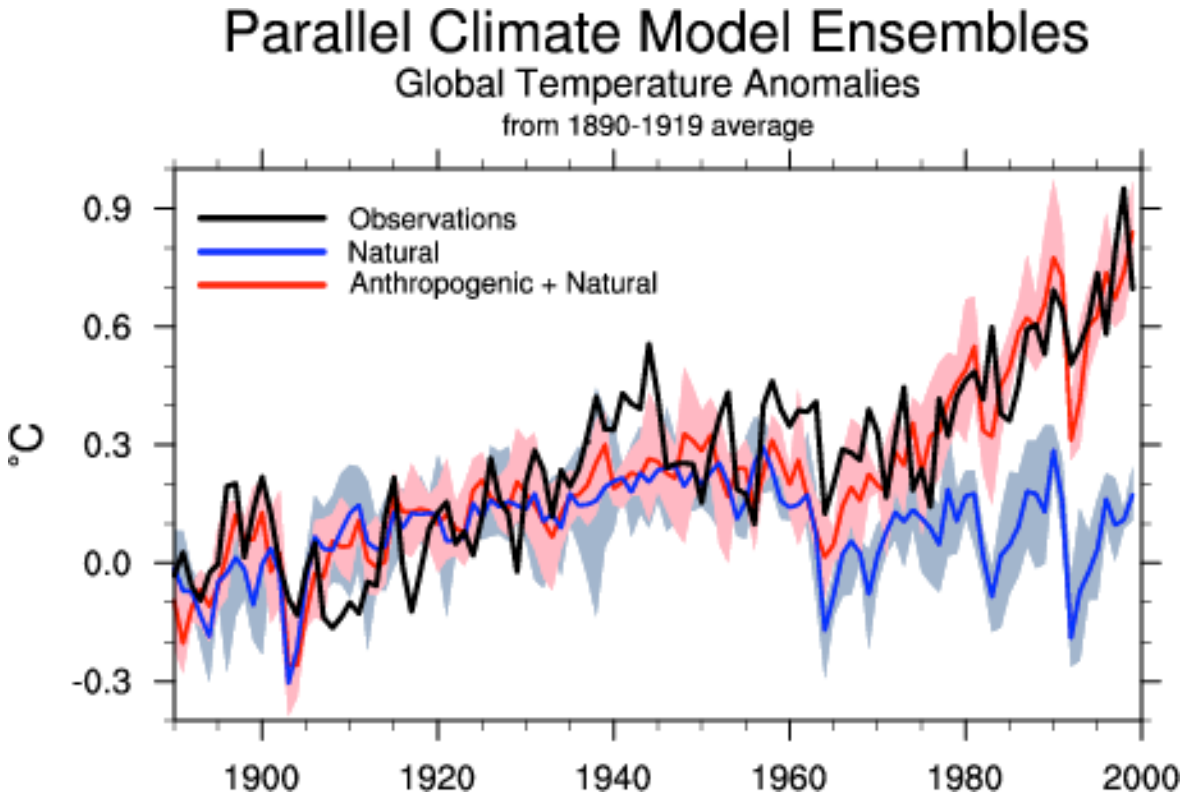
Class #6 Friday 4 February 2011

•What did we discuss last time?

- Loose Ends
  - Functions versus Procedures (returns 1 value, returns many values)
  - Psi/Chi - Streamfunction, Velocity Potential
  - WRAPIT, still not working, will show you later

