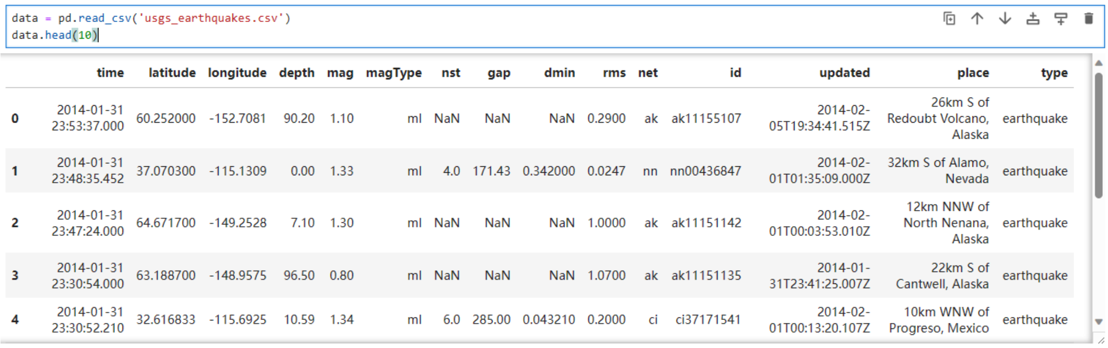
Assignment 04

YuGuobin 12332284

1. Global Earthquakes

1.1. Use the file provided (usgs_earthquakes.csv) to recreate the following map. Use the mag column for magnitude. **[10 points]**

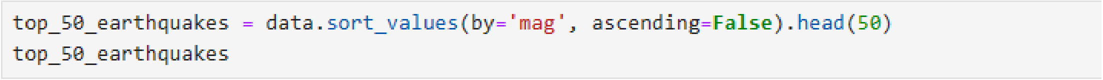
a) Read the earthquakes data and name it “data”.



```
data = pd.read_csv('usgs_earthquakes.csv')
data.head(10)
```

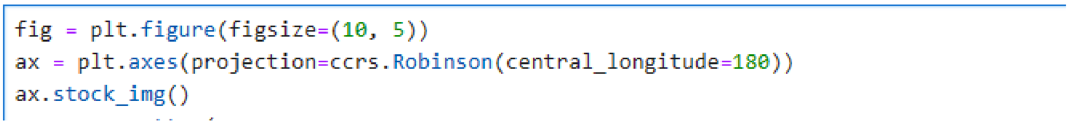
	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net	id	updated	place	type
0	2014-01-31 23:53:37.000	60.252000	-152.7081	90.20	1.10	ml	NaN	NaN	NaN	0.2900	ak	ak11155107	2014-02-05T19:34:41.515Z	26km S of Redoubt Volcano, Alaska	earthquake
1	2014-01-31 23:48:35.452	37.070300	-115.1309	0.00	1.33	ml	4.0	171.43	0.342000	0.0247	nn	nn00436847	2014-02-01T01:35:09.000Z	32km S of Alamo, Nevada	earthquake
2	2014-01-31 23:47:24.000	64.671700	-149.2528	7.10	1.30	ml	NaN	NaN	NaN	1.0000	ak	ak11151142	2014-02-01T00:03:53.010Z	12km NNW of North Nenana, Alaska	earthquake
3	2014-01-31 23:30:54.000	63.188700	-148.9575	96.50	0.80	ml	NaN	NaN	NaN	1.0700	ak	ak11151135	2014-01-31T23:41:25.007Z	22km S of Cantwell, Alaska	earthquake
4	2014-01-31 23:30:52.210	32.616833	-115.6925	10.59	1.34	ml	6.0	285.00	0.043210	0.2000	ci	ci37171541	2014-02-01T00:13:20.107Z	10km WNW of Progreso, Mexico	earthquake

b) Use the "sort_values" function to sort the "mag" from largest to smallest and select the first 50 data.



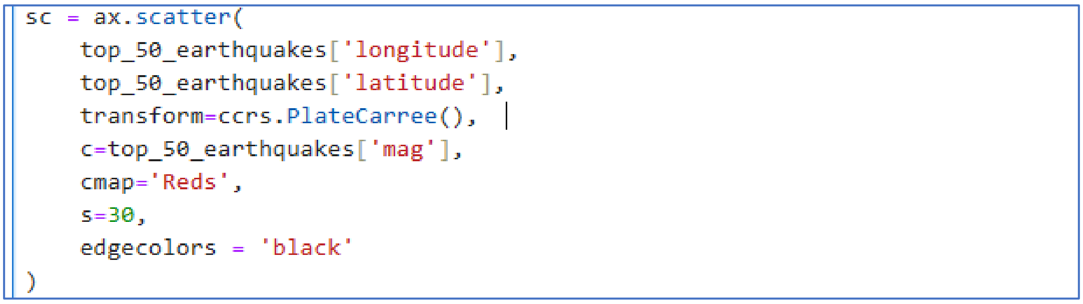
```
top_50_earthquakes = data.sort_values(by='mag', ascending=False).head(50)
top_50_earthquakes
```

c) Create a figure named "fig", set the map projection to "Robinson" with a central longitude of 180, and use "ax.stock_img()" to set the earth background.



```
fig = plt.figure(figsize=(10, 5))
ax = plt.axes(projection=ccrs.Robinson(central_longitude=180))
ax.stock_img()
```

