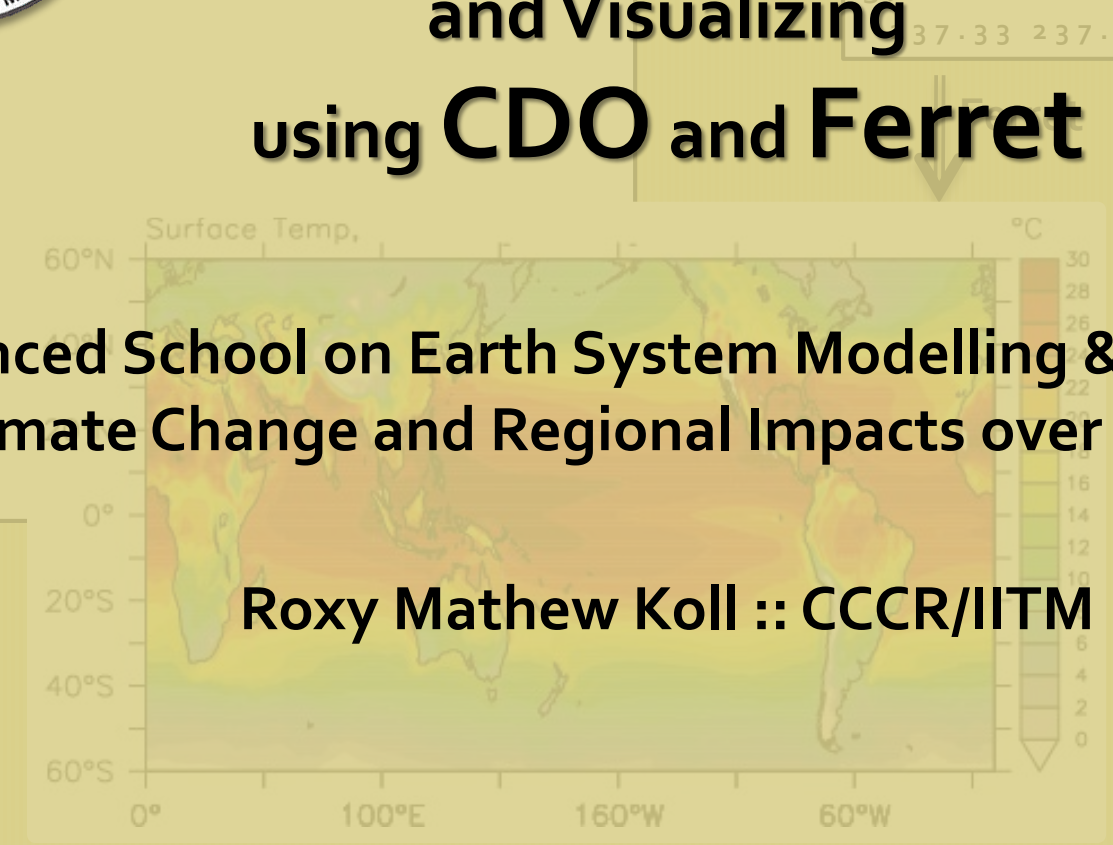
July 2016

==== Analyze and Visualize ====

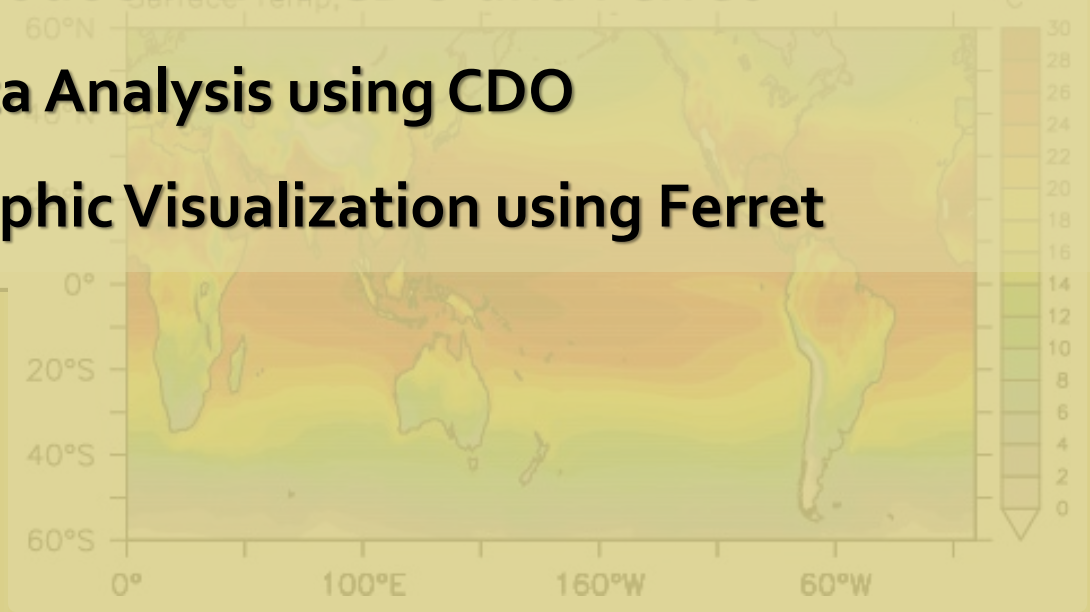
Data Analysis, Model Diagnostics and Visualizing using CDO and Ferret



Advanced School on Earth System Modelling & Workshop on Climate Change and Regional Impacts over South Asia

Roxy Mathew Koll :: CCCR/IITM

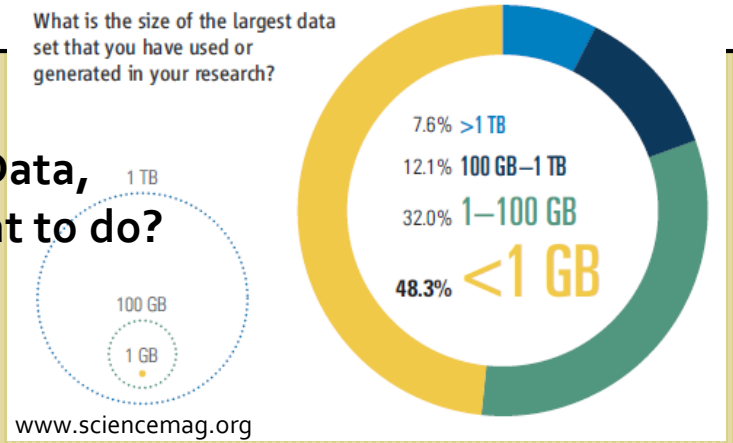
1. Data Analysis and Visualization in Scientific Research
2. Data Attributes, Formats and netCDF
3. Common Tools for Data Analysis and Visualization
4. Introduction to CDO and Ferret
5. Data Analysis using CDO
6. Graphic Visualization using Ferret



Dealing with Data

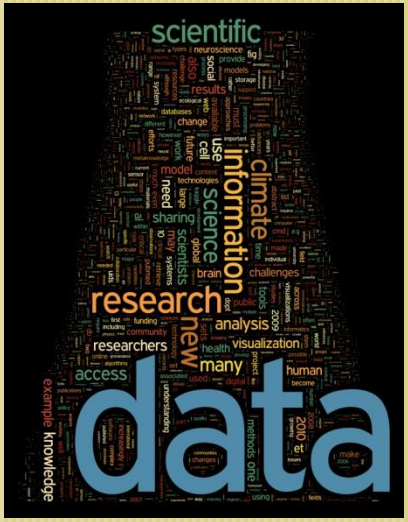
"If we can use and reuse scientific data better, the opportunities are myriad".

Lot of Data,
What to do?



How to do?
Who will do?!

