

Usando GMT en Python: Una introducción a la librería PyGMT



+



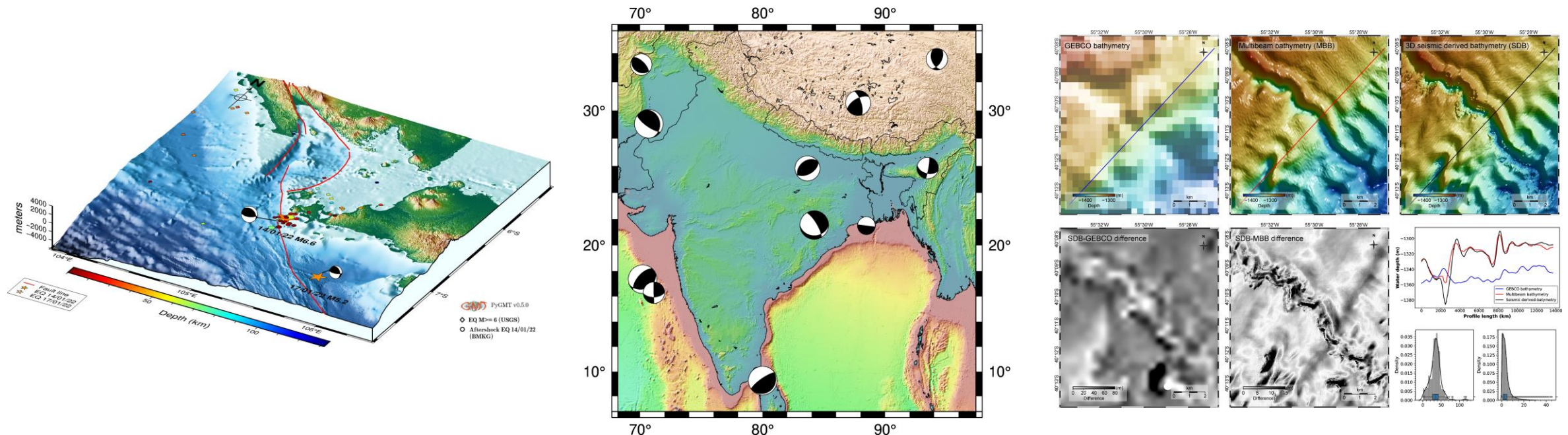
Sebastian Principi

Grupo de prospección Geológica y Geofísica Lacustre y Marina del IGEBA

sebaprincipi@gmail.com

¿Por qué usar PyGMT?

- Sintaxis: En líneas generales es mas literal y mas entendible para usuarios nuevos de GMT
- Python: Si ya se tiene una idea del lenguaje, es muy fácil empezar a crear scripts
- Integración con otras librerías de Python: la principal ventaja, es posible usarlo en conjunto con una gran numero de librerías para análisis de datos geográficos, estadísticos y de manejo de grillas
- Al igual que GMT, se tiene como producto figuras de alta calidad para publicaciones científicas



¿Cómo lo usamos?

