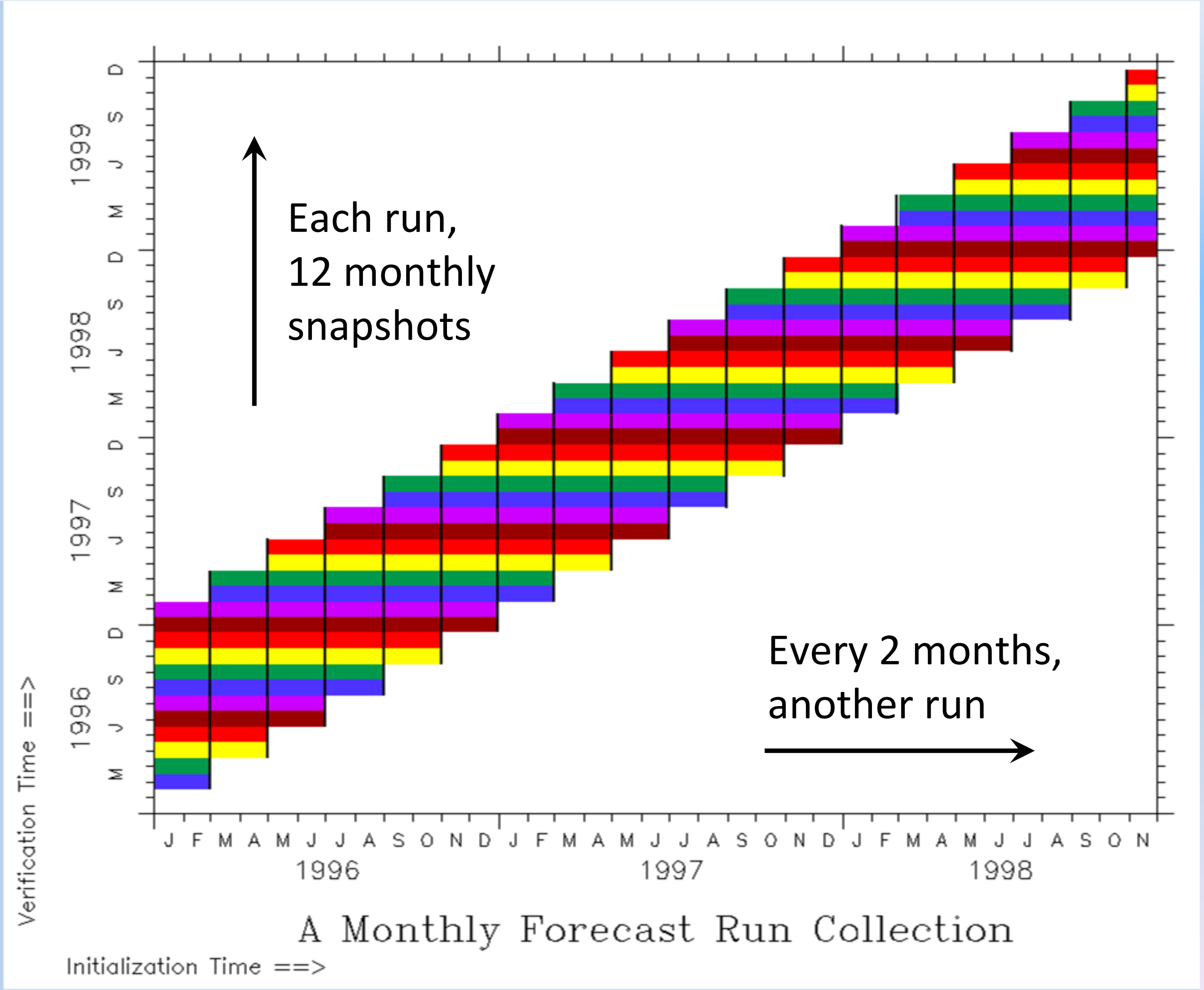
Working with Forecast Models in PyFerret and Ferret

June 2015

A plot to illustrate the data output by a Forecast Model Run Collection (FMRC). Every colored cell is a model snapshot.



Each model output (starting at one initialization time) is a "dataset". How to aggregate these many individual datasets into a single dataset of higher dimensionality?

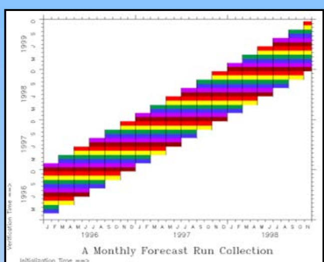
- use the Ferret command
  `DEFINE AGGREGATION/F`

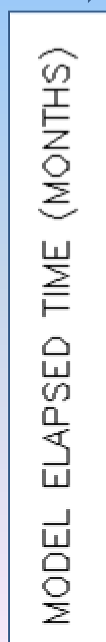
  or
- serve the collection via OPeNDAP using Unidata's THREDDS Data Server (TDS)

