Updates on Climatology Calculation Support in the GeoCAT Ecosystem

Transition from NCL to Python

Heather R. Craker,

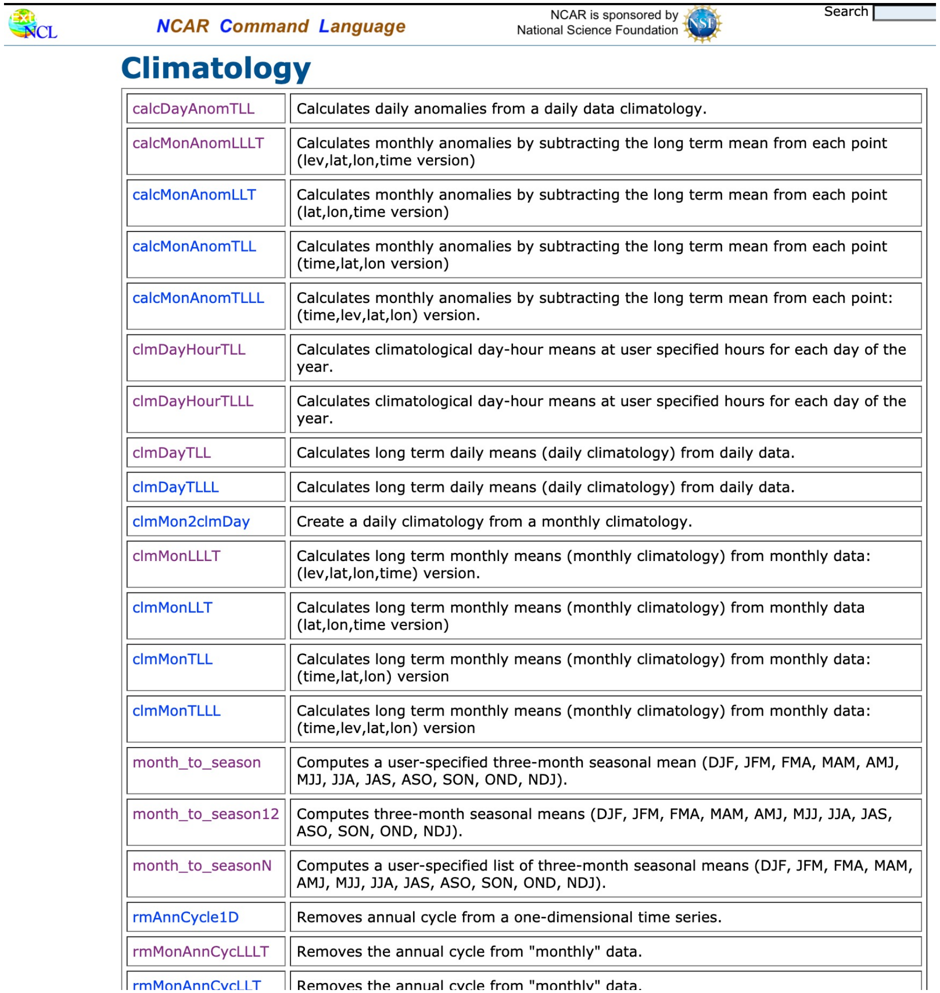
SIParCS/GeoCAT Intern

July 12, 2021

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

NCL Climatology Functions

- Climatology Calculations
  - Daily, monthly, seasonally climatological averages
- Anomaly Calculations
- Removing Annual Cycles
- Monthly Standard Deviation Calculations



NCL NCAR Command Language

NCAR is sponsored by National Science Foundation

Search

Climatology

calcDayAnomTLL	Calculates daily anomalies from a daily data climatology.
calcMonAnomLLLT	Calculates monthly anomalies by subtracting the long term mean from each point (lev,lat,lon,time version)
calcMonAnomLLT	Calculates monthly anomalies by subtracting the long term mean from each point (lat,lon,time version)
calcMonAnomTLL	Calculates monthly anomalies by subtracting the long term mean from each point (time,lat,lon version)
calcMonAnomTLLL	Calculates monthly anomalies by subtracting the long term mean from each point: (time,lev,lat,lon) version.
clmDayHourTLL	Calculates climatological day-hour means at user specified hours for each day of the year.
clmDayHourTLLL	Calculates climatological day-hour means at user specified hours for each day of the year.
clmDayTLL	Calculates long term daily means (daily climatology) from daily data.
clmDayTLLL	Calculates long term daily means (daily climatology) from daily data.
clmMon2clmDay	Create a daily climatology from a monthly climatology.
clmMonLLLT	Calculates long term monthly means (monthly climatology) from monthly data: (lev,lat,lon,time) version.
clmMonLLT	Calculates long term monthly means (monthly climatology) from monthly data (lat,lon,time version)
clmMonTLL	Calculates long term monthly means (monthly climatology) from monthly data: (time,lat,lon) version
clmMonTLLL	Calculates long term monthly means (monthly climatology) from monthly data: (time,lev,lat,lon) version
month_to_season	Computes a user-specified three-month seasonal mean (DJF, JFM, FMA, MAM, AMJ, MJJ, JJA, JAS, ASO, SON, OND, NDJ).
month_to_season12	Computes three-month seasonal means (DJF, JFM, FMA, MAM, AMJ, MJJ, JJA, JAS, ASO, SON, OND, NDJ).
month_to_seasonN	Computes a user-specified list of three-month seasonal means (DJF, JFM, FMA, MAM, AMJ, MJJ, JJA, JAS, ASO, SON, OND, NDJ).
rmAnnCycle1D	Removes annual cycle from a one-dimensional time series.
rmMonAnnCycLLLT	Removes the annual cycle from "monthly" data.
rmMonAnnCycLLT	Removes the annual cycle from "monthly" data.