1. NCL Graphics
2. NCO
3. Grads printing issue

Next time Basic statistical analysis



XY Plots

gsn_csm_xy

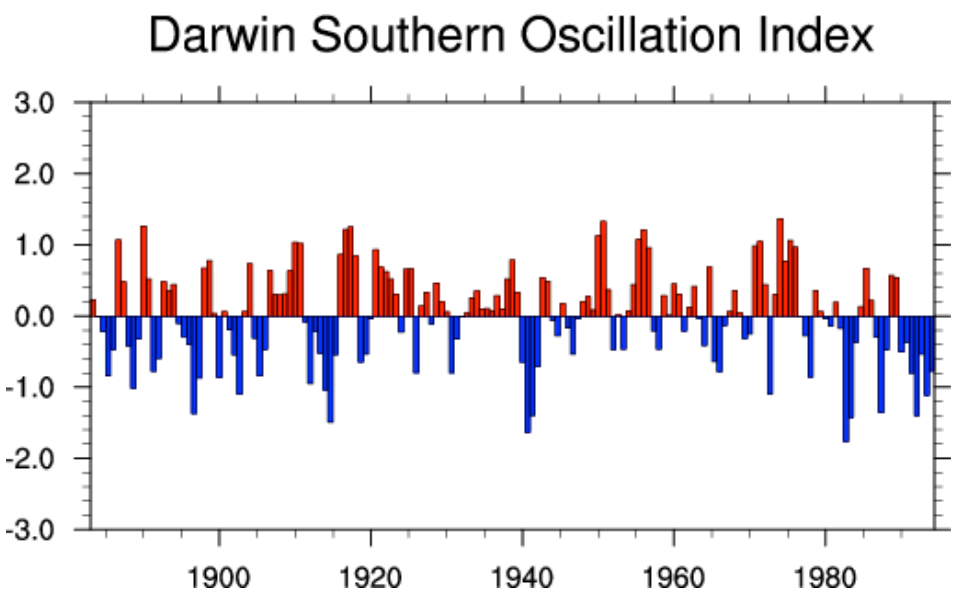
gsn_csm_y

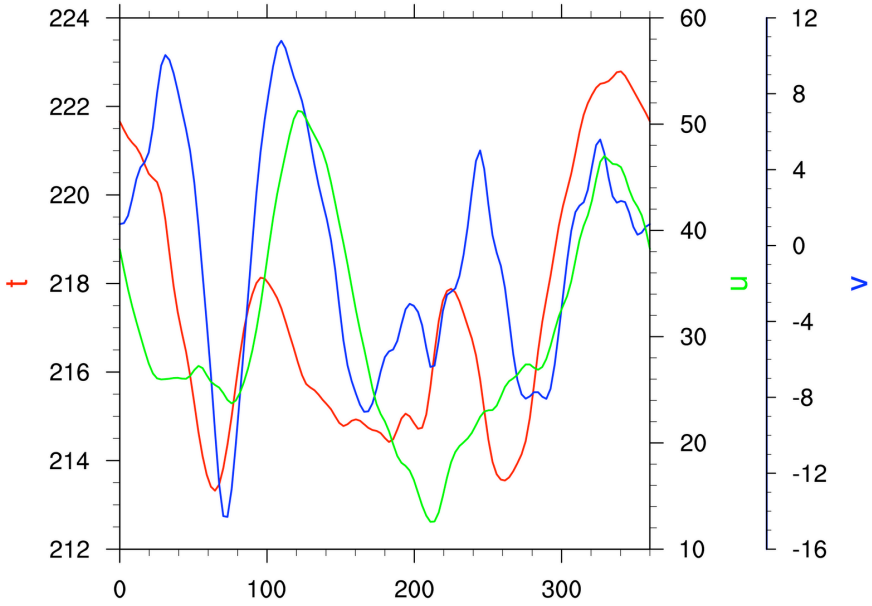
gsn_csm_xy2

gsn_csm_x2y

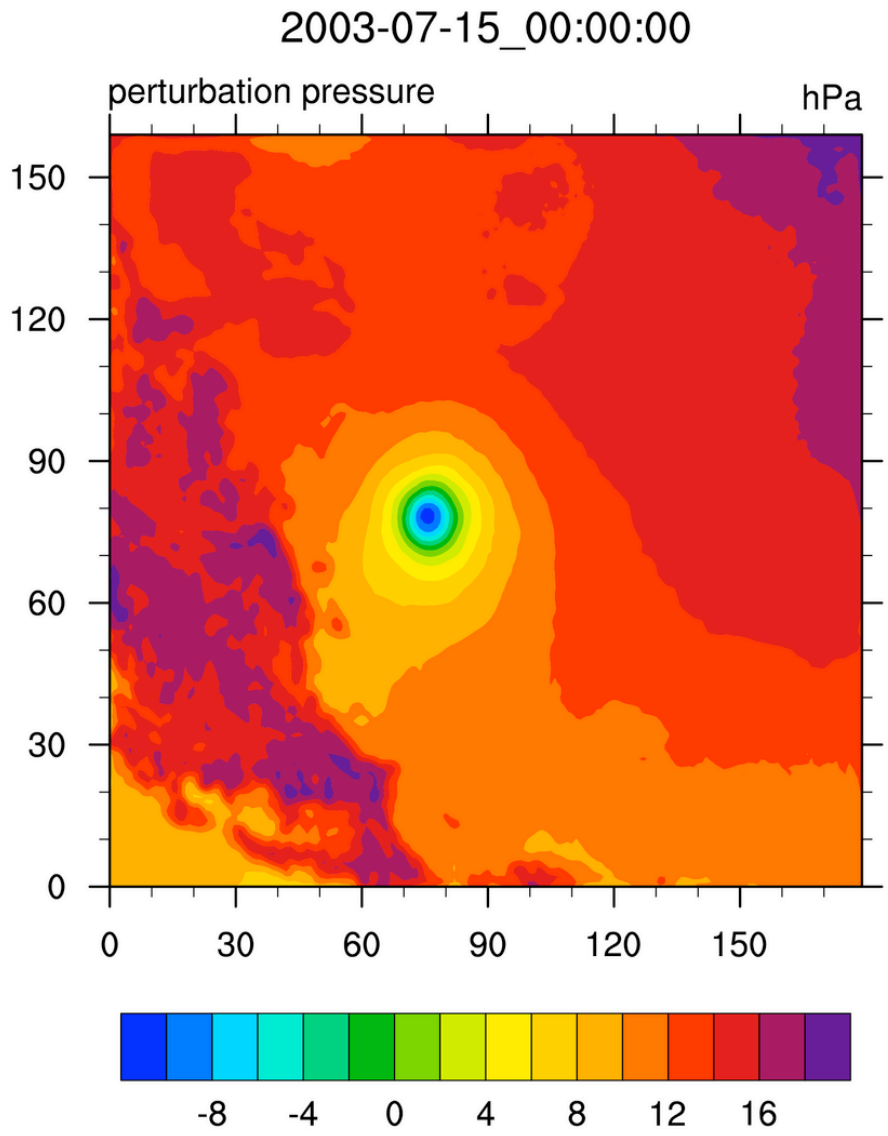
gsn_csm_x2y2

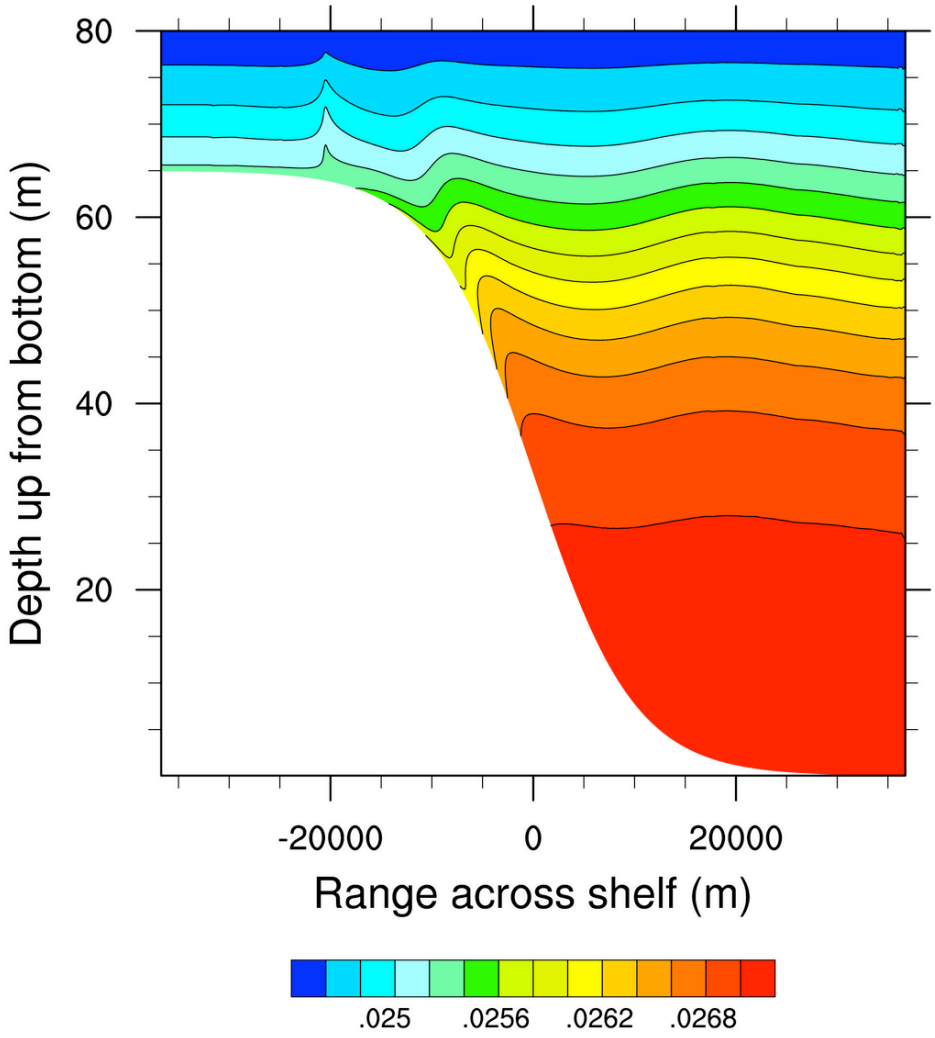
gsn_csm_xy3





Contour plots
gsn_csm_contour

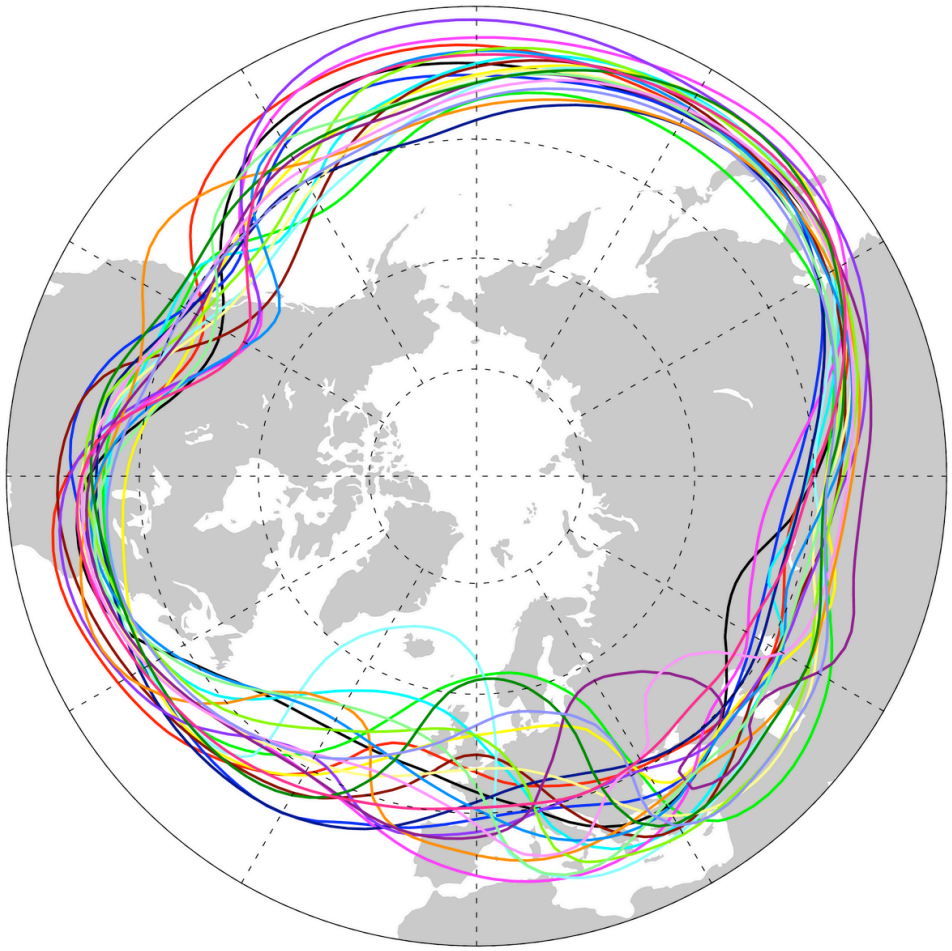


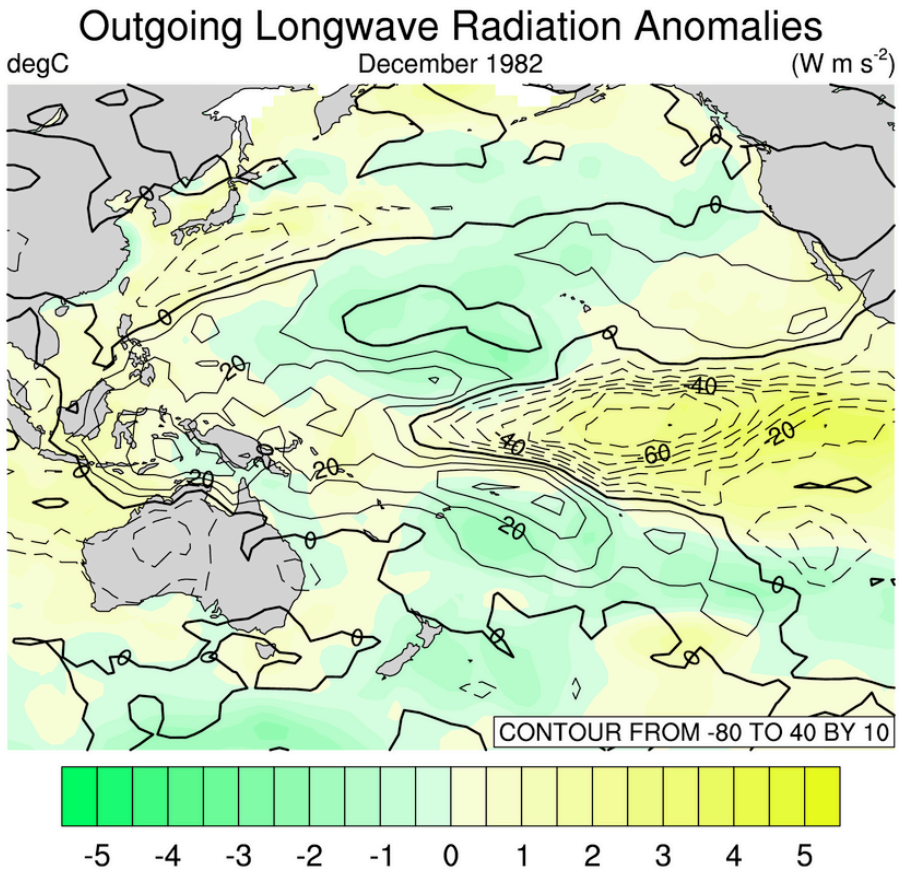


Contour over maps

gsn_csm_contour_map • gsn_csm_contour_map_ce

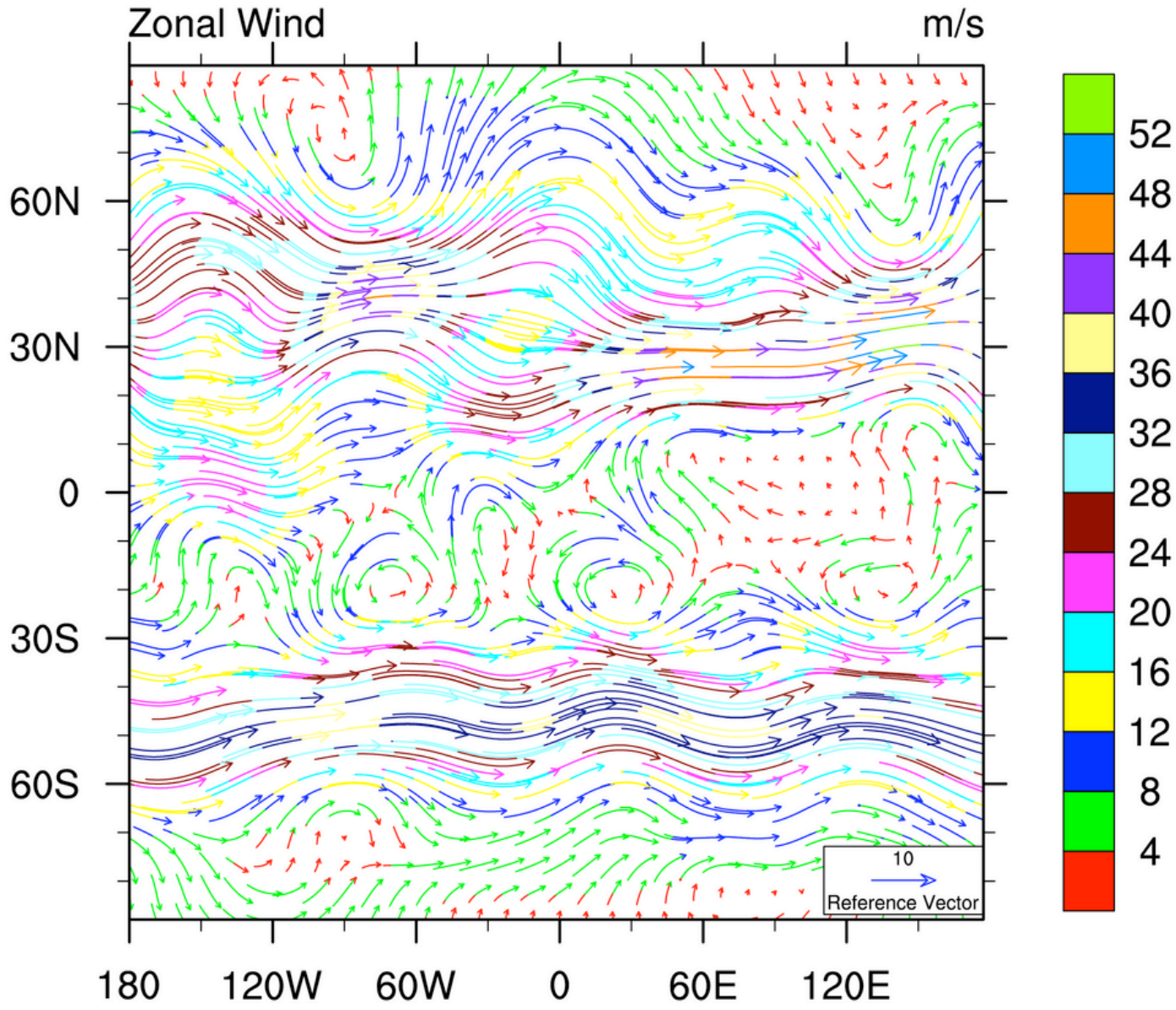
•gsn_csm_contour_map_polar • gsn_csm_contour_map_overlay



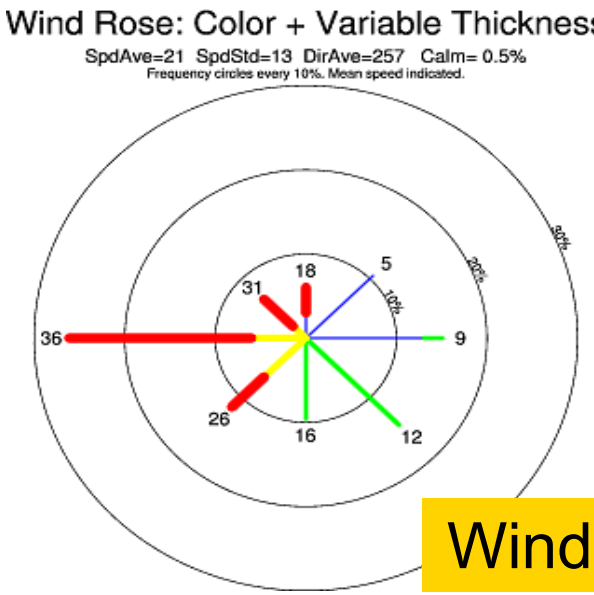


Vector plots

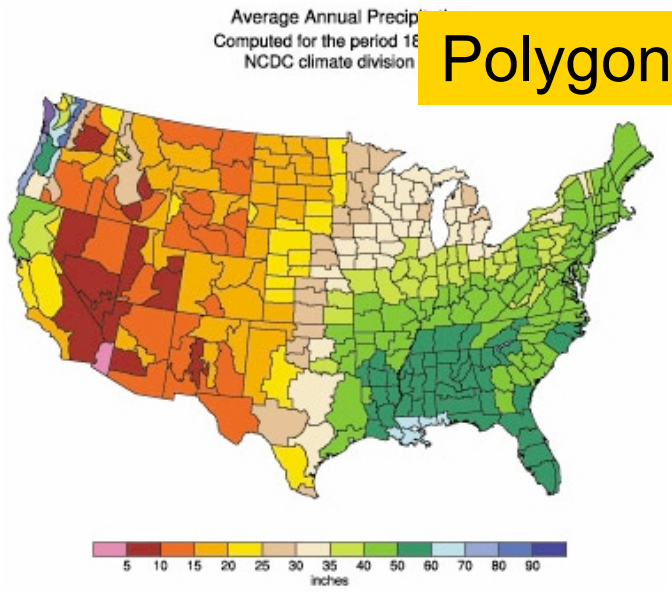
gsn_csm_vector • gsn_csm_pres_hgt_vector • gsn_csm_vector_scalar



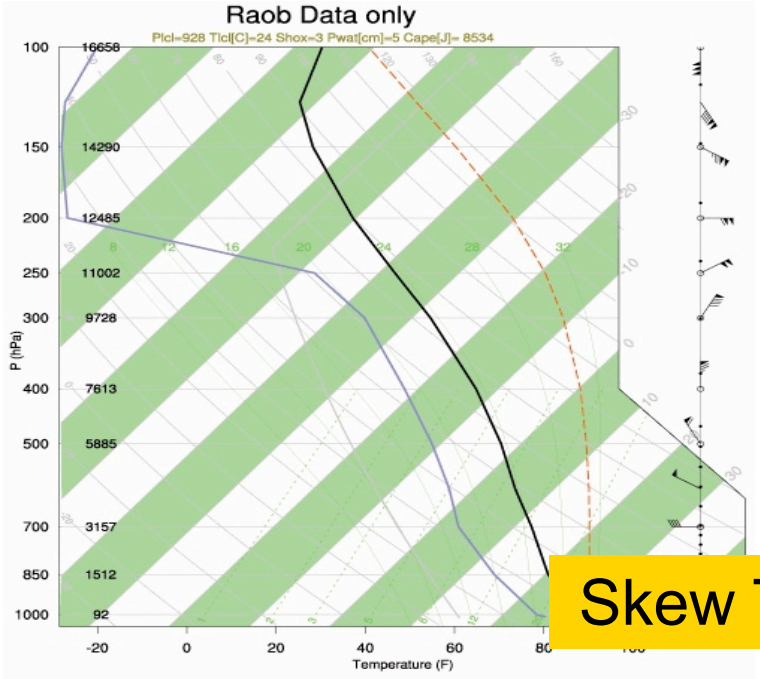
Special Templates and Scripts



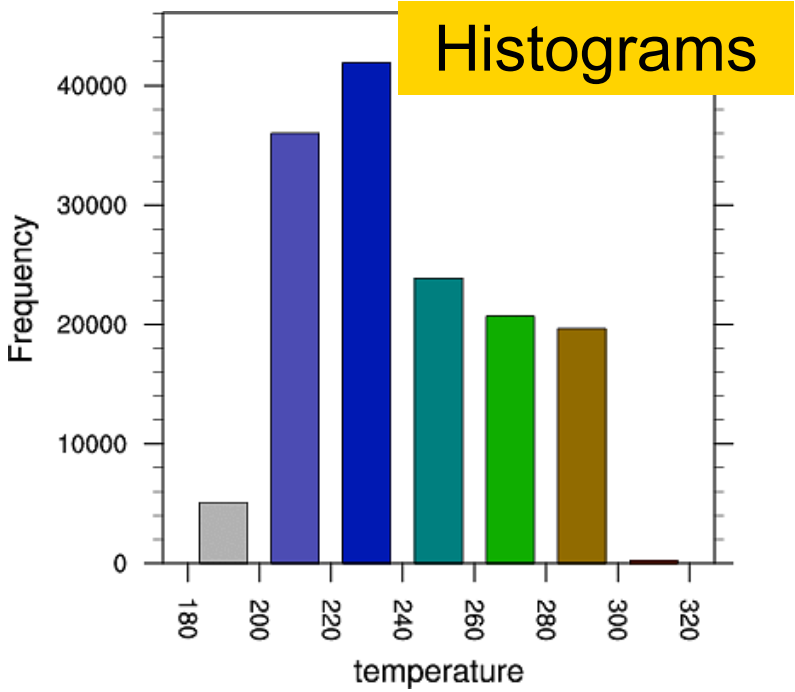
Wind Rose



Polygons



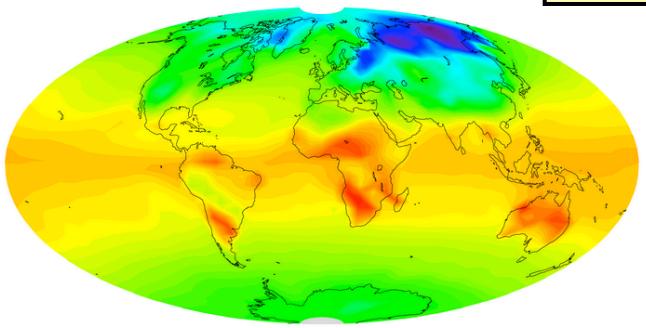
Skew T



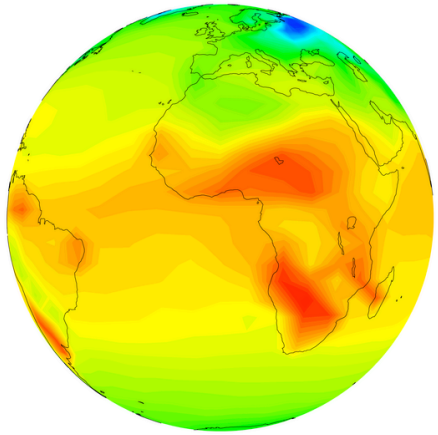
Histograms

16 map projections (8 here)

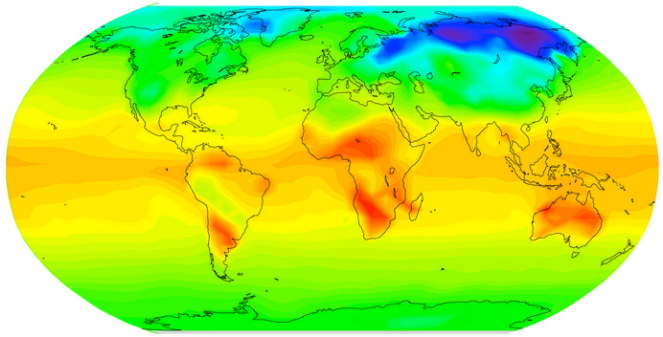
Aitoff



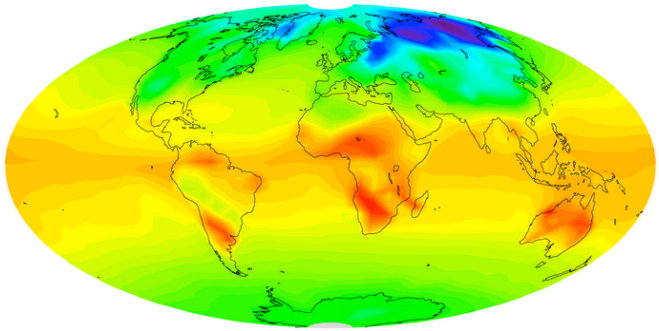
Satellite



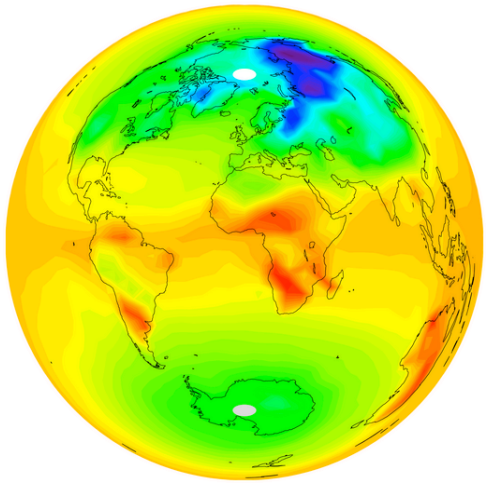
Robinson



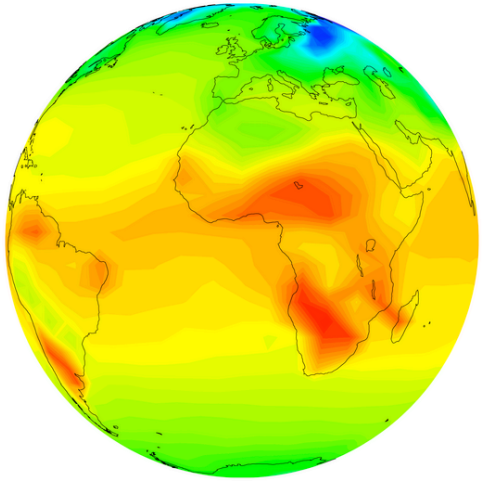
Hammer



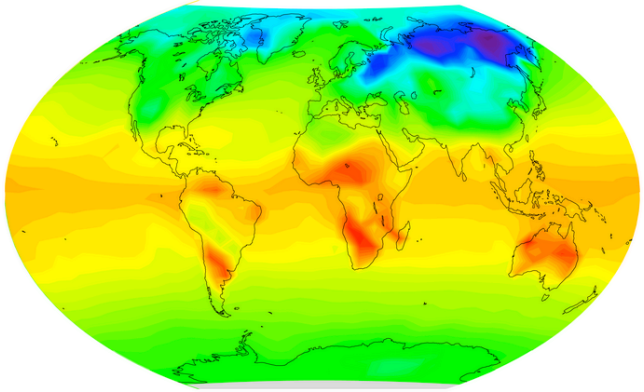
LambertEqualArea



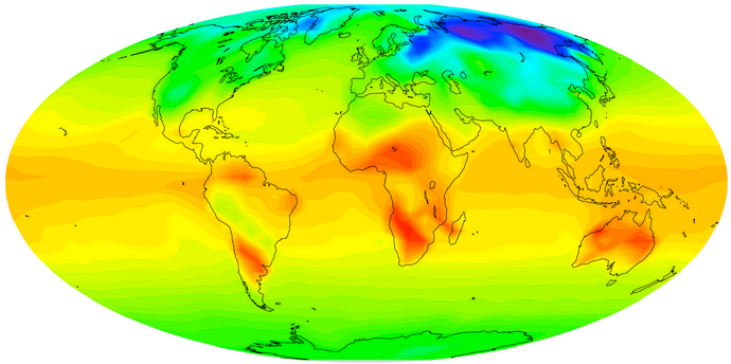
Orthographic



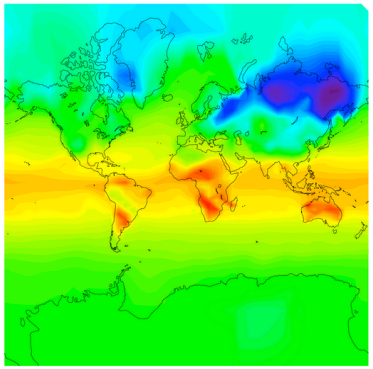
WinkelTripel



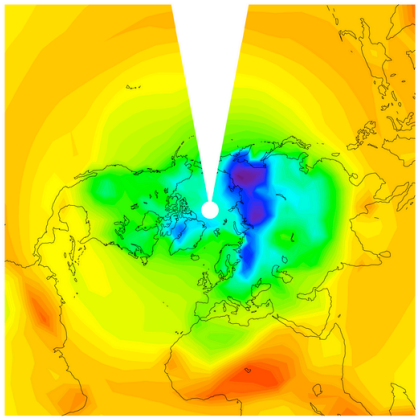
Mollweide



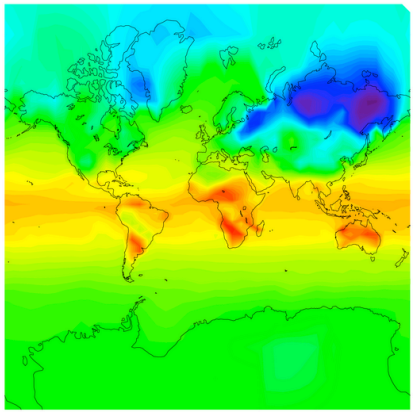
Mercator



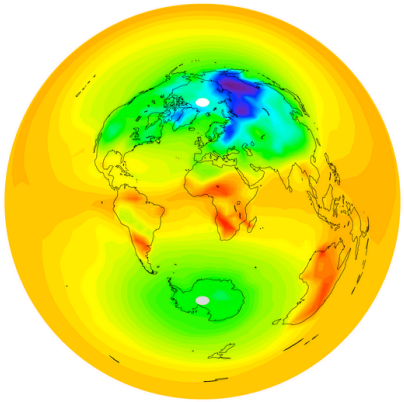
LambertConformal



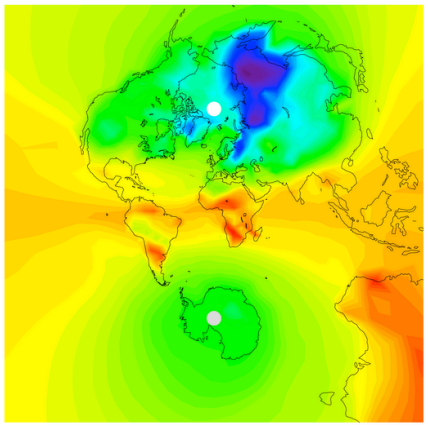
RotatedMercator



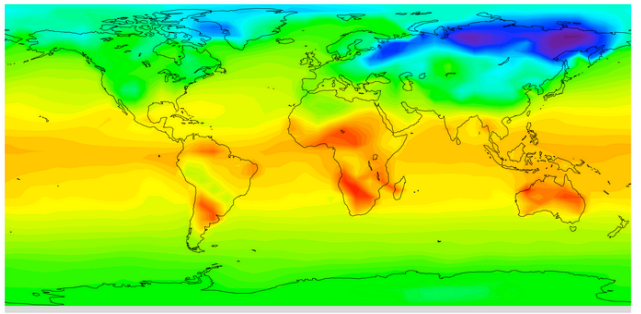
AzimuthalEquidistant



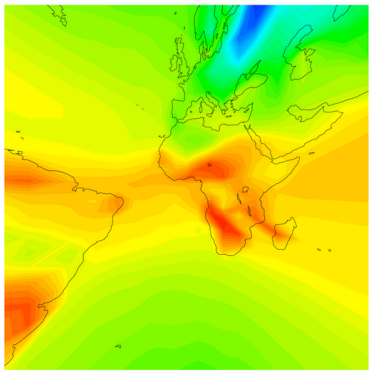
Stereographic



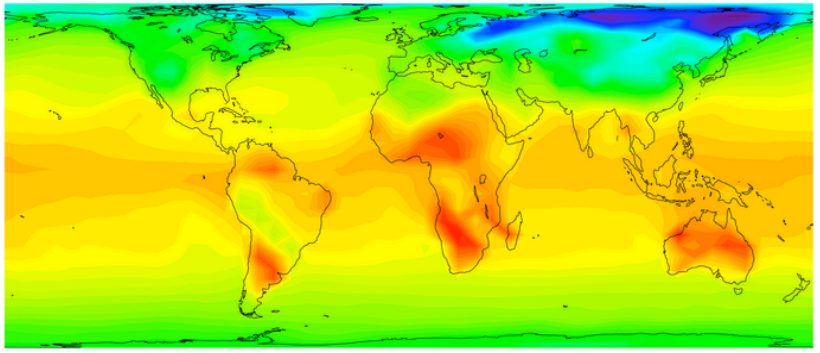
CylindricalEquidistant



Gnomonic



CylindricalEqualArea

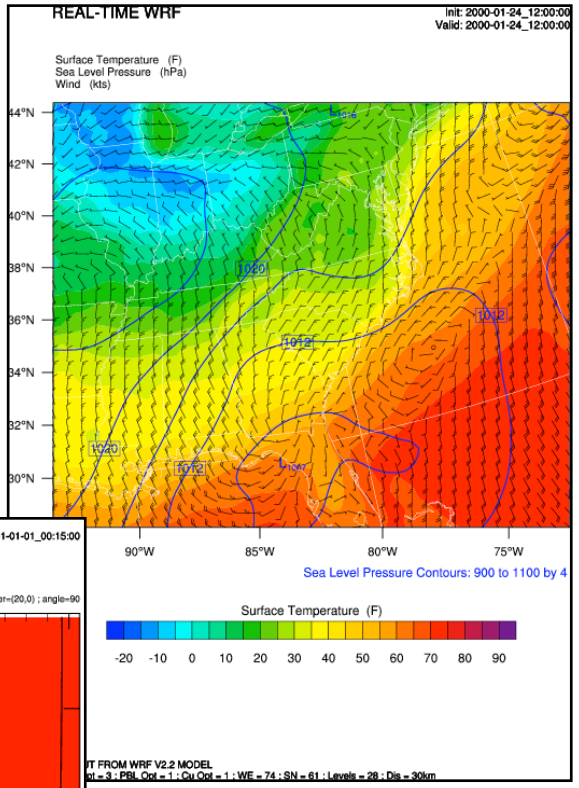


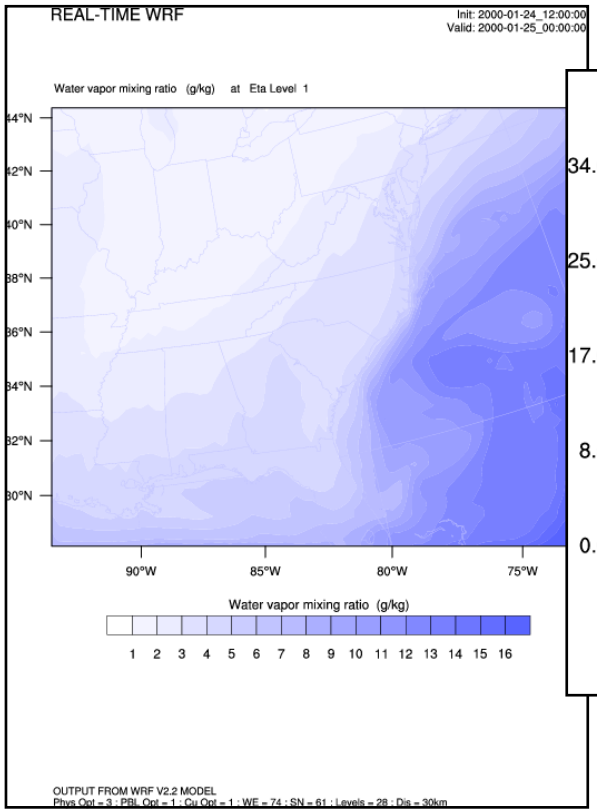
WRF Plots

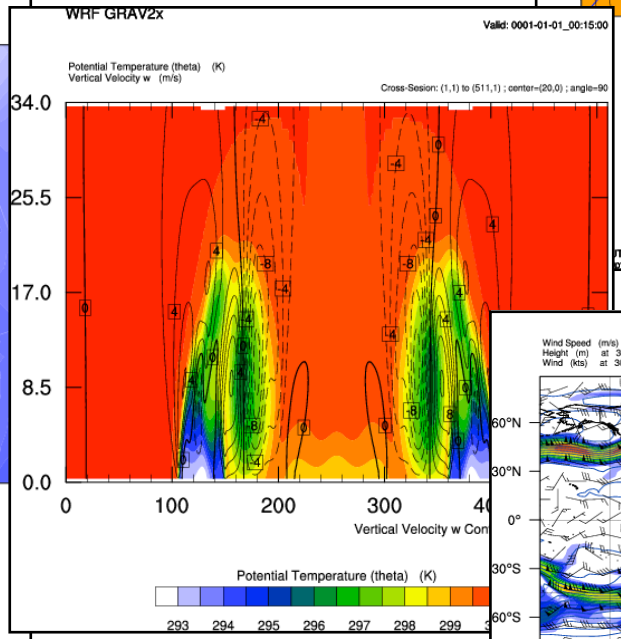
wrf_contour • wrf_map • wrf_vector •

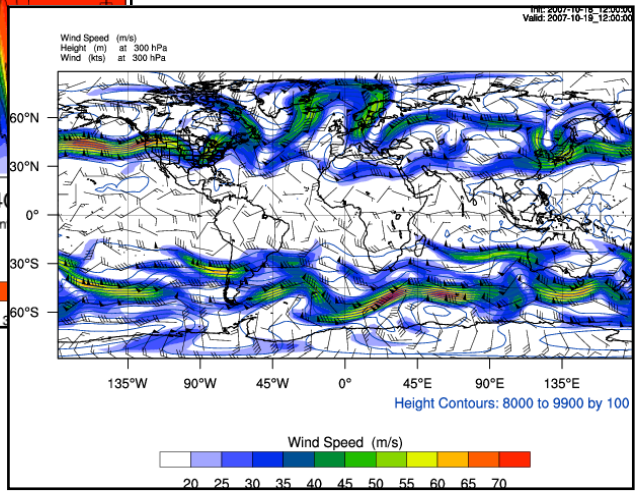
wrf_map_overlays • wrf_overlays

http://www.mmm.ucar.edu/wrf/OnLineTutorial/Graphics/NCL/NCL_examples.htm

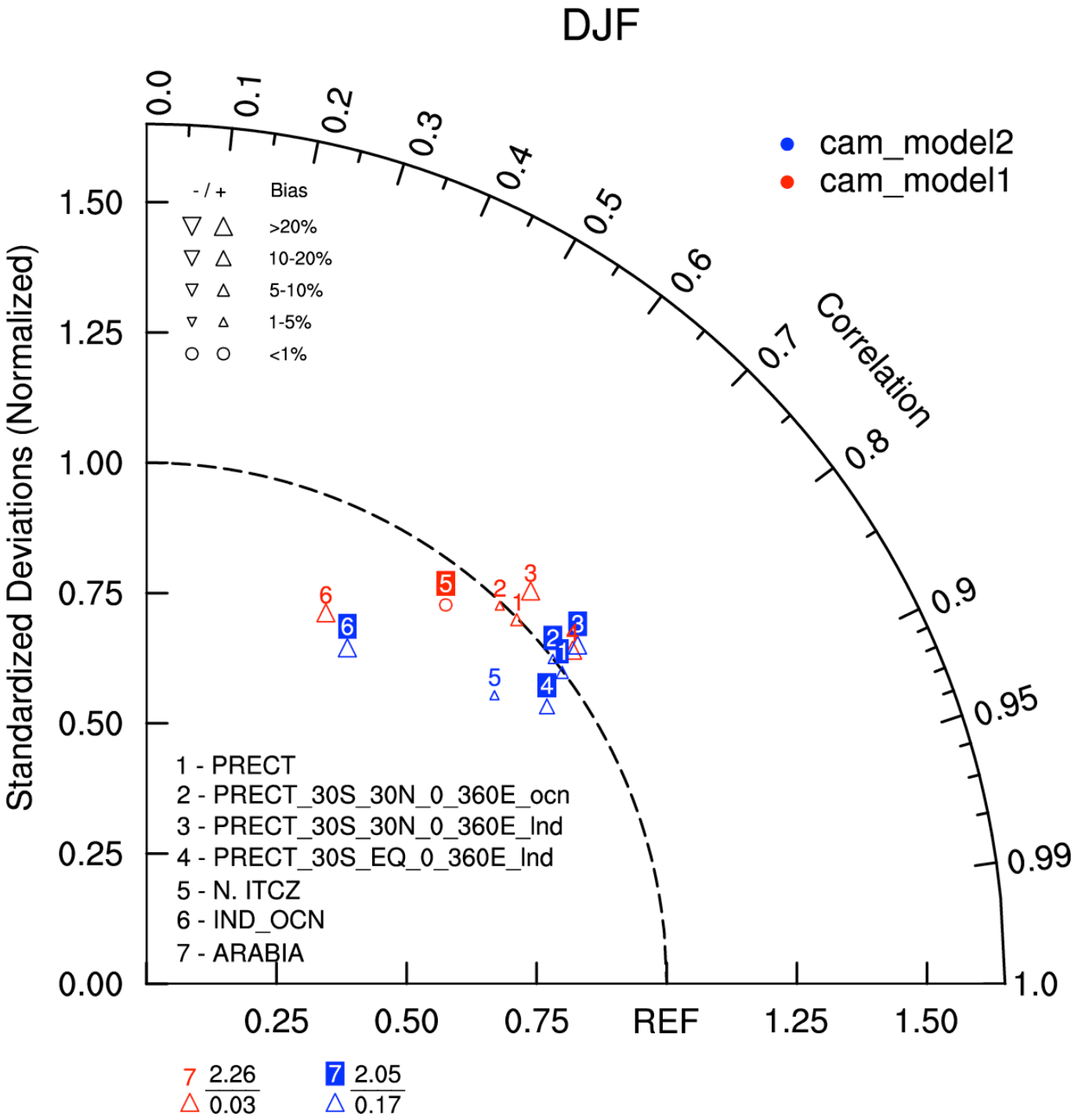






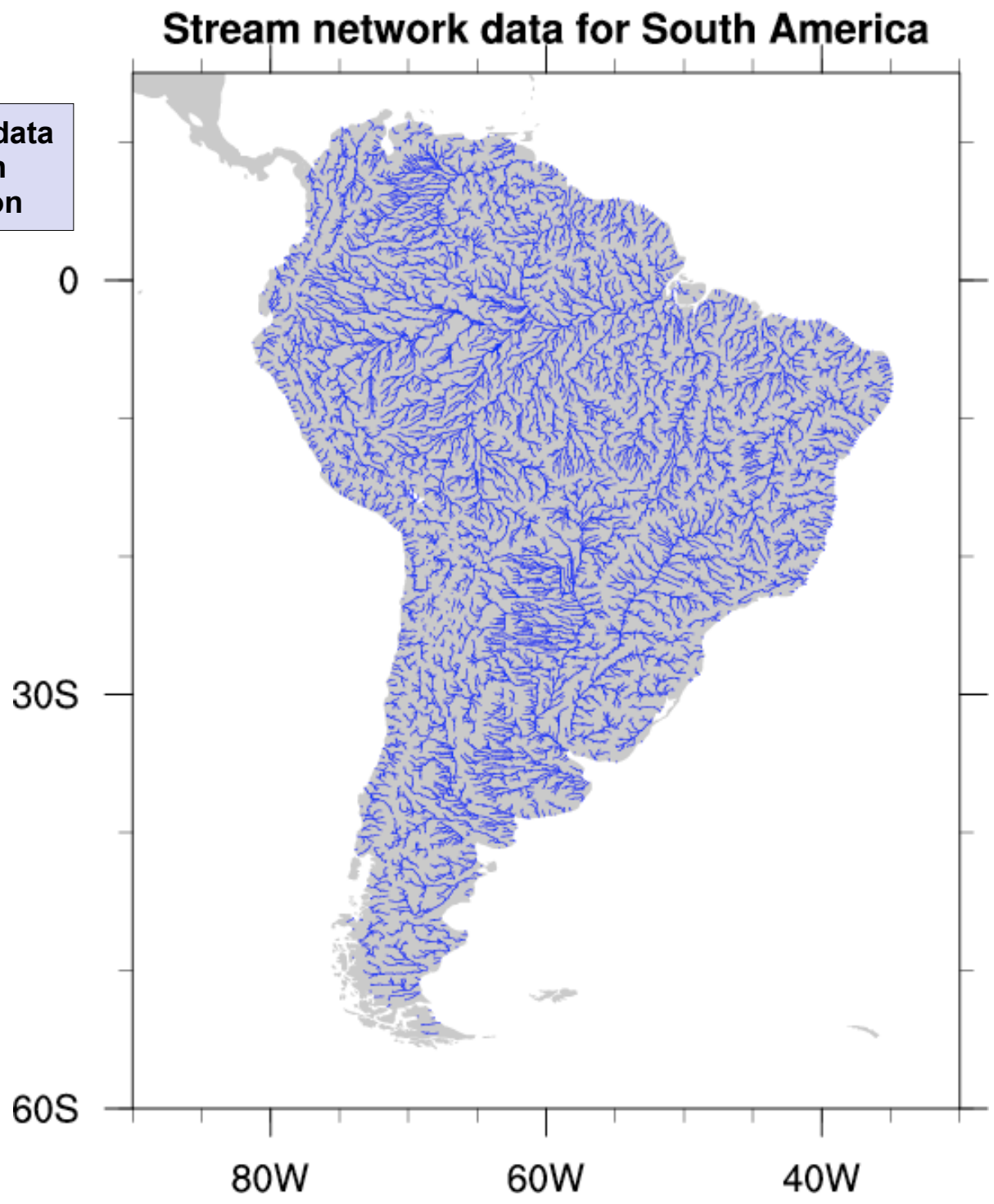


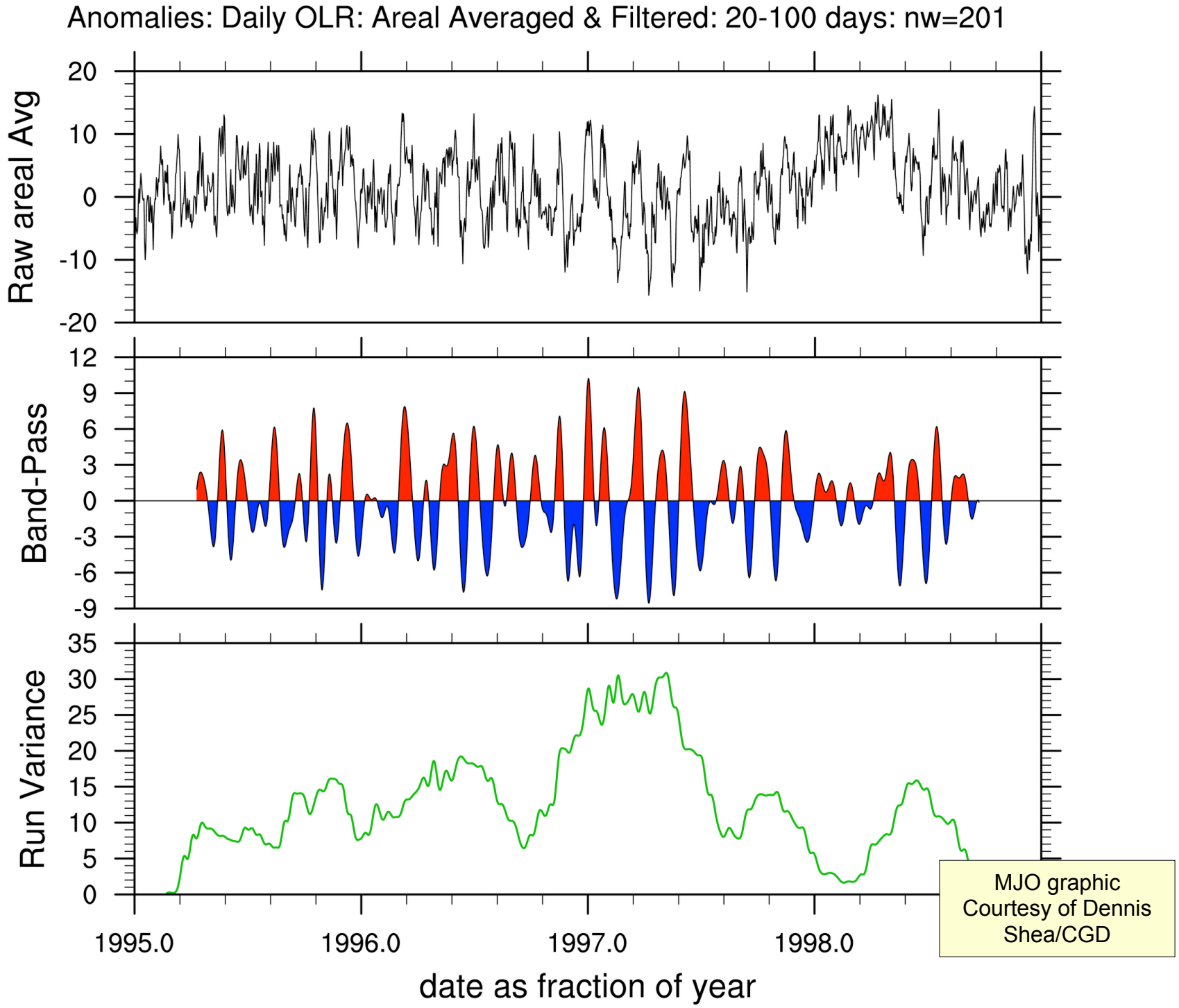
Taylor Diagram

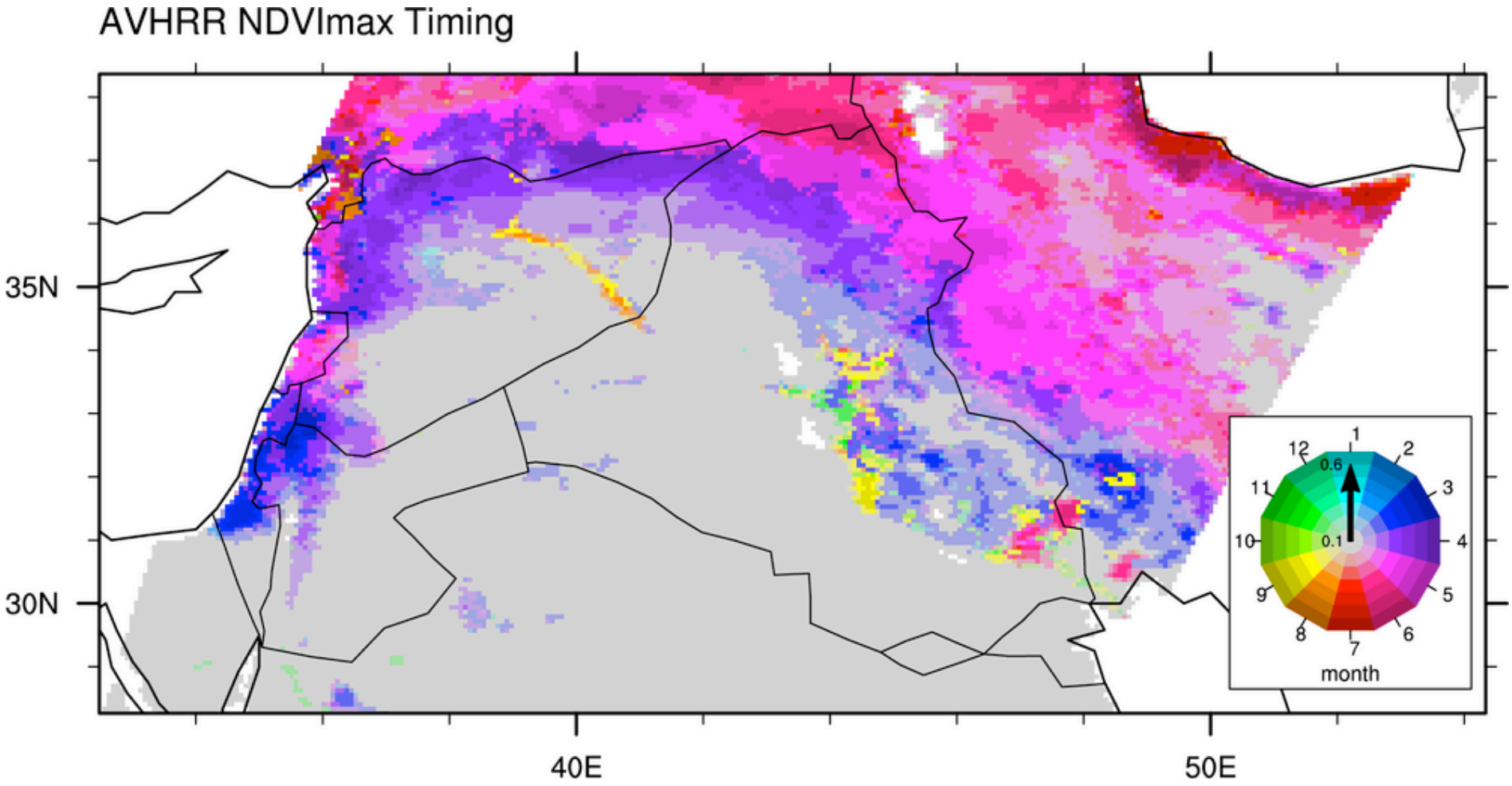


Taylor diagram Courtesy of Dennis Shea and Adam Phillips, CGD

Using “shapefile” data to add your own lat/lon information

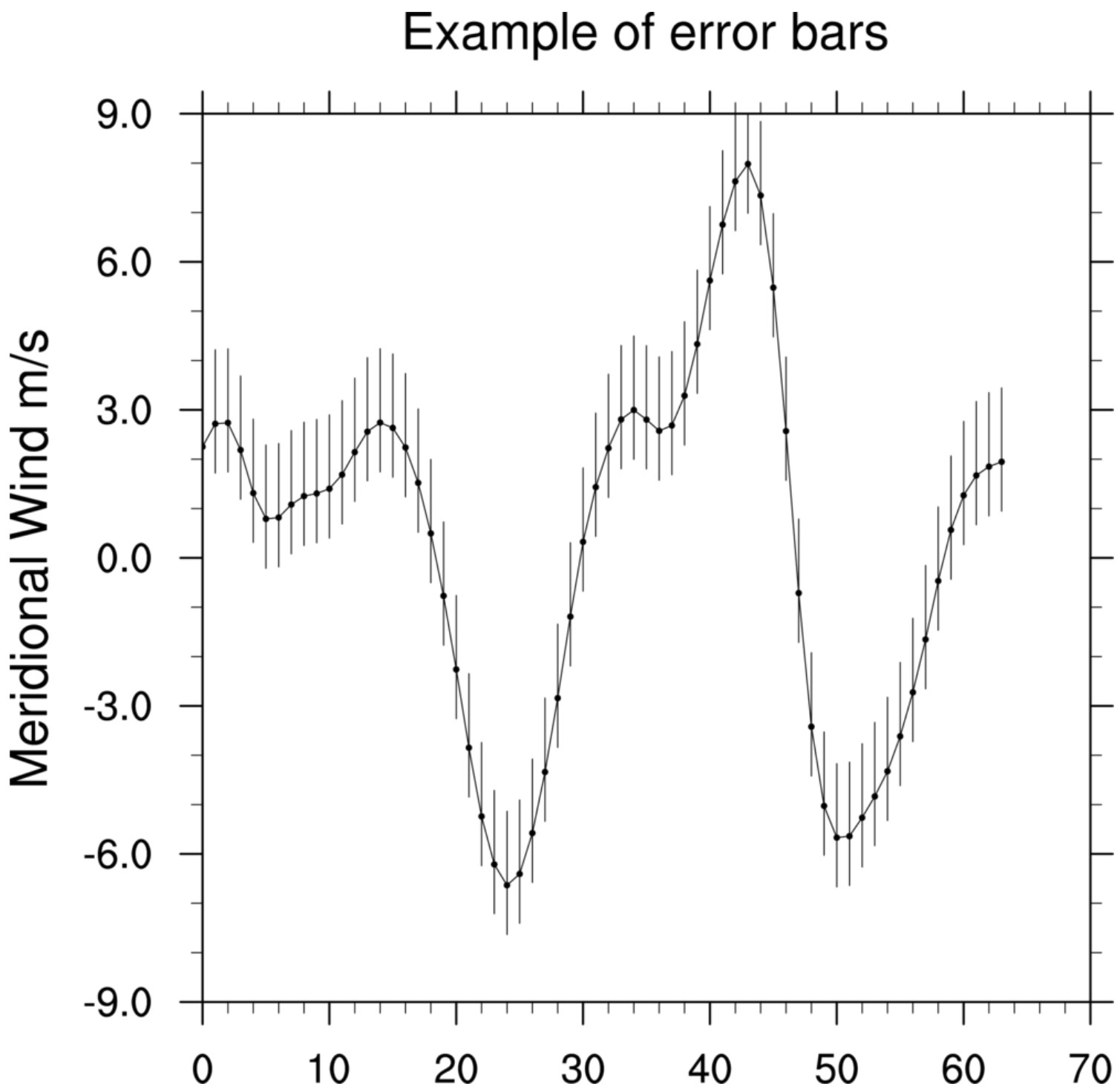


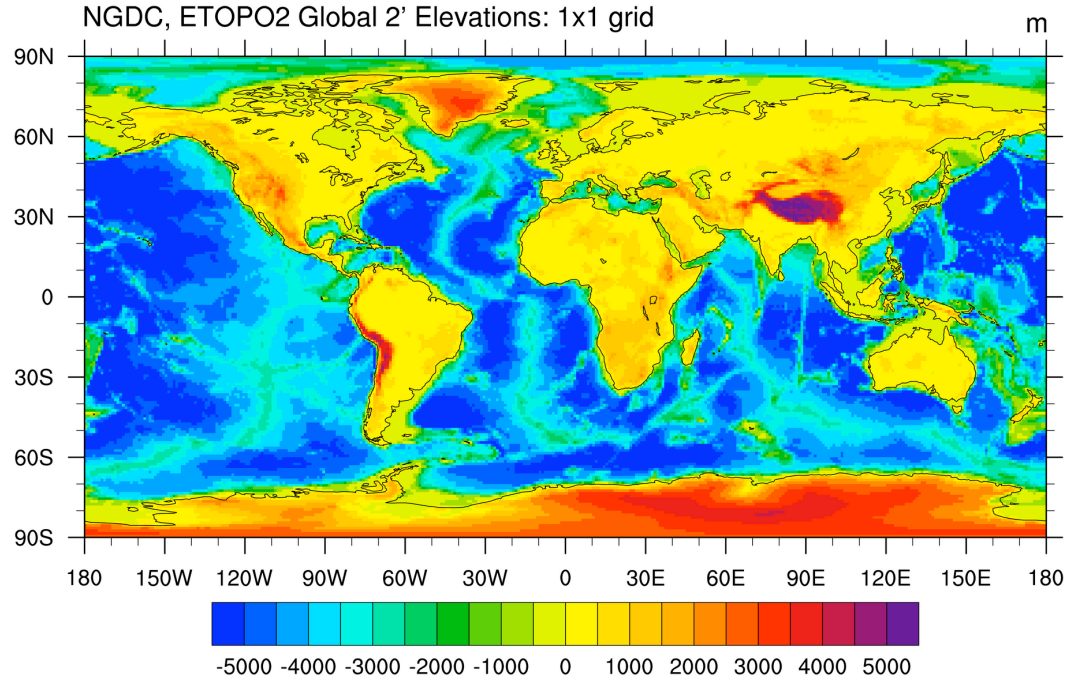

Anomalies: Daily OLR: Areal Averaged & Filtered: 20-100 days: nw=201
Raw areal Avg
20
10
0
-10
-20
Band-Pass
12
9
6
3
0
-3
-6
-9
Run Variance
35
30
25
20
15
10
5
0
1995.0
1996.0
1997.0
1998.0
date as fraction of year
MJO graphic
Courtesy of Dennis
Shea/CGD

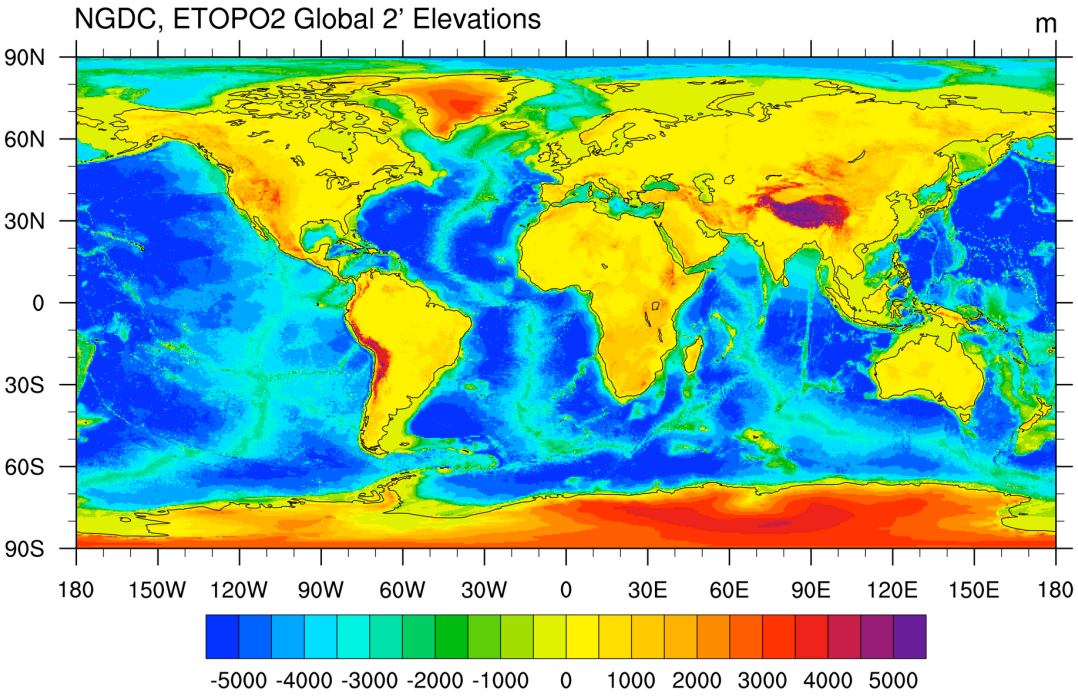


Evans plot - Created by Jason Evans of Yale University.

An Evans plot is a way to visualize spatially, two variables of interest, one of which provides some measure of "importance".

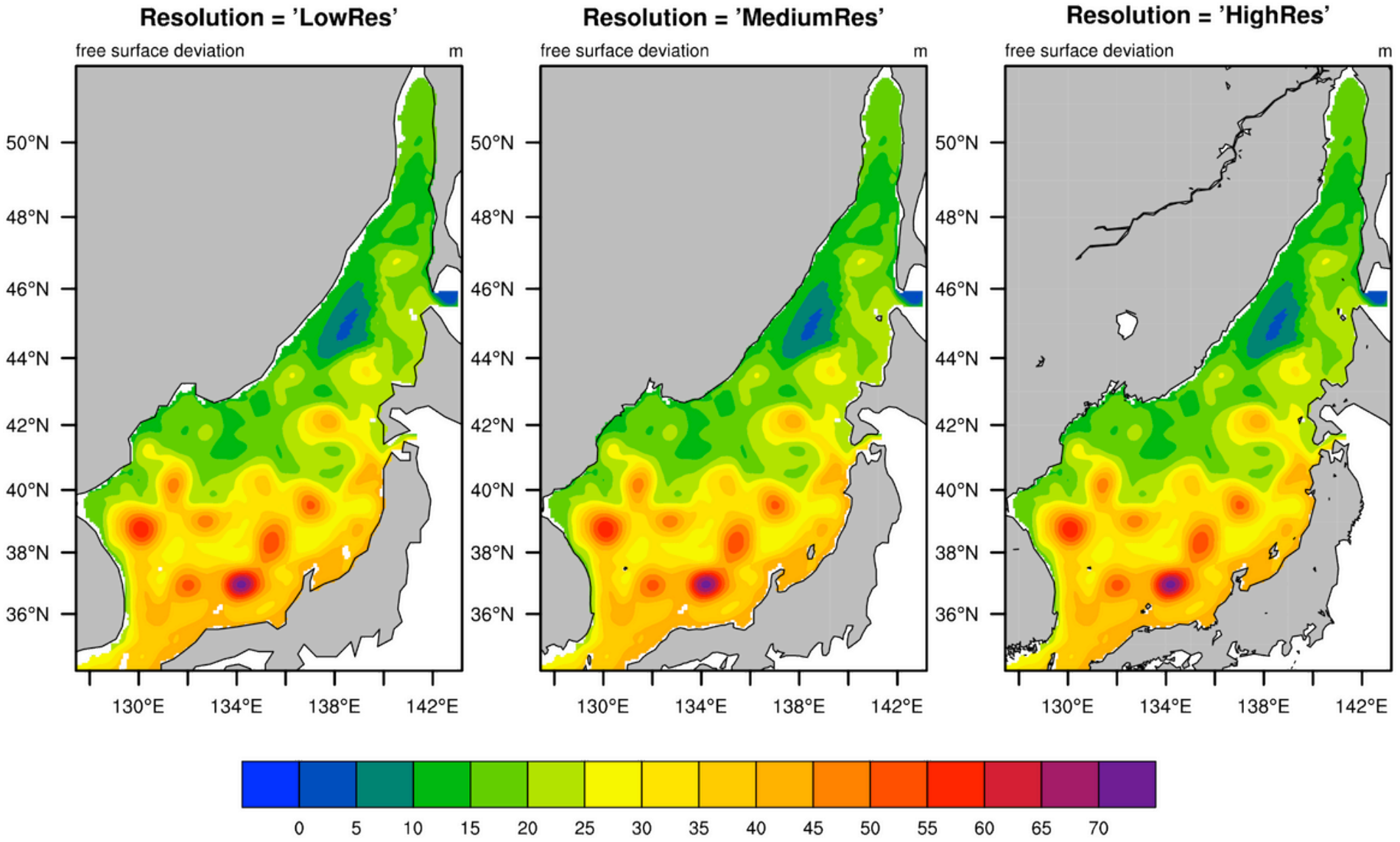

Example of error bars
Meridional Wind m/s
9.0
6.0
3.0
0.0
-3.0
-6.0
-9.0
0
10
20
30
40
50
60
70





Interpolating from a higher resolution grid to a lower resolution using conservative remapping courtesy Dennis Shea NCAR/CGD

Comparison of coastline resolutions



First two map databases built-in; high-resolution available as simple download

NCL Graphics - the basics

- Minimum steps needed to create a plot
- How resources (plot options) work
- Special topics:
  - resizing
  - paneling
  - annotations
  - function codes (superscripts, subscripts)
  - creating images for web/PowerPoint
- Exercises, example scripts, and data
- Useful documentation links

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  y = sin(0.0628*ispan(0,100,1))   ; 101 points

  wks = gsn_open_wks("ps","test")  ; 'test.ps'
  gsn_define_colormap(wks,"rainbow")

  res                  = True          ; plot options
  res@xyLineColor = "Blue"         ; line color
  plot = gsn_csm_y(wks,y,res)    ; no X values
end
```

1. Load the necessary libraries

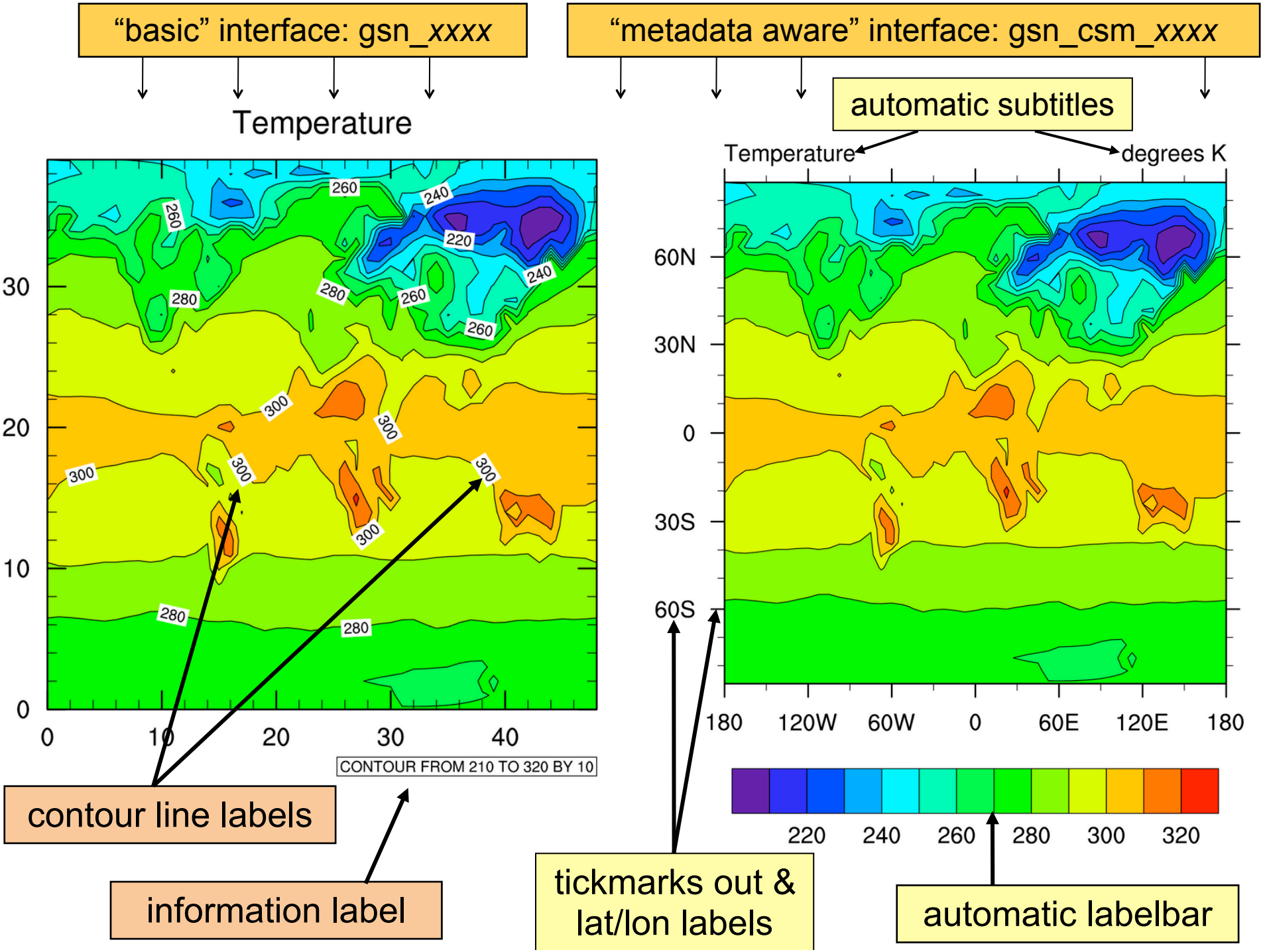
1.5 Get some data!

2. Open a workstation

3. Change color map

4. Set plot options

5. Call the graphical function


"basic" interface: gsn_xxxx
"metadata aware" interface: gsn_csm_xxxx
Temperature
automatic subtitles
Temperature
degrees K
60N
30N
0
30S
60S
180
120W
60W
0
60E
120E
180
220
240
260
280
300
320
CONTOUR FROM 210 TO 320 BY 10
contour line labels
information label
tickmarks out & lat/lon labels
automatic labelbar

Step 2: Open graphics "workstation"

- Can be PostScript (PS/EPS), PDF, X11 window, or NCGM (**new: PNG**)
- Has a default color map associated with it,

```
wks = gsn_open_wks("x11","test") ; X11 window

wks = gsn_open_wks("ps","test")  ; "test.ps"

wks = gsn_open_wks("png","wrf")  ; "wrf.00001.png"

wks = gsn_open_wks("pdf","slp")  ; "slp.pdf"

wks = gsn_open_wks("ncgm","cn")  ; "cn.ncgm"
```

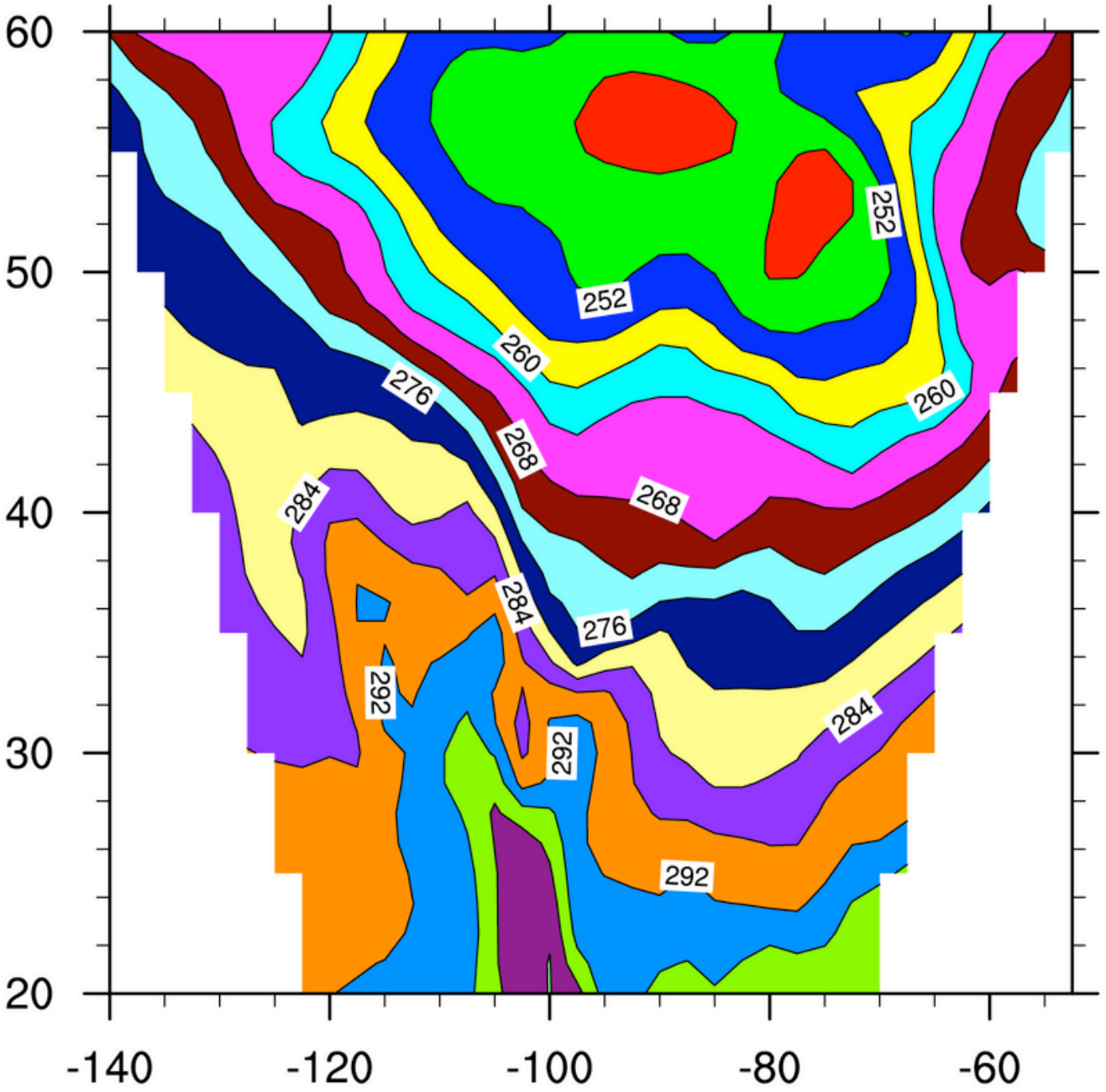
Step 3:
Change the color map (opt'l)

- Do this before drawing to the frame.

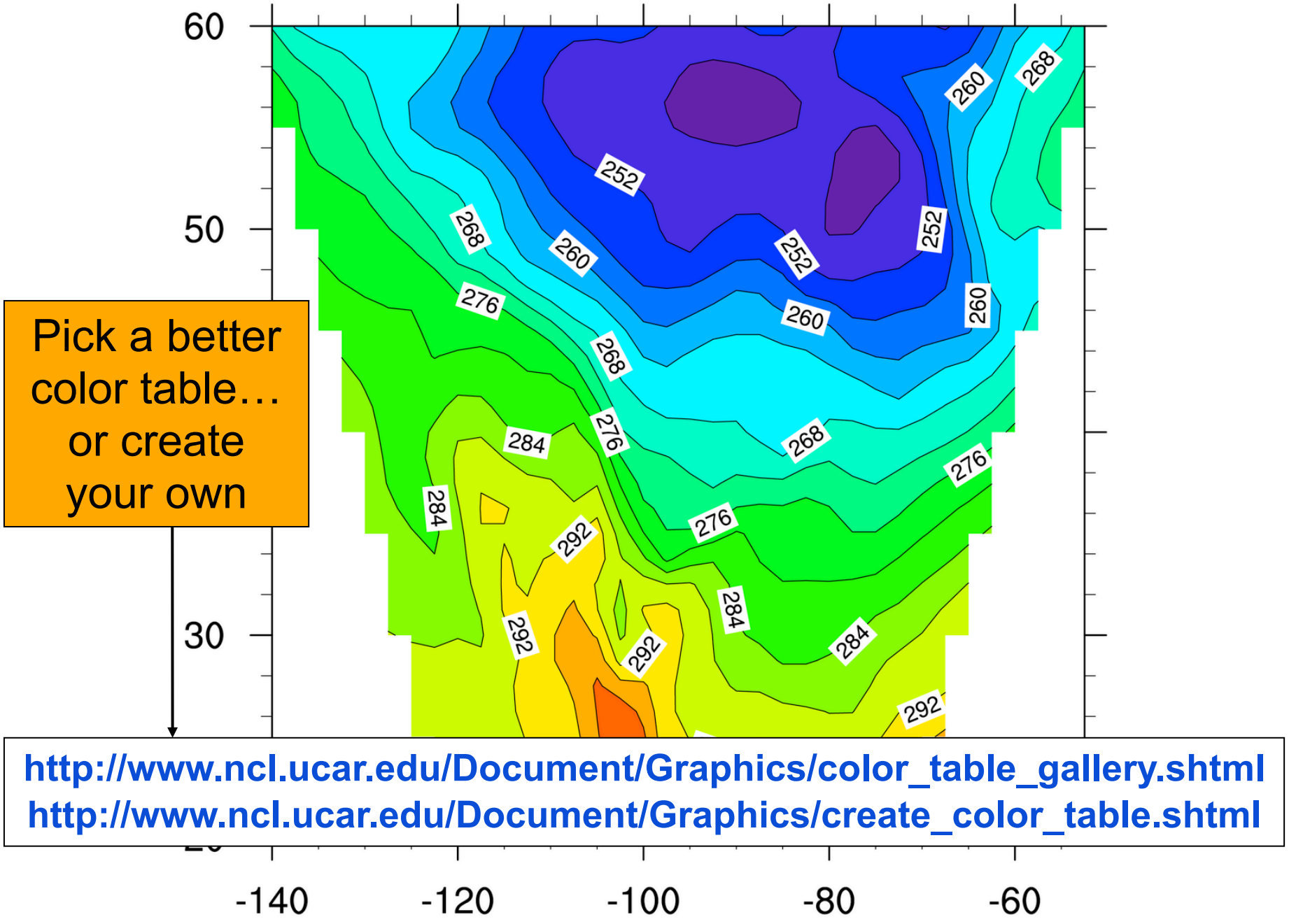
```
gsn_define_colormap(wks,"rainbow")
```

- If you use the same color map a lot, can put in ".hluresfile" (more later)
- Can use one of the other 90+ color maps, or create your own.
- If you don't change the color map, here's what you'll get…

Default color table (yuck)



Better color table



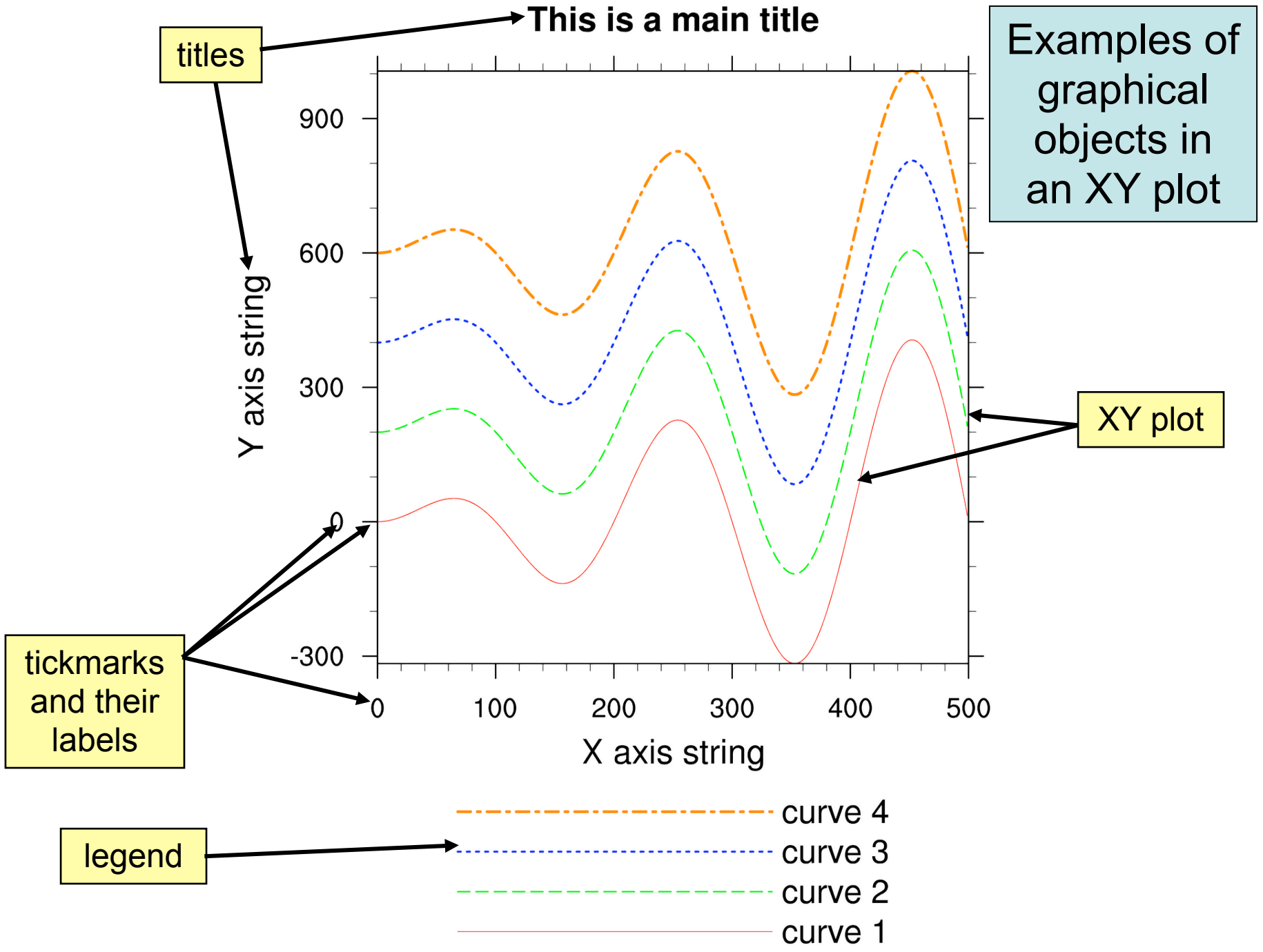
Default color table

Index 0 is the background color

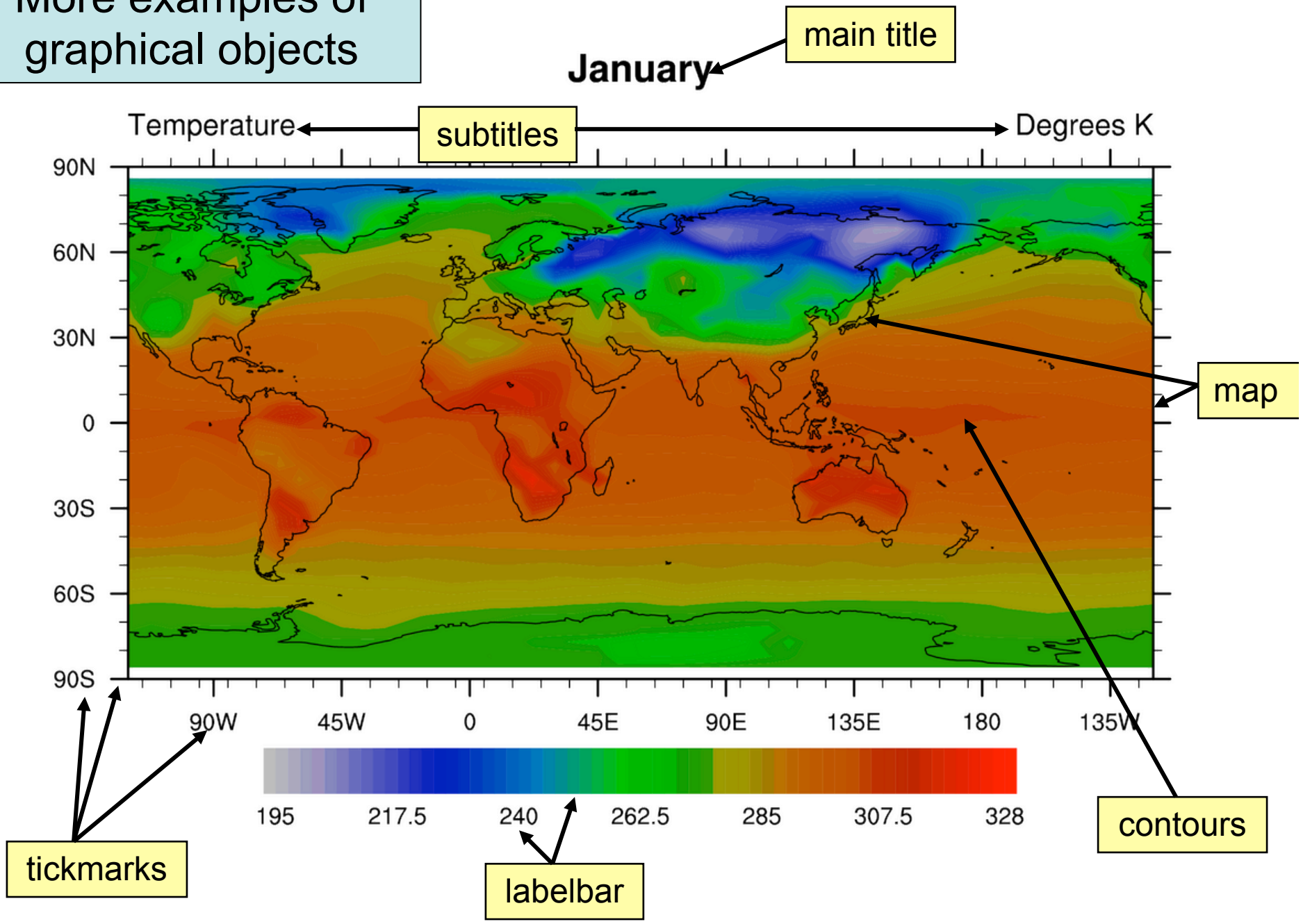
Index 1 is the foreground color

Step 4:
Set optional resources

- Resources are the heart of your NCL graphics code.
- There are over 1,400 resources!
- Resources are grouped by object type.
- There are 11 “graphical” objects: contours, labelbars, legends, maps, primitives, streamlines, text strings, tickmarks, titles, vectors, XY plots


Examples of graphical objects in an XY plot
titles
This is a main title
Y axis string
900
600
300
0
-300
0
100
200
300
400
500
X axis string
XY plot
tickmarks and their labels
legend
curve 4
curve 3
curve 2
curve 1

More examples of graphical objects



How a resource is constructed

- Starts with 2 or 3 lower-case letters based on object it is associated with. Some examples:

  "xy" - XY Plot "cn" - Contour plot

  "vc" - Vector plot "ti" - Title

  "tm" - Tickmark "lb" - Labelbar

- Made up of full words; first letter capitalized:
  - "xyLineColor", "cnFillOn", "tiMainString", "vcRefMagnitudeF", "gsnMaximize"
- Some have an "F" on the end to indicate a floating point resource: "xyLineThicknessF"
- "gsn" – special resources

How a resource is constructed (cont'd)

- Resources are set by attaching them as attributes to an NCL *logical* variable:

  res = True ; can name it whatever you want

  res@mpMinLatF = 30 ; decimal not necessary
- Most have default values.
- There are many types:
  - res@tiMainString = "This is a title"
  - res@tmXBLabelFontHeightF = 0.01
  - res@cnLineLabelsOn = True
  - res@xyLineColors = (/5,7,11/)

http://www.ncl.ucar.edu/Document/Graphics/Resources/

How a resource is constructed (cont'd)

- Resources across objects are similarly named for easier recollection:
  - xyLineColor, cnLineColor, gsLineColor, mpGridLineColor, tmBorderLineColor
  - tiMainFontHeightF, tmXBLabelFontHeightF, lbLabelFontHeightF, cnLineLabelFontHeightF
  - xyDashPattern, mpPerimLineDashPattern, lbBoxLineDashPattern, cnLineDashPattern

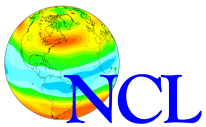
and so on…

Step 5:
Draw the graphics

- Call one of the gsn_csm_*xxxxx* functions from the second library we loaded.
- Some examples:

```
xy   = gsn_csm_xy(wks,x,y,res)
plot = gsn_csm_contour(wks,data,res)
plot = gsn_csm_vector(wks,u,v,res)
map  = gsn_csm_vector_map(wks,u,v,res)
phgt = gsn_csm_pres_hgt(wks,data,res)
```

http://www.ncl.ucar.edu/Document/Graphics/Interfaces/



Example *xy1c.ncl*

gsn_csm_xy

- X values added
- Line color changed (using “named” color)
- Line thickness increased
- “long_name” attributes set
- Resource introduced:
  - xyLineThicknessF - sets line thickness

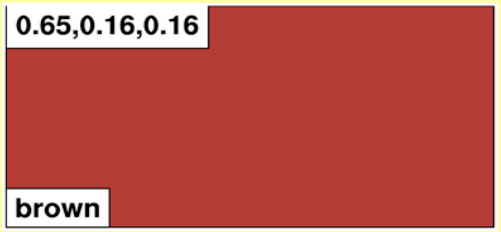
```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  x = ispan(-50,50,1)          ; Create some X and
  y = sin(0.0628*x)            ; Y data.
  x@long_name = "X values"     ; Add long_name attributes to
  y@long_name = "Sine values"  ; see what happens to plot.

  wks = gsn_open_wks("ps","xy1c")     ; "xy1c.ps"

; Set some XY plot resources.

  res                  = True
  res@xyLineColor      = "brown" ; or closest match
  res@xyLineThicknessF = 3       ; 3 times thicker
                                 ; default is 1

  plot = gsn_csm_xy(wks,x,y,res)
end
```

Order of resources not important

Default color table (again)

You can use "named" colors with color resources, but that color must be in your color table.

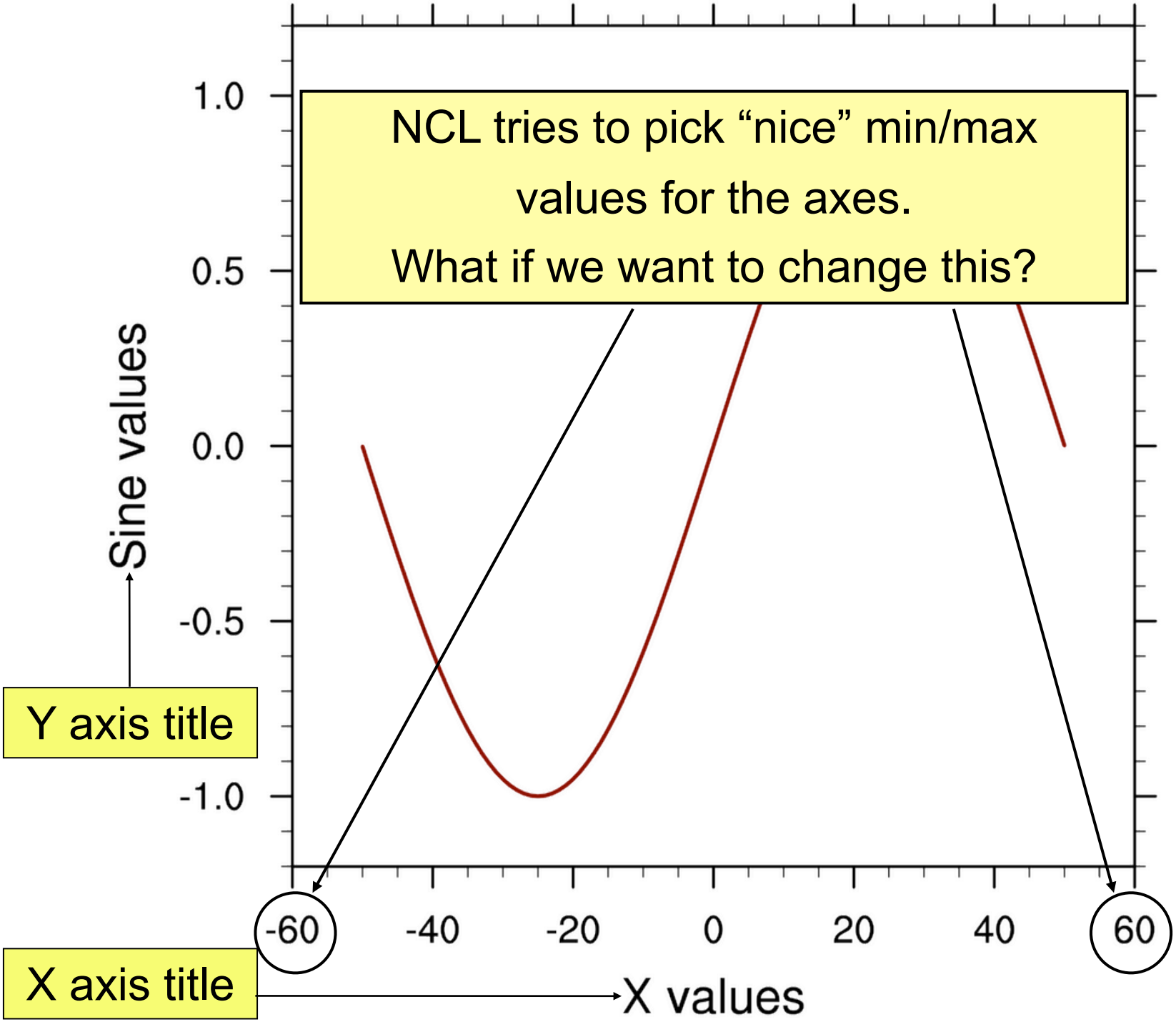
res@xyLineColor = "brown"



Closest match is used, which is the color associated with index 8 in this case.

650 named colors:

http://www.ncl.ucar.edu/Document/Graphics/named_colors.shtml

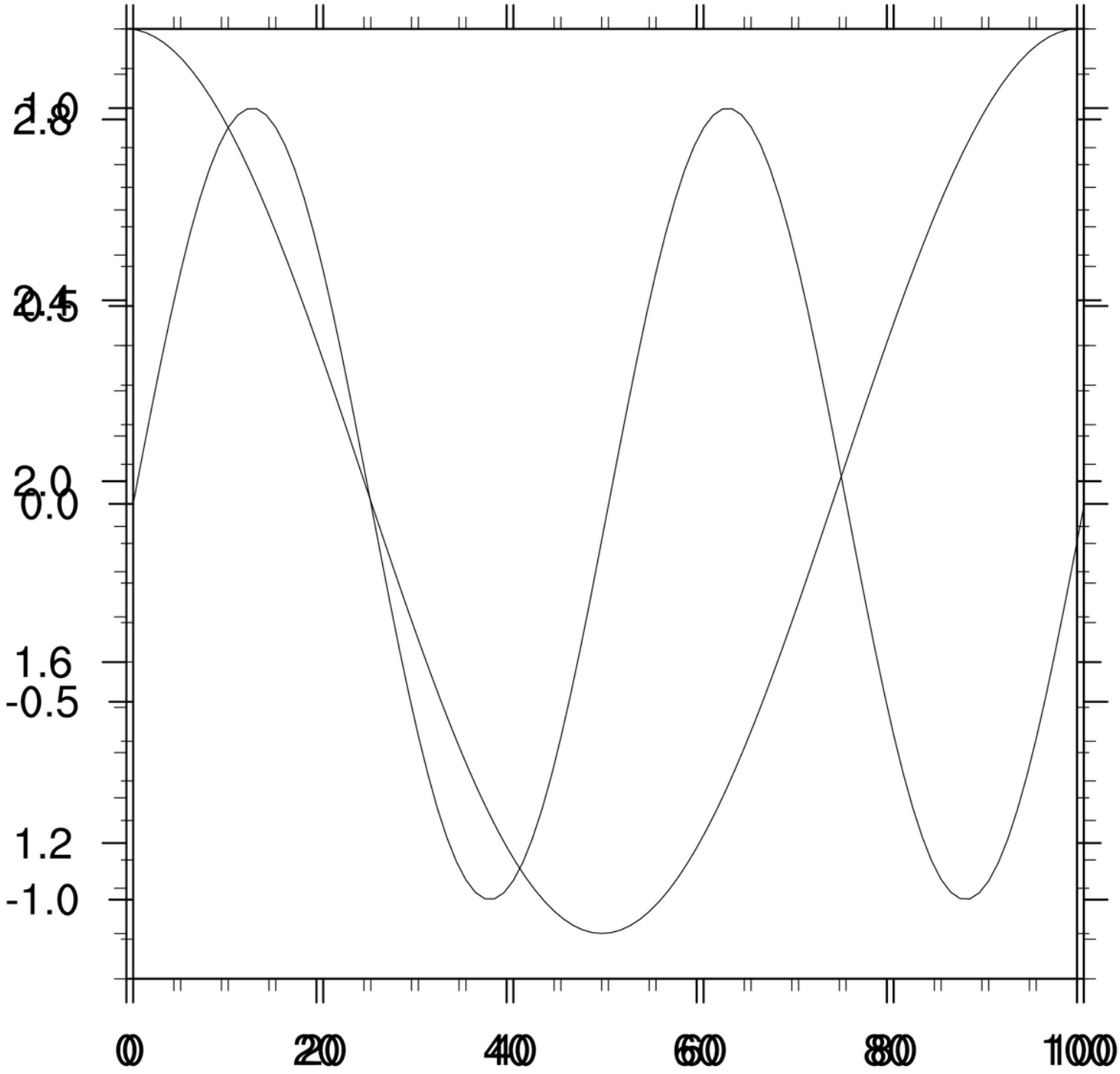

NCL tries to pick “nice” min/max values for the axes.
What if we want to change this?
1.0
0.5
0.0
-0.5
-1.0
Sine values
Y axis title
-60
-40
-20
0
20
40
60
X values
X axis title

Special topic: "frame" procedure

- By default, main plotting functions draw the plot and advance the frame (page).
- If you want to continue drawing on same frame (page), then you need to turn off frame advance.
- This can be accomplished with special resource "gsnFrame" and special procedure "frame".

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  y1 = sin(0.1256*ispan(0,100,1))
  y2 = cos(0.0628*ispan(0,100,1)) + 2.


  wks = gsn_open_wks("ps","xy")

  res  = True
  res@gsnFrame = False      ; Don't advance the frame
  plot = gsn_csm_y(wks,y1,res)
  plot = gsn_csm_y(wks,y2,res)
end
```

Set to False don't advance frame


2.8
2.4
2.0
1.6
1.2
0
20
40
60
80
100

Example *contour1d.ncl*
gsn_csm_contour
gsn_define_colormap

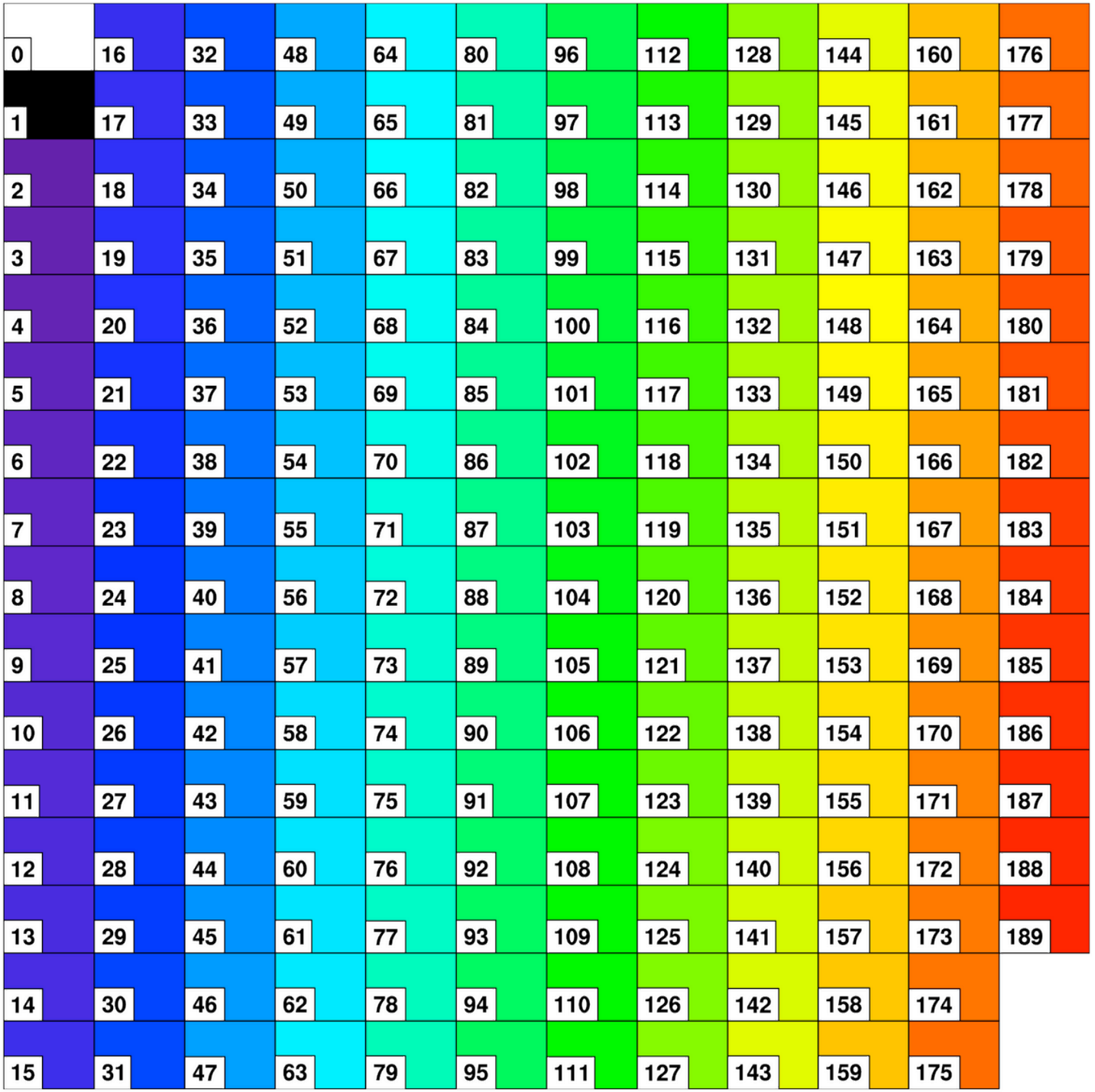
- Color map changed
- Full color map spanned
- Main title added
- Resources introduced:
  - gsnSpreadColors - if True, span full color map when contour fill (or vector fill) turned on

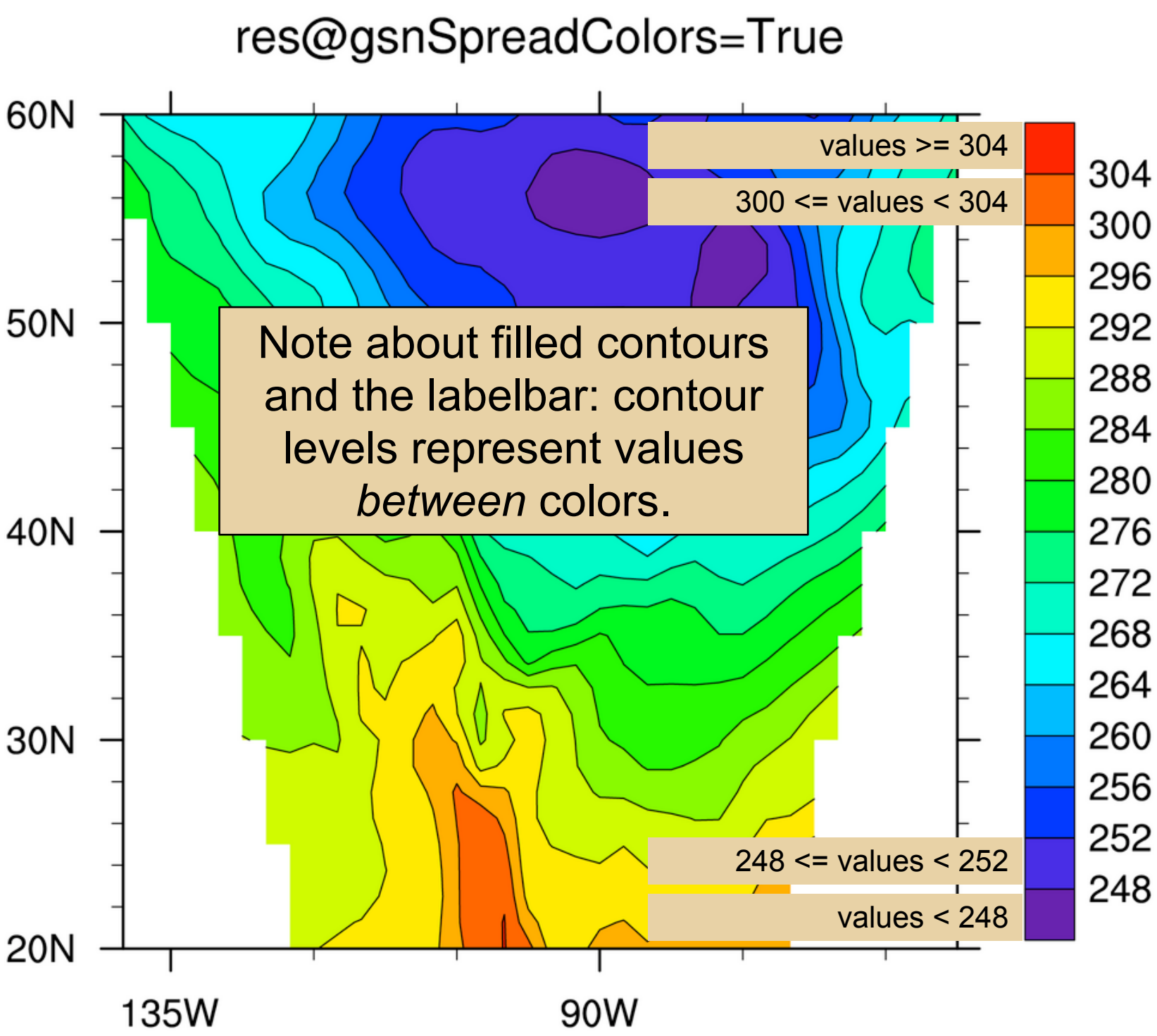
```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  tf = addfile("Tstorm.cdf","r")
  T  = tf->t(0,:,:)
  T&lon@units = “degrees_east” ; Add some units
  T&lat@units = “degrees_north”
```

