Generation of derived variable and use of transformations

Data Visualization of Marine Met data using FERRET

27 – 31 August, 2018

Derived variables

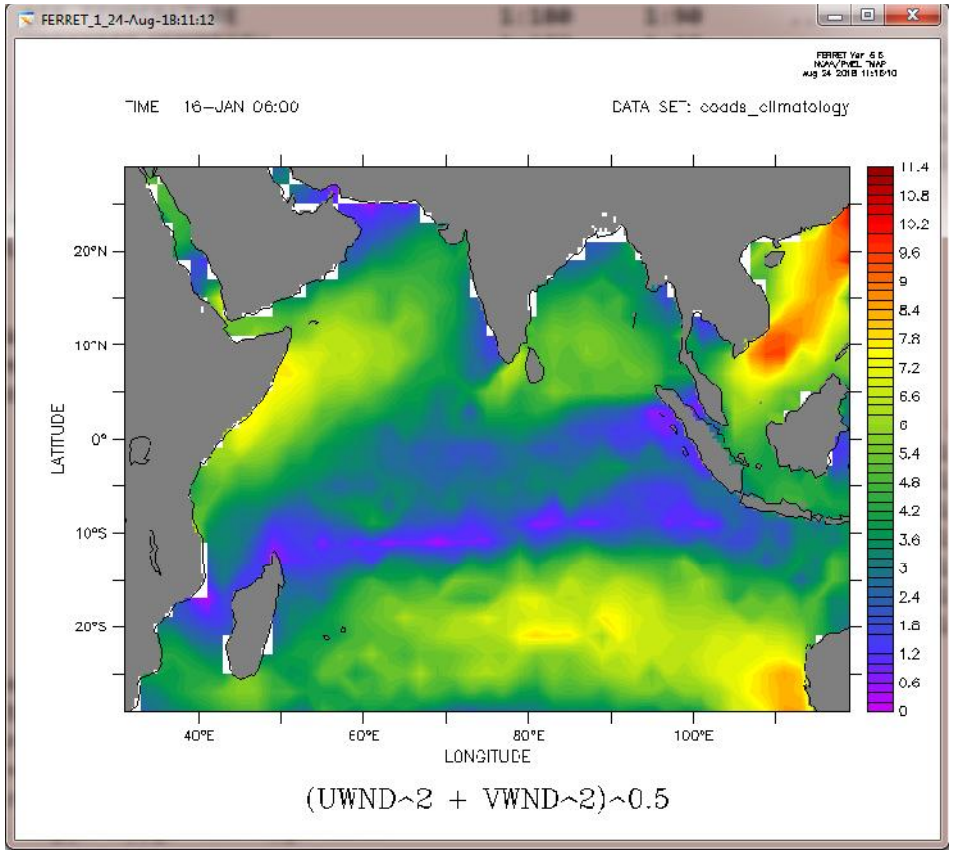
- LET command is used for creating any new variables from the existing variables in the NetCDF file loaded
  - **Eg: abstract expressions** are created from combinations of known variables as arbitrary **expressions**.
    - Yes? Let diff = airt - sst
- One should note that the new variable being created from existing variables should be of same grid size.
- If not then re-gridding need to done for setting them on the same platform for obtaining the variables.
- If no arguments are supplied to the variables then the new variables creates hold the same grid definition of the existing variables in the file
  - Eg: let diff = temp[z=0] – temp[z=5] only creates one level for z
  - Let diff = temp[z=0] – temp creates diff for all z levels in the file.