NCL to Python Approach

Wrap the Fortran in GeoCAT-f2py

- Pros
  - Relatively quick to do
  - Preserves functionality exactly
- Cons
  - Doesn't allow parallelization
  - Relies on code base that is in maintenance mode
  - Not compatible with Windows OS

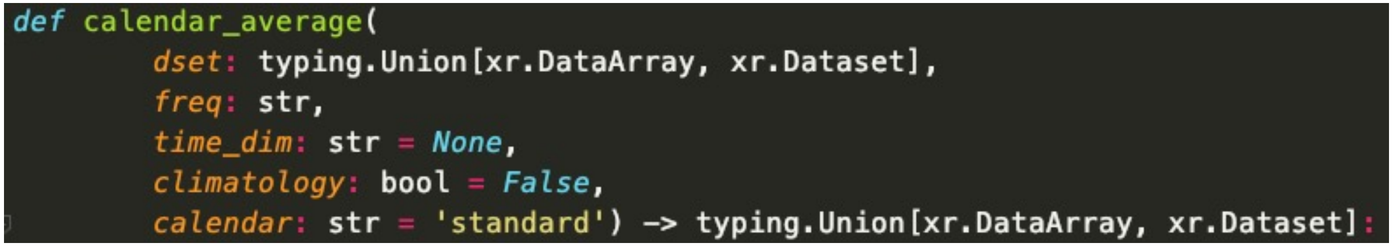
Reimplement in pure Python

- Pros
  - Can add Dask compatibility
  - Allows for additional improvements and features
  - In a code base that is in development mode
- Cons
  - Much more time needed

Implementation Process

1. Seeing what was already available in Python and GeoCAT
2. Asking users what they wanted to see
3. Deciding which functionality to implement first
   - Started with daily and moved to monthly and seasonally
4. Decide if any improvements/functionality needs to be added
   - Make function names more intuitive
   - Add non-standard calendar support
   - Make function coordinate order agnostic
5. Making a proof of concept
   - PR #158 open in GeoCAT-comp https://bit.ly/3rcxNs8
6. Getting user feed back (that's why I'm here today!)

Function Signature



```
def calendar_average(
        dset: typing.Union[xr.DataArray, xr.Dataset],
        freq: str,
        time_dim: str = None,
        climatology: bool = False,
        calendar: str = 'standard') -> typing.Union[xr.DataArray, xr.Dataset]:
```

- Freq: Frequency alias. (hour, day, month, season, year)
- Time_dim: the name of the time coordinate dimension
- Climatology: False = compute averages, True = compute climatologies
- Calendar: the name of the calendar (noleap, all_leap, 360_days, Gregorian, julian)

What we have so far

- Compatible with xarray.Dataset and xarray.DataArray
- Data can be aggregated into hourly, daily, monthly, seasonally, and yearly climatological means
  - Going from month to seasonal data is weighted by number of days in each month
- Sequential means are available
  - i.e. return the average high temp for each day in a multi year period
- Time coordinates are centered within the time bounds
  - Would love feed back on this explicitly

Jupyter Lab Demo

Demo Repository: https://github.com/hCraker/calendar_average_demo

What we are working on

- Support for non-standard calendars
  - Which ones would you all like to see supported?
- Refining how time coordinate is manipulated
  - i.e. should day/monthly means have time coordinate at start or middle of day/month?
- Add functionality for user to calculate averages for specific day-hours and seasons
  - i.e. only calculate summer and winter averages versus all four seasons

Thank you!

Have questions/suggestions?

- Ask them today!
- If they are longer, send me an email or Zulip message
  - hcraker@ucar.edu
- Look at the PR for this function to participate in discussion
  - PR #158 in GeoCAT-comp
  - https://bit.ly/3rcxNs8

Curious about overall project?

- See our website
  - https://geocat.ucar.edu/
- Look at our documentation page for GeoCAT-comp
  - https://geocat-comp.readthedocs.io/en/latest/
- Check out our GitHub
  - https://github.com/NCAR/geocat-comp

Thank you to Max Grover, Anderson Banihirwe, and Deepak Cherian!