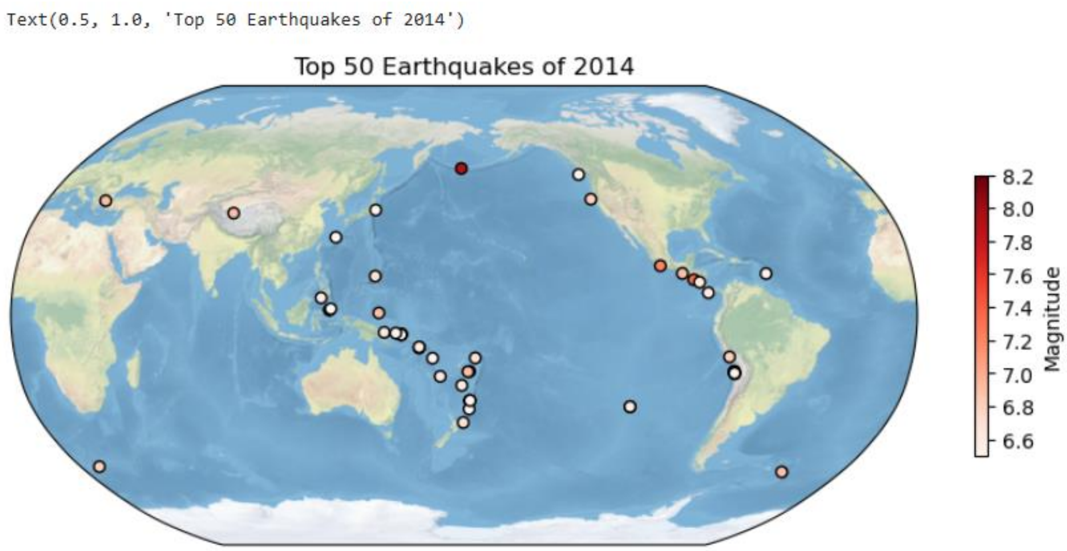
d) Use the function of “ax.scatter()” to draw points, set the x, y to the “latitude” and “longitude” of the “top_50_earthquakes”, set the geographic coordinate to "PlateCarree()", set the color parameter "c" to stretch according to the "mag", set the color bar "cmap" to "Reds", Set the size of the point "s" to 30 and the outer border "edgecolors" to "black".



```
sc = ax.scatter(
    top_50_earthquakes['longitude'],
    top_50_earthquakes['latitude'],
    transform=ccrs.PlateCarree(),
    c=top_50_earthquakes['mag'],
    cmap='Reds',
    s=30,
    edgecolors = 'black'
)
```

e) Add a "colorbar" to "fig" and set the title.

```
fig.colorbar(sc,ax=ax,location='right',anchor=(0,0.5),
shrink=0.5,label='Magnitude',ticks =[8.2,8.0,7.8,7.6,7.4,7.2,7.0,6.8,6.6])
ax.set_title('Top 50 Earthquakes of 2014')
```



```
Text(0.5, 1.0, 'Top 50 Earthquakes of 2014')
```

I got inspired by reading:

https://matplotlib.org/stable/users/explain/colors/colormaps.html#sphx-glr-users-explain-colors-colormaps-py

https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.colorbar.html#matplotlib.pyplot.colorbar,

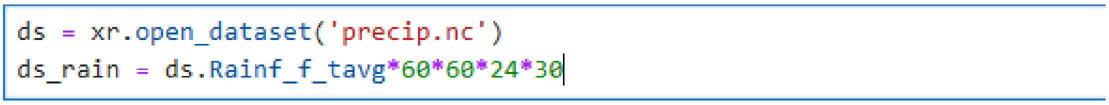
https://matplotlib.org/stable/api/_as_gen/matplotlib.axes.Axes.scatter.html#matplotlib.axes.Axes.scatter

https://scitools.org.uk/cartopy/docs/latest/gallery/lines_and_polygons/global_map.html#sphx-glr-gallery-lines-and-polygons-global-map-py