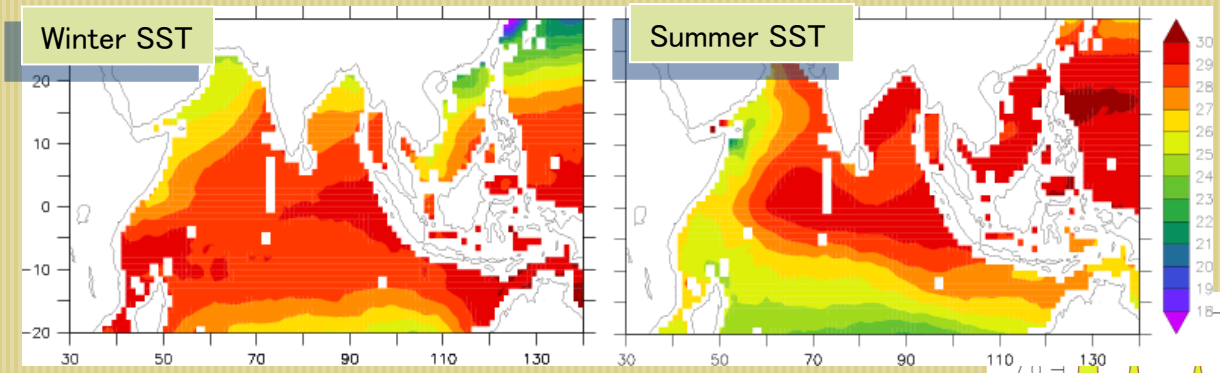
Do you want to make it easy? faster?



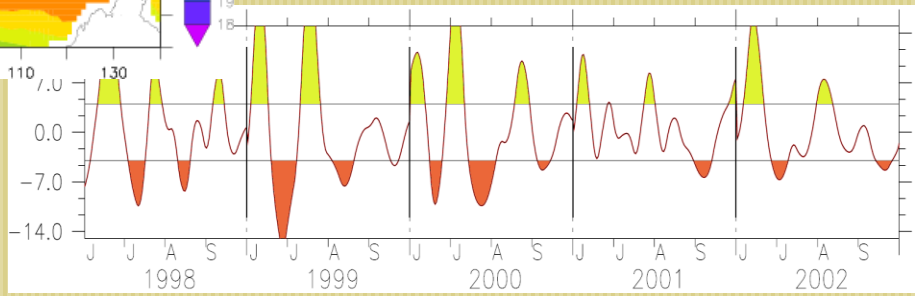
Dealing with Data

Data in Scientific Research

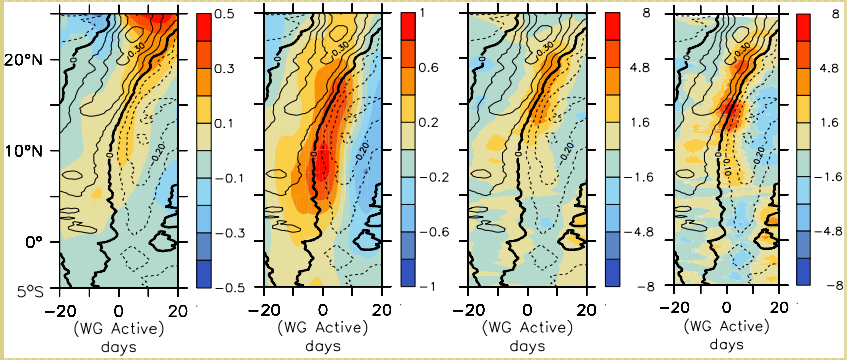
Climatological Analysis

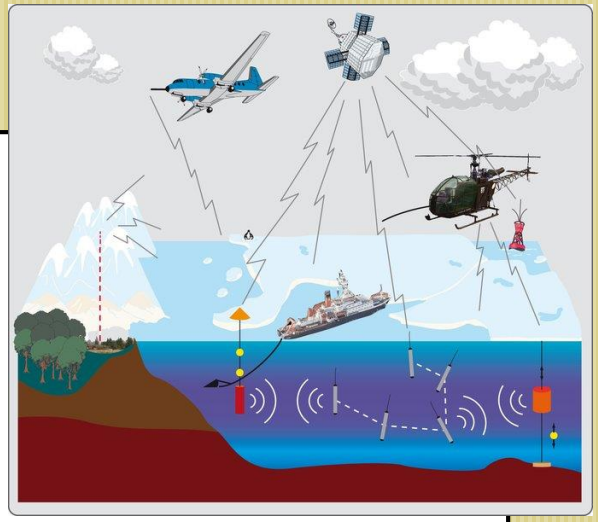


Time series Analysis



Extended Analysis:

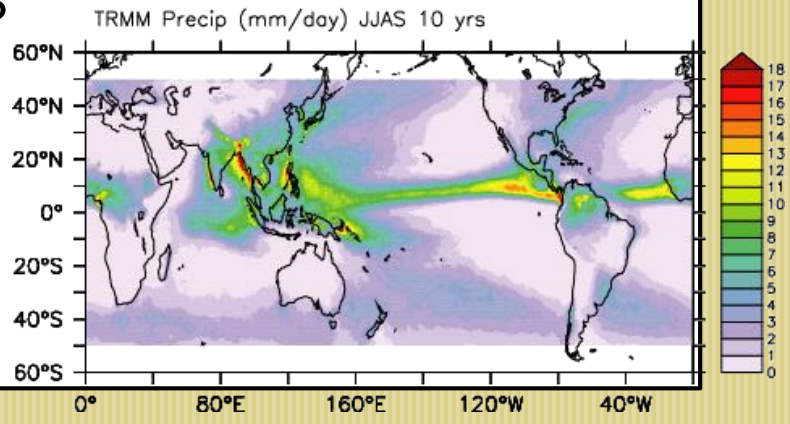




Attributes of the Data

Defines attributes of the data sets used,
e.g. resolution (x,y,t), coverage (spatial scale).

1. Where do the data come from?
    - direct sampling of atmos /ocean/ surface
    - derived from remote sensing
    - model
2. What geographic area does your data/model cover?
    - Eg: Indian Ocean? Monsoon region?
3. What time period does your data/model cover?
    - June-September? Which years?
4. What is the area your variable measured over (resolution of your grid boxes)?
    - Regional processes captured?



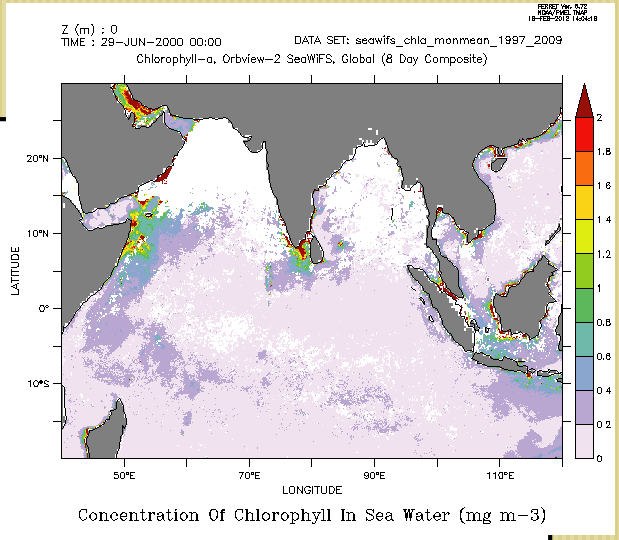
Attributes of the Data

Basic netcdf utility, ncdump:

ncdump –h file.nc

```
netcdf file.nc {
dimensions:
          lon = 192 ;
          lat = 96 ;
          lev = 1 ;
          time = UNLIMITED ; // (10 currently)
variables:
     double lon(lon) ;
          lon:long_name = "longitude" ;
          lon:units = "degrees_east" ;
     double lat(lat) ;
          lat:long_name = "latitude" ;
          lat:units = "degrees_north" ;
     double lev(lev) ;
          lev:long_name = "pressure" ;
          lev:units = "Pa" ;
     double time(time) ;
          time:units = "day as %Y%m%d.%f" ;
     float q(time, lev, lat, lon) ;
          q:long_name = "specific humidity" ;
          q:units = "kg/kg" ;
          q:code = 133 ;
          q:table = 128 ;
          q:grid_type = "gaussian" ;
// global attributes:
     :CDO = "Climate Data Operators version 0.9.5 " ;
     :source = "ECHAM5.2" ;
     :institution = "Max-Planck-Institute for Meteorology" ;
}
```

Attributes of the Data



Describes strengths / limitations
of data sets or models used.

1. Why was data set or model selected?
   - "It was available at the data server" ??
   - "My professor told me to use it" ??!

2. How accurate are the data?
   - Are they equally accurate in all parts of the world under all conditions?
   - What factors may impact confidence in the data?

3. What kind of analysis/techniques are you going to do?

```
247.21 244.83 242.70 240.88 239.51 238.76
238.52 238.58 238.68 238.82 238.99 239.18
239.17 238.95 238.78 238.42 238.04 237.71
237.43 237.33 237.53 238.16 239.29 240.95
243.37 245.92 248.69 251.52 252.06 257.99
260.38 262.42 263.97 265.29 265.57 266.06
265.64 264.20 263.74 263.20 262.42 261.77
261.32 260.99 260.78 260.71 260.72 261.14
261.53 261.57 263.04 263.72 264.30 265.09
265.23 264.64 264.52 263.75 261.33 259.16
257.07 255.06 252.64 249.88 245.79 244.97
244.23 243.53 242.89 242.34 241.87 241.45
241.12 240.87 240.69 240.56 240.47 240.42
240.41 240.45 240.55 240.71 240.96 241.32
241.75 242.28 242.88 243.58 244.34 245.15
```

NCL

Matlab

CDO

Fortran

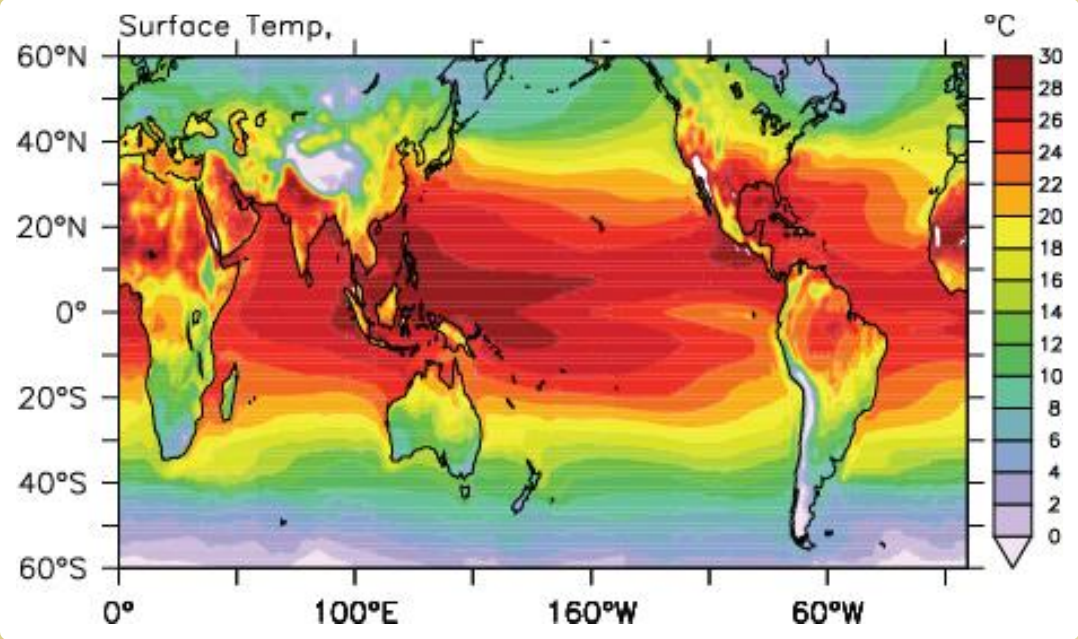
NCO

```
J >
  243.53 242.89 242.34 241.87 241.45
J >
  264.64 264.52 263.75 261.33 259.16
A >
  264.20 263.74 263.20 262.42 261.77
S >
  237.33 237.53 238.16 239.29 240.95
```

GraDs

Ferret

GMT

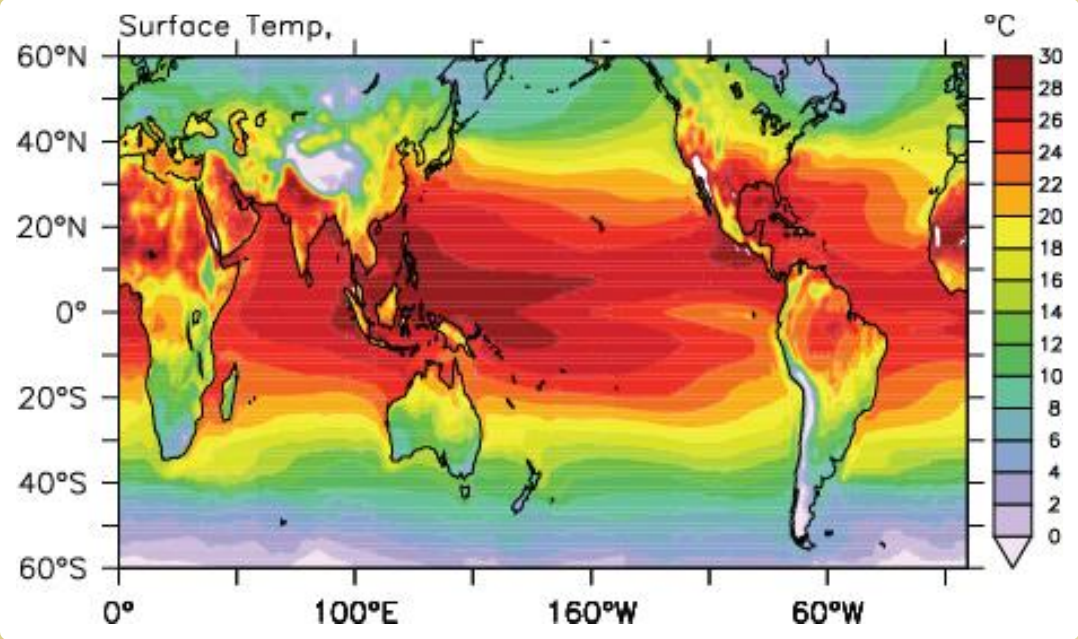


```
247.21 244.83 242.70 240.88 239.51 238.76
238.52 238.58 238.68 238.82 238.99 239.18
239.17 238.95 238.78 238.42 238.04 237.71
237.43 237.33 237.53 238.16 239.29 240.95
243.37 245.92 248.69 251.52 252.06 257.99
260.38 262.42 263.97 265.29 265.57 266.06
265.64 264.20 263.74 263.20 262.42 261.77
261.32 260.99 260.78 260.71 260.72 261.14
261.53 261.57 263.04 263.72 264.30 265.09
265.23 264.64 264.52 263.75 261.33 259.16
257.07 255.06 252.64 249.88 245.79 244.97
244.23 243.53 242.89 242.34 241.87 241.45
241.12 240.87 240.69 240.56 240.47 240.42
240.41 240.45 240.55 240.71 240.96 241.32
241.75 242.28 242.88 243.58 244.34 245.15
```

CDO

```
J>
  243.53 242.89 242.34 241.87 241.45
J>
  264.64 264.52 263.75 261.33 259.16
A>
  264.20 263.74 263.20 262.42 261.77
S>
  237.33 237.53 238.16 239.29 240.95
```

Ferret



CDO – Climate Data Operators

CDO is a "single command-line" tool with a collection of operators (650+) to manipulate and analyze climate data.
- Max-Planck-Institute for Meteorology

Current officially released version is cdo 1.7.2
https://code.zmaw.de/projects/cdo

Supported file formats: **GRIB** 1/2, **netCDF** 3/4, srv, ext, and ieg

Supported grid types: rectangular, curvilinear and unstructured

Installing CDO

```
bash$   tar -xvf cdo.tar

bash$   cd cdo

bash$   ./configure --with-netcdf=/usr/local/lib

bash$   make install
```

Magic Word*

* Usage: **cdo** , That's all!

```
bash$ cdo <options> <operator> input.nc out.nc
```

This is all you need to know about CDO

Operators

There are more than 650 operators available.

Categories	Description	Example
File information	Print information about datasets	`cdo sinfo file.nc`
File operations	Copy, split and merge datasets	`cdo mergetime f1995.nc f1996.nc out.nc`
Selection	Select parts of a dataset	`cdo seldate,1996-06-15 f1996.nc out.nc`
Comparison	Compare datasets	`cdo eq`
Modification	Modify datasets	`cdo settaxis,1990-06-15,00:00,1mon input.nc out.nc`
Arithmetic	Arithmeticly process datasets	`cdo add f1995.nc f1996.nc out.nc`
Statistical values	Ensemble, field, vertical and time statistic	`cdo monmean input.nc out.nc`
Regression	Detrend of time series	
Interpolation	Field, vertical and time interpolation	
Transformation	Spectral transformation	etc.

Global options for all operators:

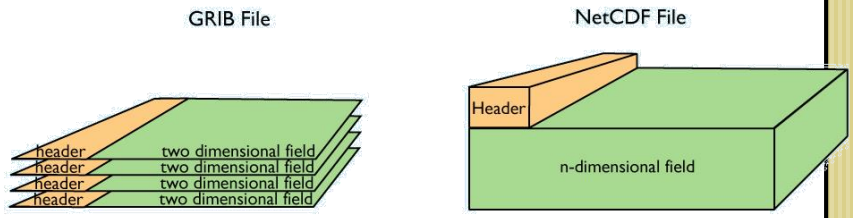
-h Help information for the operators

Eg: `cdo -h <operator>`

-f <format>

Format of the output file (grb, nc, srv, ext, ieg)

Eg: `cdo -f nc copy input.grb out.nc`



-m <missval>

Set the default missing value (default: -9e+33)

-a Converts from relative to absolute time axis

Eg: `cdo -a -f nc copy input.grb out.nc`

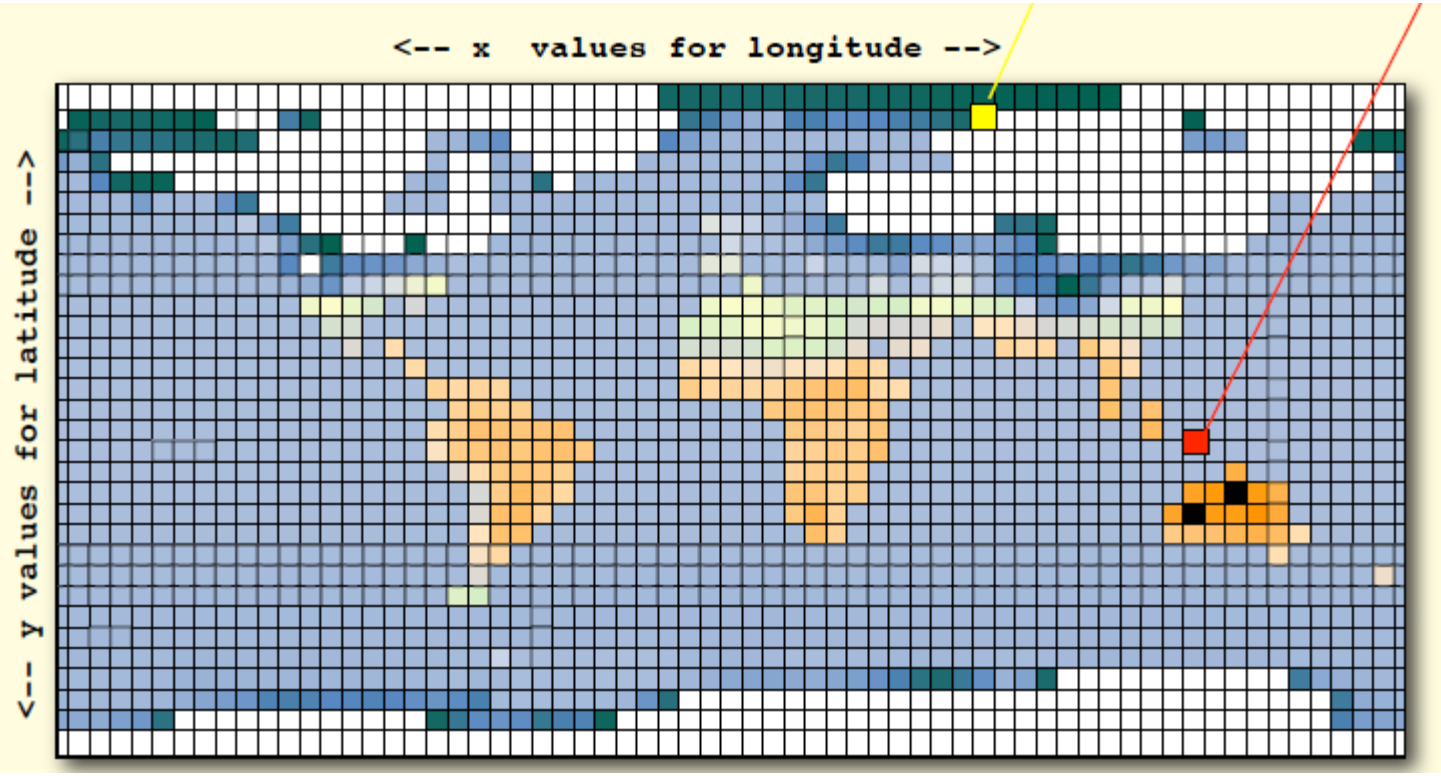
-r Converts from absolute to relative time axis

Eg: `cdo -r -f nc copy input.grb out.nc`

File Information Eg: info, sinfo

```
bash$ cdo info ts.nc
```

```
-1 : Date       Time  Code Level Size Miss  : Minimum Mean Maximum
 1 : 1978-01-02 00:00 169  0     2048 0     : 226.98 268.46 311.08
```



If you are running a model, the min/max can point you to the location where the model is getting unstable.

File Information Eg: pardes, griddes, showyear

```
bash$ cdo pardes data.nc
```

display the parameters stored in the data file

```
bash$ cdo griddes data.nc
```

shows the grid properties

```
bash$ cdo showyear data.nc
```

display the years

Data Selection Eg: seltimestep

```
bash$ cdo seltimestep,1/5 ts.nc ts5.nc

-1 : Date Time Code Level Size Miss : Minimum Mean Maximum
 1 : 2001-01-31 18:00 167 0 18432 0 : 227.84 276.36 307.02
 2 : 2001-02-28 18:00 167 0 18432 0 : 221.51 276.02 307.52
 3 : 2001-03-31 18:00 167 0 18432 0 : 212.20 276.55 306.41
 4 : 2001-04-30 18:00 167 0 18432 0 : 206.65 277.81 311.49
 5 : 2001-05-31 18:00 167 0 18432 0 : 206.30 279.51 314.02
cdo info : Processed 1 variable 5 timesteps. ( 0.00s )
```

Simple calculations Eg: fldmean, yearmean, selname

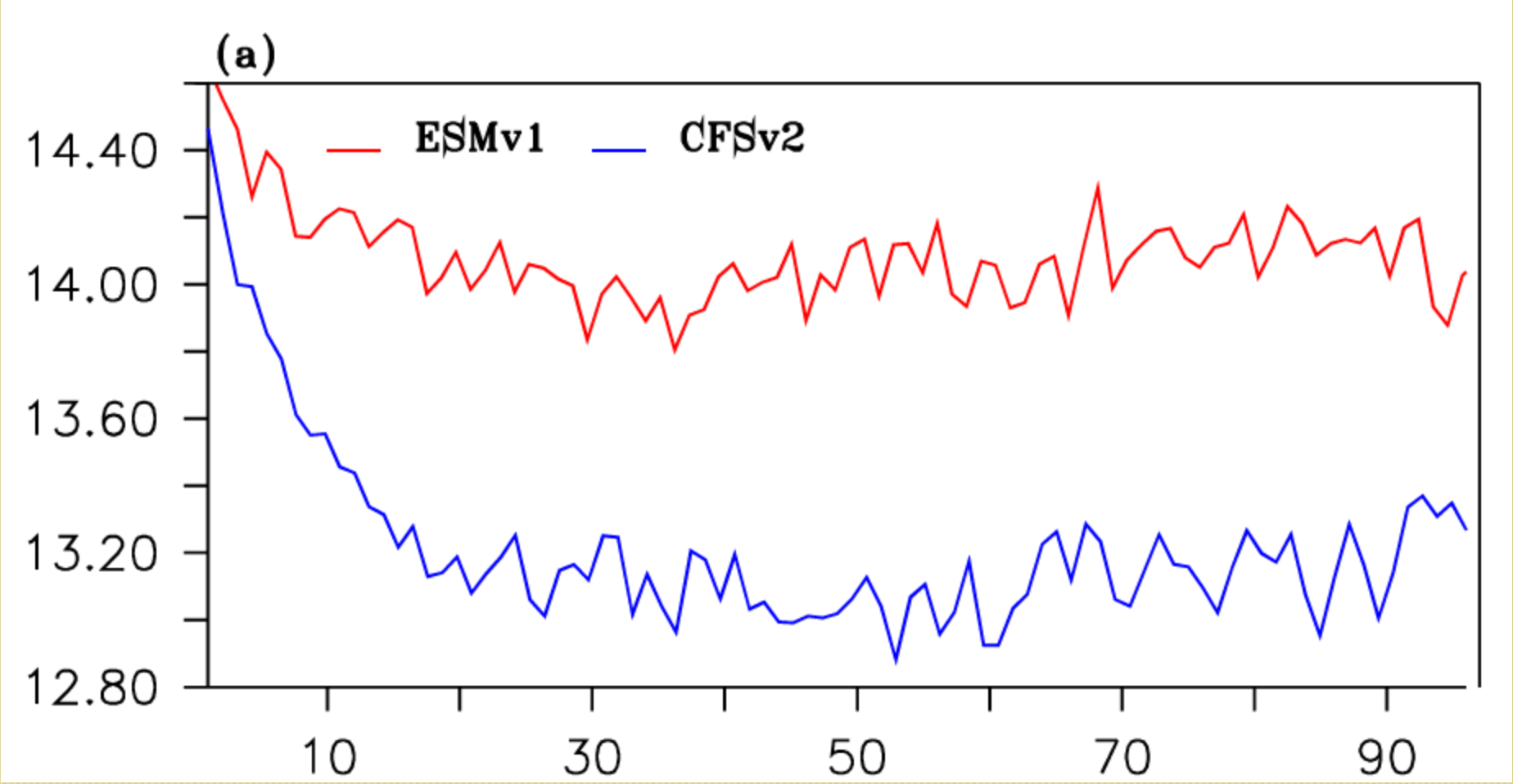
calculate global annual means of surface temperature

```
bash$ cdo selname,ts data.nc out1.nc
bash$ cdo yearmean out1.nc out2.nc
bash$ cdo fldmean out2.nc out3.nc
```

calculate and display the global annual means

```
bash$ cdo -s output -fldmean -yearmean -selname,ts data.nc
```

Model diagnostics
Task: Compare global mean surface temp. CTL and Sensitivity

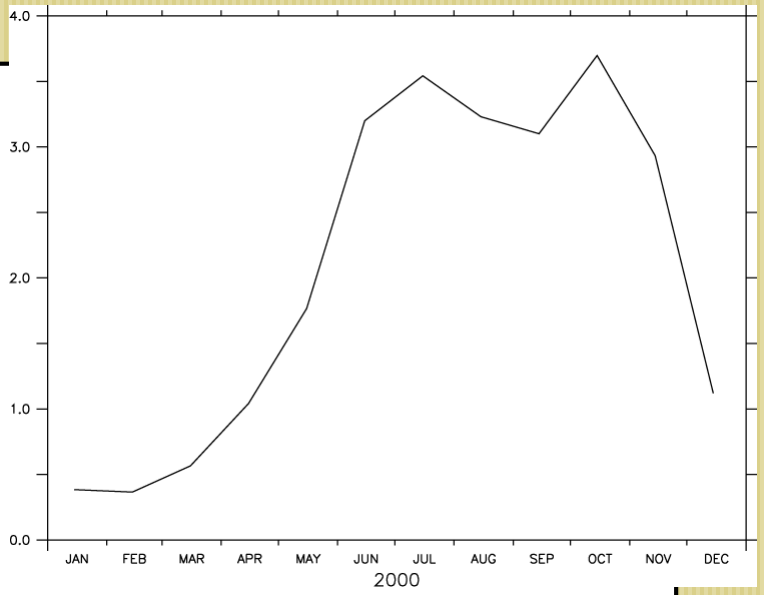


Piping

→ Reduces unnecessary disk I/O

→ Parallel processing

Annual Cycle of precipitation

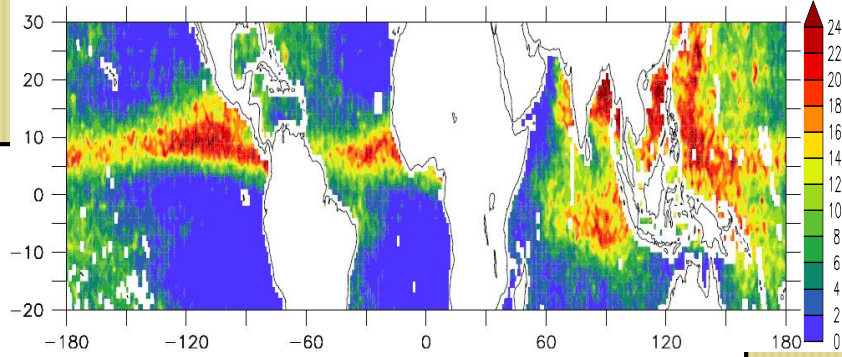


Step by Step:

```
bash$ cdo sellonlatbox,75,85,10,15 input.nc out_box.nc

bash$ cdo fldmean out_box.nc out_box_fldmean.nc

bash$ cdo ymonmean out_box_fldmean.nc out_box_ymonmean.nc
```

Piping:

```
bash$ cdo ymonmean -fldmean -sellonlatbox,75,85,10,15

                    input.nc out_box_ymonmean.nc
```



Piping

→ Reduces unnecessary disk I/O

→ Parallel processing

Eg: Standard deviation of JJAS precipitation anomalies

Step by Step:

```
bash$ cdo selmon,6,7,8,9 input.nc out_jjas.nc

bash$ cdo timmean out_jjas.nc out_jjas_mean.nc

bash$ cdo sub out_jjas.nc out_jjas_mean.nc  out_jjas_anom.nc

bash$ cdo timstd out_jjas_anom.nc out_jjas_std.nc
```

Piping:

```
bash$ cdo –timstd -sub -selmon,6,7,8,9  input.nc
       -timmean –selmon,6,7,8,9 input.nc out_jjas_std.nc
```