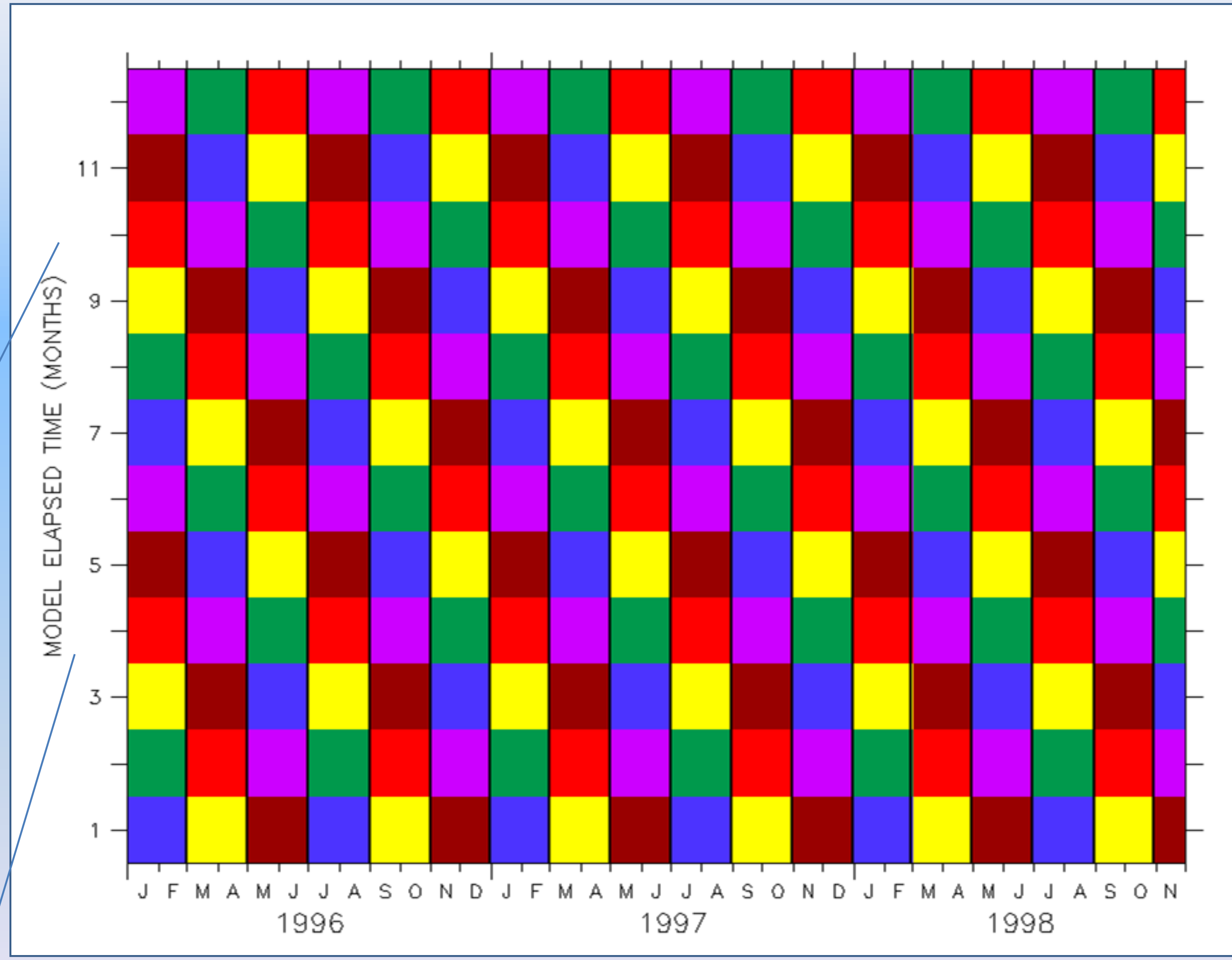
many 4D files ➔ become a single 5D dataset

Native ("compact") form of FMRC

as run …







Forecast Initialization Time

In the native, compact form the time step values are a 2D variable. Note that the same forecasted date/time occurs at lags in multiple forecast runs.

Forecast Series Axis →

Model Run Time Axis ↓

	RUN 1	RUN 2	RUN3	RUN4	RUN5	RUN6	RUN7
1	744.	2184.	3648.	5112.	6576.	8040.	9528.
2	1440.	2904.	4368.	5856.	7320.	8784.	10200.
3	2184.	3648.	5112.	6576.	8040.	9528.	10944.
4	2904.	4368.	5856.	7320.	8784.	10200.	11664.
5	3648.	5112.	6576.	8040.	9528.	10944.	12408.
6	4368.	5856.	7320.	8784.	10200.	11664.	13128.
7	5112.	6576.	8040.	9528.	10944.	12408.	13872.
8	5856.	7320.	8784.	10200.	11664.	13128.	14616.
9	6576.	8040.	9528.	10944.	12408.	13872.	15336.
10	7320.	8784.	10200.	11664.	13128.	14616.	16080.
11	8040.	9528.	10944.	12408.	13872.	15336.	16800.
12	8784.	10200.	11664.	13128.	14616.	16080.	17544.

netCDF CDL markup language showing how 'auxiliary coordinates' are defined in a CF file

```
dimensions:
    ntime = 10
    lev = 15;   // number of levels
float temp(ntime,lev);
    temp: long_name = “temperature”;
    temp: coordinates = “depth”;
float depth(lev);
     depth: units = “meters”;
     depth: axis = “Z”;
```