2. Explore a netCDF dataset

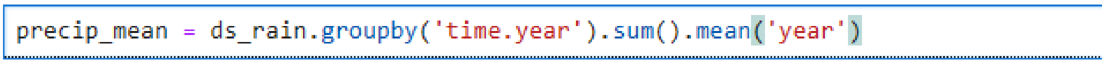
2.1. **[10 points]** Make a global map of a certain variable. Your figure should contain: a project, x label and ticks, y label and ticks, title, gridlines, legend, colorbar, masks or features, annotations, and text box (1 point each).

a) Read the “rainfall flux” data and named it “ds”, convert the unit of “rainfall flux” from "kg m-2 s-1" to "mm".



```
ds = xr.open_dataset('precip.nc')
ds_rain = ds.Rainf_f_tavg*60*60*24*30
```

b) Grouping the data according to year, and calculate the annual rainfall flux summation, then calculate the average of 10 years.



```
precip_mean = ds_rain.groupby('time.year').sum().mean('year')
```

Make Map:

a) Set the coordinate system to "PlateCarree".

```
fig = plt.figure(figsize=(12,8), dpi=120)
proj = ccrs.PlateCarree()
ax = plt.axes(projection=proj)
```

b) Use the "ax.gridlines ()" function to set the gridlines.

```
gl=ax.gridlines(crs=ccrs.PlateCarree(),
draw_labels=True,linewidth=0.5,color=’gray’,alpha=0.8,
zorder=3,linestyle=’--’)
```

c) Use the "mticker.FixedLocator ()" function to redefine the axis labels, removing the top and right labels.

```
gl.ylocator = mticker.FixedLocator(np.arange(-60,91,30))
gl.xlocator = mticker.FixedLocator([-135,-90, -45, 0, 45, 90,135])
gl.top_labels = False
gl.right_labels = False
```

d) plot the data using the "plot ()" method.

```
sc=precip_mean.plot(ax=ax,vmax=100,vmin=-
100 ,transform=ccrs.PlateCarree(),cmap=’RdBu’,add_colorbar=False,z
order=1)
```

e) Use the "fig.colorbar()" function to add a colorbar.

```
cb = fig.colorbar(sc, ax=ax, location=’right’, anchor=(0, 0.5),
extend =’both’ , shrink=0.5,label=’rainfall_flux (mm)’)
```

f) Use the "ax.add_feature ()" function to add coastline elements.

```
ax.add_feature(cfeature.OCEAN,                    edgecolor=’black’,
facecolor=’white’,linewidths=0.5,zorder=2)
```

g) Use the "ax.set_title()" function to set the title.

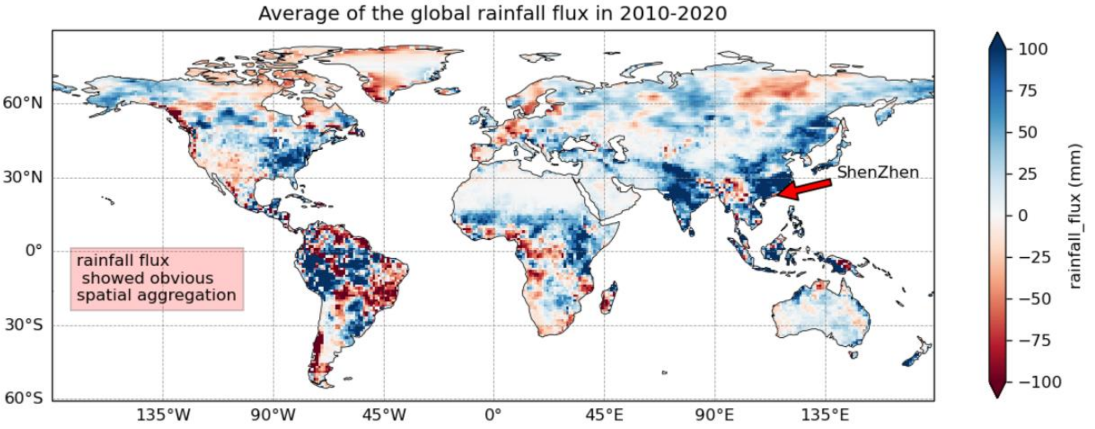
```
ax.set_title(’Average of the global rainfall flux in 2010-2020’)
```

h) Use the "ax.annotate ()" function to set the annotation to mark the location of Shenzhen.

```
ax.annotate(’ShenZhen’,  xy=(114.1,  22.5),  xycoords=’data’,
xytext=(140, 30),arrowprops=dict(facecolor=’red’, shrink=0.05))
```

i) Use the "ax.text()" function to add a text box at the specified location.

```
ax.text(-170,  -20,  ″rainfall  flux\n  showed  obvious  \nspatial
aggregation″, color=’black’,bbox=dict(facecolor=’red’, alpha=0.2))
```