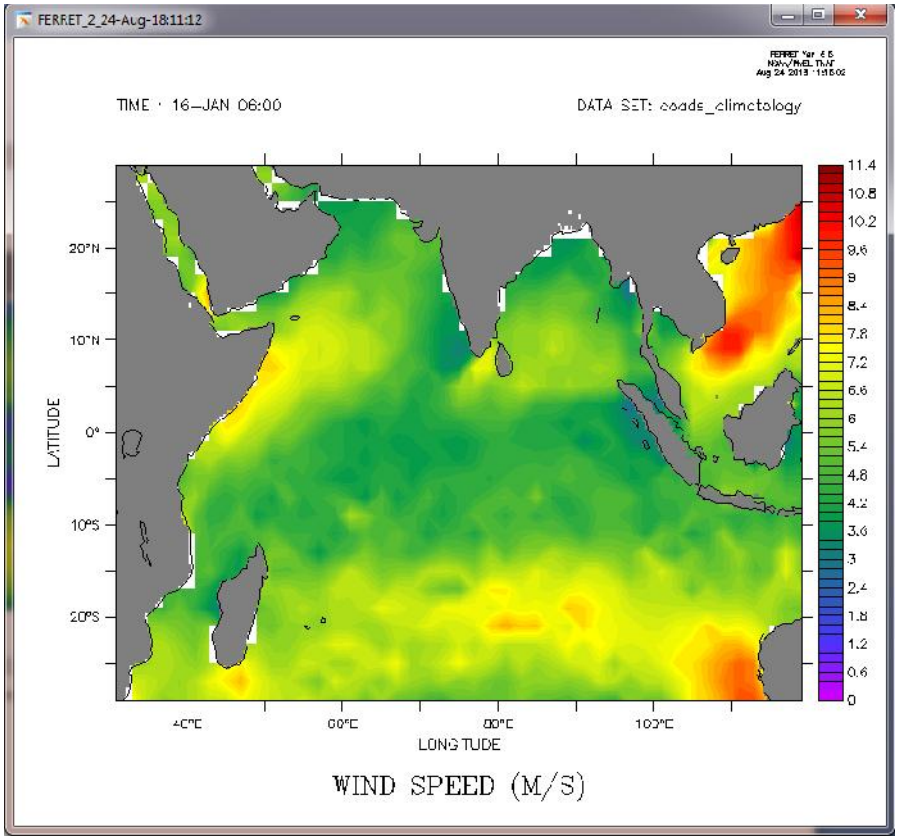
Examples of derived variables

- Let us derive a magnitude from individual u and v components.
  - Magnitude is obtained by sqrt($u^2 + v^2$)
  - Let us define the new variable as “mag” and assign the expression to this varible.
- Eg:- yes? Use coads_climatology

```
yes? Let mag = (uwnd^2 + vwnd^2)^0.5
yes? Set window 1
 ! Let us compare this with original wspd varibale in built
yes? Fill mag[l=1];go fland; go land
yes? Set window 2
yes? Fill wspd[l=1];go fland; go land
```

- We observe slight differences, which can be attributed to interpolation.


FERRET_1_24-Aug-18:11:12
TIME 16-JAN 06:00
DATA SET: coads_climatology
20°N
10°N
0°
10°S
20°S
LATITUDE
40°E
60°E
80°E
100°E
LONGITUDE
11.4
10.8
10.2
9.6
9
8.4
7.8
7.2
6.6
6
5.4
4.8
4.2
3.6
3
2.4
1.8
1.2
0.6
0
(UWND^2 + VWND^2)^0.5