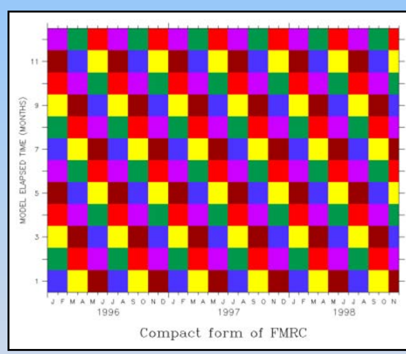
← depth in a layered model

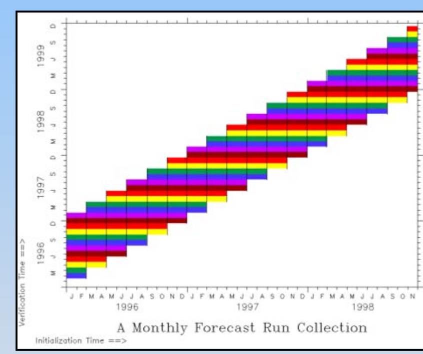
2D times in an an FMRC →

```
dimensions:
    ntime = 10
    nrun  = 15;   // number of runs
float temp(nrun,ntime);
    temp: long_name = “temperature”;
    temp: coordinates = “times”;
float times(nrun, ntime);
    times: units = “days since 01-01-1996”;
```

Ferret syntax: how to regrid using an auxiliary coordinate

- Depth: **temp[GZ(density) = My_density_axis]**
- Time2d: **temp[GT(tf_times)= TF_CAL_T @FMRC]**

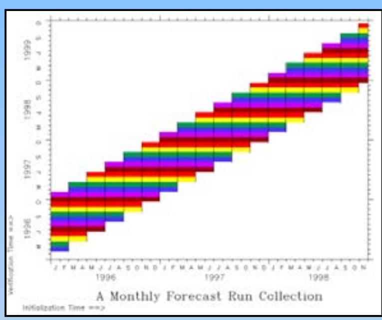


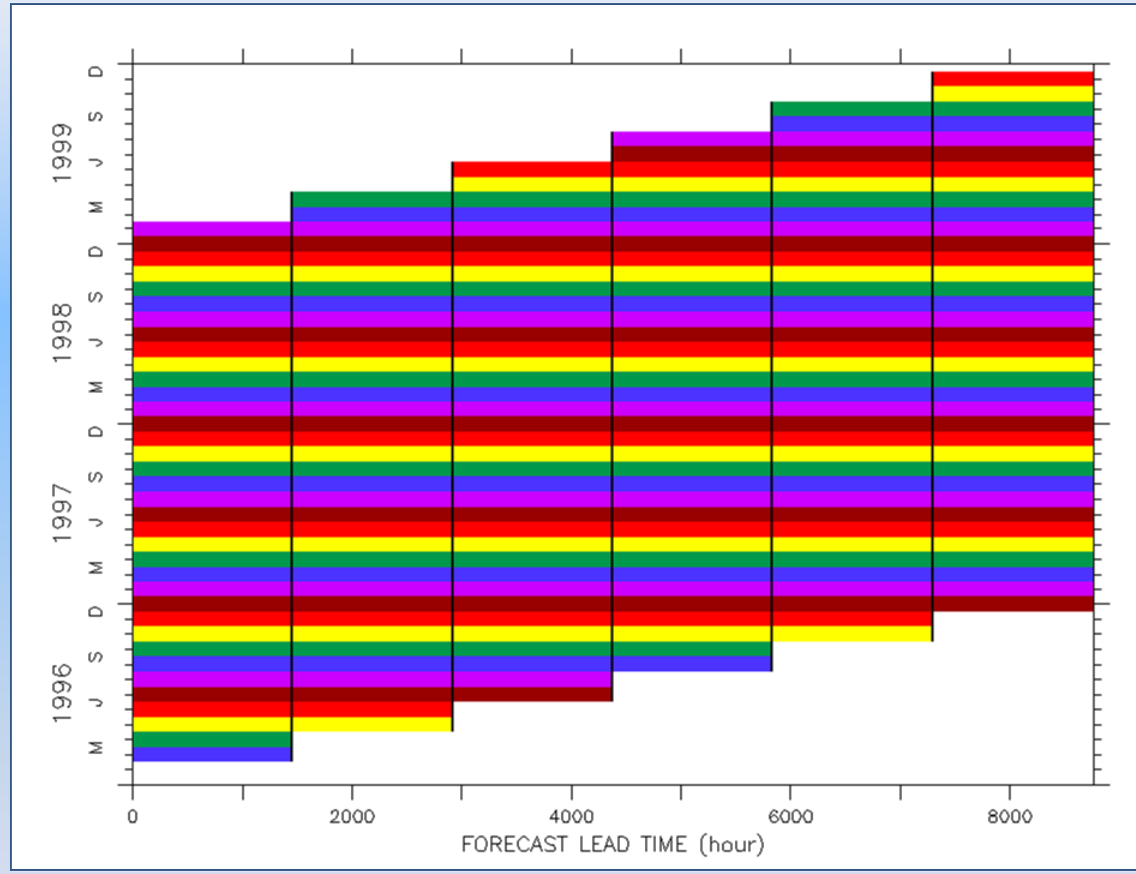


Regridding in a manner analogous to a depth-to-density transformation, we convert the FMRC compact form into diagonal form

Forecast Skill Visualization of an FMRC

Our Goal!





Similarly also replace the Forecast time axis with a lead (lag) time axis

temp[GT(tf_times)=TF_CAL_T ,GF(tf_times)=TF_LAG_F]

That's the background.

Now, what does it feel like to analyze forecast model outputs using Ferret?