I got inspired by reading:

https://scitools.org.uk/cartopy/docs/v0.13/matplotlib/gridliner.html

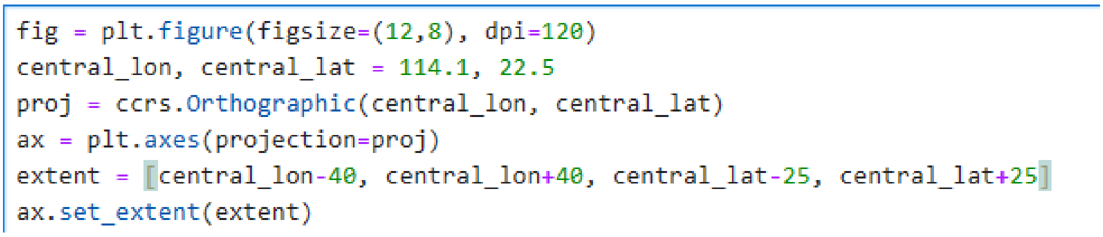
https://matplotlib.org/stable/api/_as_gen/matplotlib.axes.Axes.text.html#matplotlib.axes.Axes.text

https://matplotlib.org/stable/api/_as_gen/matplotlib.axes.Axes.annotate.html#matplotlib.axes.Axes.annotate

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 (1point each)**.**

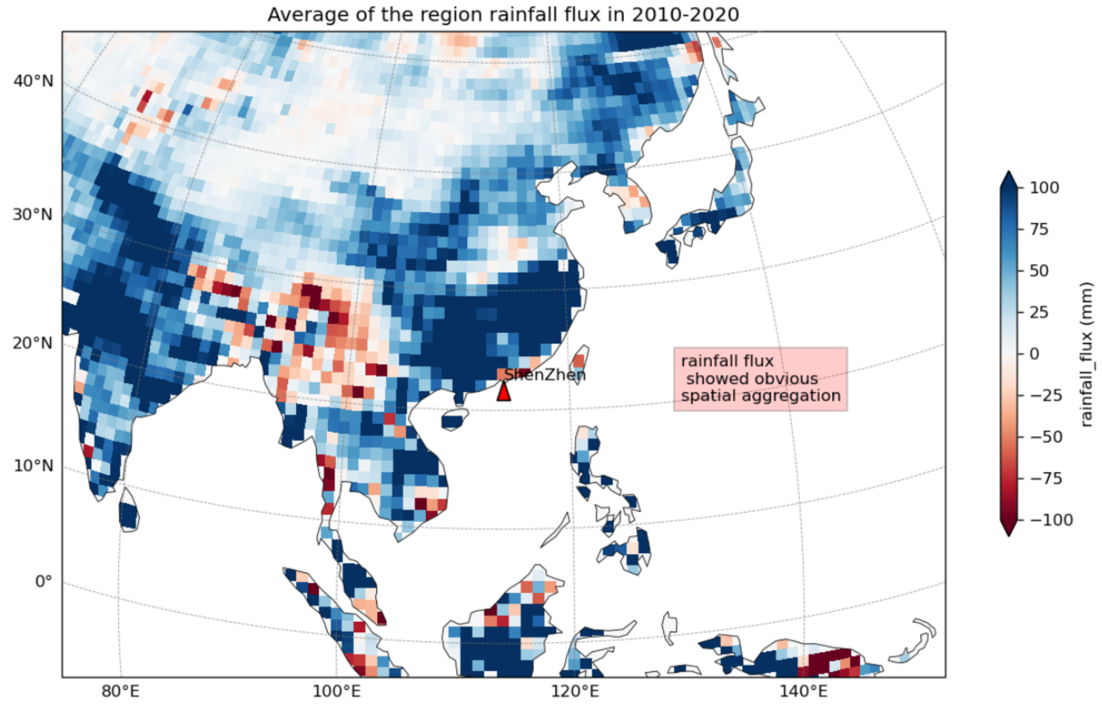
a) The longitude and latitude of Shenzhen are defined as the central longitude and latitude lines of the map.

b) Set the map projection to "Orthographic" and use the "ax.set_setent ()" function to set the scope of the map.



```
fig = plt.figure(figsize=(12,8), dpi=120)
central_lon, central_lat = 114.1, 22.5
proj = ccrs.Orthographic(central_lon, central_lat)
ax = plt.axes(projection=proj)
extent = [central_lon-40, central_lon+40, central_lat-25, central_lat+25]
ax.set_extent(extent)
```

c) The drawing of the remaining elements is similar to the method in 2.1.


Average of the region rainfall flux in 2010-2020
40°N
30°N
20°N
10°N
0°
80°E
100°E
120°E
140°E
ShenZhen
rainfall flux
showed obvious
spatial aggregation
100
75
50
25
0
−25
−50
−75
−100
rainfall_flux (mm)