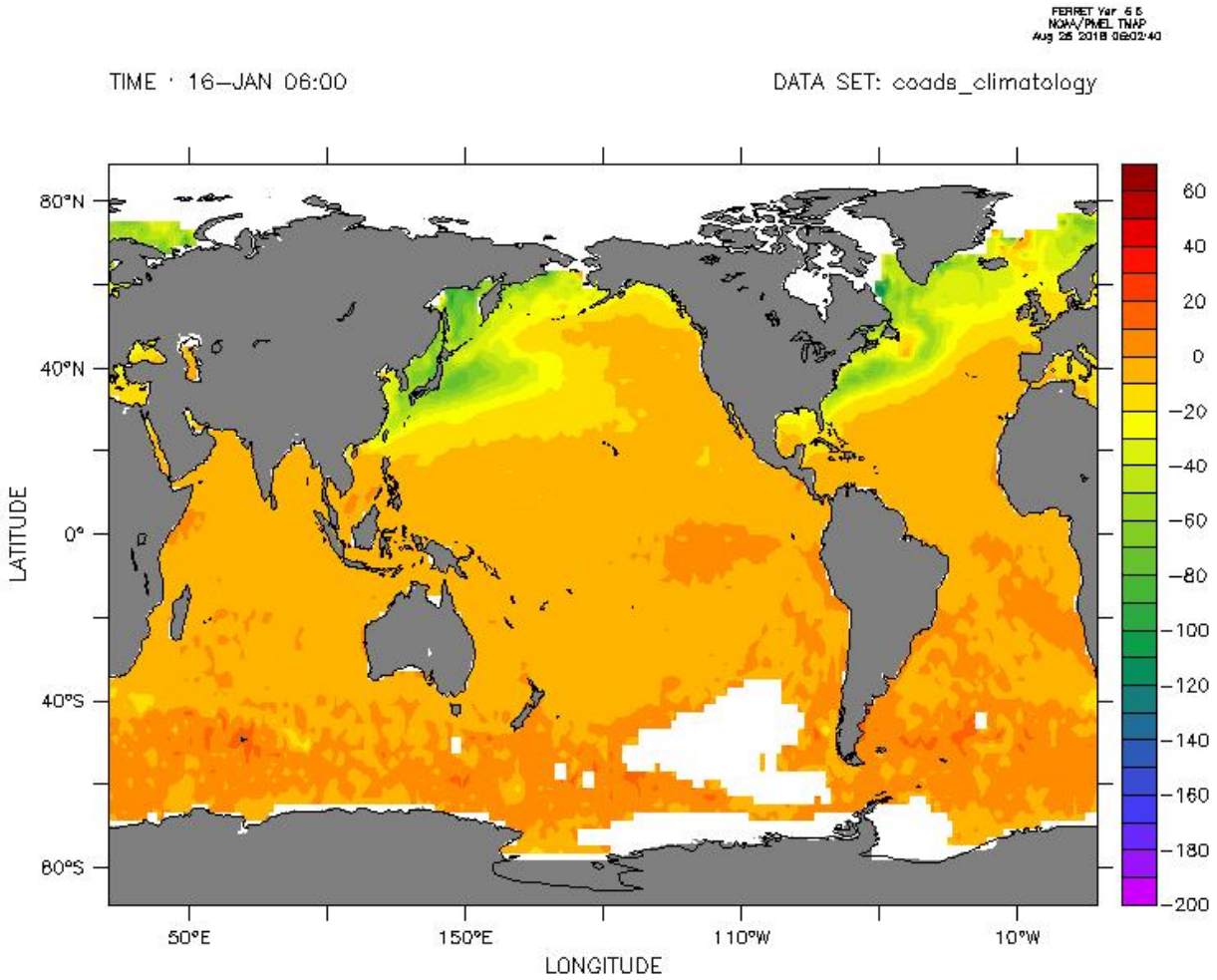

FERRET_2_24-Aug-18:11:12
TIME 16-JAN 06:00
DATA SET: coads_climatology
20°N
10°N
0°
10°S
20°S
LATITUDE
LONGITUDE
11.4
10.8
10.2
9.6
9
7.8
7.2
6.6
6
4.8
3.6
3
1.8
1.2
0.6
0
WIND SPEED (M/S)

Another example …..

- Let us derive sensible heat using wind speed, air temperature and sea surface temperature.
  - We use the following formula for deriving the sensible heat
    - Sensible Heat = k * *(air temp –sea surface temp)* * wind speed
  - Let us define the new variable as “sens_heat” and assign the expression to this variable.
- Eg:- yes? Use coads_climatology

```
yes? Let kappa = 1.004
yes? Let sens_heat = kappa * (airt – sst) * wspd
 ! Let us plot and see how the sensible heat looks like
yes? Fill sens_heat[l=1];
yes? Contour/ov sens_heat[l=1]; go fland; go land
```

- We observe the figure for the entire global ocean as we have not set any bounds.



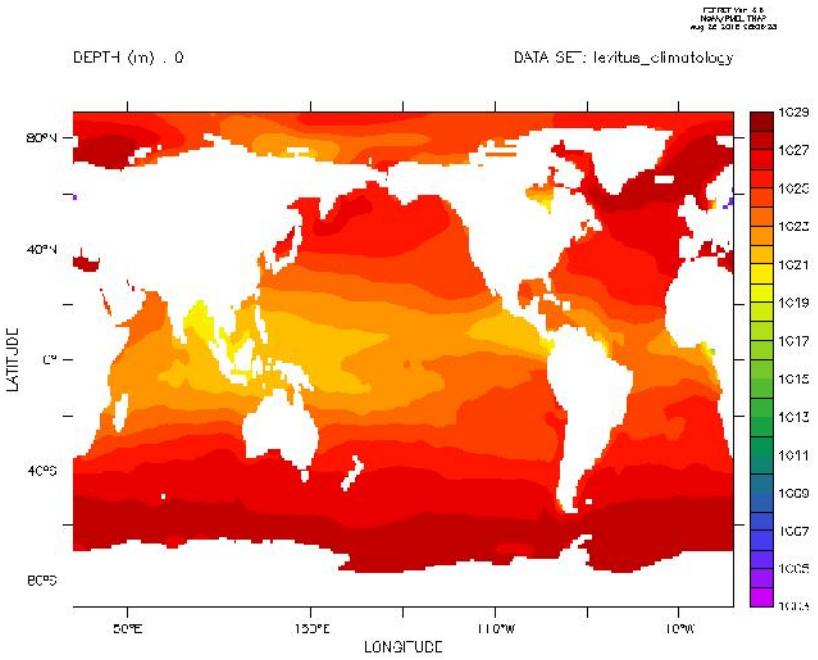
KAPPA * (AIRT − SST) * WSPD

Using built in functions

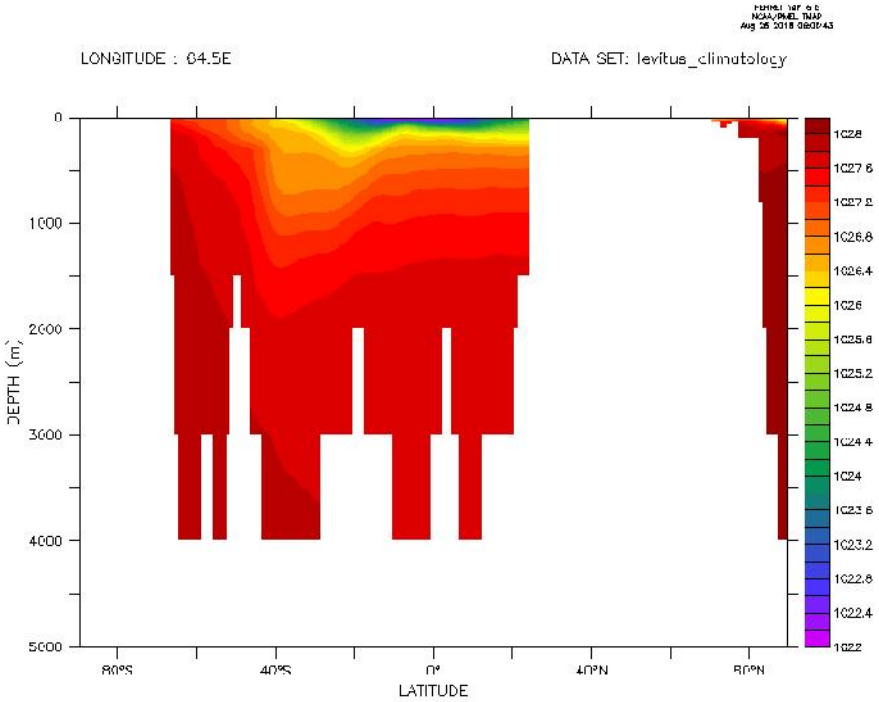
- There are many built in functions provided with in ferret which one can use for deriving a new variables.
- The list of them can be obtained using
  - Yes? Show functions
    - EXP(X)
    - exponential e(X)
    - LOG(X)
    - base 10 log(X)
    - MAX(A,B)
    - point-by-point greater of A and B
    - MIN(A,B)
    - point-by-point lesser of A and B
    - INT(X)
    - truncate to integer
    - ABS(X)
    - absolute value
    - SIN(theta)
- All these can be used for creating new variables

Examples using built in functions

- To derive the density from temperature and salinity we use the build in function called rho_un(s,t,d) as shown below
  - Yes? Let den = rho_un(salt,temp,0)
- This derived variable den will have the same dimensions as that of temperature and salinity
  - Yes? show grid den
    - GRID GMS1
    - name axis # pts start end
    - XAXLEVITR LONGITUDE 360mr 20.5E 19.5E(379.5)
    - YAXLEVITR LATITUDE 180 r 89.5S 89.5N
    - ZAXLEVITR DEPTH (m) 20 i- 0 5000
    - normal T
- The plotting of density can be done the same way as that of temperature and salinity.