Regridding/Remapping: griddes, remapbil

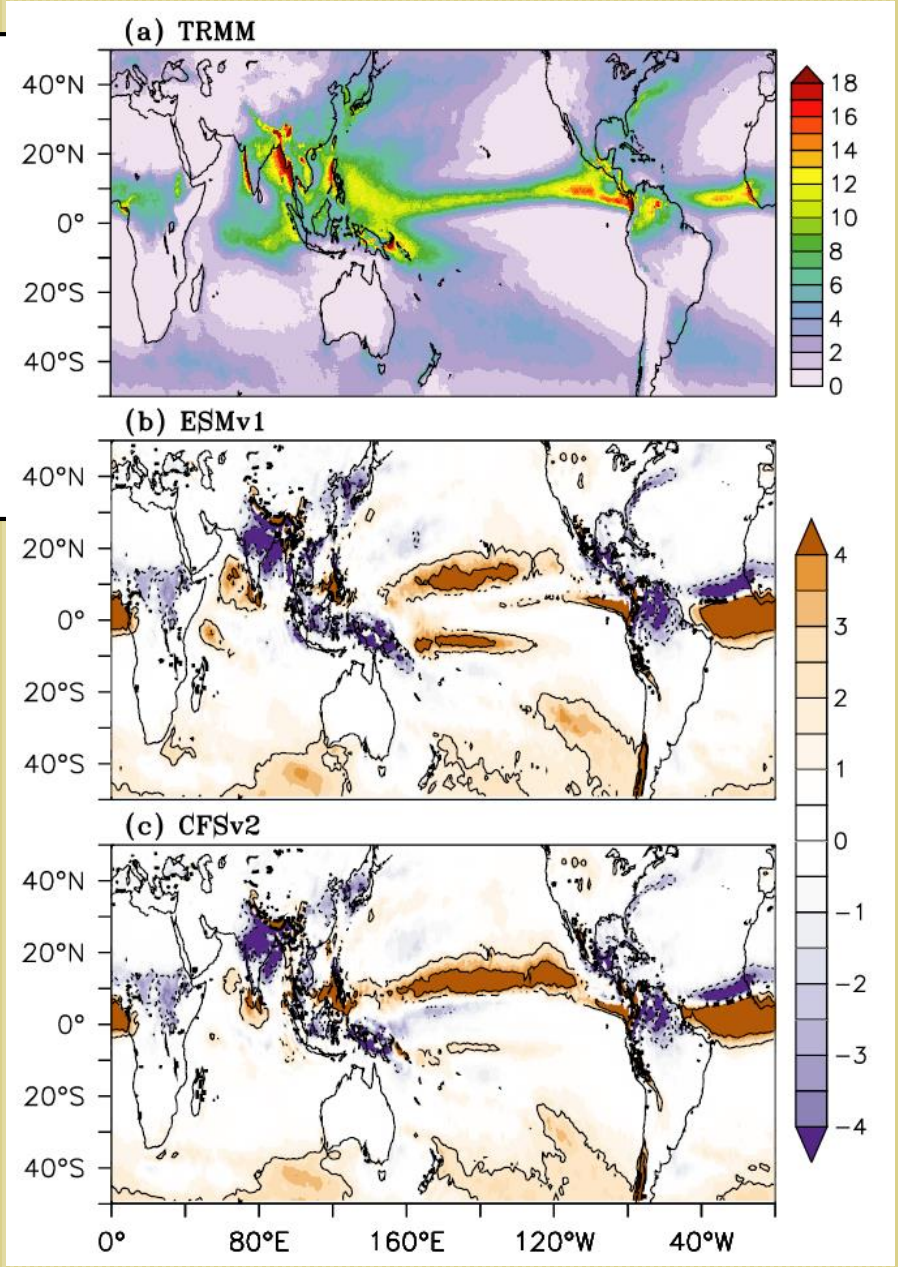
Required for Model diagnostics to compare datasets with different grids

```
bash$ cdo griddes obs_data.nc > obsgrid
```

```
bash$ cdo remapbil,obsgrid mod_data.nc mod_data_obsgrid.nc
```

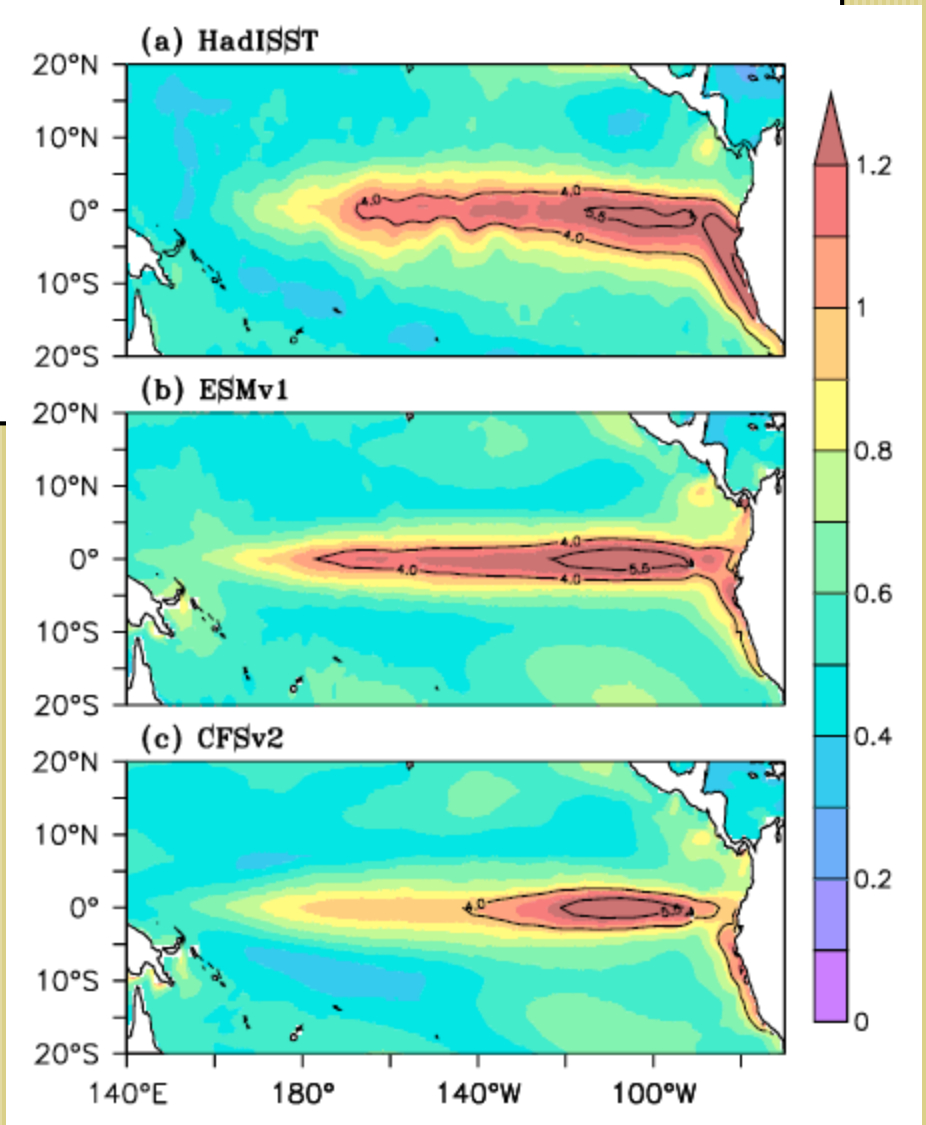
Model diagnostics
Eg: Mean precipitation

Task: Prepare precipitation bias maps



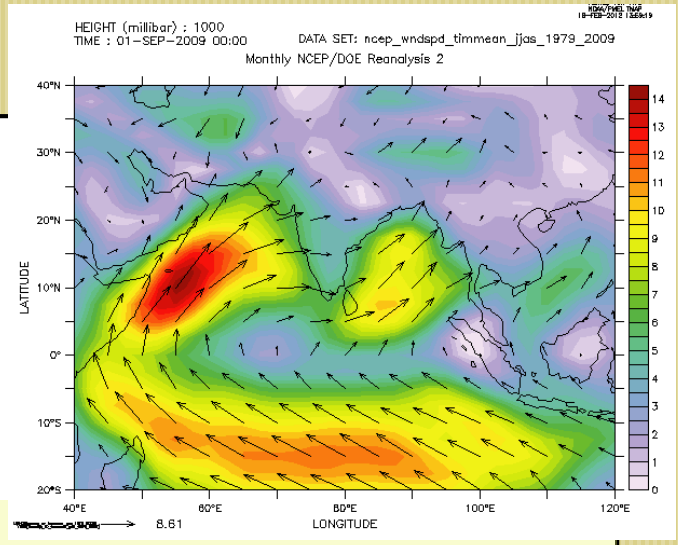
Model diagnostics
Eg: ENSO variability

Task: Prepare SST standard deviations and compare



Arithmetic example: sqr, sqrt

wind speed = $\sqrt{(u^2 + v^2)}$



Step by Step:

```
bash$ cdo sqr uwind.nc uwind_sqr.nc

bash$ cdo sqr vwind.nc vwind_sqr.nc

bash$ cdo add uwind_sqr.nc vwind_sqr.nc wind_add.nc

bash$ cdo sqrt wind_add.nc wind_spd.nc
```

