labeling and annotation.

Subsequently, we add x and y labels, ticks, and a title using the **ax.set_xlabel()**, **ax.set_ylabel()**, **ax.text()**, and **plt.title()** functions. We add gridlines, a legend, and a color bar using the **ax.gridlines()**, **ax.legend()**, and **plt.colorbar()** functions, setting the appropriate parameters. Finally, we add annotations and a text box to the figure using the **plt.annotate()** function, specifying attributes such as arrows, font size, and box style. Lastly, the resulting plot was displayed using **plt.show()** as shown below:



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**).

Answer: First, the code begins by importing the required libraries, including `xarray`, `matplotlib`, and `cartopy`. Then, the `xr.open_dataset()` function is used to read the netCDF file named "**GLDAS_NOAH025_M.A202308.021.nc**," and the dataset is stored in the variable **ds**. Next, the surface average temperature variable is extracted from the dataset, the range of the Chinese region is defined, and data for the Chinese region is extracted using the `sel` function.

Afterward, a canvas and axes are created using the `plt.subplots()` function, and the projection of the axes is set to `PlateCarree`, specifying the size of the figure. The surface average temperature of the Chinese region is plotted on the map using the `ax.pcolormesh()` function, with longitude,

latitude, and surface average temperature as parameters, and a color map is set. Map features such as coastlines, land, and lakes are added using the `ax.add_feature()` function.

Next, a loop iterates through the dictionary of cities and their corresponding latitude and longitude information. The `ax.plot()` function is used to mark different cities on the map, and a legend is added. Labels for x and y axes, ticks, and a title are added using `ax.set_xlabel()`, `ax.set_ylabel()`, `ax.text()`, and `plt.title()` functions. Subsequently, gridlines are added with the `ax.gridlines()` function, a legend is added with the `ax.legend()` function, and a color bar is added using the `plt.colorbar()` function with appropriate parameters. Finally, annotations and text boxes are added to the figure, possibly using the `plt.annotate()` function, with settings for arrows, font size, and text box style. Ultimately, the `plt.show()` function is used to display the graph, and the result is as follows:

