

# Visualizing Geospatial Data with GMT



# Generic Mapping Tools (GMT)

- The **objective** of this session is to introduce you to geospatial visualization using GMT
- Introduction ~30 min
  - What are GMT?
  - Examples of GMT maps/uses
- Part 1 - Getting Started ~60 min
  - Basic Syntax and commands
  - Map Projections/Coordinate Systems
  - Gridding and contouring datasets
  - Color/Illumination/Intensities
- Part 2 - Independent Mapping Project ~60 min

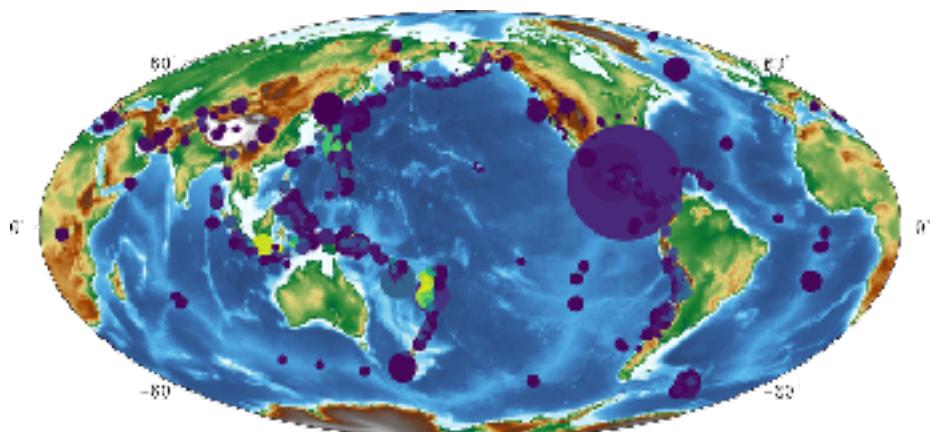


# What are GMT?

- GMT is an open source collection of about 80 command-line tools for manipulating geographic and Cartesian data sets (including filtering, trend fitting, gridding, projecting, etc.) and producing PostScript illustrations ranging from simple x-y plots via contour maps to artificially illuminated surfaces and 3D perspective views
- It makes publication quality maps and figures.

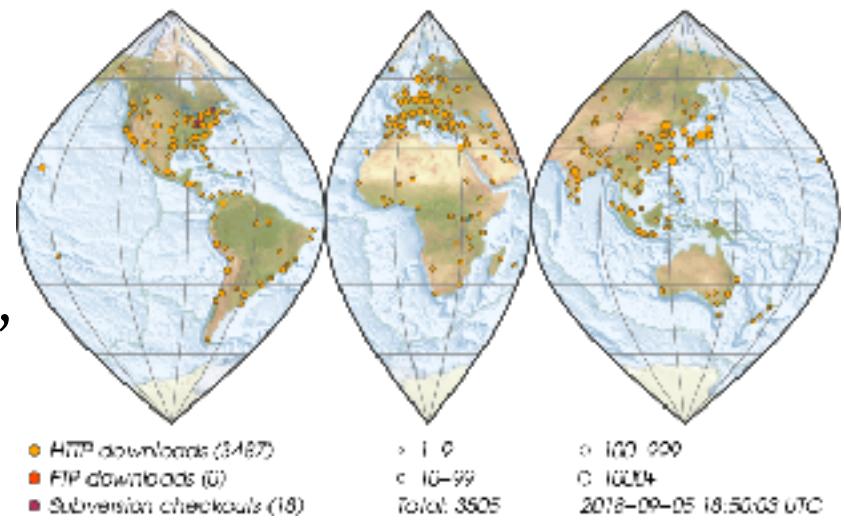
# Why GMT?

- PROS
  - Open Source (free!)
  - Powerful, flexible, and robust
  - High quality (.ps) output suitable for publication
- CONS
  - Steep learning curve
  - Not user friendly



# Who uses GMT?

- Most users are Earth, ocean or planetary scientists
- Also used in
  - medical research,
  - engineering,
  - physics,
  - mathematics,
  - social and biological sciences,
- And by
  - geographers,
  - fisheries institutes,
  - oil companies,
  - a wide range of government agencies,
  - innumerable hobbyists.



# Resources

- GMT web page - <http://gmt.soest.hawaii.edu>
  - Documentation and man pages
  - Discussion forum
  - Examples (included on box)
  - Tutorial
  - GMT Technical Reference and Cookbook
  - Podcast

The screenshot shows the GMT website homepage. At the top, there is a navigation bar with links: Overview, Activity, Roadmap, Issues, News, Wiki, Documentation, Download, Developer, Forums, and Repository. A search bar is located at the top right. Below the navigation bar, the main content area has a title "What is GMT?". The text describes GMT as an open source collection of about 80 command-line tools for manipulating geographic and Cartesian data sets. It mentions filtering, trend fitting, gridding, projecting, etc., and producing Postscript illustrations ranging from simple x-y plots via contour maps to artificially illuminated surfaces and 3D perspective views. The text also notes that the GMT supplements add another 40 more specialized and discipline-specific tools. It supports over 10 map projections and transformations and requires support data such as coastlines, rivers, and political boundaries and potentially DEM country polygons. The text credits Paul Wessel, Walter H. F. Smith, Imhoff Schmitt, Jaqcobsen and Flexner-Wilson, with help from a global set of volunteers, and is supported by the National Science Foundation. It is released under the GNU Lesser General Public License version 3 or any later version.

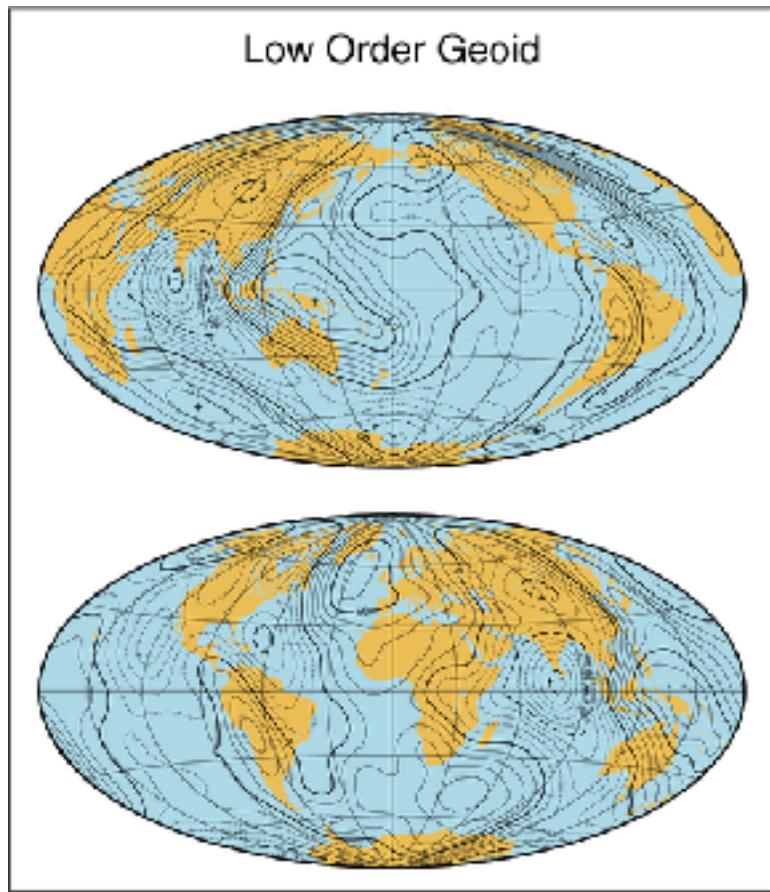
The "Wiki" tab is highlighted in blue. To the right of the main content, there is a sidebar with sections for "Latest Releases" (listing GMT-5.2.1 (2015-11-12) and GMT-4.5.14 (2015-11-01)), "Resources" (links to Installing Instructions, Building Instructions, Mailing Lists, Tickets, Podcasts, References, and Postdoc Opportunity), and "Wiki" (links to Start page, Index by title, and Index by date).

# Python API - [www.gmtpython.xyz](http://www.gmtpython.xyz)

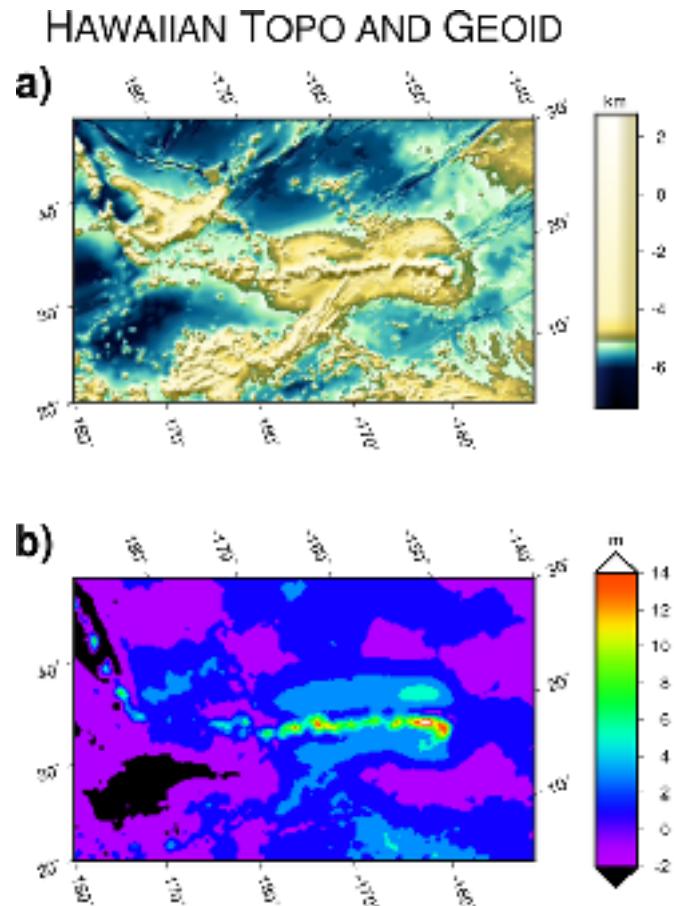
- Goal is modern API for Python programmers who want to use GMT
- Package is still in early stages of design and implementation
- This course will be taught using the Python API even though it is incomplete
  - Seamless integration
  - Only an introduction
  - The future of GMT
- Use distribution version for full functionality