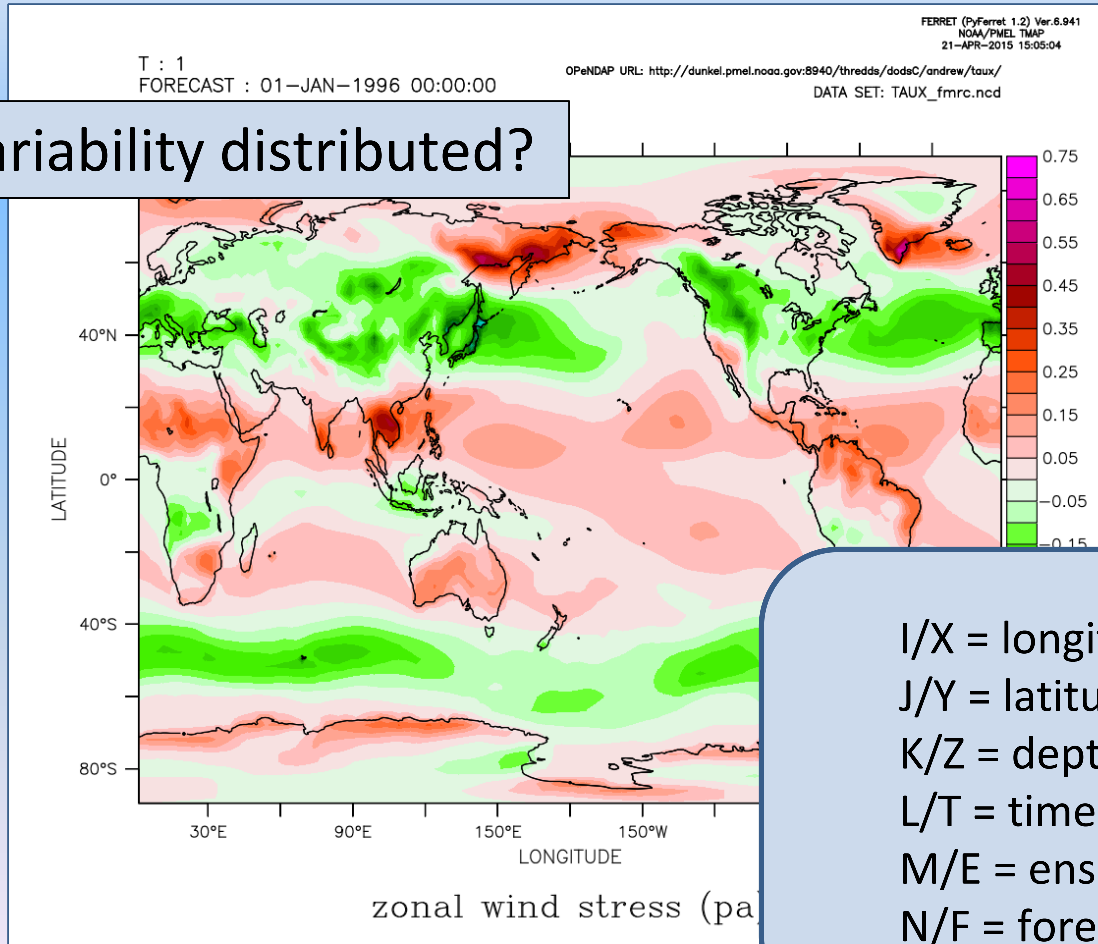
Start up the dataset as you would any other. “N” is the forecast index

> pyferret

yes? USE “http://server/path/myfiles/TAUX_fmrc.ncd” ! aggregation via TDS

yes? FILL tau_x[L=1,N=1]

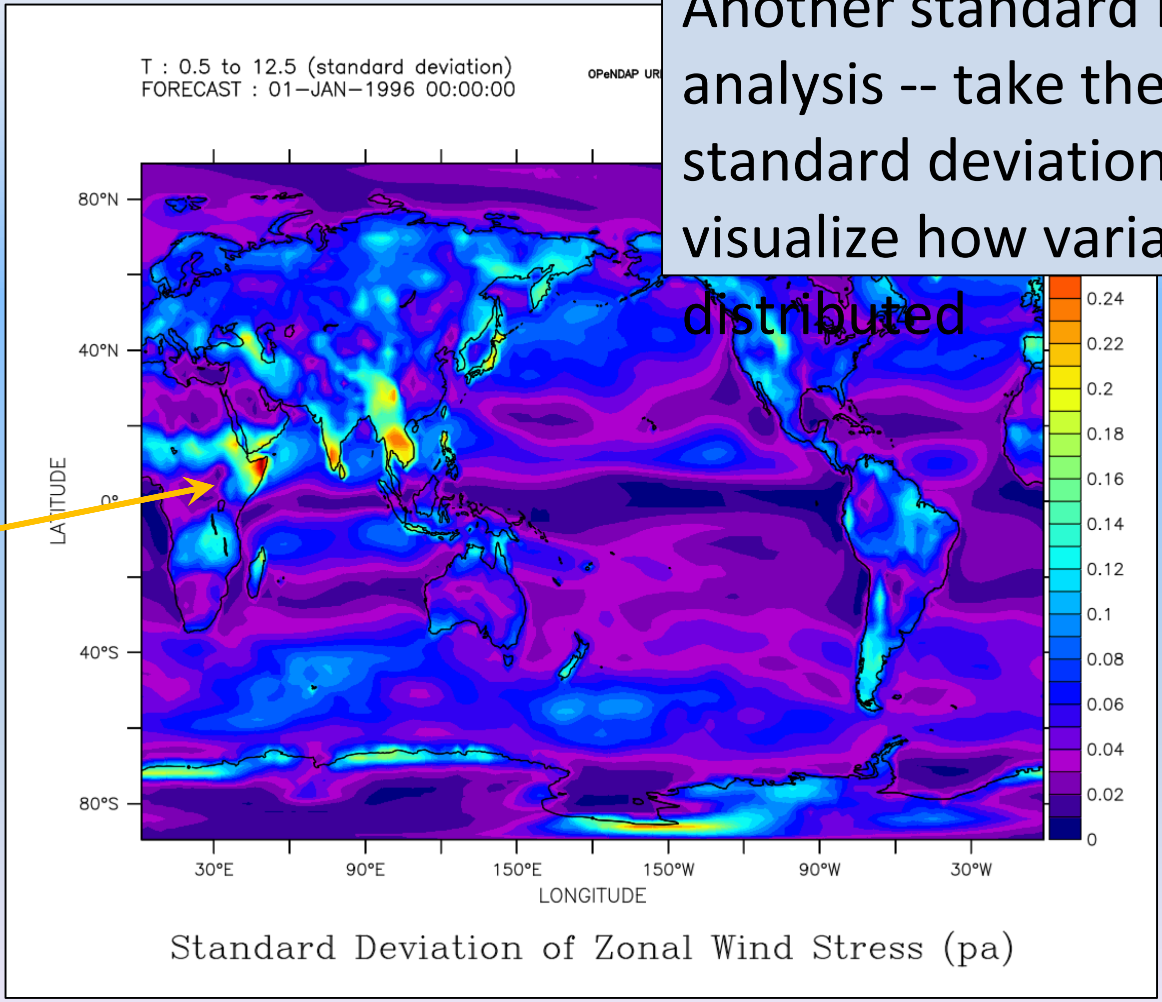
How is the variability distributed?



- I/X = longitude
- J/Y = latitude
- K/Z = depth
- L/T = time
- M/E = ensemble member
- N/F = forecast run

yes? FILL/N=1 tau_x[L=@std]

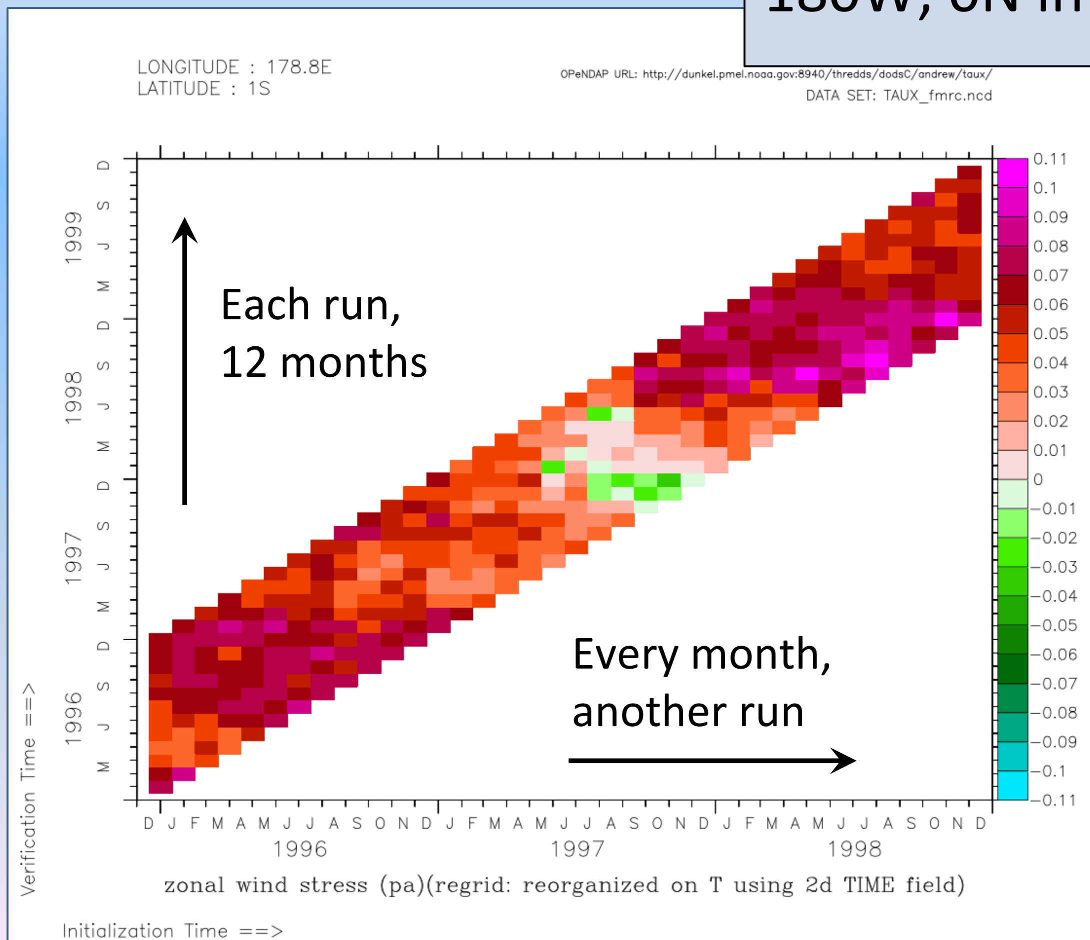
Another standard Ferret analysis -- take the time-standard deviation to visualize how variability is distributed