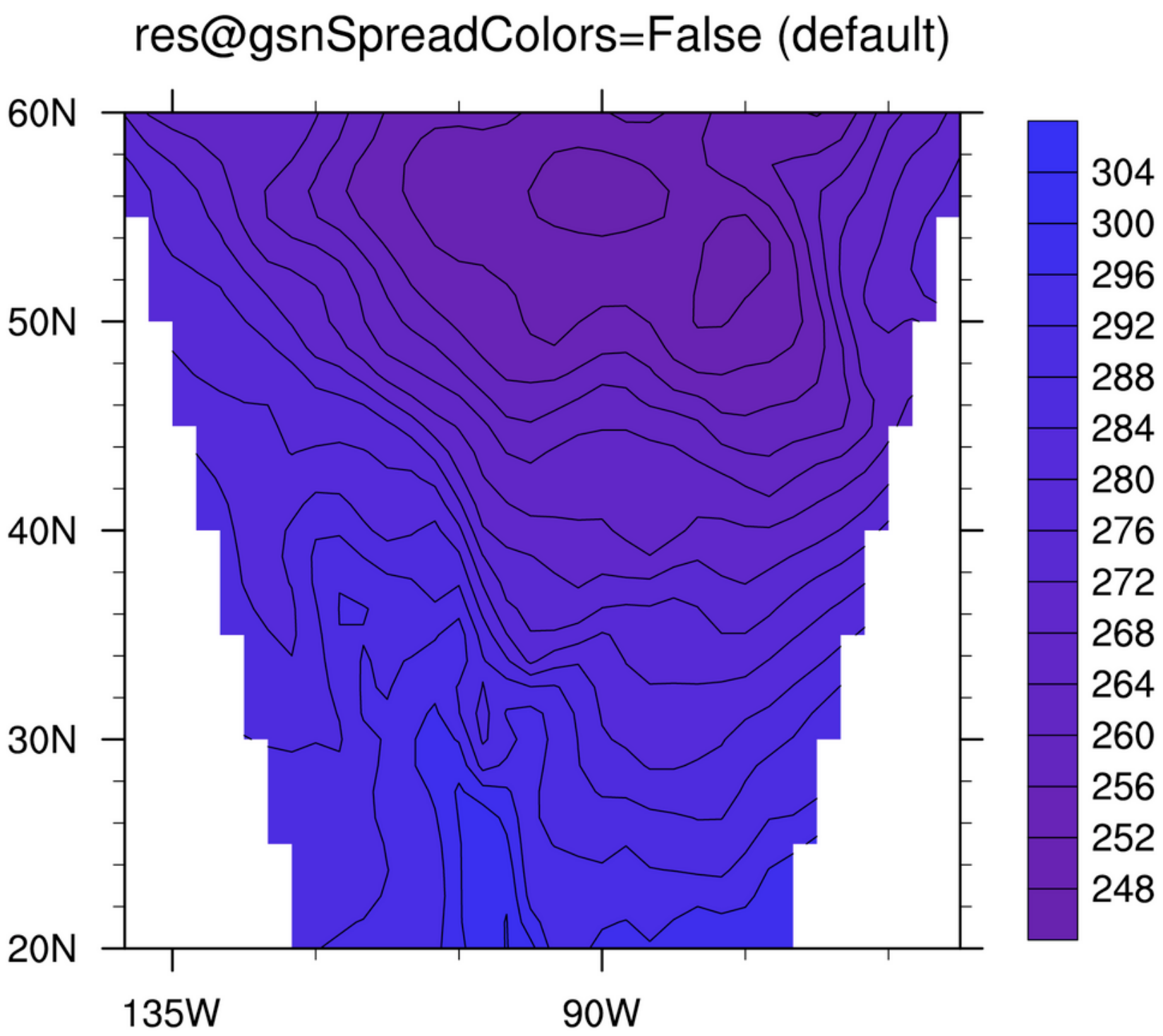
Better to set color map in “.hluresfile”

```
  wks = gsn_open_wks("ps","contour1d")
  gsn_define_colormap(wks,"rainbow")    ; Change color map

  res                  = True
  res@cnFillOn         = True       ; Turn on contour fill
  res@gsnSpreadColors = True        ; Span full color map

  res@lbOrientation    = “Vertical” ; Move labelbar
  res@tiMainString     = ”res@gsnSpreadColors=True” ; Main title

  plot = gsn_csm_contour(wks,T,res)
end
```

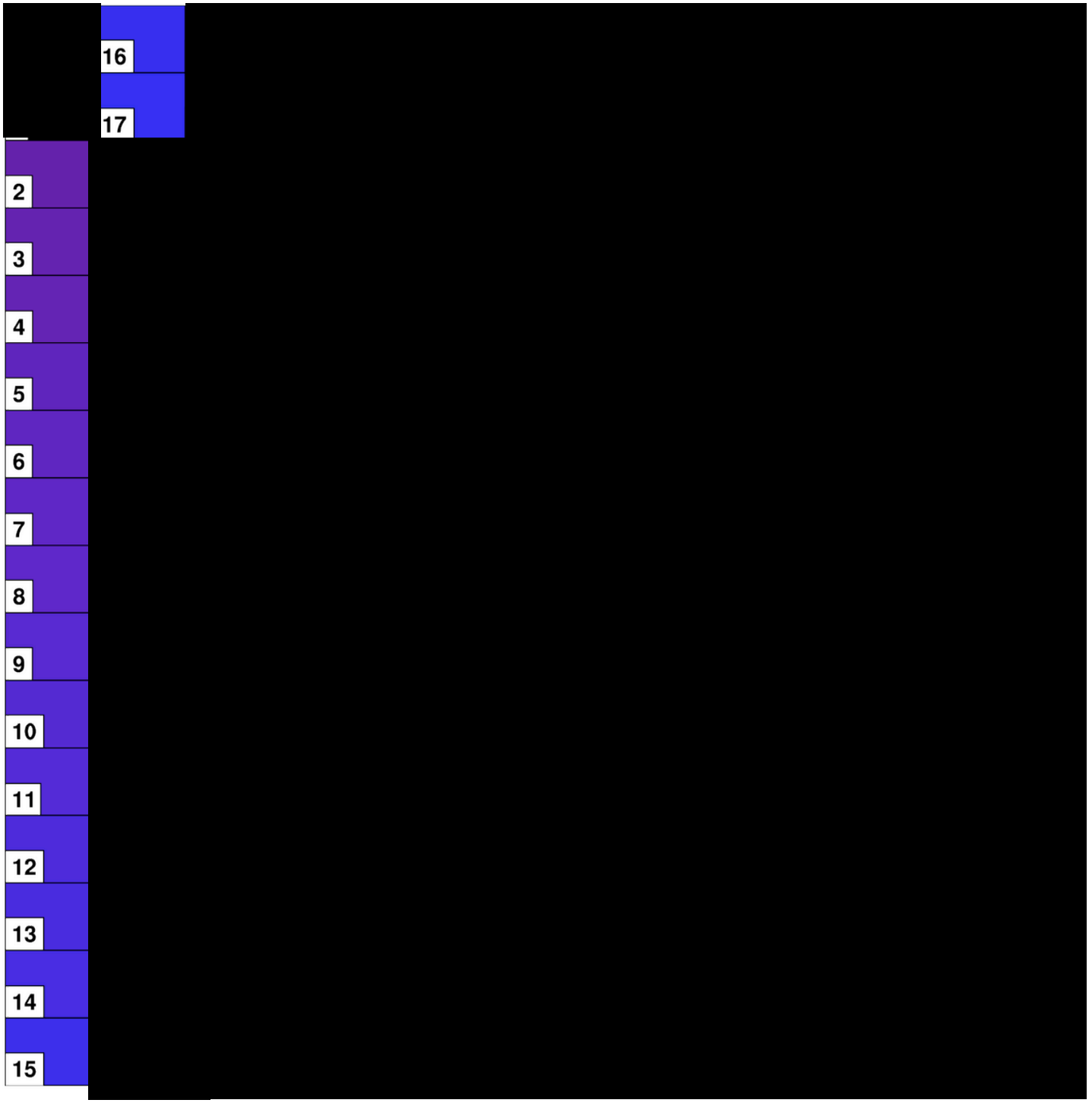
“rainbow” color map




res@gsnSpreadColors=True
60N
50N
40N
30N
20N
135W
90W
values >= 304
300 <= values < 304
248 <= values < 252
values < 248
304
300
296
292
288
284
280
276
272
268
264
260
256
252
248
Note about filled contours and the labelbar: contour levels represent values *between* colors.

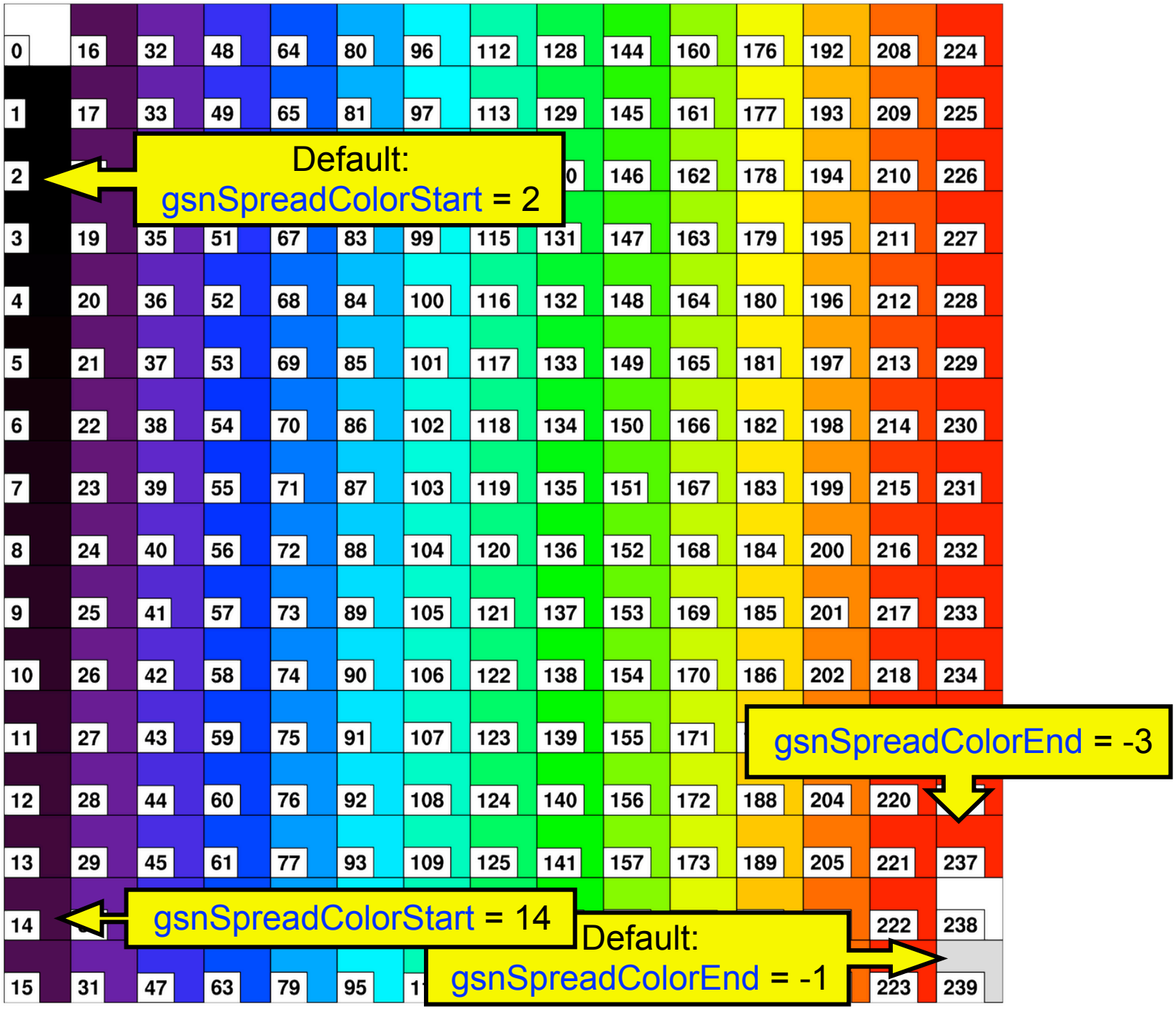

res@gsnSpreadColors=False (default)
60N
50N
40N
30N
20N
135W
90W
304
300
296
292
288
284
280
276
272
268
264
260
256
252
248

“rainbow” color map first set of colors are purple/ dark blue

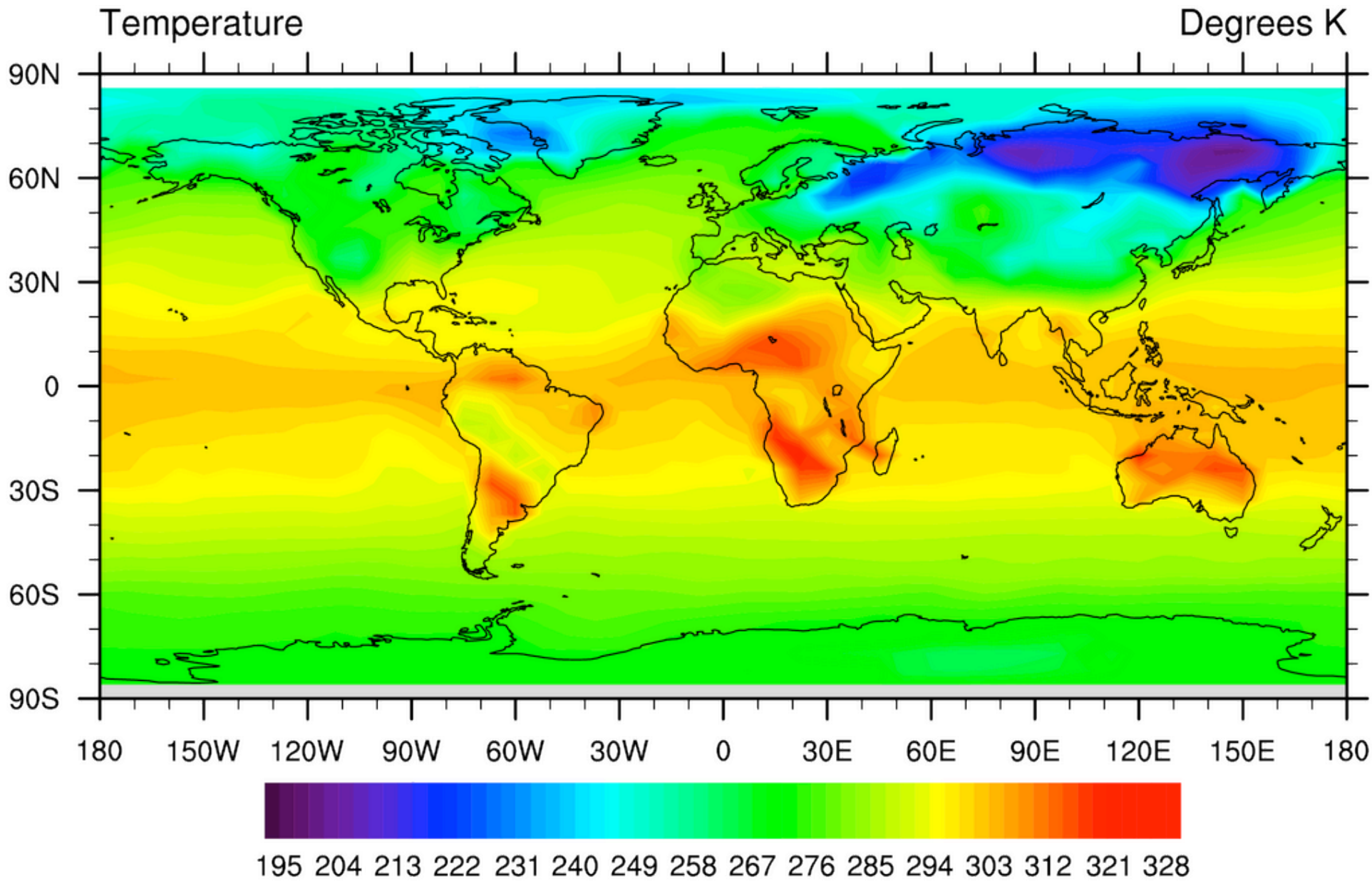


Example *contour2d.ncl*
gsn_csm_contour_map

- Contour and labelbar box lines turned off
- Only part of color map spanned
- Resources introduced:
  - gsnSpreadColorStart, gsnSpreadColorEnd - indicates portion of color map to span
  - cnLinesOn - turns contour lines on/off
  - lbBoxLinesOn - turns labelbar box lines on/off
- Will set these last two to False


Default:
gsnSpreadColorStart = 2
gsnSpreadColorEnd = -3
gsnSpreadColorStart = 14
Default:
gsnSpreadColorEnd = -1

```
. . .
  tf = addfile("meccatemp.cdf","r")
  T  = tf->t(0,:,:)

  wks = gsn_open_wks("ps","contour2c")
  gsn_define_colormap(wks,"rainbow+white+gray")

  res                          = True
  res@gsnAddCyclic             = False
  res@cnLevelSelectionMode     = "ManualLevels"
  res@cnMinLevelValF           = 195.0      ; Min contour
  res@cnMaxLevelValF           = 328.0      ; Max contour
  res@cnLevelSpacingF          = 2.25       ; Spacing

  res@gsnSpreadColors          = True   ; Span full color map
  res@gsnSpreadColorStart      = 14     ; Start at color index 14
  res@gsnSpreadColorEnd        = -3     ; Stop at 3rd color from end

  res@cnFillOn                 = True       ; Turn on contour fill
  res@cnLinesOn                = False      ; Turn off contour lines
  res@lbLabelAutoStride        = True       ; Control labelbar labels
  res@lbBoxLinesOn             = False      ; Turn off lbar box lines
  res@gsnSpreadColors          = True       ; Span full color map

  plot = gsn_csm_contour_map(wks,T,res)
```


Temperature
Degrees K
90N
60N
30N
0
30S
60S
90S
180
150W
120W
90W
60W
30W
0
30E
60E
90E
120E
150E
180
195 204 213 222 231 240 249 258 267 276 285 294 303 312 321 328

Example: 2D lat/lon arrays

- Assume file is from sea ice model: “iceh_mavg.0014-02.nc”
- Has a variable “hi” w/no coordinate arrays

```
Dimensions and sizes:   [lat | 384] x [lon | 320]
Coordinates:
Number Of Attributes: 7
  time :          4804
  units :         m
  long_name :     grid box mean ice thickness
  coordinates :   i j time
  _FillValue :    1e+30
  time_rep :      averaged
```

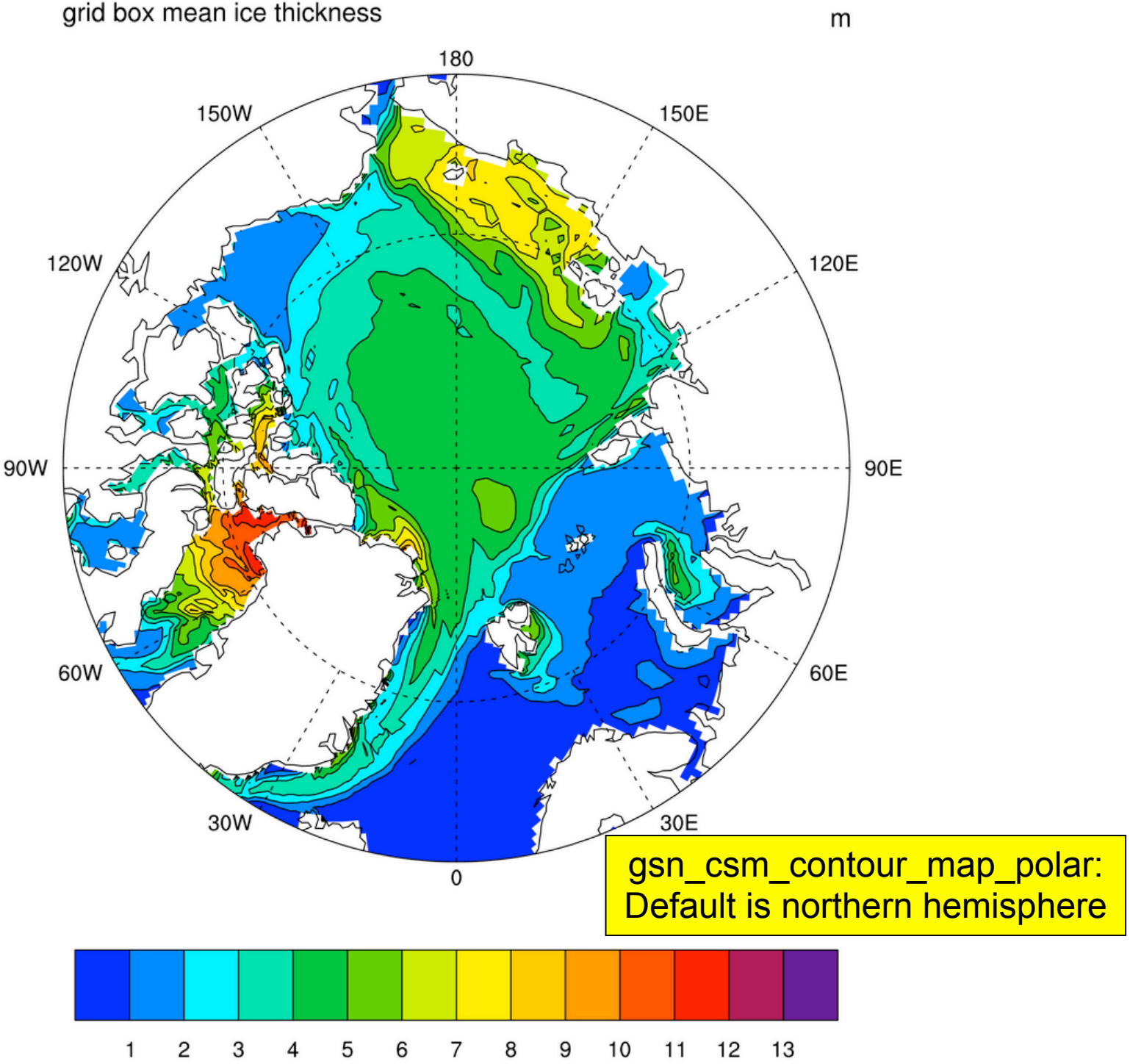
- File does have two-dimensional lat/lon arrays

```
float TLON ( lat, lon )
          long_name :     grid center longitude
          units :         degrees_east
float TLAT ( lat, lon )
          long_name :     grid center latitude
          units :         degrees_north
```