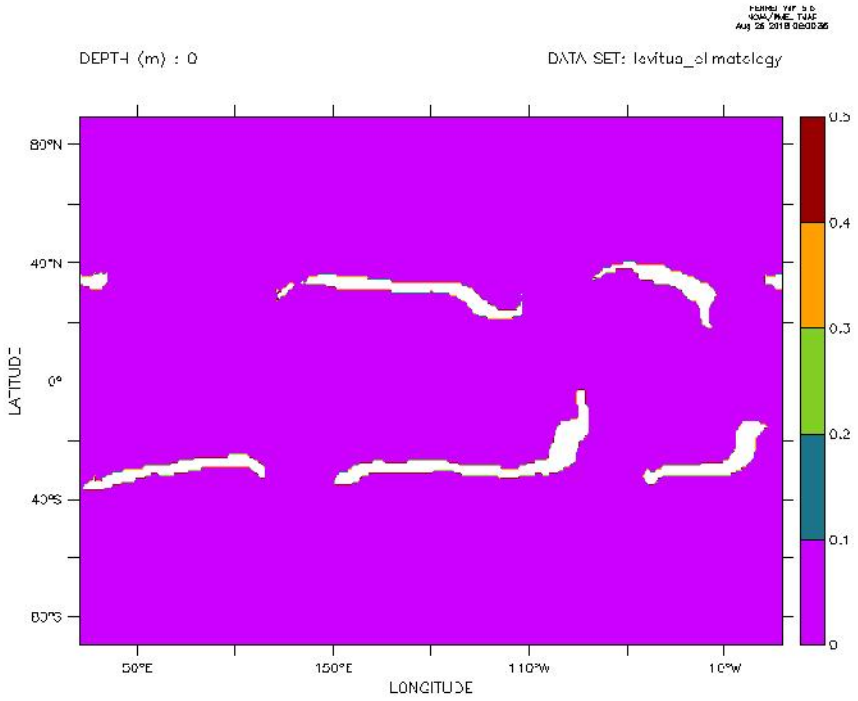
RHO_UN(SALT,TEMP,0)



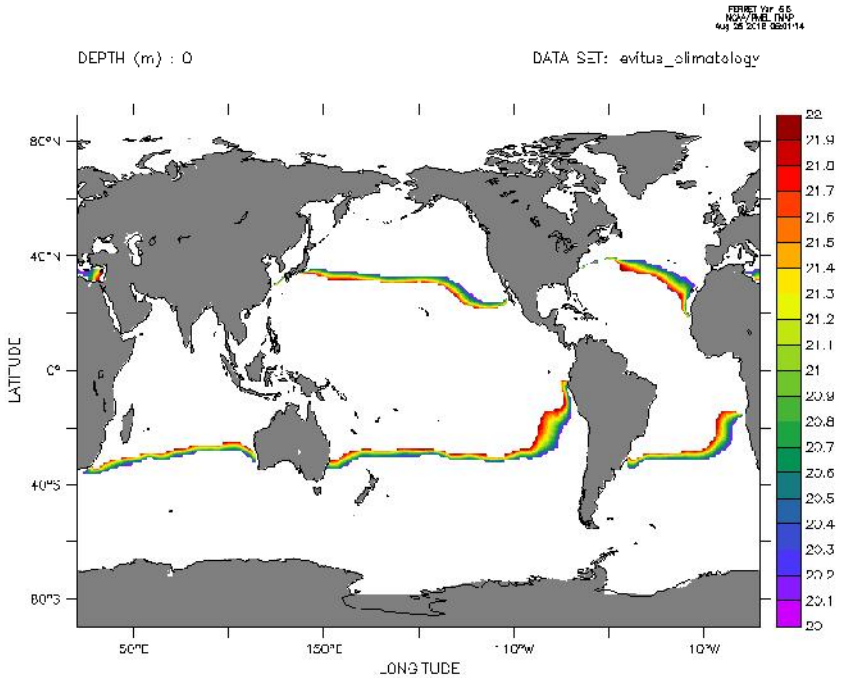
RHO_UN(SALT,TEMP,0)

Using let with “if then else”

- Some conditional assignment can be done and new variable can be creating, by using both let command and “if then else”.
- Generally this can be used for creating mask or eliminating some of the unwanted levels of data in the desired variables.
  - Eg: If we want to capture only region of temperature between 20 and 22 deg C we give.
    - Yes? let tem20deg = if temp gt 20 and temp le 22 then temp
- We can also create mask to be used for eliminating unwanted stuff.
  - Yes? let mask = if temp gt 20 and temp le 22 then 1 else 0
  - Yes? fill/level=(20,22,0.1) temp[l=1,k=1]*mask[l=1,k=1];go fland;go land



IF TEMP GT 20 AND TEMP LE 22 THEN 1 ELSE 0



TEMP[L=1,K=1]*MASK[L=1,K=1]

Using Transformations

- A transformation is an operation performed on a variable along a particular axis and is specified with the syntax "@trn" where "trn" is the name of a transformation.
- Transformations (e.g., averaging, integrating, etc.) may be specified along the axes of a variable.
- Some transformations
  - E.g., averaging, minimum, maximum) reduce a range of data to a point;
  - Others (e.g., differentiating or smoothers) retain the range.
- When transformations are specified along more than one axis of a single variable the order of execution is X axis first, then Y, Z, T, E, and F.