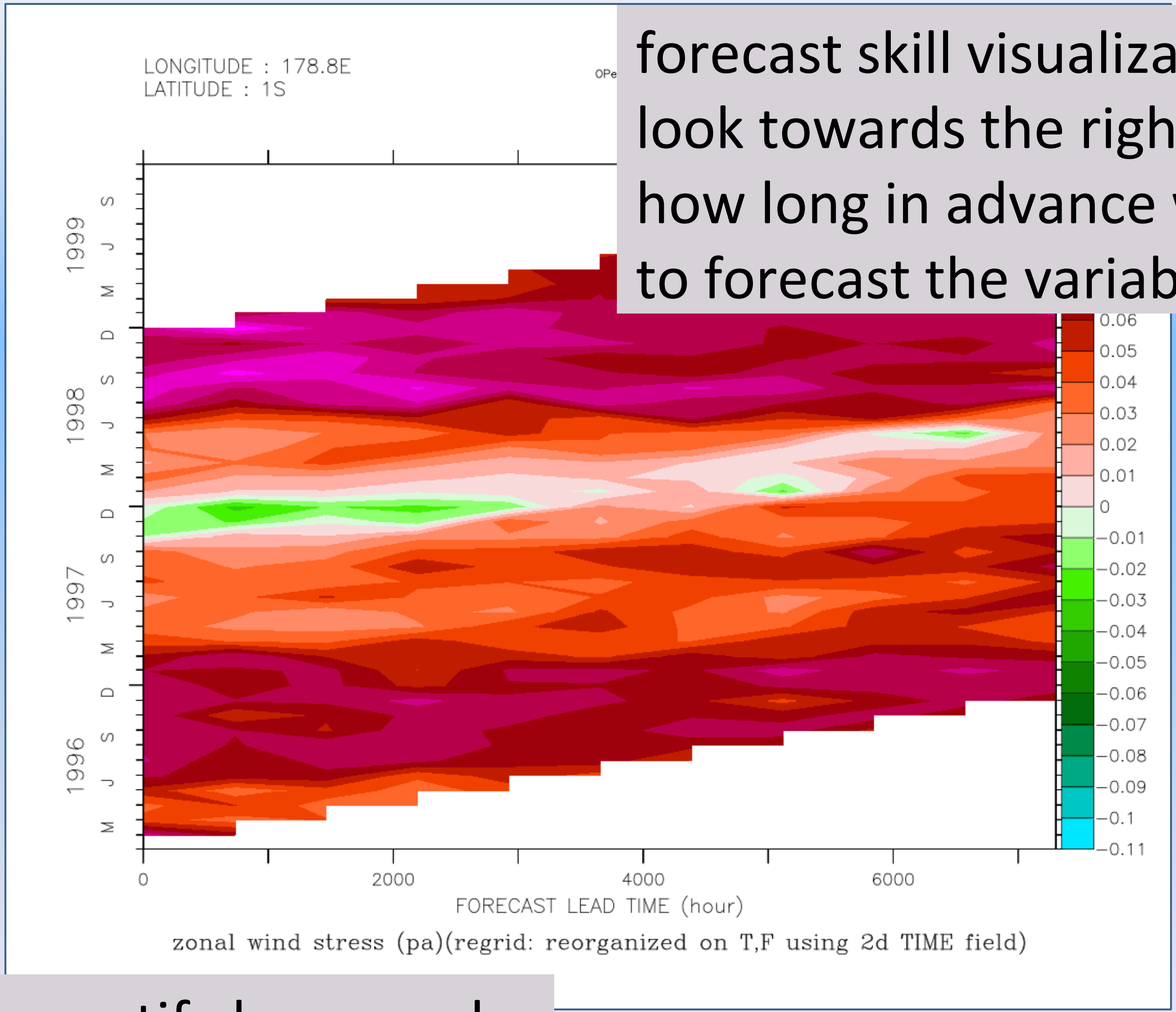
high variability

SHADE/X=180/Y=0 tau_x[GT(times)=TF_CAL_T]

View the forecast series at 180W, 0N in ‘diagonal’ form



FILL/X=180/Y=0   tau_x[GT(times)=TF_CAL_T,GF(times)=TF_LAG_F]



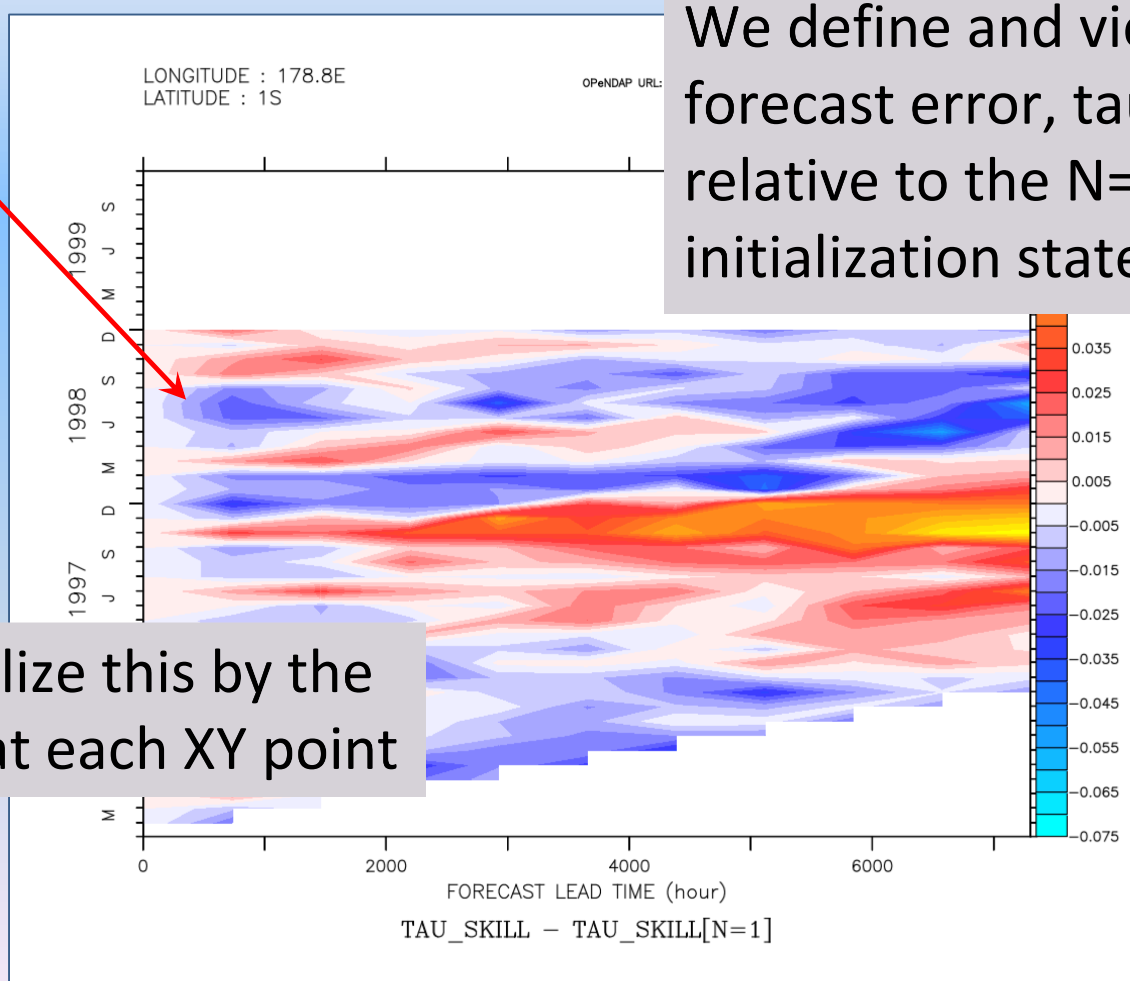
forecast skill visualization: as we look towards the right, we see how long in advance we were able to forecast the variable

Can we quantify how good our forecasts were?

```
LET tau_tf = tau_x[gt(times)=TF_CAL_T,gf(times)=TF_LAG_F]
LET tau_fe = tau_tf - tau_tf[N=1]
FILL/X=180/Y=0  tau_fe
```

We define and view the forecast error, tau_fe, relative to the N=1 initialization state

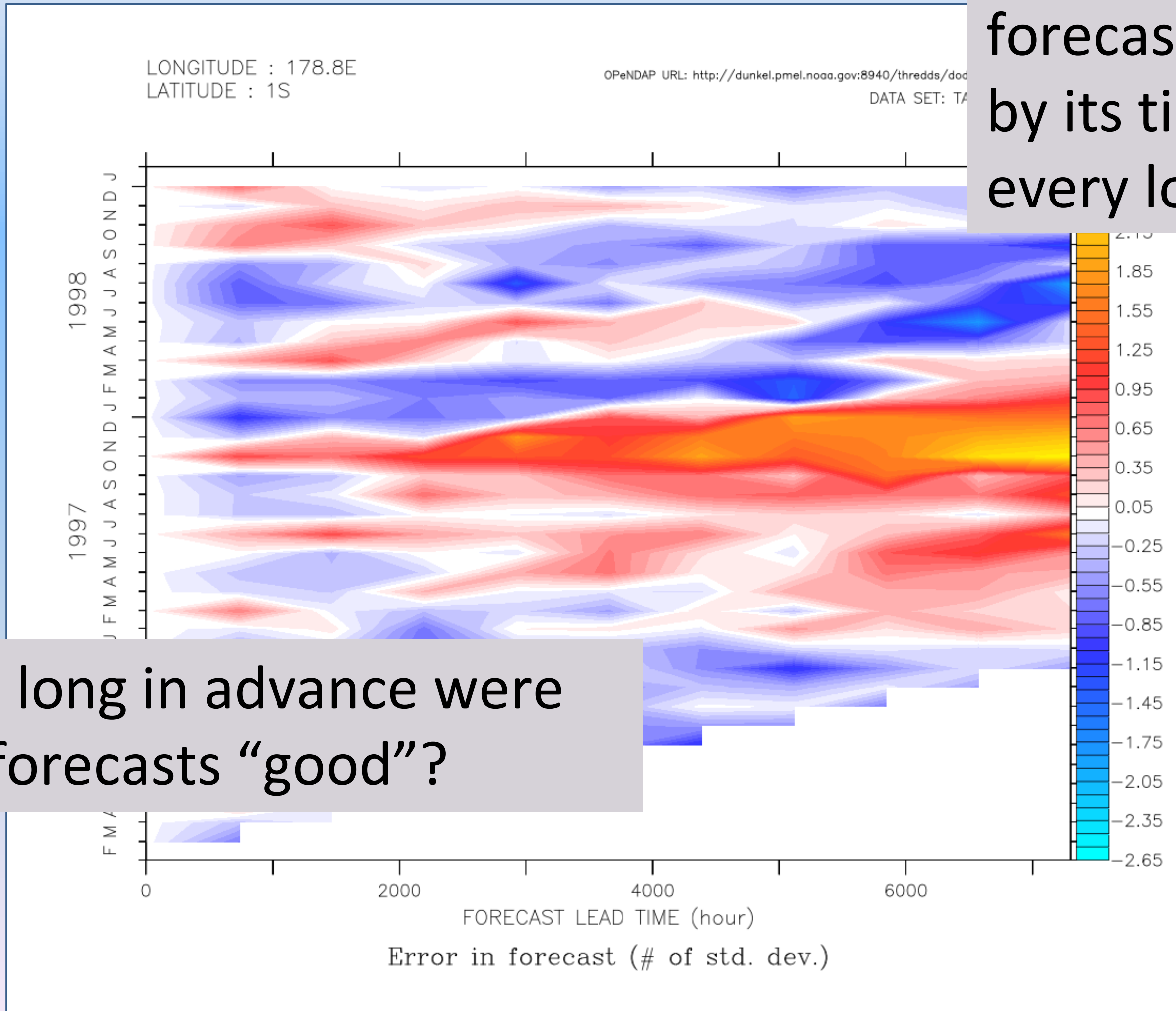
Is there a season cycle to the errors?