Piping:

```
bash$ cdo sqrt -add -sqr uwind.nc -sqr vwind.nc wind_spd.nc
```

Country based statistics: country_mask

```
bash$ cdo ifthen -eqc,<code> countrymask.nc in.nc out.nc
```

Country ba

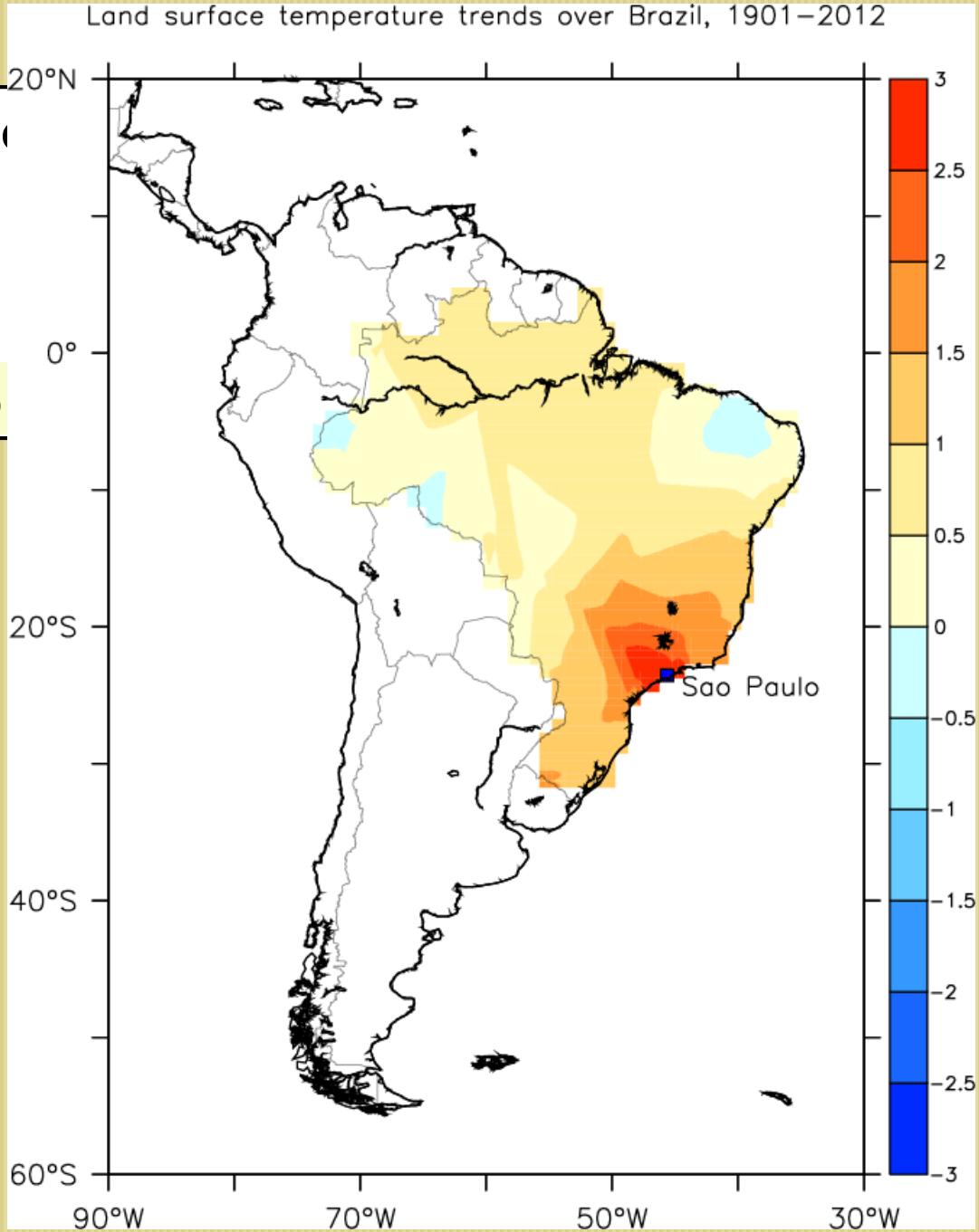
```
bash$ cdo i
```

1 afghanistan	38 chad	74 guam	112 malawi	148 portugal	185 tonga
2 albania	39 chile	75 guatemala	113 malaysia	149 puerto_rico	186 trinidad
3 algeria	40 china	76 guinea	114 maldives	150 qatar	187 tunisia
4 american_samoa	41 colombia	77 guinea-bissau	115 mali	151	188 turkey
5 andorra	42 comoros	78 guyana	116 malta	republic_of_palau	189 turkmenistan
6 angola	43 congo	79 haiti	117 martinique	152 reunion	190 tuvalu
7 anguilla	44 costa_rica	80 hawaii	118 mauritania	153 romania	191 uganda
8 antigua	45 cuba	81 honduras	119 mauritius	154 russia1	192 ukraine
9 argentina	46 cyprus	82 hong_kong	120 mexico	154 russia2	193
10 armenia	47 czechoslovakia	83 hungary	121 micronesia	154 russia3	united_arab_emirat
11 australia	48 denmark	84 iceland	122 moldova	157 rwanda	es
12 austria	49 djibouti	85 india	123 mongolia	158 sao	194
13 azerbaijan	50 dominica	86 indonesia	124 montserrat	159 saudi_arabia	united_kingdom
14 bahamas	51 dominican_rep.	87 iran	125 morocco	160 senegal	195 uruguay
15 bahrain	52 ecuador	88 iraq	126 mozambique	161 seychelles	196 uzebkistan
16 bangladesh	53 egypt	89 ireland	127 namibia	162 sierra_leone	197 vanuatu
17 barbados	54 el_salvador	90 isa-	128 nauru	163 singapore	198 venezuela
18 belarus	55	neutral_zone	129 nepal	164	199 vietnam
19 belgium	equatorial_guinea	91 israel	130 netherlands	solomon_islands	200 virgin_islands
20 belize	56 estonia	92 italy	131	165 somalia	201 wake_island
21 benin	57 ethiopia	93 ivory_coast	netherlands_antilles	166 south_africa	202 wallis
22 bermuda	58	94 jamaica	132 new_caledonia	167 south_korea	203 west_bank
23 bhutan	falkland_islands	95 japan	133 new_zealand	168 spain	204
24 bolivia	59 fiji	96 jordan	134 nicaragua	169 sri_lanka	western_sahara
25 botswana	60 finland	97 kazakhstan	135 niger	170 st._helena	205
26 brazil	61 france	98 kenya	136 nigeria	171 st._lucia	western_samoa
27 brunei	62 french_guiana	99 kiribati	137 north_korea	172 st._pierre	206 yemen
28 bulgaria	63	100 kuwait	138 norway	173 st._vincent	207 yugoslavia
29 burkina_faso	french_polynesia	101 kyrgyzstan	139 oman	174 sudan	208 zaire
30	64 gabon	102 laos	140 pakistan	175 surinam	209 zambia
burma_myanmar	65 gambia	103 latvia	141 panama	176 swaziland	210 zimbabwe
31 burundi	66 gaza_strip	104 lebanon	142	177 sweden	211 USA
32 cambodia	67 georgia_ssr	105 lesotho	papua_new_guinea	178 switzerland	
33 cameroon	68 germany	106 liberia	143 paraguay	179 syria	
34 canada	69 ghana	107 libya	144 peru	180 taiwan	
35 cape_verde	70 greece	108 lithuania	145 philippines	181 tajikistan	
36 cayman_islands	71 greenland	109 luxembourg	146 pitcairn_island	182 tanzania	
37	72 grenada	110 macau	147 poland	183 thailand	
central_african_rep	73 guadeloupe	111 madagascar		184 togo	

