Dimension = lat, long, time

How to subset Arrays

position

.isel( ) = select data w/ values or slices (integers)

.sel( ) = select coordinate label/value w/ values or slices

interp( ) = interpolate w/ coordinate labels

DATA VARIABLES (there are 2 in our data set)

tempanomaly (time, lat, lon)

time_bnds (time, nv)

GEOGRAPHY

[min lat, min lon, max lat, max lon]

U.S. -135 -60 20 55

ax.set_extent([-135, -60, 20, 55]) → this is used with creating a map

lat = slice(25, 50), lon = slice(-125, -67) → this is used with xarray, subselect array

HELPFUL CONTEXT (for code in repo "Data Transformation")

In order to plot a figure use matplot (Python library)
we need to first call on the data `ds`
then select a data variable `tempanomaly`
then subset from the array `.sel`

Next, we specify the coordinate values i.e. `lat` `time`
we can create plot types after this info is present
`.plot` will let us create 2D plots

GROUPBY

We can groupby specific keys, this could be useful for when we look at data by specific time
i.e. lets group by month (Januarys, Februarys, Marchs, etc.)

`.groupby` we can use an accessor which basically allows us to enable functionality `.dt`

`ds.tempanomaly.groupby(ds.time.dt.month)` (additional funct. of the month)

Helpful for this dataset → year
month
season (DecJanFeb) = 'DJF'

Output to NetCDF

After we've selected a subset of data, done any extra analysis, etc. we can output to a new file

1) convert to a dataset `.to_dataset()`
2) output to netcdf file `.to_netcdf()`

* can do it in 1 step

`name_of_array.to_dataset().to_netcdf('name.nc')`

SUMMARY

1) Open dataset xarray
2) Subset data, to get to a particular area of interest `.sel()`
3) Compute any extra analysis i.e. rolling average
4) Plot the data, create map, output as a new file (netcdf file)
5) Ready to use in ParaView

Troubleshoot Output

1) niño_34 = ds.sel(lat, lon)
   - ↳ getting specific location
   - ↳ in our data this is **anomaly_temp_USA**

2) gb = niño34.datavariable.groupby('time.month)
   - ↳ group by month
   - ↳ in our data this is **ga**

3) niño34_anom = gb - gb.mean(dim = 'time')
   - ↳ take data - average = anomalies
   - ↳ in our data this is **avg-us_anomaly**

4) index_niño34 = anom.weighted(Ale).mean(dim=[lat,lon])
   - ↳ this takes weighted data
   - ↳ not in our data because we don't have that available

5) index_niño34_rolling = index_niño34.rolling(time=5).mean
   - ↳ find the 5 month running avg