# GMT Examples

## *Contour Maps*

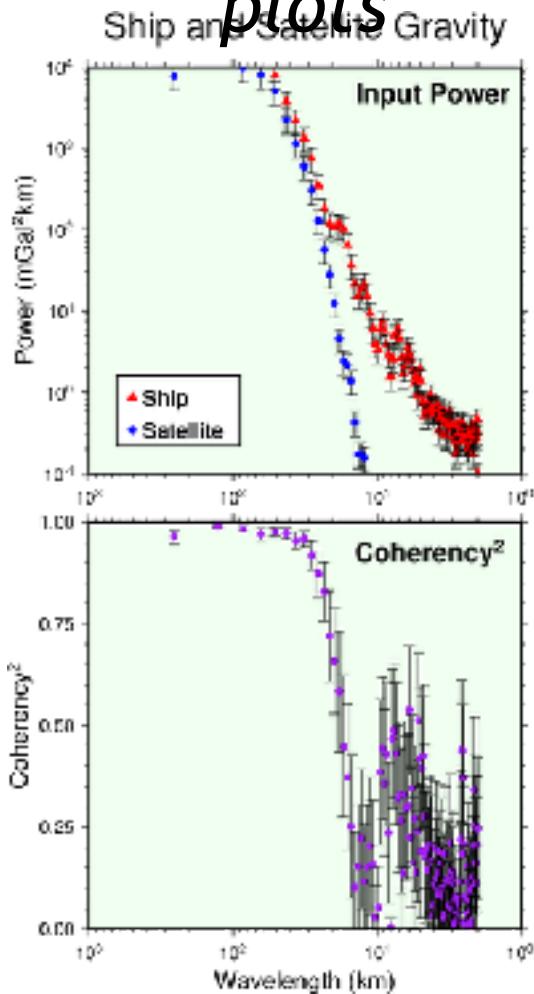


## *Topo and Geoid Maps*



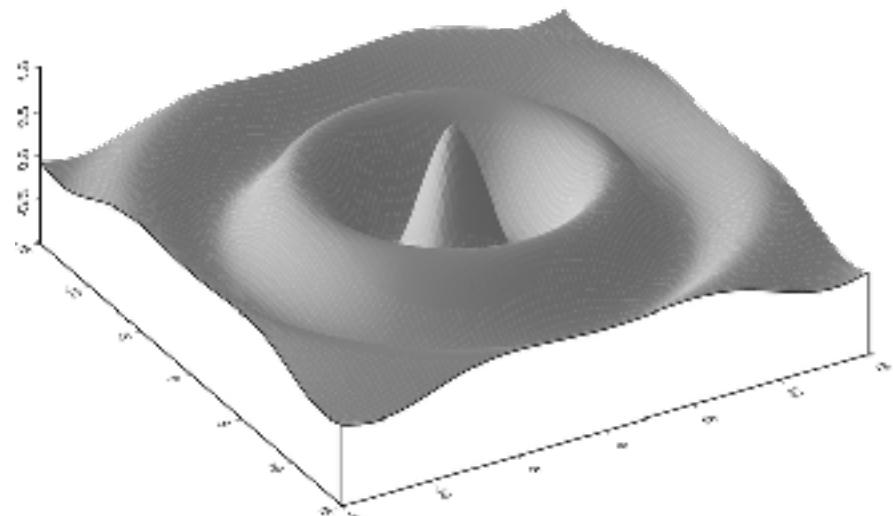
# GMT Examples

## *Spectral Estimation and xy plots*



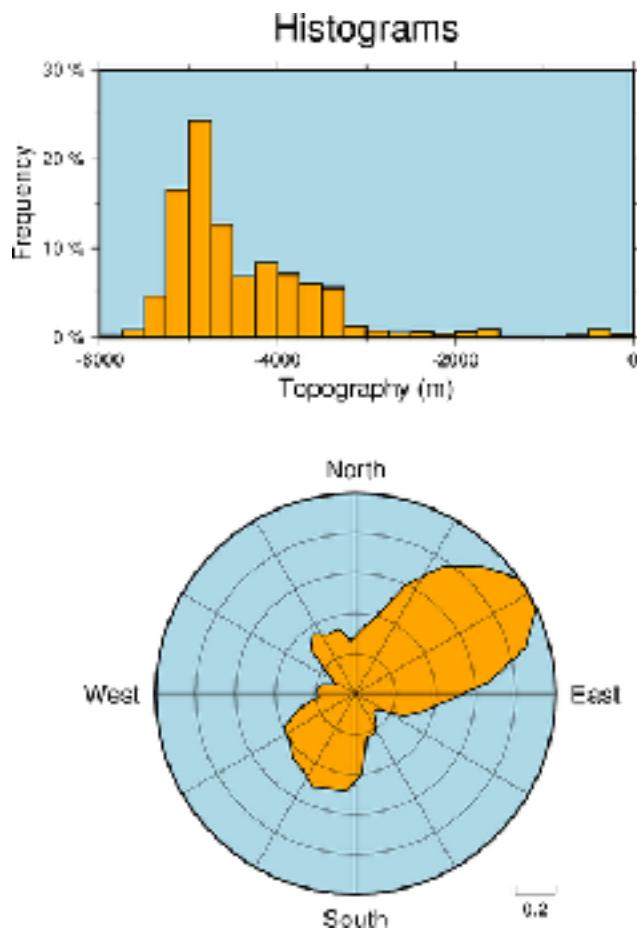
## *3D Perspective Mesh Plot*

$$z(r) = \cos(2\pi r/8) \cdot e^{-r/10}$$

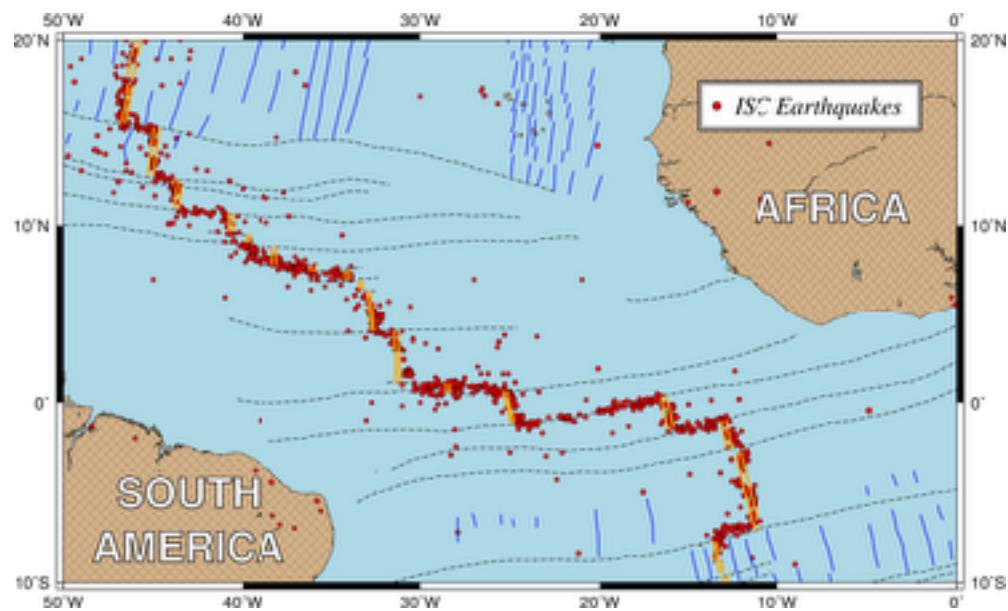


# GMT Examples

## *Plotting of Histograms*

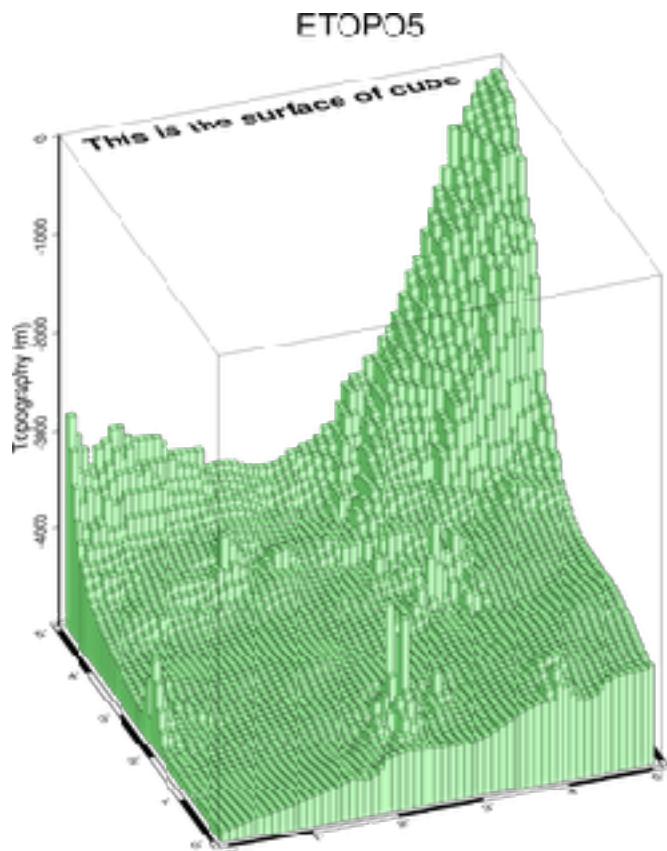


## *A Simple Location Map*

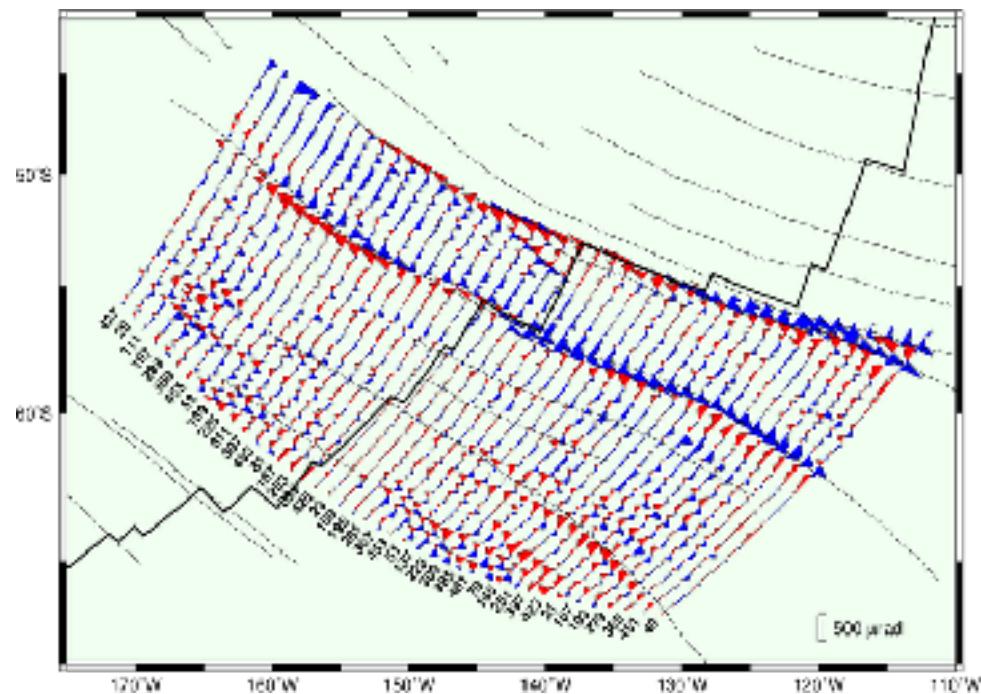


# GMT Examples

## 3D Histogram



## Time Series Along Tracks

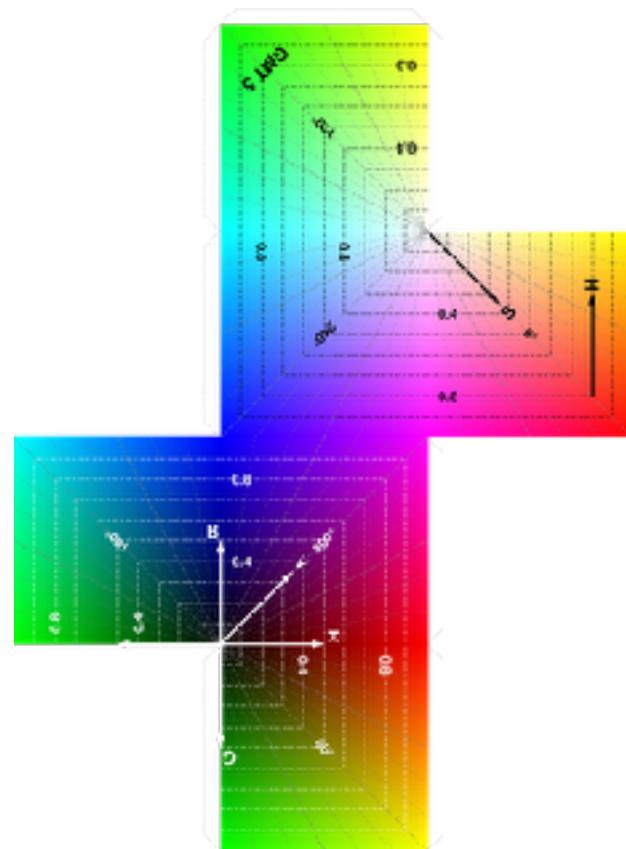
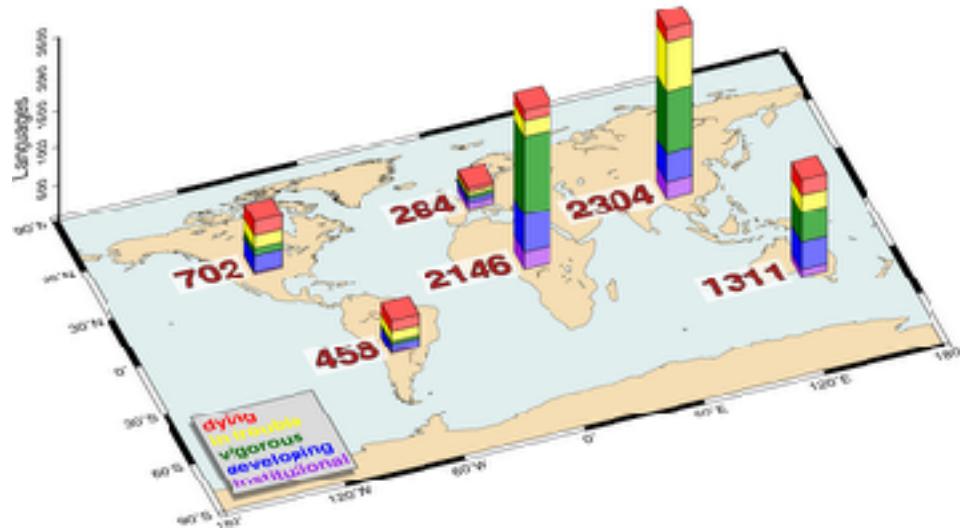


# GMT Examples

*Geographical Bar Graph Plot*

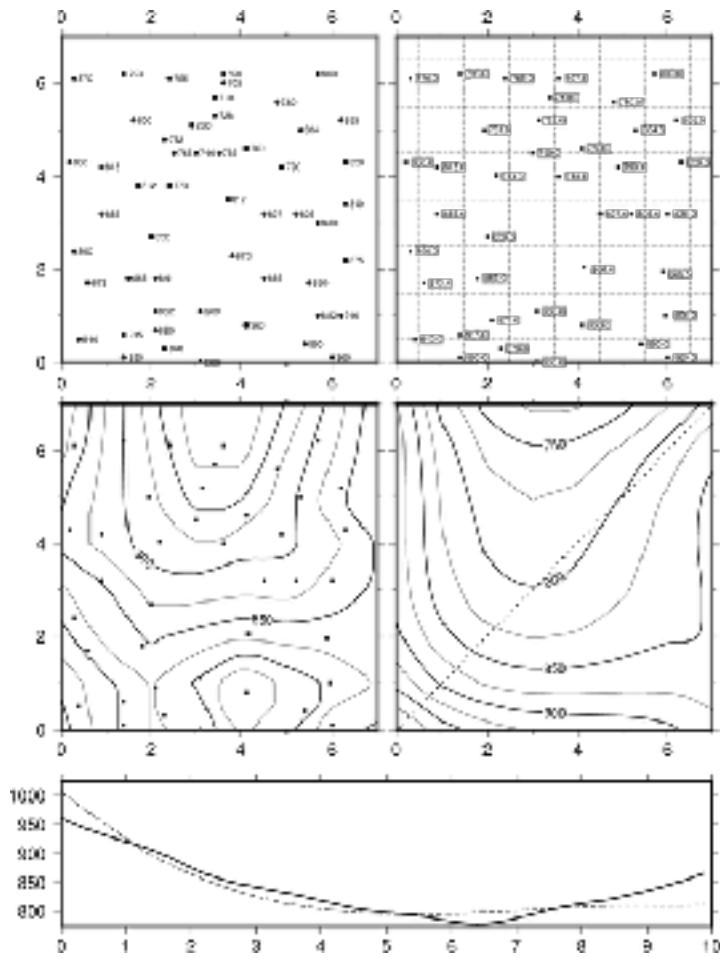
*A 3D RGB Color Cube*

World Languages By Continent

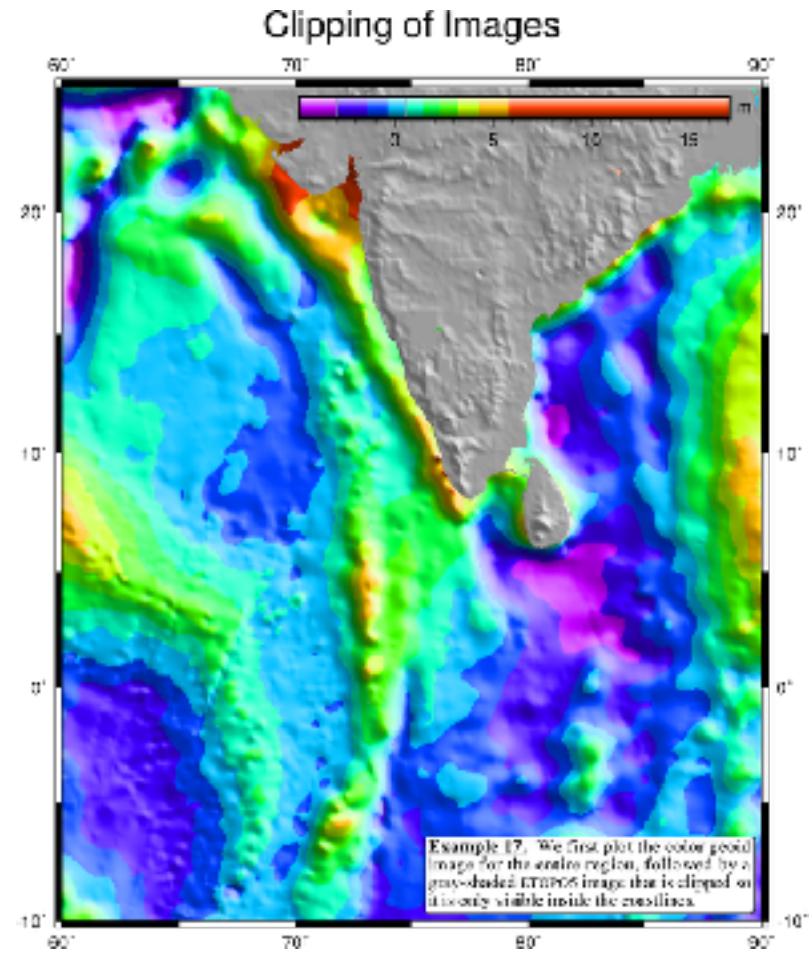


# GMT Examples

## *Gridding of Data and Trend Surfaces*

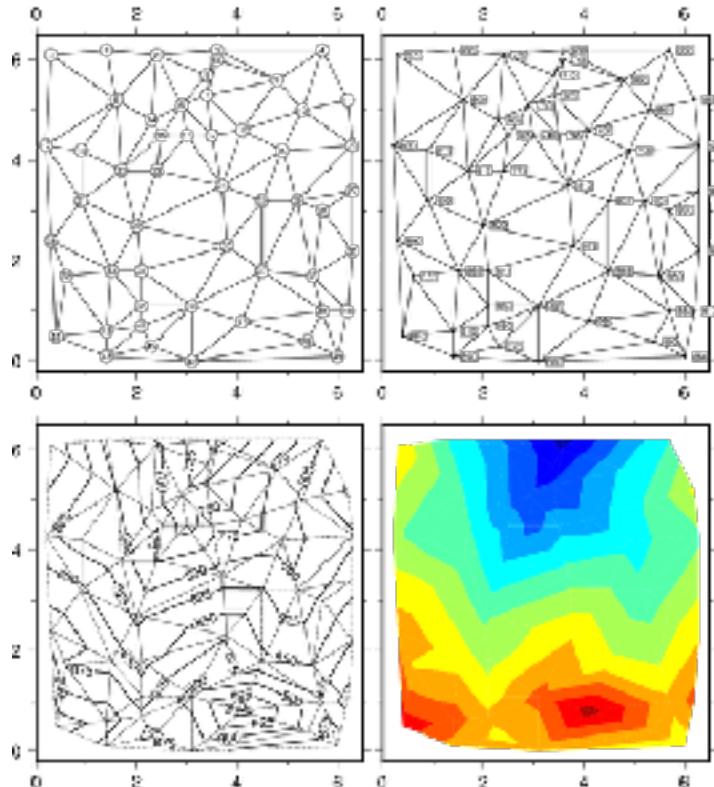


## *Images Clipped by Coastlines*



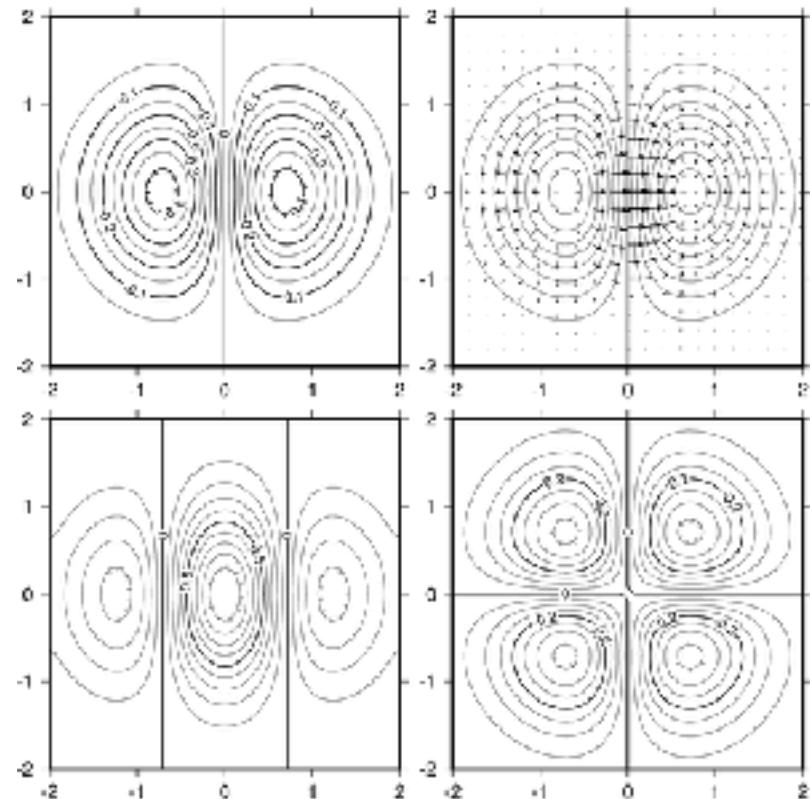
# GMT Examples

*Optimal  
Triangulation of  
Data  
Delaunay Triangulation*



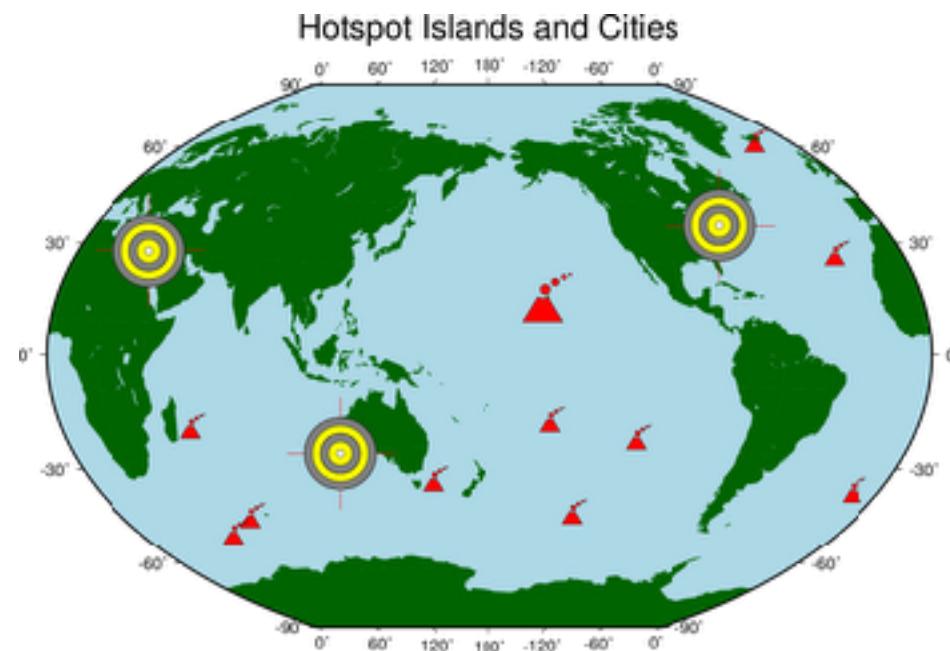
*Plotting of  
Vector Fields*

$$z(x,y) = x \cdot \exp(-x^2 - y^2)$$

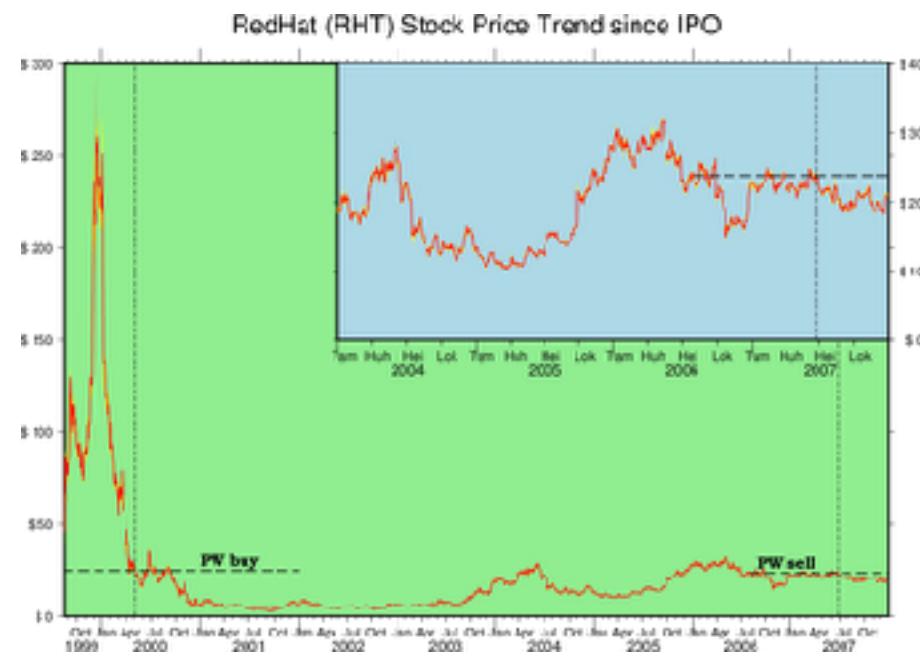


# GMT Examples

*Custom Plot and  
Symbols*

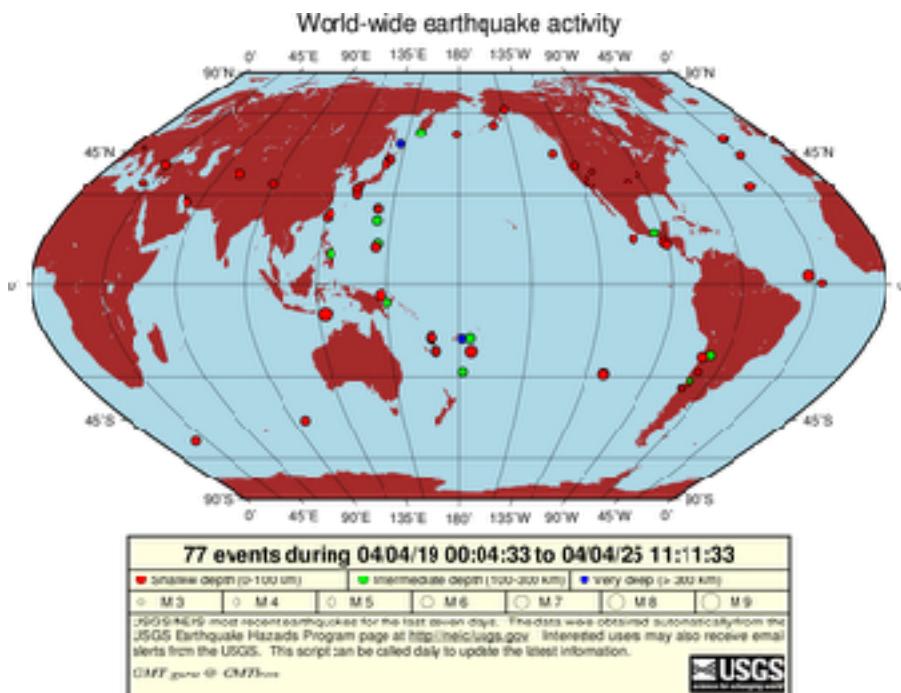


*Time Series of  
Red Hat Stock  
Price*

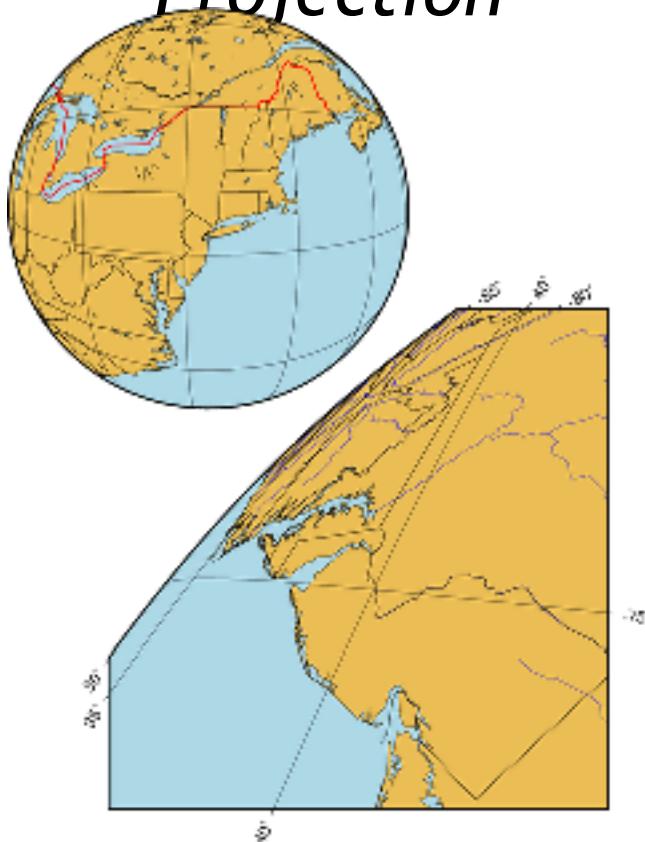


# GMT Examples

## Worldwide Seismicity Last 7 days

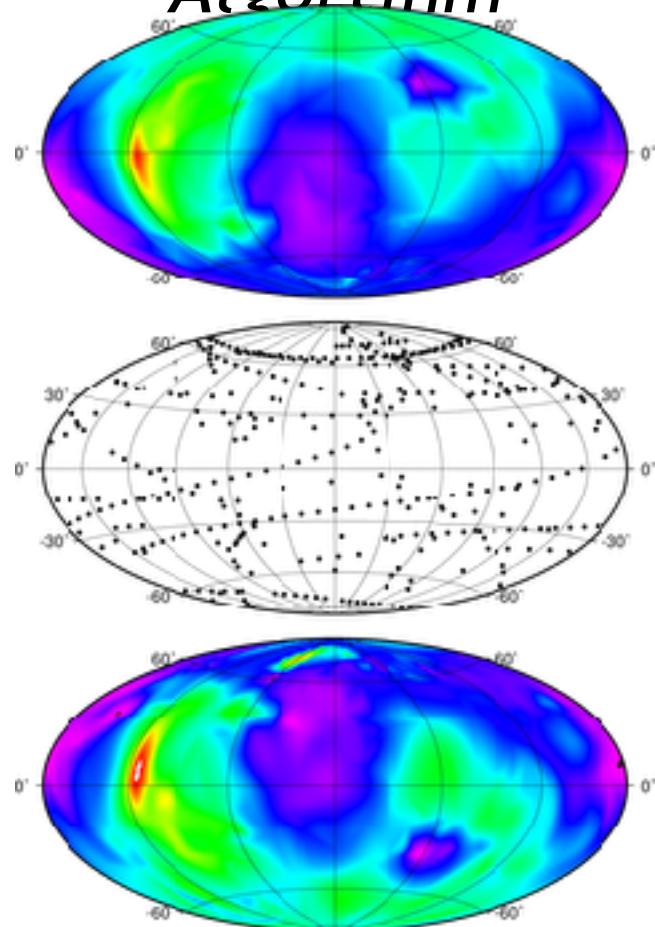