Country based statistics: (

E.g., Brazil

```
bash$ cdo ifthen -eqc,26
```

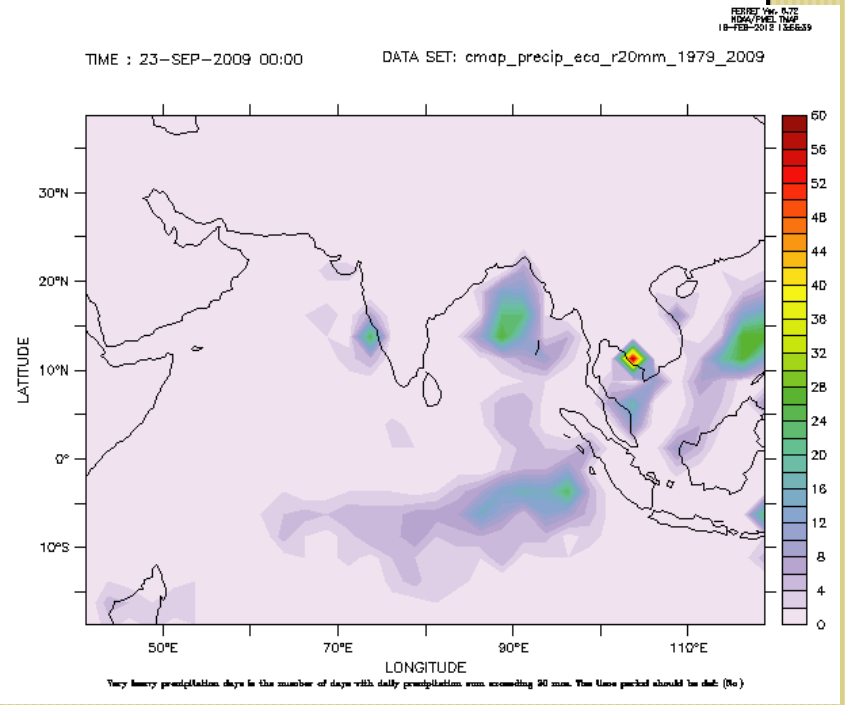


Indexes

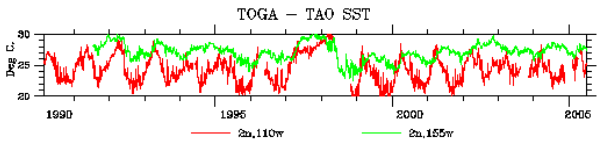
Eg:
ECAR20MM

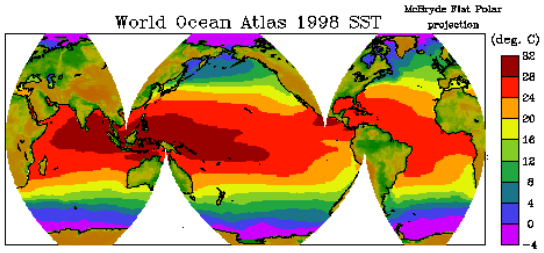
- Number of days with precip >20 mm

```
bash$ cdo eca_r20mm input.nc output.nc
```



Ferret





Ferret is an interactive visualization and analysis environment for gridded and non-gridded data

- PMEL/NOAA

Current officially released version is Ferret 6.96

http://www.ferret.noaa.gov

Supported file formats: **netCDF**, binary, ascii, etc.

Installing and running Ferret

Download tarfiles:

fer_executables.tar.gz (Ferret and utilities)
fer_environment.tar.gz (support files)
fer_dsets.tar.gz (sample data sets)

Follow the procedures given at the Ferret website:

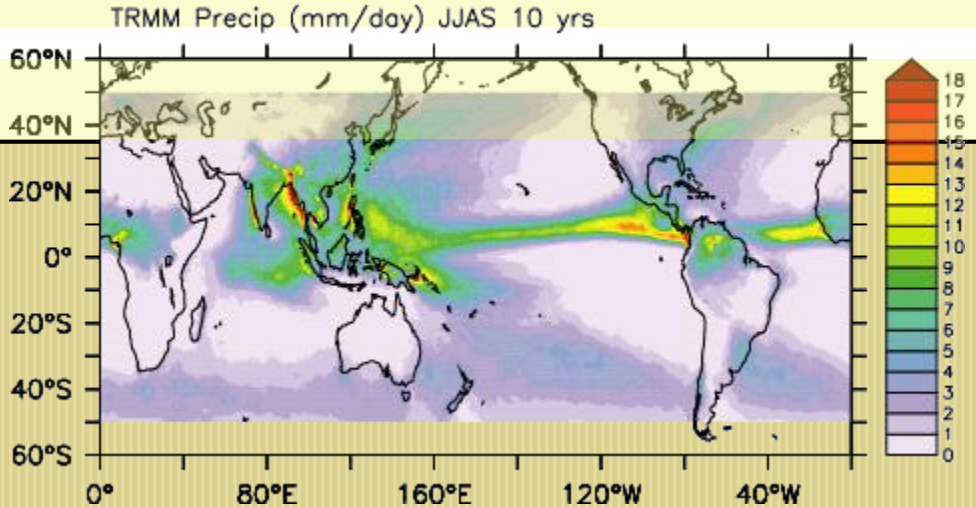
untar the downloaded files and run installation file

```
bash$ ferret
yes?
```

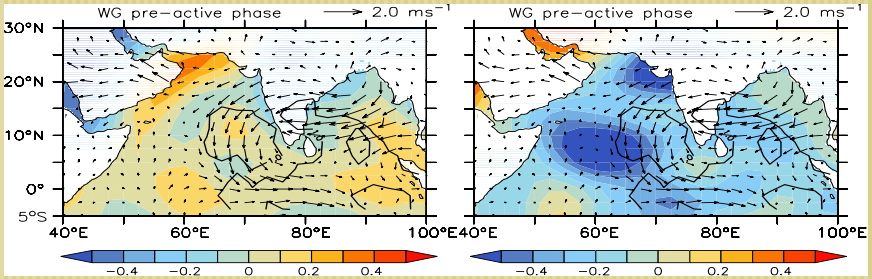
Visualizing your processed data in Ferret

Let's use the data processed by CDO, for monsoon precip.:

```
bash$ ferret

yes? set data out_jjas_mean.nc                    !open file
yes?  show data                    !will show the details of the file.
yes? fill precip                                  !plot variable

yes? go land                                      !draw land bounds

yes? quit
```



Ferret Usage



Attributes	Command	Example
1D plot	plot	`yes? plot sst_ann_cyc`

2D plot	shade/fill
	contour
	vector
		`yes? fill sst_jjas`
		`yes? contour/overlay precip_jjas`
		`yes? vector/overlay u_jjas,v_jjas`

Overlay	/overlay	
Land	go land go fland	`yes? go land`
Label	label x,y,centre,angle,size text	`yes? label 50,85,0,0,12 "SST"`

Others	levels
	limits:
	x=40:120/y=-20:50
	hlimits=40:120/vlimits=-20:50
	time=01-jun-2000
	color palette
	palette=grey_scale
		`yes? fill/levels=(10,30,1) sst_jjas`
		`yes? fill/x=40:120/y=-20:30 sst_jjas`
		`yes? vector/overlay/k=1 u_jjas,v_jjas`
		`yes? fill/pal=grey_scale sst_jjas`

Saving your plots in Ferret

```
yes? frame/file=filename.gif
```

Alternatively, you can save images as **postscript** file using the metafile options in ferret.

Before plotting:

```
yes? set mode metafile
```

After plotting:

```
yes? cancel mode metafile
```

```
yes? sp Fprint -o filename.ps metafile.plt
```

Reference and Assignments