Lets normalize this by the variability at each XY point

```
LET tau_stddev = tau_tf[N=1,L=@std]
LET/TITLE=...  tau_nfe = tau_fe/tau_stddev
FILL/Y=180/Y=0/T=...   tau_nfe
```

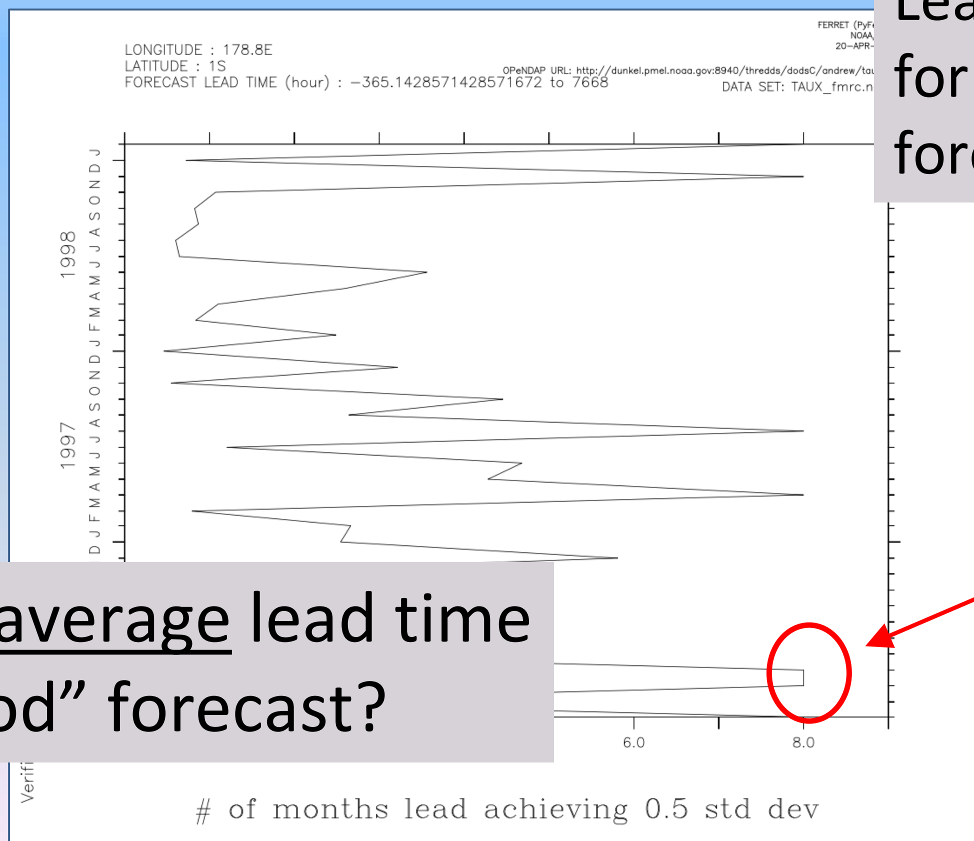
We normalize the forecast error, dividing by its time-std dev at every location



How long in advance were our forecasts "good"?

Lets say "good" == abs. val. of error within 0.5 std dev of final value

```
LET tau_abs = ABS(tau_nfe)
LET tau_fcst_lead = tau_abs[F=@loc:.5] / 730
LET/TITLE="…"  tau_skill = MISSING(tau_fcst_lead,8)  ! cap at 8 mo
PLOT/X=180/Y=0   tau_skill
```



Lead time (months) for achieving "good" forecast error

cap

What is the average lead time to get a "good" forecast?

The average lead time at which the forecast becomes accurate to 0.5 std dev at X=180, Y=0

```
LIST /X=180/Y=0 tau_skill[L=@ave]
         VARIABLE : # of months lead achieving 0.5 std dev
         FILENAME : TAUX_fmrc.ncd
         FILEPATH : http://dunkel.pmel.noaa.gov:8940/thredds/dodsC/andrew/taux/
         LONGITUDE: 178.8E
         LATITUDE : 1S
         TIME     : 17-JAN-1996 12:00 to 16-DEC-1999 00:00 (averaged)

                 4.165
```

Now lets see how our skill is distributed globally

yes? FILL tau_skill[L=@ave]

Global distribution of "good" forecast -- lead time in months