Available Transformations are

- Yes? Show transformations ! Will give list of transformations

variable transforms e.g.SST[T=1-jan:15-mar@DDC]

code	description	code	description
@ITP	interpolated	@SHN	Hanning smoothed
@AVE	averaged	@SPZ	Parzen smoothed
@VAR	variance	@FAV	ave-filled
@SUM	summed	@FLN	linear-filled
@RSU	running sum	@FNR	nearest-filled
@SHF	shifted	@NGD	number of valid
@MIN	minimum	@NBD	number flagged bad
@MAX	maximum	@LOC	location
@DDC	centered derivative	@WEQ	weighted equal
@DDF	forward derivative	@CDA	closest dist above
@DDB	backwards derivative	@CDB	closest dist below
@DIN	integrated	@CIA	closest index above
@IIN	indef. integ.	@CIB	closest index below
@SBX	box smoothed	@EVN	event mask
@SBN	binomial smoothed	@MED	median smoothed
@SWL	Welch smoothed		

regridding transforms e.g.SST[GX=x5deg@AVE]

code	description	code	description
@LIN	lin. interp.	@MOD	modulo ave
@AVE	box avgd	@MODVAR	modulo var
@XACT	exact match	@MODNGD	# gd mod pts
@ASN	index assn	@MODNBD	# bad mod pt
@VAR	variance	@MODSUM	modulo sum
@MIN	minimum	@MODMIN	modulo min
@MAX	maximum	@MODMAX	modulo min
@SUM	sum	@NRST	nearst coord
@NGD	# gd pts		

Examples of Transformations

- If we want to find the depth of 20 degrees isotherms from the temperature data sets.

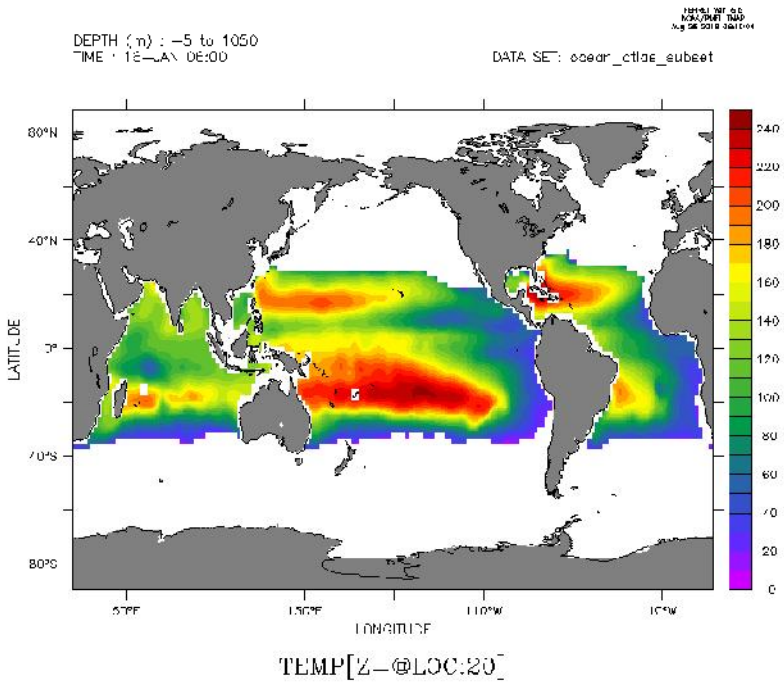
yes? use ocean_atlas_subset

yes? let d20 = temp[z=@loc:20] ! Define variable using “let” command

yes? fill d20[l=1]

yes? fill d20[l=1];go fland;go land

- This will yield the depth and display the depth of 20 deg isotherm across the global ocean.



Examples continued …

- If we want to find heat content up to depth of 300 mts.

yes? use ocean_atlas_subset

yes? let cp = 4000.5

Yes? Let rho = 1024.5

yes? Let htcnt = rho*cp*temp[z=0:300@din]

yes? Fill htcnt[l=1];go fland;go land

This will yield the heat content up to 300 mts across the global ocean.