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f = addfile("iceh_mavg.0014-02.nc","r")
  hi          = f->hi(0,:,:)                    ;
  printVarSummary(hi)                           ;

  wks = gsn_open_wks("ps","ice")
  gsn_define_colormap(wks,"BlAqGrYeOrReVi

  res                       = True              ; Plot mods desired

  res@sfXArray              = f->TLON  ; 2D lat/lon arrays, must
  res@sfYArray              = f->TLAT  ; be same dimensions as "hi"

  res@cnFillOn              = True              ; Turn on color fill
  res@mpMinLatF             = 65                ; Specify min lat
  res@gsnSpreadColors       = True              ; Use full colormap

  plot = gsn_csm_contour_map_polar(wks,hi,res)
end
```

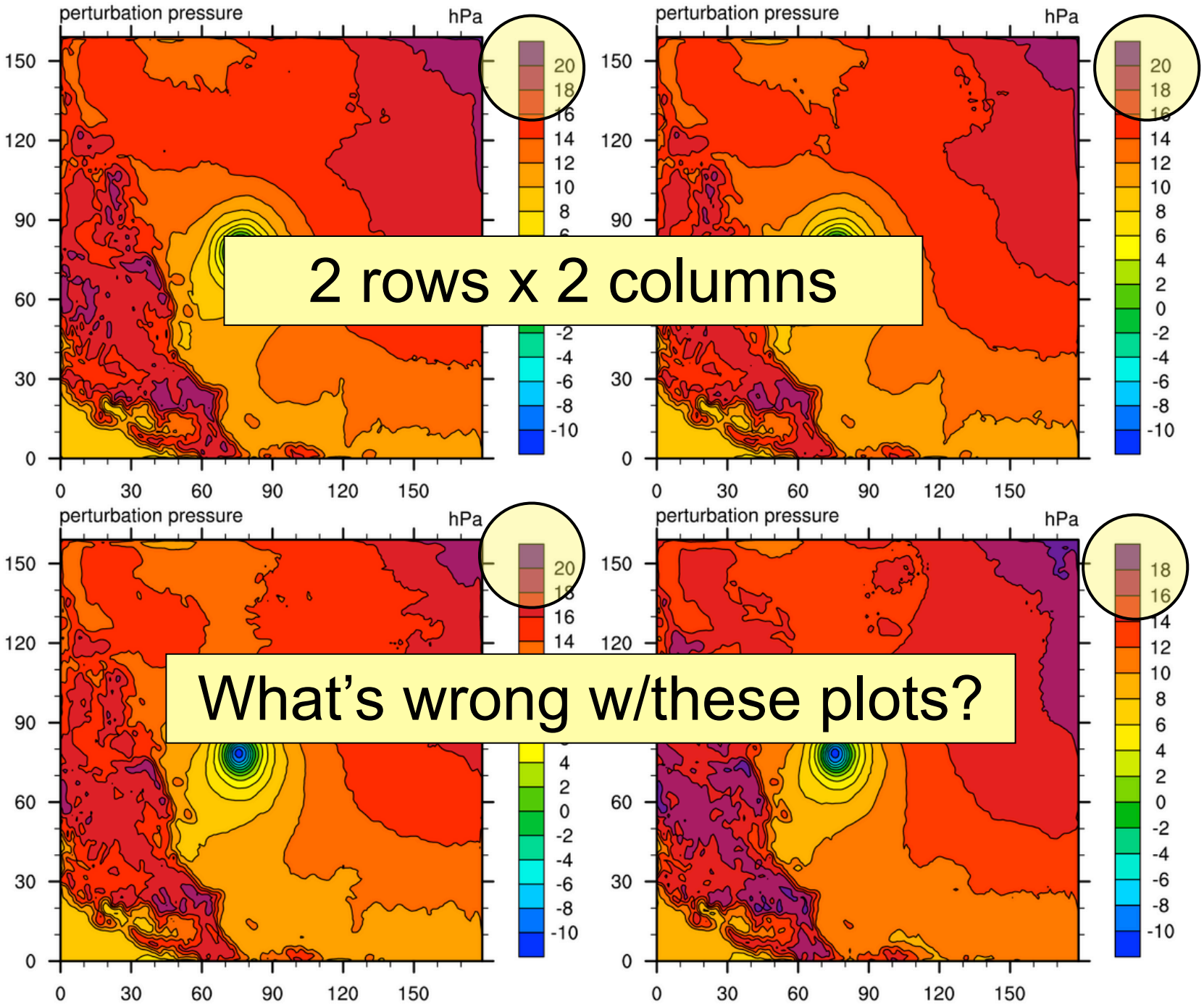
Setting sfX/YArray is equivalent to:

```
hi@lat2d = f->TLAT
hi@lon2d = f->TLON
```



gsn_csm_contour_map_polar:
Default is northern hemisphere

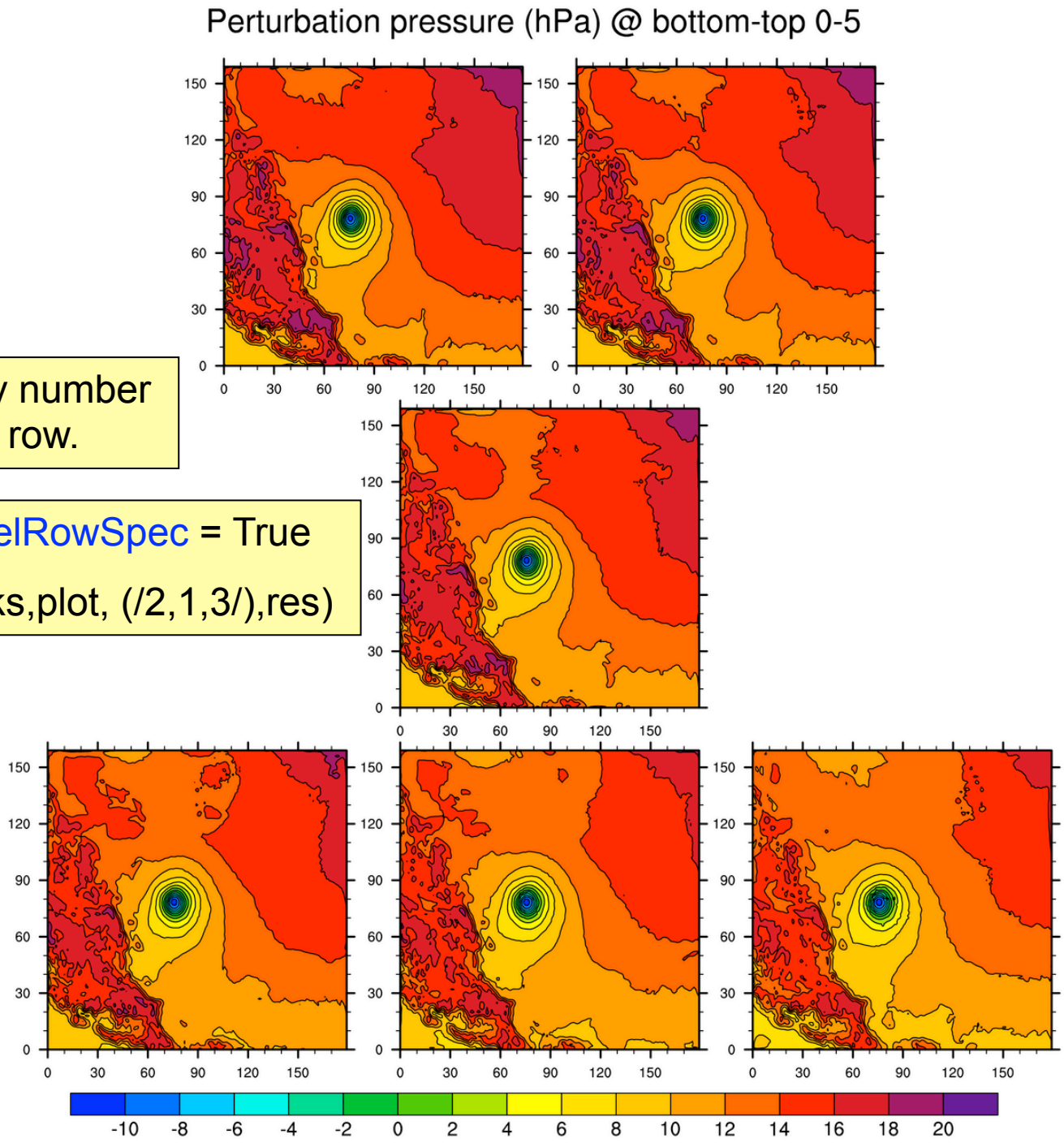
```
. . .
begin
  f = addfile ("wrfout_d01_2003-07-15_00:00:00.nc", "r")
  p        = f->P(0,:,:,:)                    ; Read pressure
  p        = p*0.01                           ; Convert to hPA
  p@units = "hPa"                             ; Update units attribute

  wks = gsn_open_wks("ps" ,"panel1b")
  gsn_define_colormap(wks,"BlAqGrYeOrReVi200")

  res                    = True               ; Plot options desired
  res@gsnDraw            = False              ; Don't draw plots
  res@gsnFrame           = False              ; Don't advance frames
  res@cnFillOn           = True               ; Turn on color
  res@gsnSpreadColors    = True               ; Use entire color map
  res@lbOrientation      = "Vertical"         ; Vertical labelbar

  plots    = new(4,graphic)
  plots(0) = gsn_csm_contour(wks,p(0,:,:),res)
  plots(1) = gsn_csm_contour(wks,p(3,:,:),res)
  plots(2) = gsn_csm_contour(wks,p(5,:,:),res)
  plots(3) = gsn_csm_contour(wks,p(7,:,:),res)

; 2 rows, 2 columns
  gsn_panel(wks,plots,(/2,2/),False)
end
```


perturbation pressure
hPa
2 rows x 2 columns
What's wrong w/these plots?

Can specify number of plots per row.

res@gsnPanelRowSpec = True

gsn_panel(wks,plot, (/2,1,3/),res)



In review…

- Five steps to create a plot
- Use X11 window while debugging script; move to PS/PDF later
- Hardest part are the resources: start simple
- Organize resources for easier debugging
- Start with an existing script if possible

Customize your graphics environment

Optional, but most highly recommended.

(Come to think of it, not really that optional!)

- Download ".hluresfile" file, put in home directory
  - Changes your default background, foreground colors from black/white to white/black
  - Changes font from **times-roman** to **helvetica**
  - Changes "function code" (default is a colon)
  - Can be used to change default color map
- Available on your lab machines:

```
cat ~/.hluresfile
```

http://www.ncl.ucar.edu/Document/Graphics/hlures.shtml

netCDF Operators [NCO]

http://nco.sourceforge.net/

Introduction and History

- Suite of Command Line Operators
- Designed to operate on netCDF/HDF files
- Each is a stand alone executable
- Very efficient for specific tasks
- Available for various computer architectures:
  - Solaris, Irix, AIX, Linux, Windows

Appending vs. Concatenation

- Appending is the
  merging of files:

  file1 = T,U,V

  file2 = PSI,CHI

  file3 = T,U,V,PSI,CHI

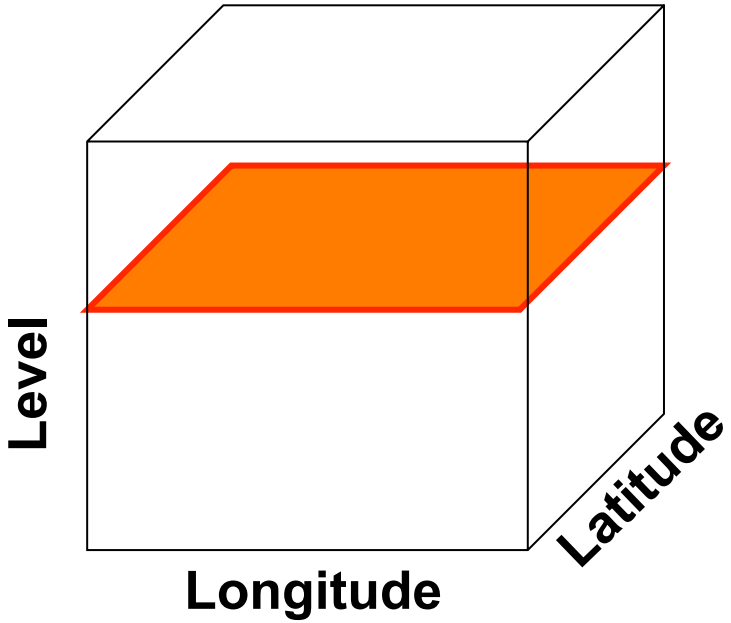
tenation is the
nation of variables
a record
sion:

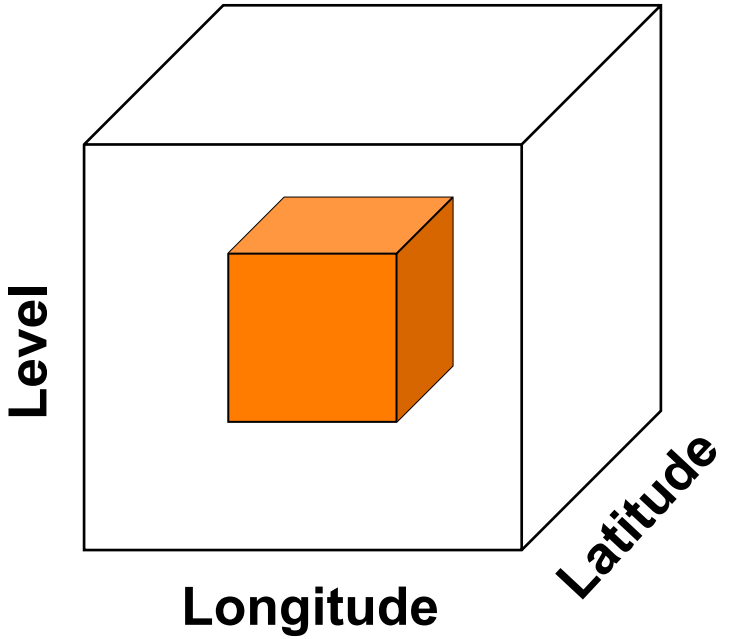
1 = T(0:12,:,:)

- file 2 = T(13:24,:,:)
- concatenated file = T (0-24,:,:)

Hyperslabs

A hyperslab is a subset of data.





Missing Values

- NCO identifies missing data by the **_FillValue** attribute. [v 3.9.2 8/2007]
- No arithmetic operations on these values.
- No longer recognizes **missing_value**
- Best to create netCDF with both **_FillValue** and **missing_value**

ncra: record averaging

- Averages record variables across an arbitrary number of input files
- The record dimension is retained as a degenerate (size 1) dimension.
- Weights each record in the input files equally
- **ncra 12.nc 01.nc 02.nc DJF.nc**

ncecat: ensemble concatenator

- Concatenates an arbitrary number of input files into a single output file. Wild characters allowed.
- Each input file is stored consecutively as a single record in the output file.
- Input files are glued together by the creation of a record dimension.
- **ncecat case-1.nc case-2.nc total.nc**
- **ncecat case*nc TOTAL.nc**

ncrcat: record concatenator

- Concatenates record variables across an arbitrary number of input files. Unix wild characters allowed
- Final record dimension is the sum of the lengths of the input files.
- Input files may vary in length, but **EACH** must have an **UNLIMITED** record dimension.
  - file1.nc ({time:1:12},:,:)
  - file2.nc ({time:13:24},:,:)
  - **ncrcat -h -O file1.nc file2.nc concat.nc**
  - concat.nc ({time:1:24},:,:)
  - **ncrcat -h -O file*.nc CONCAT.nc**

ncdiff: differencer

- File1 - File2 = File3
- Common dimensions must be the same size.
- For anomalies, the time dimension of the mean file must be removed.
- File2 should be a subset of File1 if they are not identical
  - ncwa -0 -a time in.nc out.nc
- **ncdiff 001.nc 002.nc diff.nc**

ncwa: weighted average

- Averages variables in a single file over arbitrary dimensions
  - options for weights, masks and normalizations

ncatted: attribute editor

- **ncatted -a att-dsc in.nc** (works on only one file at a time)

  att-dsc = att-nm, var-nm, mode, att-type, attval(order dependent)

  att-nm: The name of the attribute to edit

  var-nm: The name of the variable to edit

  mode: d=delete, a=append, c=create, m=mod, o=overwrite

  att-type: f=float, d=double, l = long, s=short, c=char

  att-val: The new value

- **ncatted -a history,global,a,c,"Add text here" in.nc**

ncks: kitchen sink

- Extracts a subset of data from an input file
- Global attributes for that output file are overwritten.
- Variable will be overwritten if it already exists in output file
- If record dimension is different, then **ncks** will create a new record dimension.
- **ncks -O -v TS,V in.nc out.nc**

ncrename

- Renames variables (-v), dimensions (-d), attributes (-a)
- **ncrename -v p,pres –v t,T in.nc out.nc**
- **ncrename –a missing_value,_FillValue –a Zaire,Congo in.nc out.nc**
- **ncrename –d longitude,lon –v longitude,lon –v rh,rhum in.nc out.nc**

ncap, ncap2

- Arithmetic processors
- **ncap2 –s 'x@valid_range=(min(x),max(x))' in.nc out.nc**
- **ncap2 -s 'lon=lon+180.0' in.nc out.nc**

Options: “-A” and “-O”

- Append variables to output file if it exists
- **ncks -A -v T,U,V in.nc out.nc**

- Will overwrite output file if it exists
- **ncks -O -v T,U,V in.nc out.nc**

Options: “-v” and “-x –v”

- Operates on only those variables listed
- **ncks -v T,U,V in.nc out.nc**

- Operates on all variables EXCEPT those listed.
- **ncks -x -v CHI,PSI in.nc out.nc**

Options: “-d” and “-h”

- Operates on a hyperslab of data
- **ncks -d lon,340.,50. -d lat,10.,35. in.nc out.nc**
- Real numbers indicate actual coordinate values
- Integer numbers indicate array indexes

- Override automatic appending of the global history attribute with the NCO command issued (which can be very long)

Options: “-R” and “r”

- Delete files retrieved from remote locations after they have been processed

- Prints current version of the operator

Options: “-c” and “-C”

- Ensures that coordinate variables are copied to any new files.
- This is the default.
- **ncks -c -v T,U,V in.nc out.nc**

- No coordinate variables are copied.
- Use this with caution, coordinate variables are very useful.
- **ncks -C -v T,U,V in.nc out.nc**