Anatomía de una figura de PyGMT

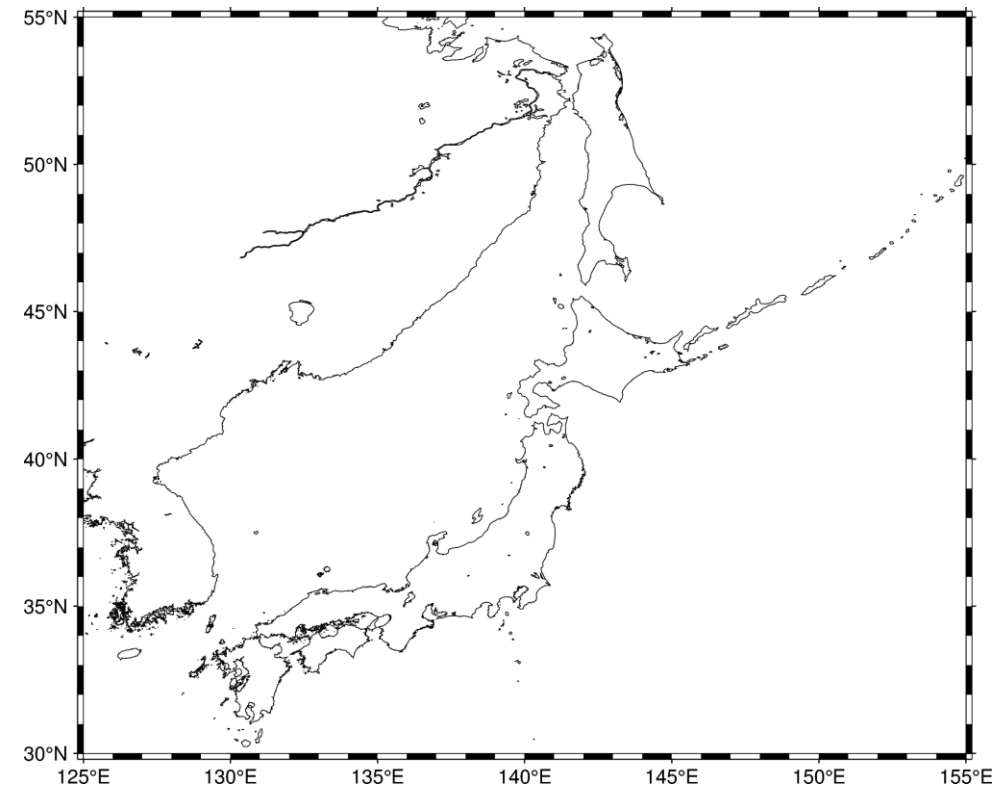
Importo la
librería

Creo la
figura

Voy
invocando
funciones
de GMT con
sus
parametros

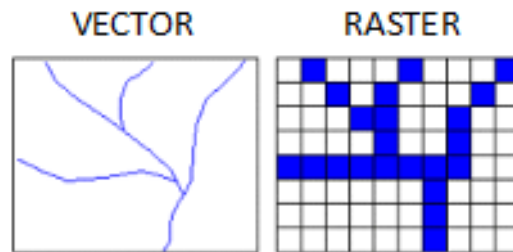
Muestro la
figura

```
8
9  import pygmt
10
11
12  fig = pygmt.Figure()
13
14
15  fig.coast(
16      region=[125, 155, 30, 55],
17      shorelines=True,
18      frame=True,
19  )
20
21
22  fig.show()
```



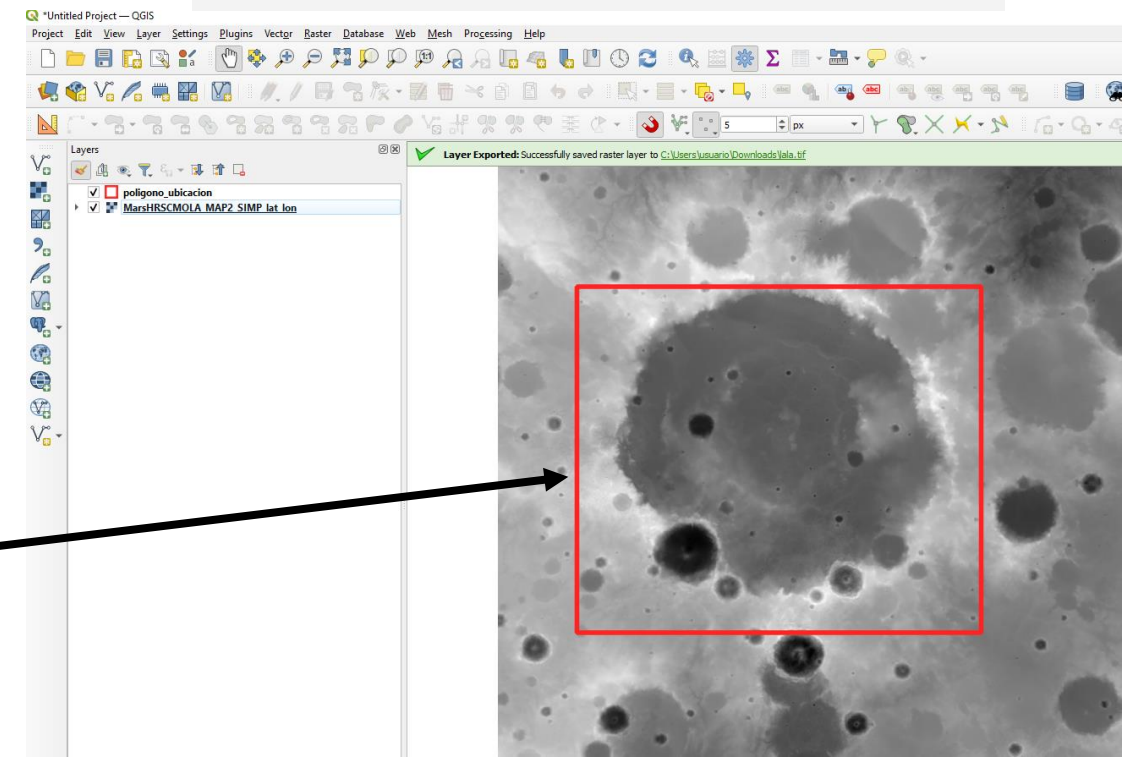
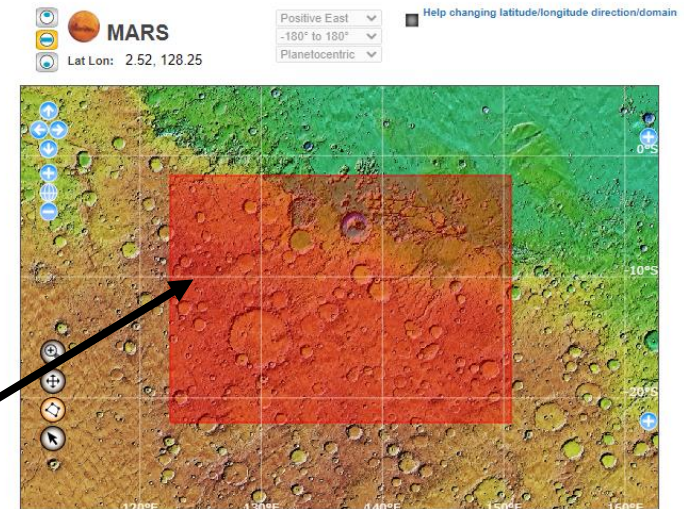
Todo muy lindo ¿Cómo lo uso con mis propios datos?

- Los datos que mas solemos usar para hacer mapas son archivos vectoriales y rasters



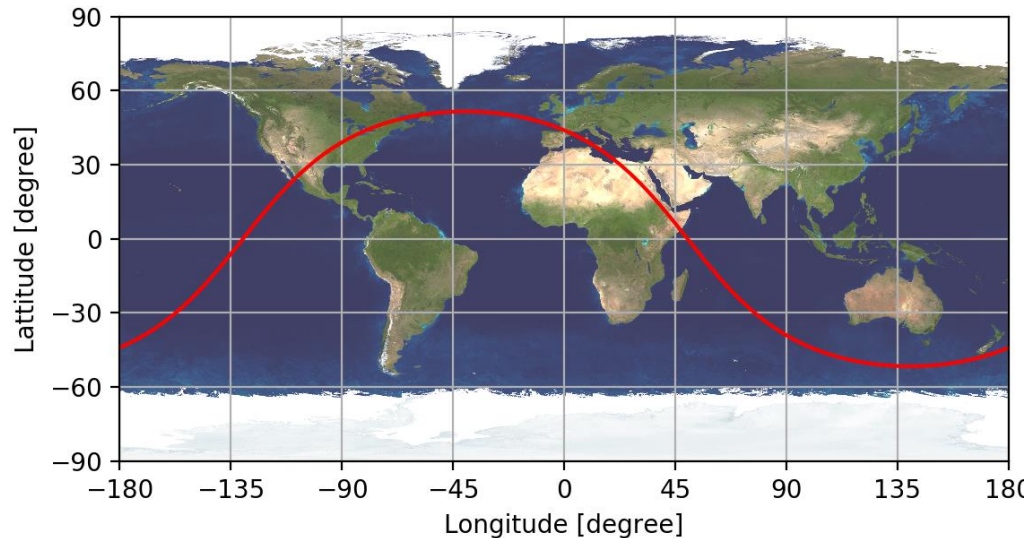
- Para hacer un ejemplo usamos:

- Raster: de un modelo de elevación digital de marte bajado de:
https://astrogeology.usgs.gov/search/map/Mars/GlobalSurveyor/MOLA/Mars_MGS_MOLA_DEM_mosaic_global_463m
- Un shapefile creado en QGis

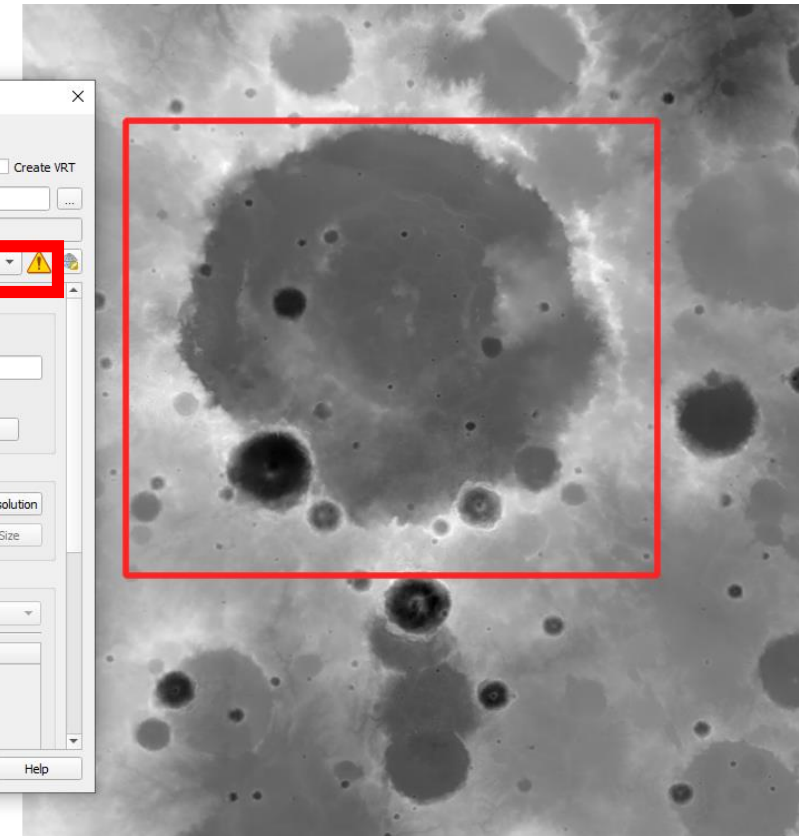
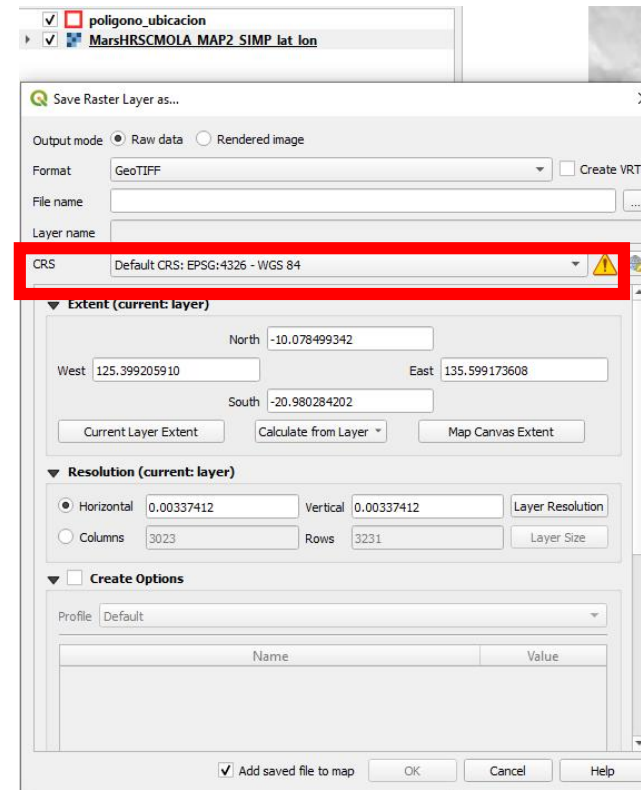


Todo muy lindo
¿Cómo lo uso con mis propios datos?

- Algo importante: Al exportar los datos de Qgis (o ArcGIS/Global Mapper, etc) hay que hacerlo en latitud / longitud (EPSG: 4326) para poder usarlo en GMT



- Proyección equirectangular (también llamada geográfica o lat/lon).
Código EPSG 4326



Ejemplo: Mapa topográfico 2D

Importo las librerías
 Rioxarray: para cargar los rasters en forma de grilla
 Geopandas: para cargar archivos vectoriales como shapefiles

Ruta archivos

Leo archivos con rioxarray (dem) y geopandas
 (shapefiles - shp)

Creo la figura

Voy invocando funciones de GMT con
 sus parametros

Guardo y muestro la figura

```
import pygmt
import rioxarray as rxr
import geopandas as gpd

###
ruta_dem=r'C:\Users\usuario\Downloads\MarsHRSCMOLA_MAP2_SIMP_Lat_Lon.tif'
ruta_shp=r'C:\Users\usuario\Downloads\poligono_ubicacion.shp'

###
dem=rxr.open_rasterio(ruta_dem,masked=True).squeeze()
shp=gpd.read_file(ruta_shp)

###
fig = pygmt.Figure()

pygmt.makecpt(
    cmap='roma',
    series=[int(dem.min()),int(dem.max()),1],
    continuous=True,
    reverse=True
)

fig.grdimage(
    grid=dem,
    projection='M12c',
    shading='a0+nt1',
    dpi=300,
    frame=True,
    cmap=True,
    nan_transparent=True
)

fig.colorbar(frame=["x+LElevacion [m]"])

fig.grdcontour(
    interval=100,
    grid=dem,
    transparency="40",
    pen="0.1p"
)

fig.plot(data= shp,
        pen="1p,black"
)

fig.basemap(
    frame=True,
    map_scale="n0.8/0.1+w200k+f+l",
    rose=["n0.85/0.9+w1.5c+f"],
)

fig.savefig("mars_v2.png", crop=True, dpi=300)
fig.show()
```

Creo paleta de colores

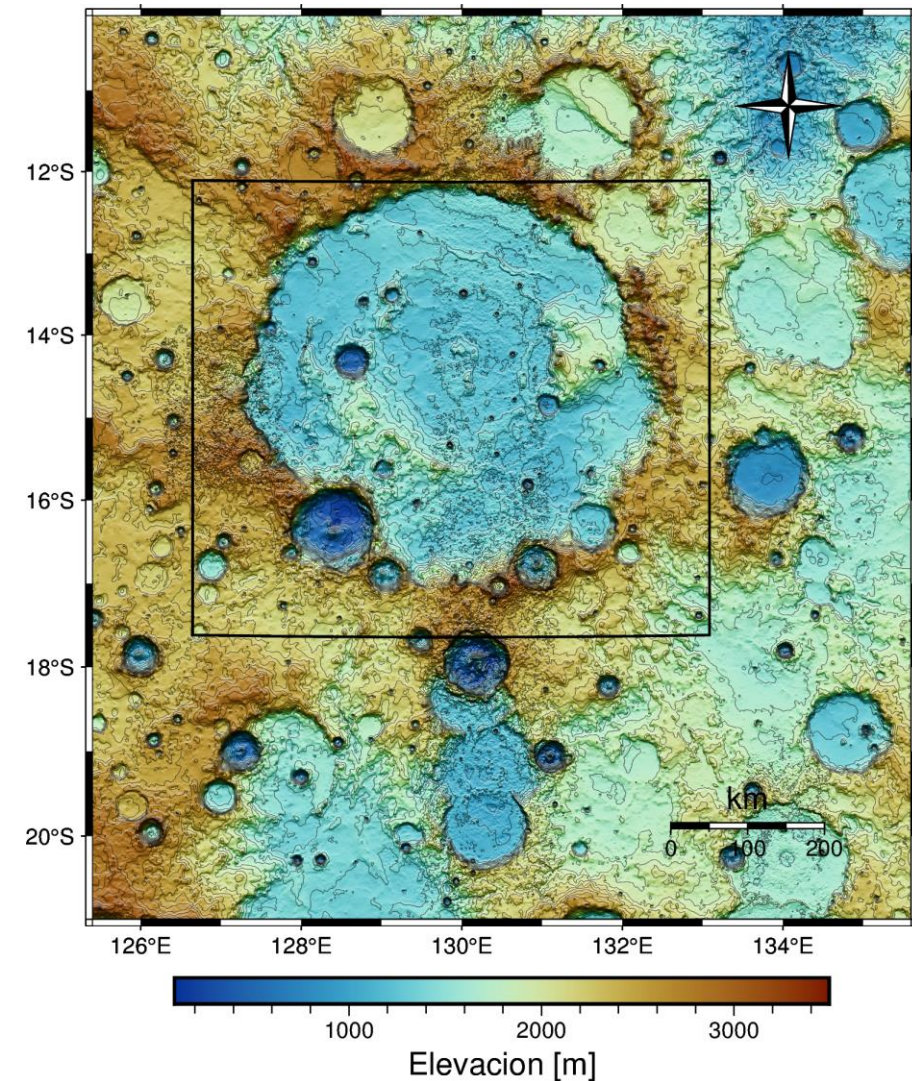
Grafico el raster (dem) con sombreado usando la paleta

Grafico la barra de colores

Curvas de nivel

Grafico el recuadro (shp)

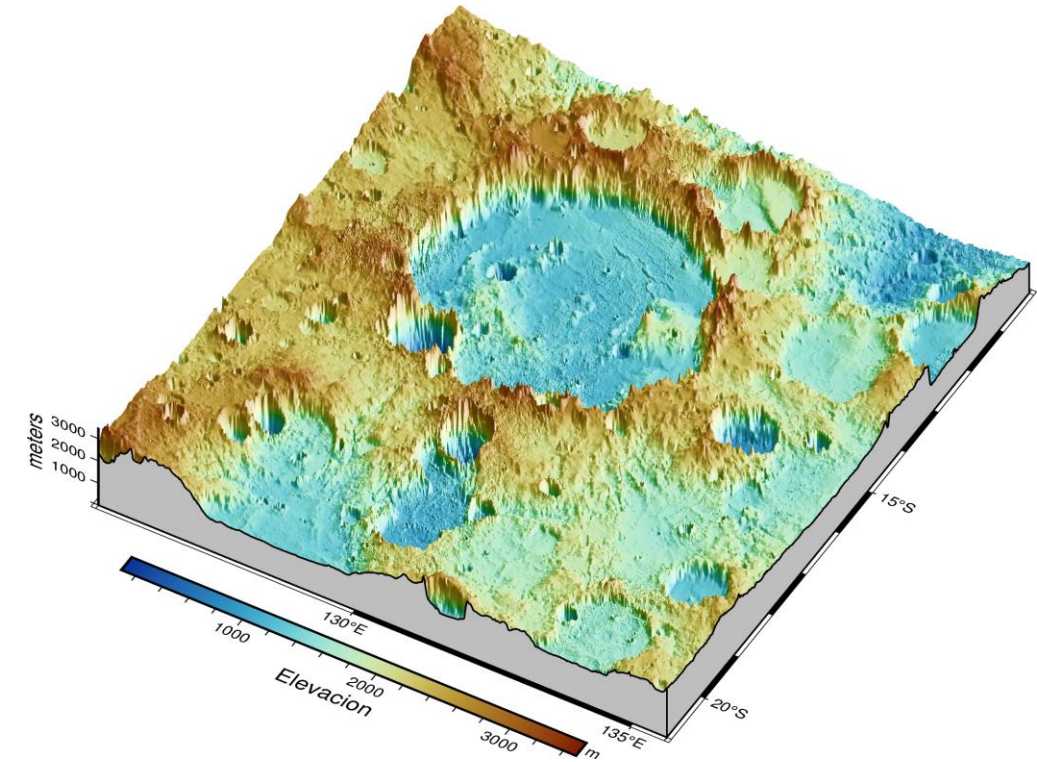
Norte y escala



Referencias:
<https://www.earthinversion.com/utilities/pygmt-high-resolution-topographic-map-in-python/>