## General Vertical Perspective Projection



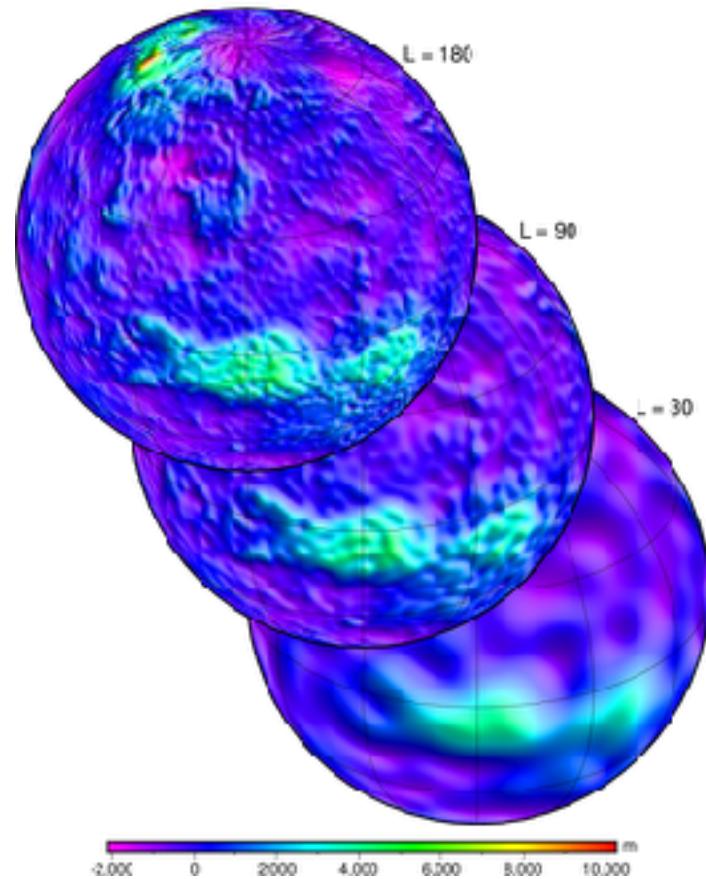
# GMT Examples

*Spherical Gridding  
using Renka's  
Algorithm*



*Evaluation of  
Spherical Harmonics  
Coefficients*

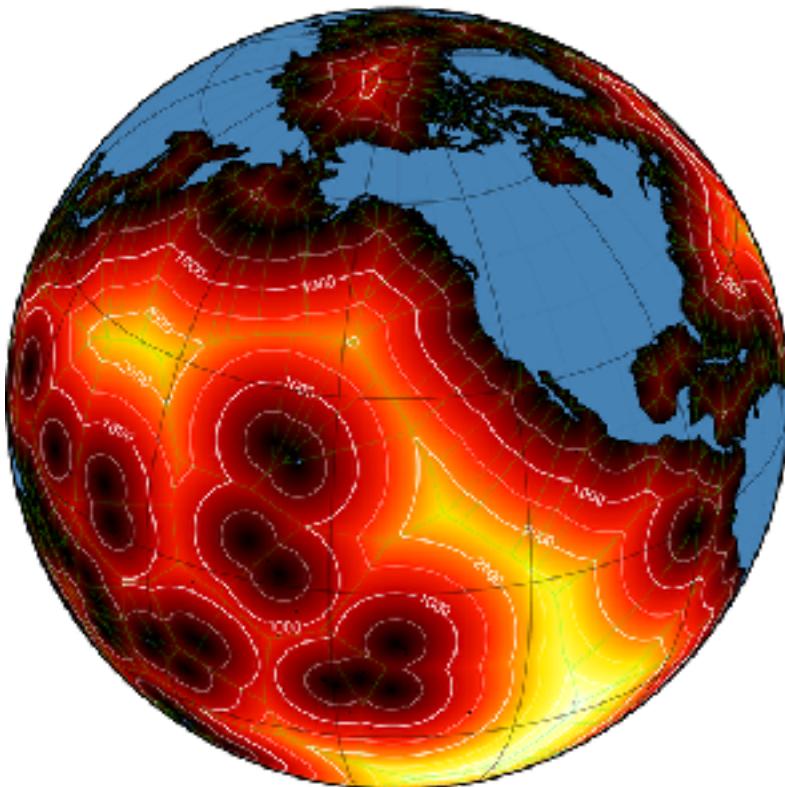
Venus Spherical Harmonic Model



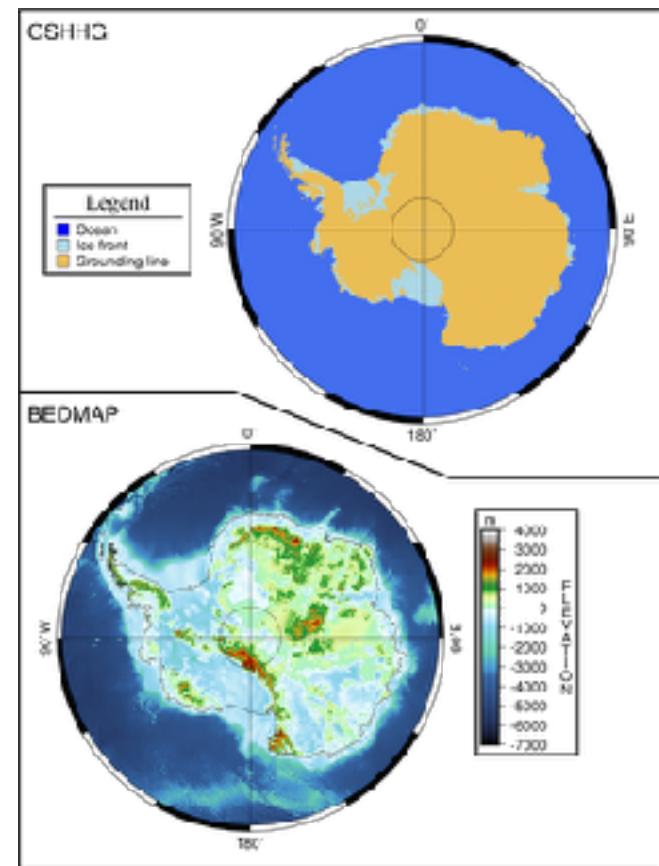
# GMT Examples

## *Spherical Triangulation and Distance Calculations*

Distances from GSHHG crude coastlines

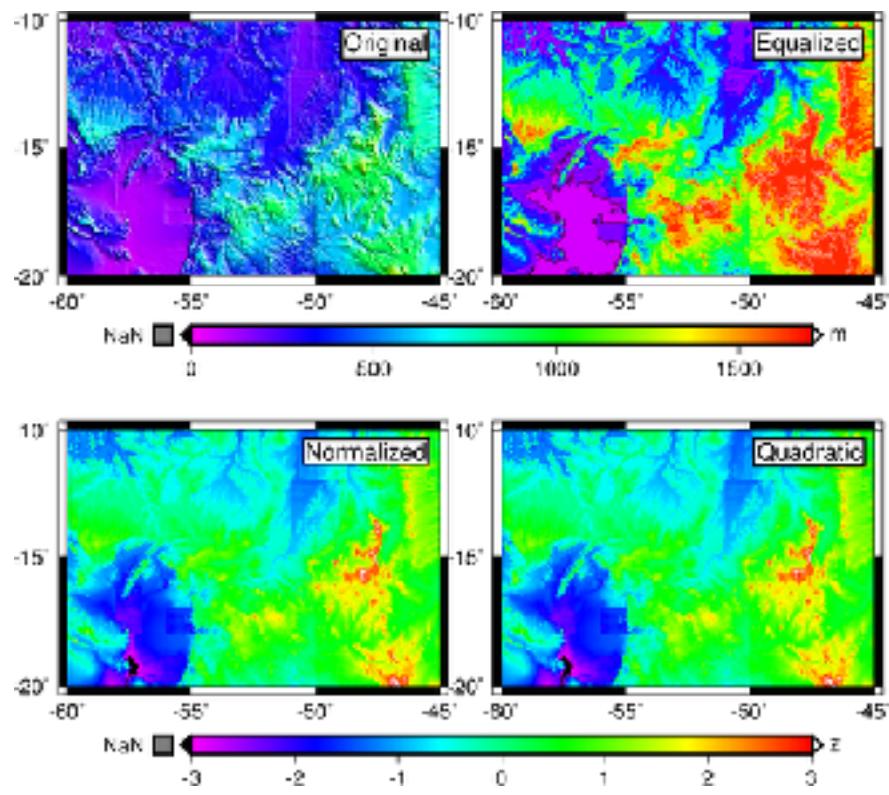


## *Antarctica and Stereographic Projections*

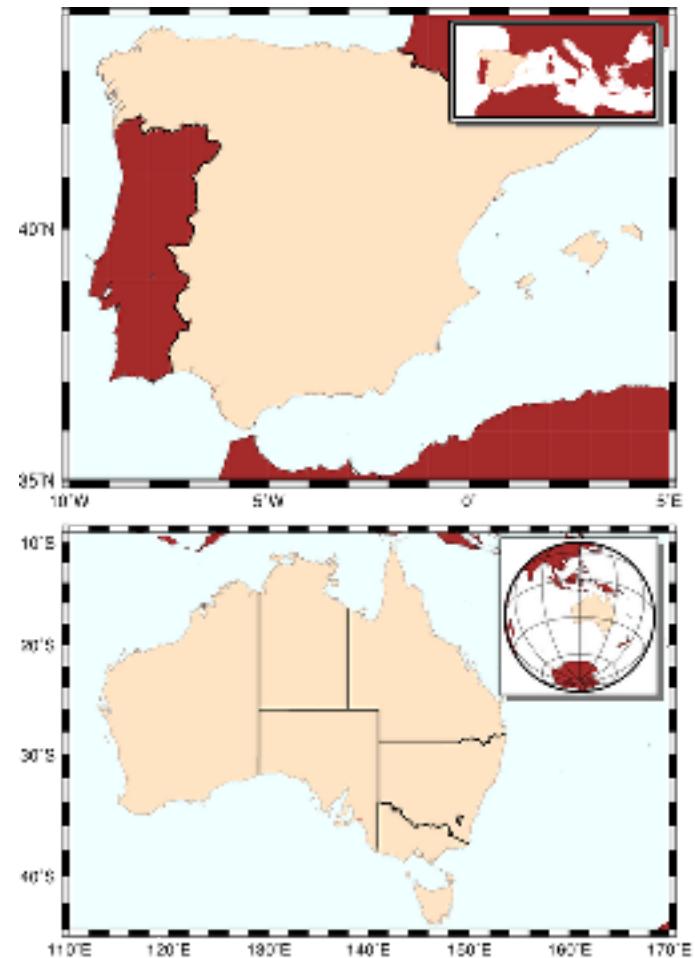


# GMT Examples

## *Histogram Equalization of Bathymetry Grids*

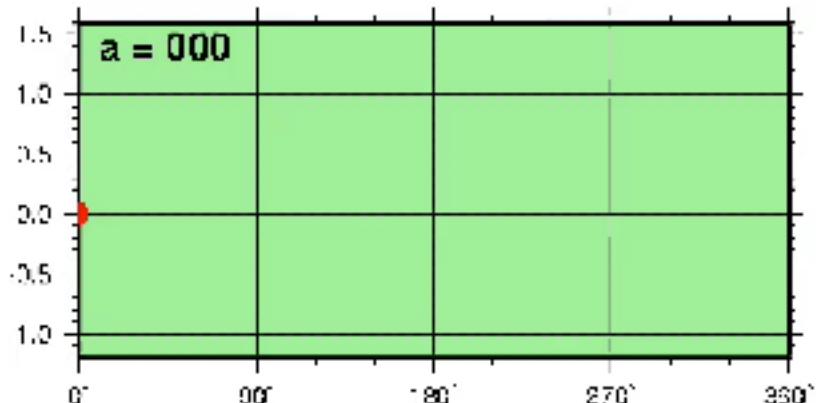


## *Map Insets*

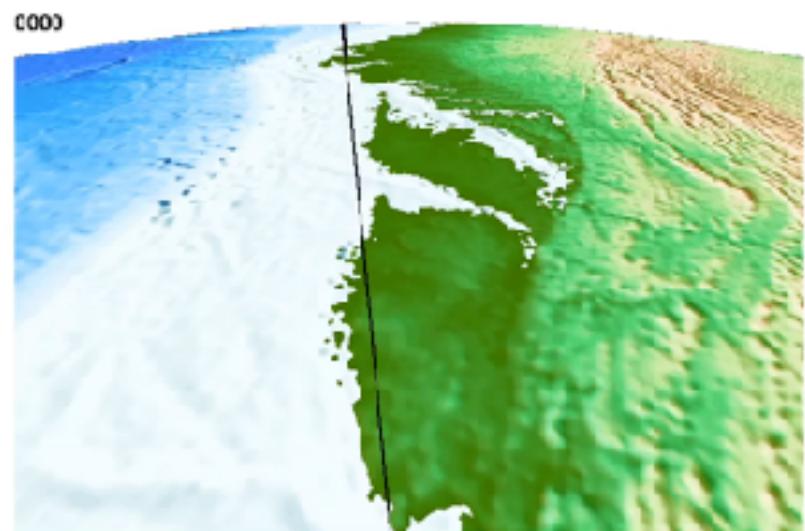


# GMT Examples

*Animation of Sine Function*



*Topography Flyover*



# Getting Started w/ GMT

- Run GMT API through Python
- Type ‘source activate gmt-python’ from a terminal window
- Type ‘jupyter-notebook’ to launch Jupyter
- Open the notebook ‘first-steps.ipynb’ available on the git repository

# My first map

## GMT command line syntax

```
gmt pscoast -R-90/-70/0/20 -JM6i -P -Ba -Gchocolate > GMT_mercator.ps
```

## GMT Python API syntax

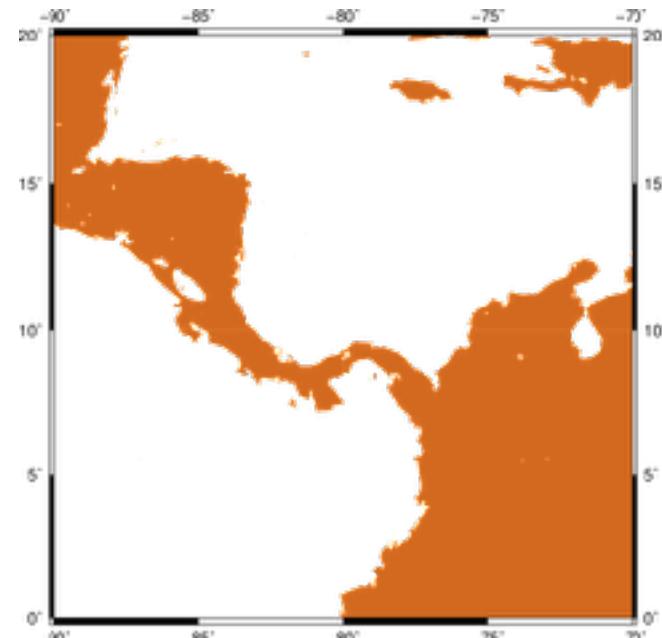
```
fig.coast(region=[-90, -70, 0, 20], projection='M6i', land='chocolate', frame=True)
```

## GMT Python API using command line syntax

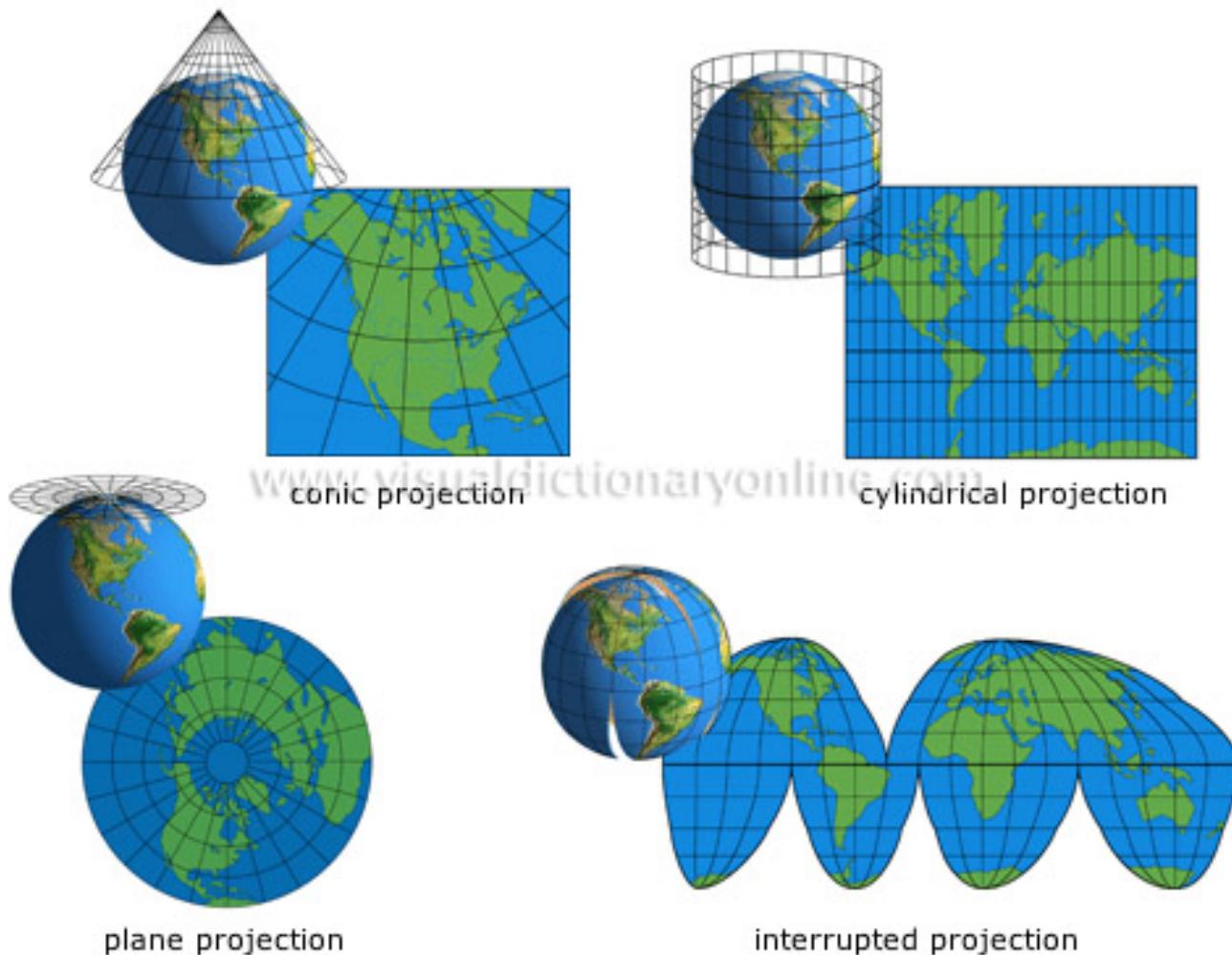
```
fig.coast(R='-90/-70/0/20', J='M6i', G='chocolate', B=True)
```

Exercises:

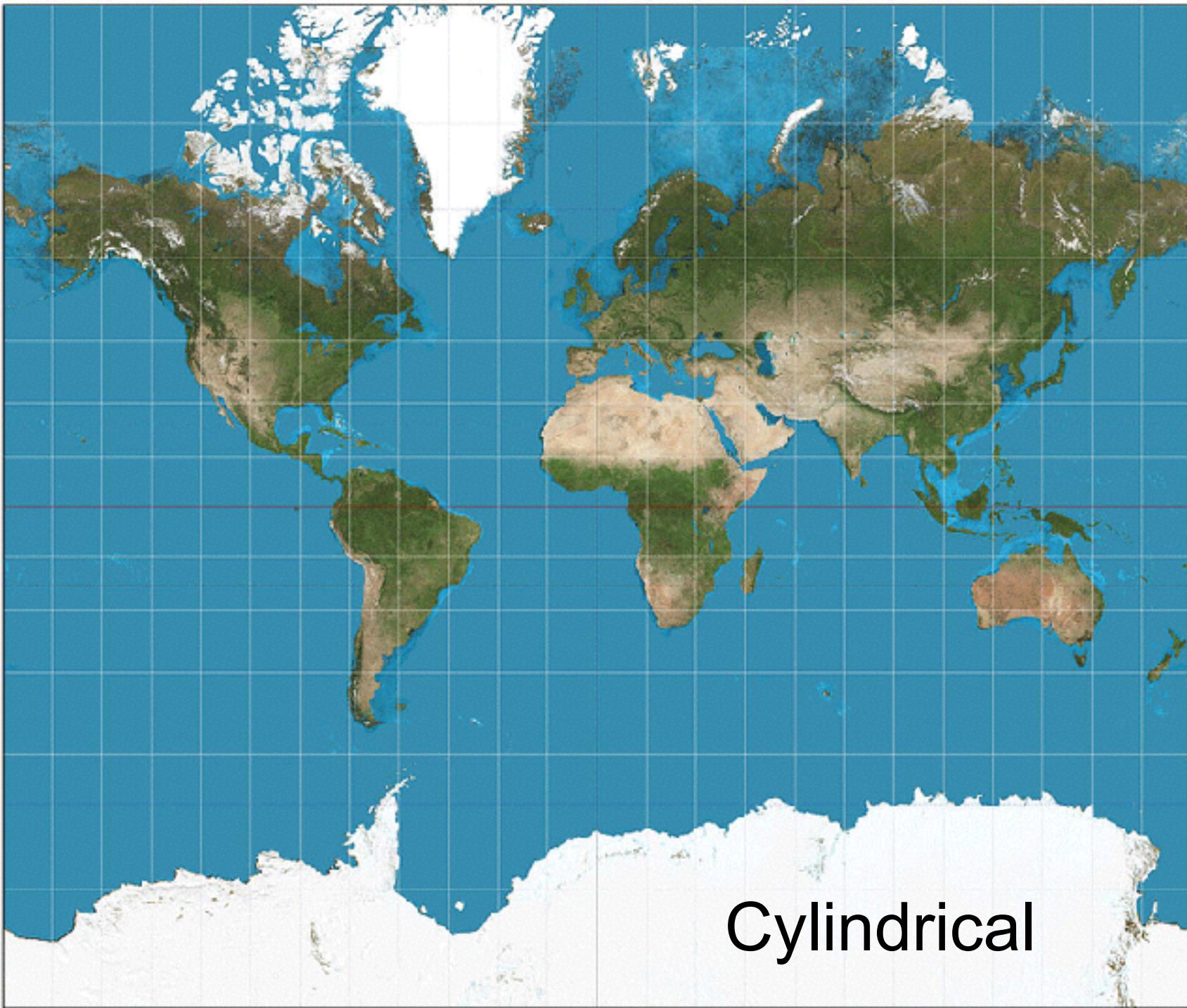
1. Pick another land color
2. Pick another water color
3. Pick a region that includes the north or south poles.
4. Try **-W0.25p** instead of (or in addition to) **-G**. This is the pen alias in GMT.
5. What is the map projection?



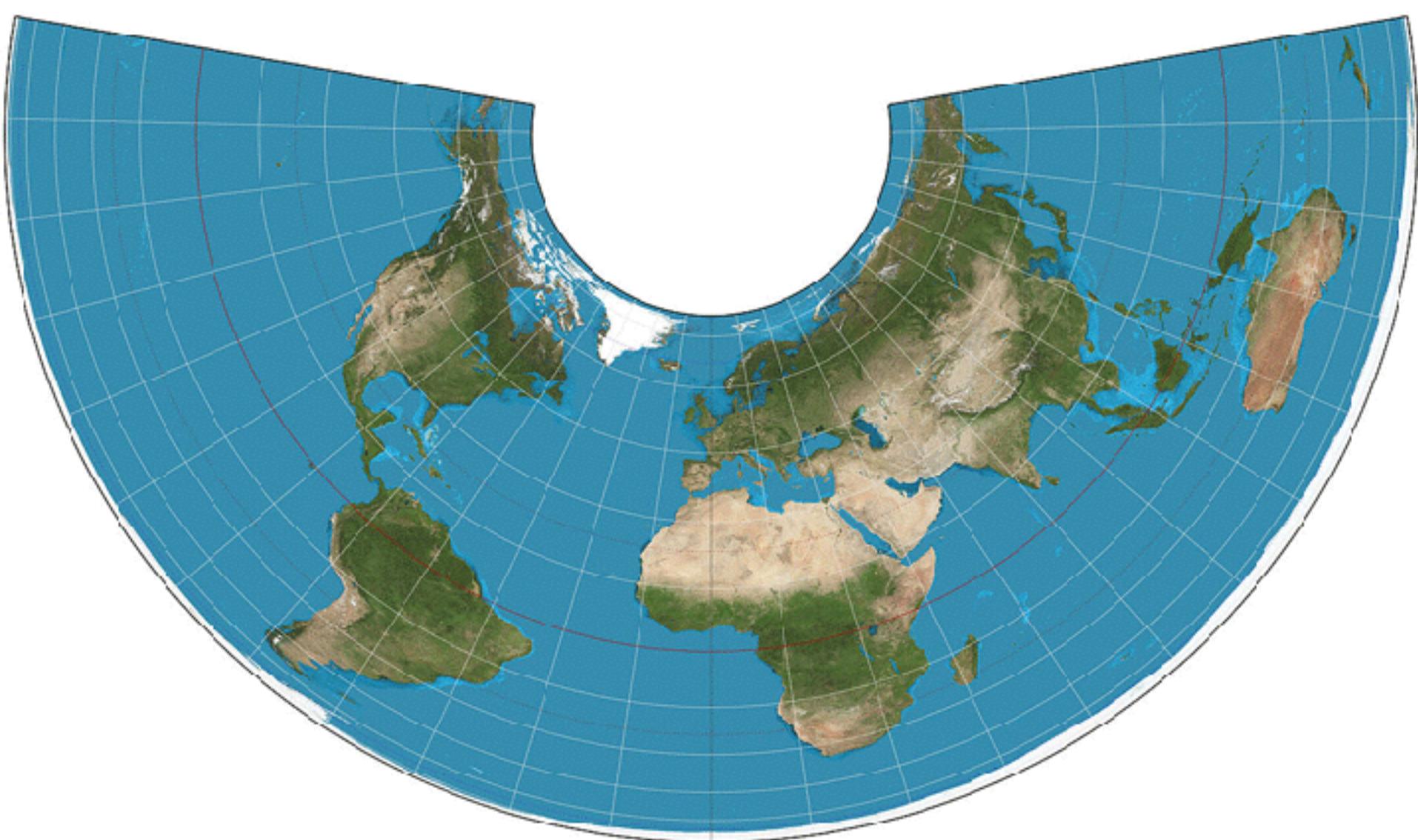
# 3 Basic Types of Map Projections



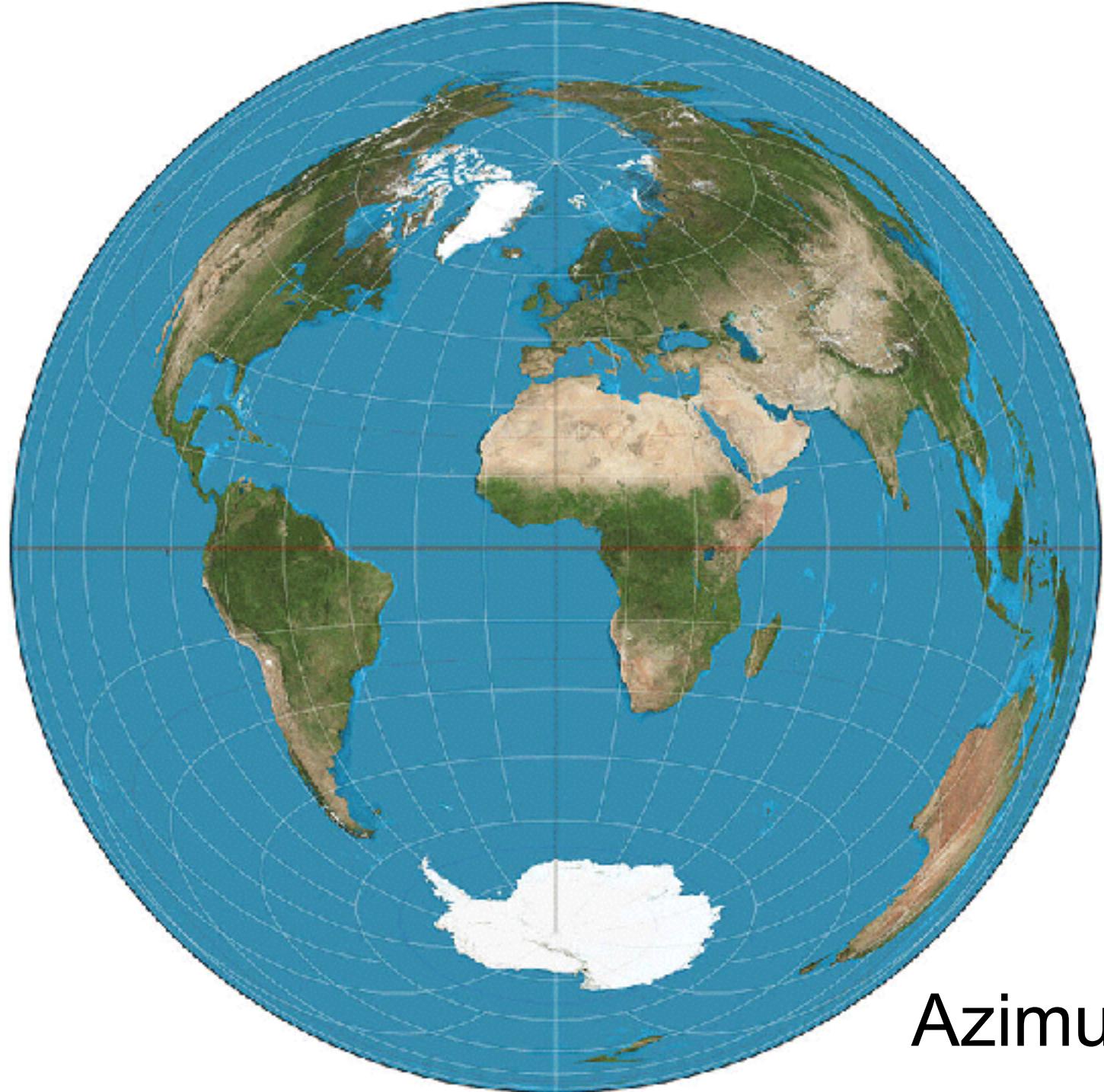
[https://en.wikipedia.org/wiki/List\\_of\\_map\\_projections](https://en.wikipedia.org/wiki/List_of_map_projections)



Cylindrical



Conical



Azimuthal

# Available GMT projections

C = Conformal  
E = Equal Area

## GMT PROJECTIONS

### GEOGRAPHIC PROJECTIONS

#### CYLINDRICAL

Basic [E]

Cassini

Equidistant

Mercator [C]

Miller

Oblique Mercator [C]

Stereographic

Transverse Mercator [C]

UTM [C]

#### CONICAL

Albers [E]

Equidistant

Lambert [C]

Polyconic

#### AZIMUTHAL

Equidistant

Gnomonic

Orthographic

Perspective

Lambert [E]

Stereographic [C]

#### THEMATIC

Eckert IV + VI [E]

Hammer [E]

Mollweide [E]

Robinson

Sinusoidal [E]

Winkel Tripel

Van der Grinten

#### OTHER

Linear

Logarithmic

Exponential

Time

Polar

# Map Projections

- <https://www.jasondavies.com/maps/transition/>

# Orthographic Projection

```
gmt pscoast -Rg -JG280/30/6i -Bag -Dc -A5000 -Gwhite -SDarkTurquoise -P >  
GMT_tut_5.ps
```

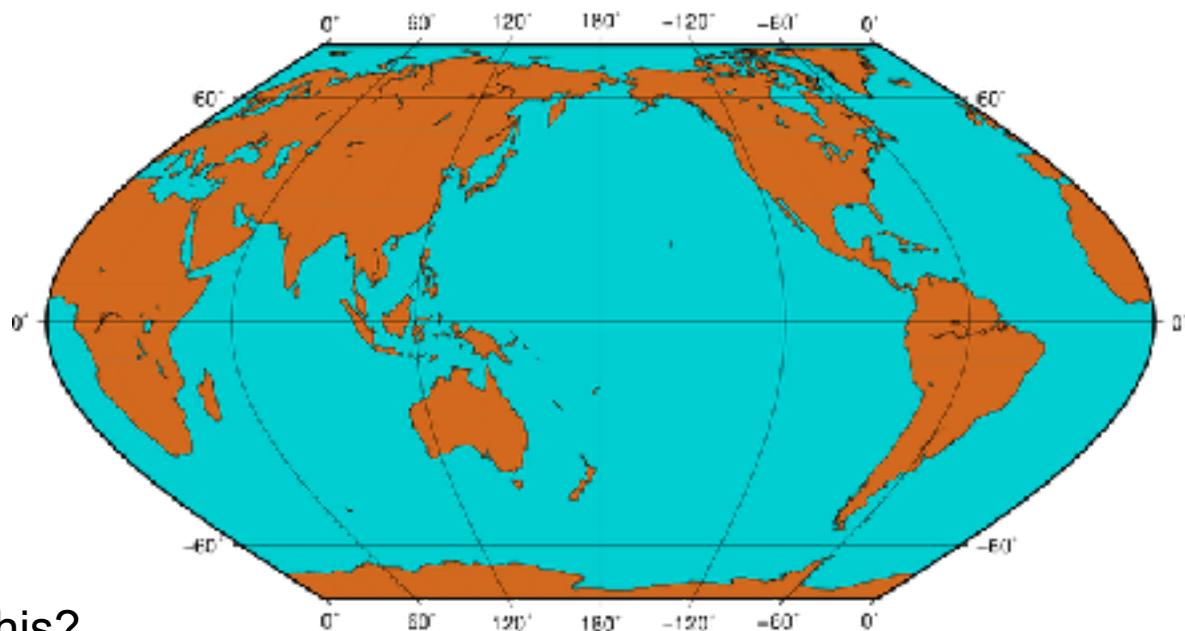


## Exercises:

1. What type of map projection is shown?
2. Write this command and create this map using Python GMT
3. Use the rectangular option in -R to make a rectangular map showing the US only.

# Eckert VI Projection

```
gmt pscoast -Rg -JKs180/9i -Bag -Dc -A5000 -Gchocolate -SDarkTurquoise  
-Wthinnest > GMT_tut_6.ps
```



Exercises:

1. What type of projection is this?
2. Rewrite the command in Python GMT
3. Center the map on Greenwich.
4. Add a map scale with **-L**.

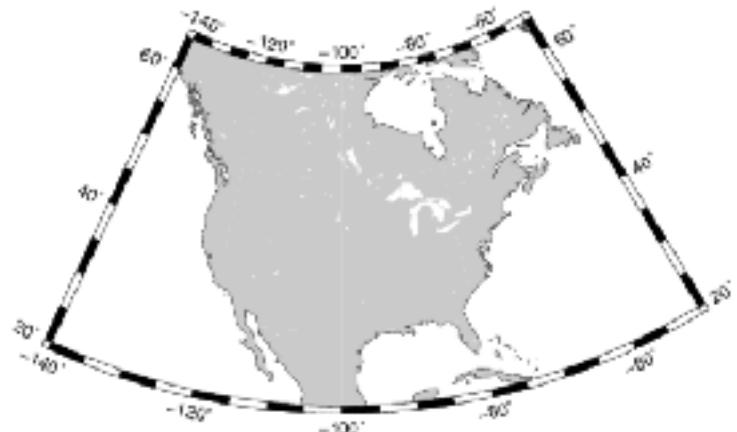
# Albers Projection

```
gmt pscoast -R-140/-50/20/65 -JB-100/35/33/45/6i -Ba20f5/a20f5 -Dc -Ggray -  
W1/0 -P > GMT_alpers.ps
```

What is the map projection?

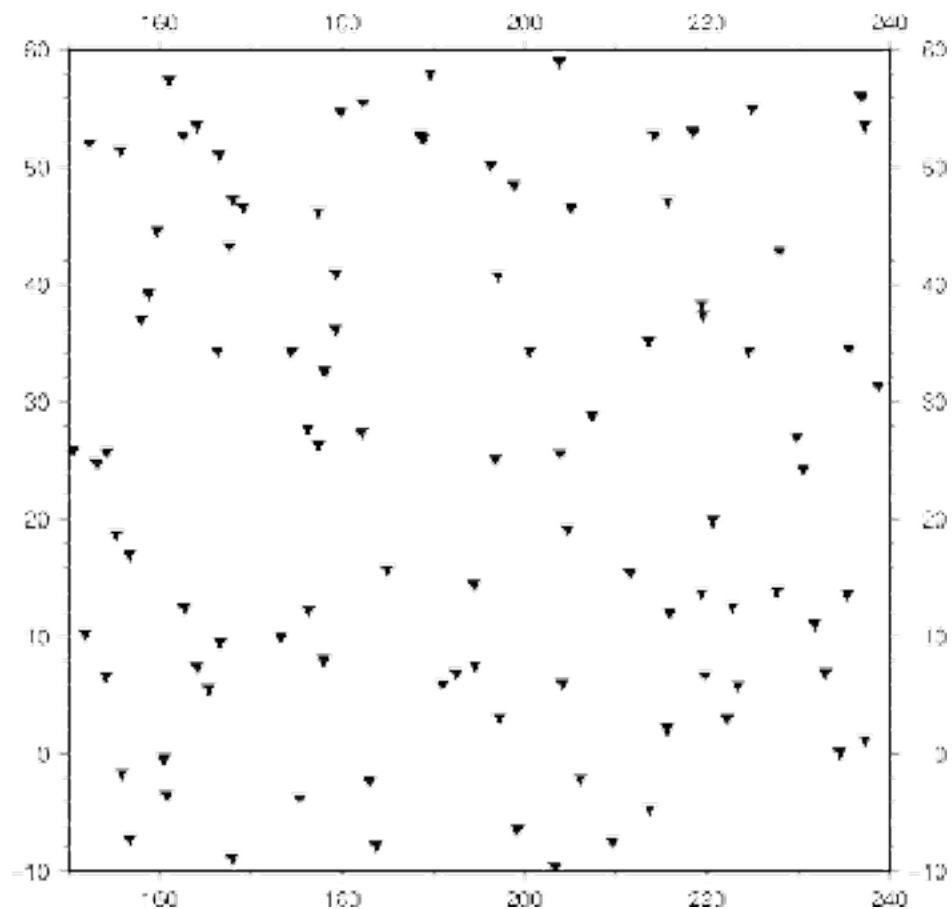
Rewrite the command in GMT Python and add the following features to your map:

- Longitudinal interval marks every 20°
- Latitudinal interval marks every 10°
- Light brown land masses
- Light blue oceans
- Intermediate resolution coastlines
- 1500 km long map scale in bottom right corner
- 7.5 inches wide
- All major rivers in blue pen
- State boundaries in dashed black pen
- Country borders in solid red
- Solid green star showing your favorite city
- Text annotation of your favorite city
- Map Title



# Linear Projection for XY plots

1. Can you recreate this plot in matplotlib? How?
2. Try plotting your example dataset from Tuesday's matplotlib tutorial.
3. Which is easier?



# Independent Mapping Project

1. Pick your favorite continent, country, or region.
2. Create a map that shows the topography/bathymetry of your area.
3. Plot some point data on your map
  - Earthquakes
  - Volcanoes
  - Airport locations
  - Beaches
  - Lightning strikes
4. Save your map for upload to the git repository
5. Display your map on a Nasa WorldWind globe

Be prepared to say what map projection you choose and why.

Some ideas include plot recent seismicity on Hawaii, plot volcano locations in Alaska, plot LIGO data on a map etc.

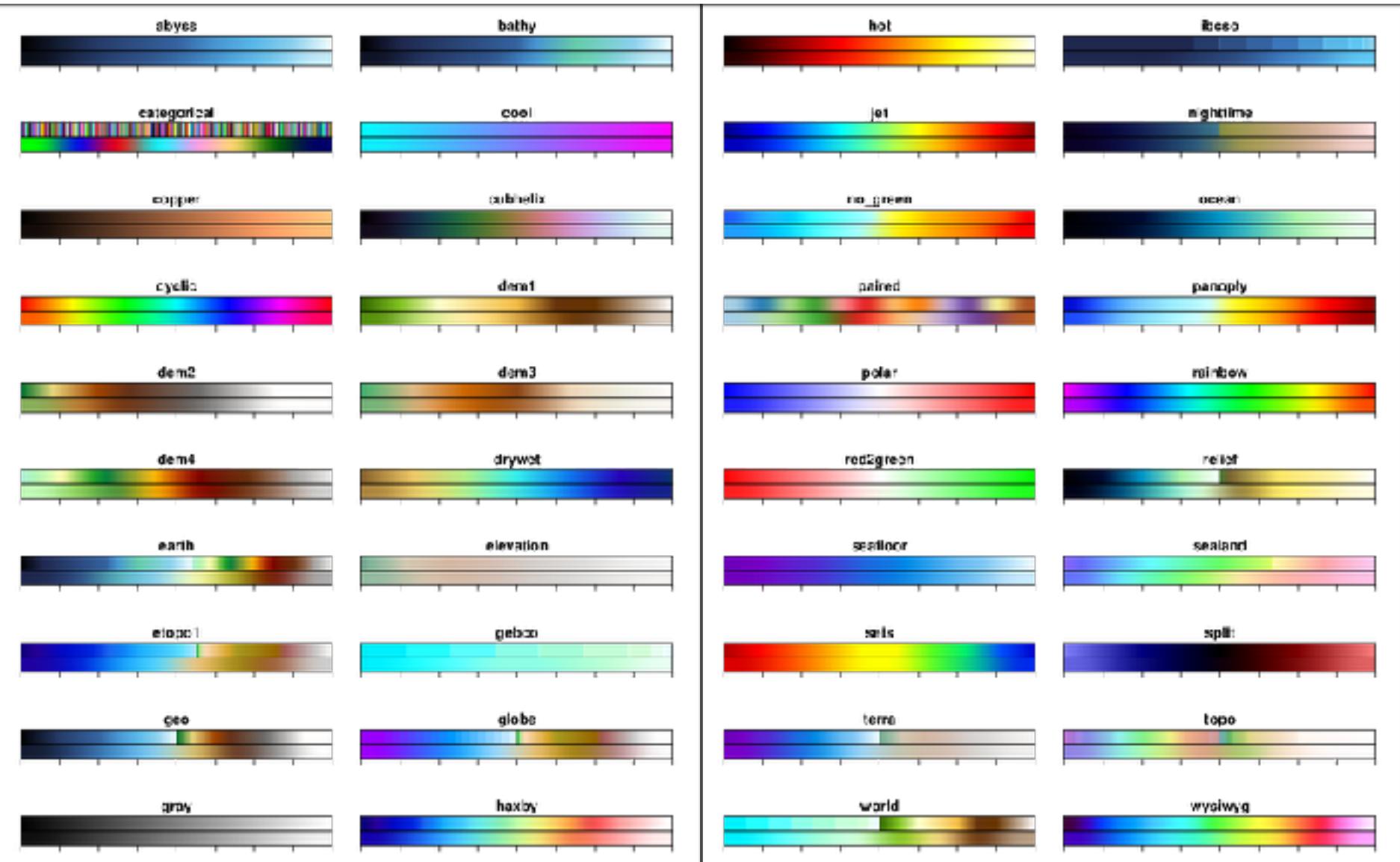
# LIGO data exercises

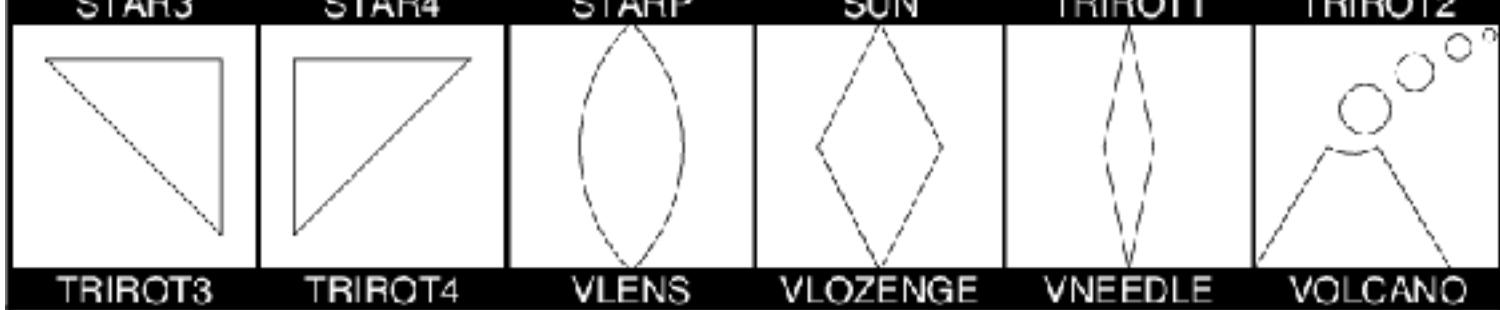
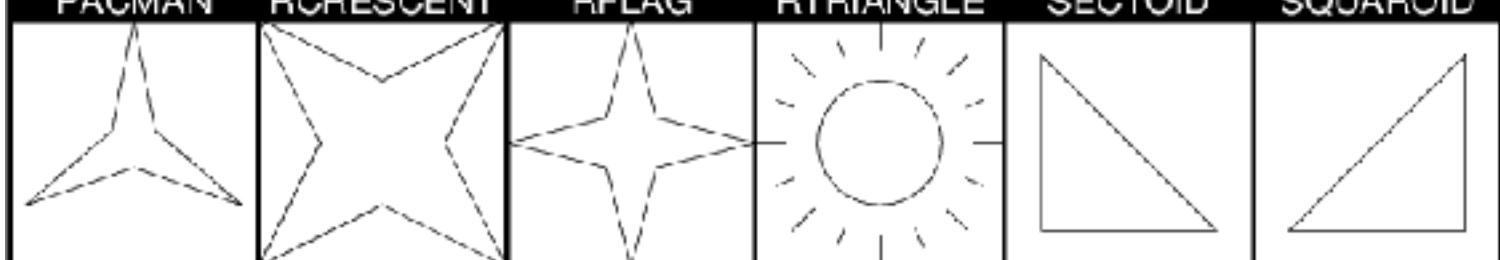
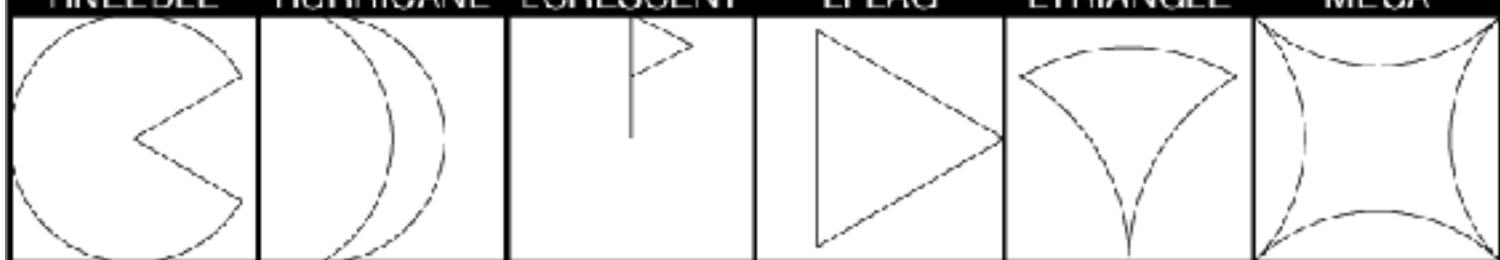
- Plot L1 data
- Plot H1 data
- Plot L1 data overlaid on H1 data
- Plot L1 and H1 station locations on a map showing their geospatial relationship
- Plot observed data as inset on map
- Plot line connecting Hanford and Livingstone stations. Is it straight? What if you change the map projection?
- Add topography to your map

# Example Datasets

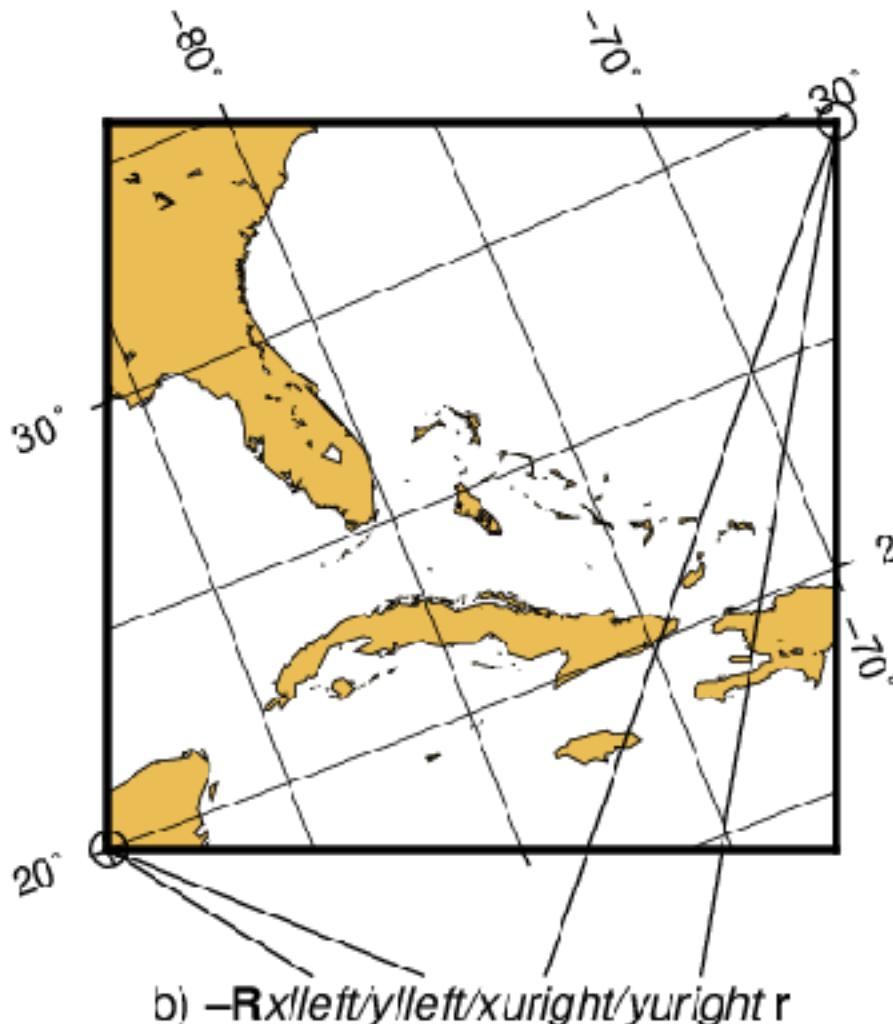
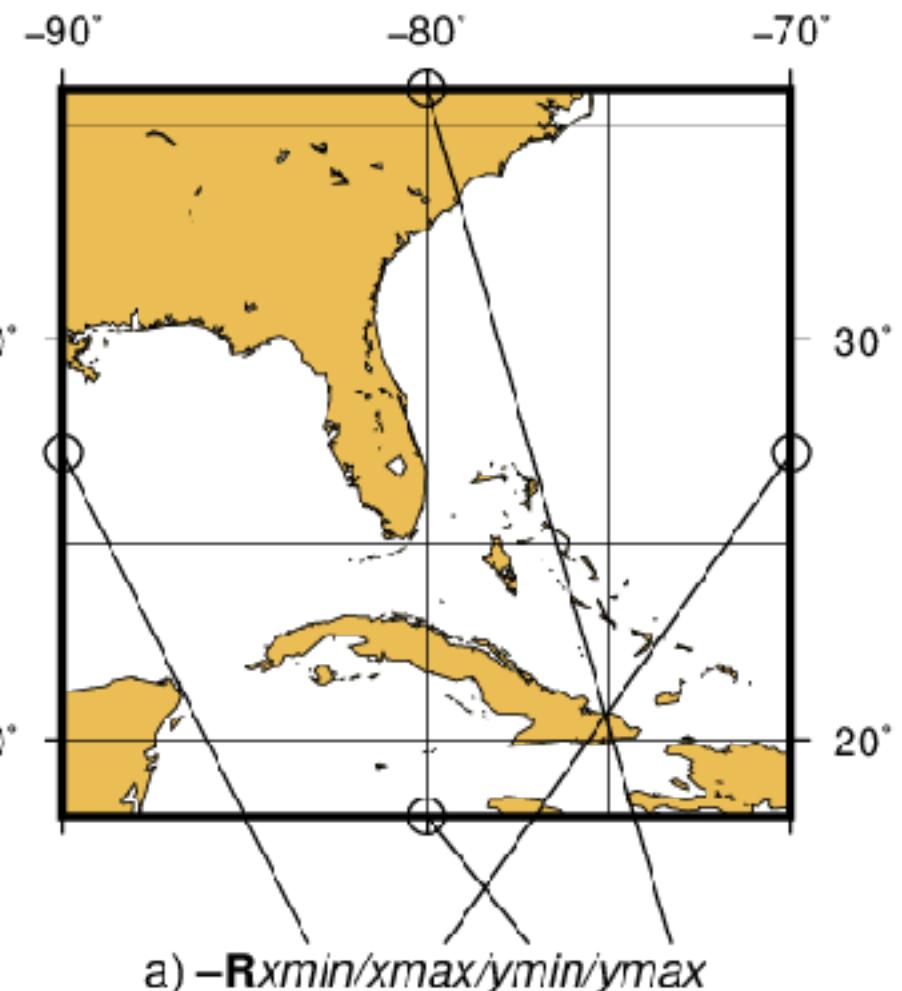
- Global topography/bathymetry data
  - <https://www.ngdc.noaa.gov/mgg/global/>
- LIGO data
  - <https://losc.ligo.org/data/>
- Earthquakes
  - <https://earthquake.usgs.gov/earthquakes/search/>
- Volcanoes
  - <https://sos.noaa.gov/datasets/volcano-locations/>

# Some Useful References

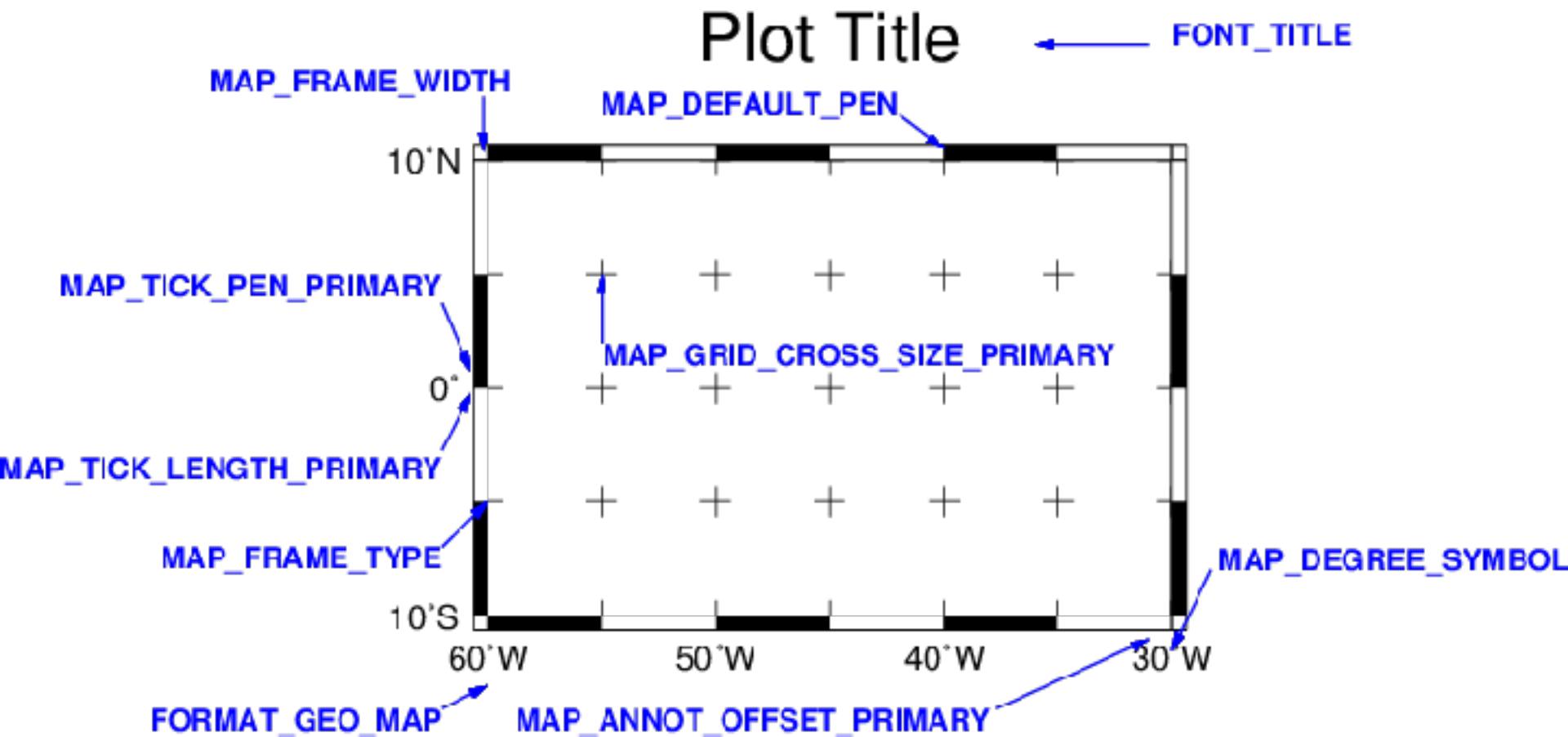




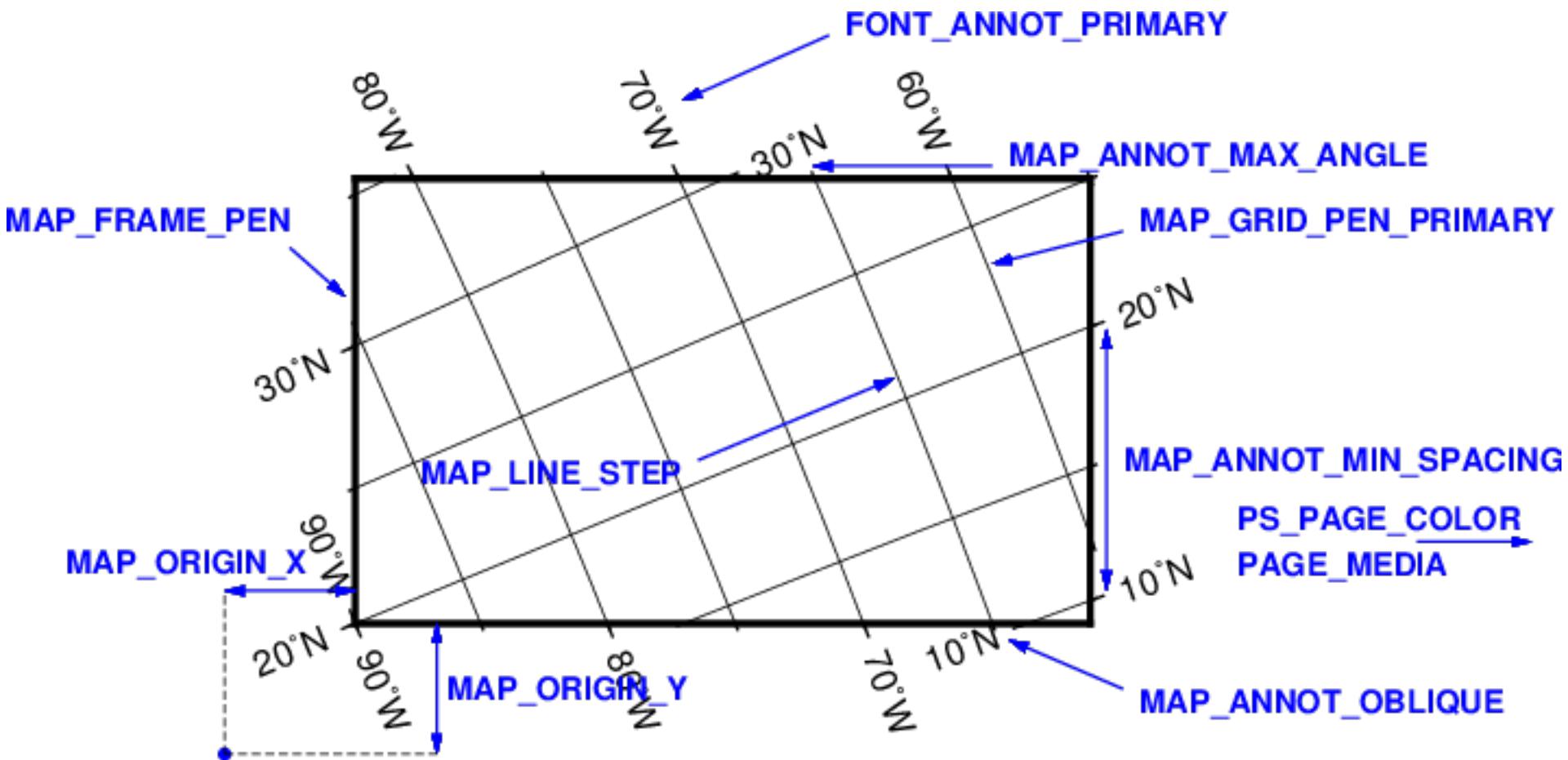
# Map Region: the -R option



# GMT defaults



# GMT defaults



# GMT defaults