Ejemplo: Mapa topográfico en perspectiva 3D

Importo las librerías	<code>import pygmt import rioarray as rxr import geopandas as gpd</code>
Ruta archivos	<code>ruta_dem=r'C:\Users\usuario\Downloads\dem_mars_4326.tif' ruta_shp=r'C:\Users\usuario\Downloads\poligono_ubicacion.shp'</code>
Leo archivos	<code>dem=rxr.open_rasterio(ruta_dem,masked=True).squeeze() shp=gpd.read_file(ruta_shp)</code>
Creo la figura	<code>fig = pygmt.Figure()</code>
Voy invocando funciones de GMT con sus parametros	<code>pygmt.makecpt(cmap='roma', series=[int(dem.min()),int(dem.max()),1], continuous=True, reverse=True) fig.grdview(grid=dem, surftype='i', projection='M12c', perspective=[150,45], zsize='2c', plane="0+ggrey", frame=['xa','yaf','z1000+lmeters','wSEnZ'],shading='+a50+nt1')</code>
Guardo y muestro la figura	<code>fig.colorbar (perspective=True, frame=['x+lElevacion','y+lm']) fig.savefig("mars_3d.png", crop=True, dpi=300) fig.show()</code>



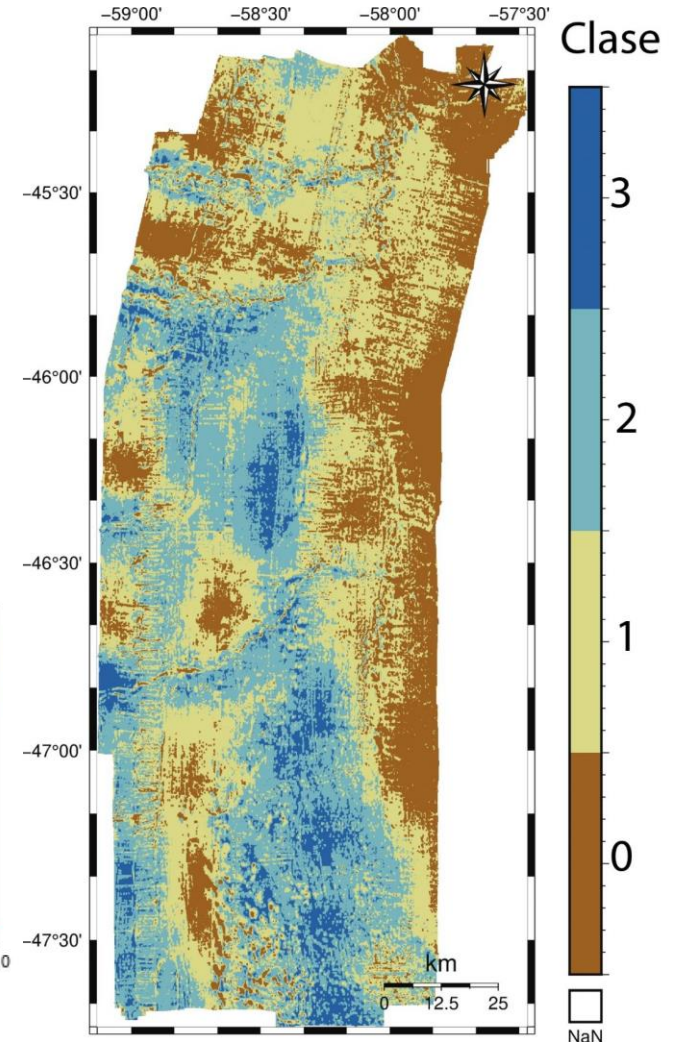
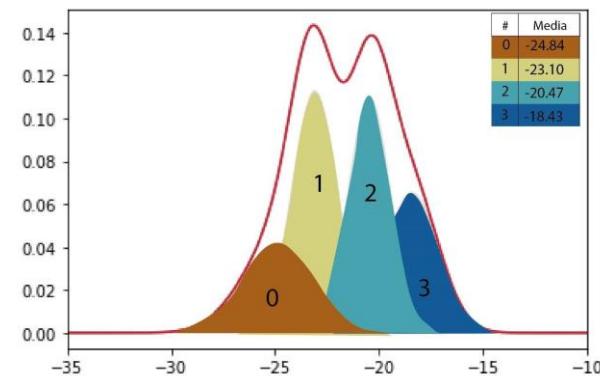
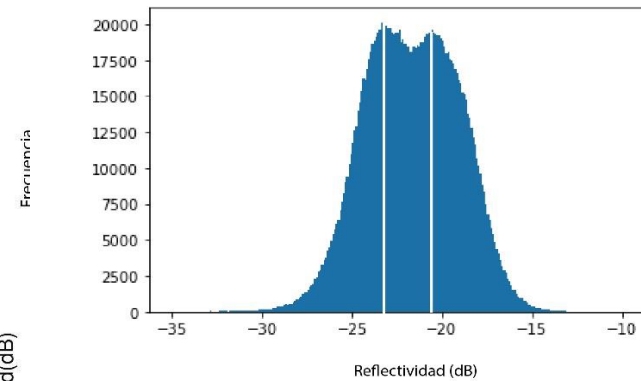
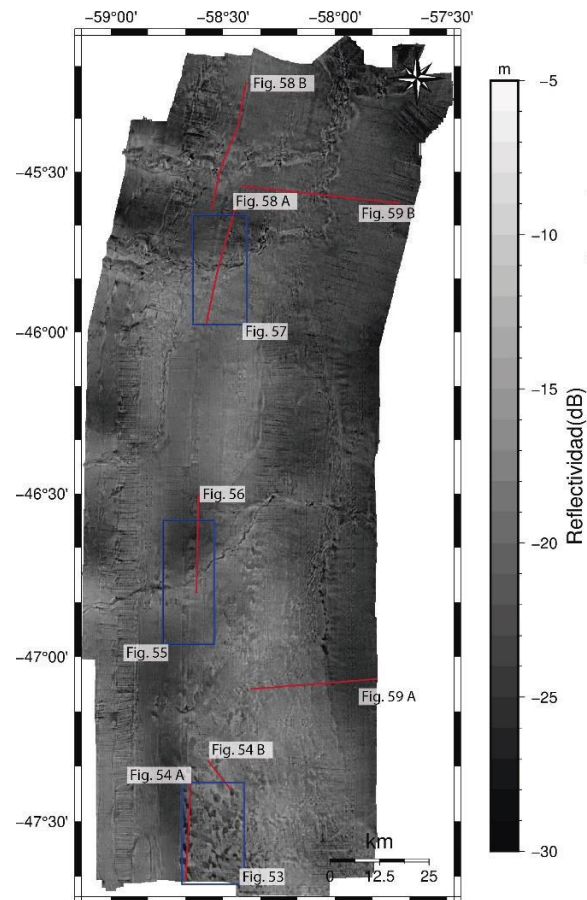
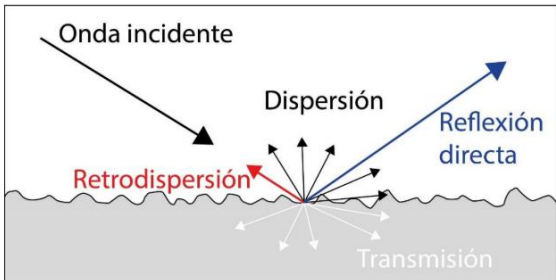
Referencias:

1. <https://www.earthinversion.com/utilities/Three-dimensional-perspective-map-of-Taiwan-using-GMT-and-PyGMT//>
2. https://www.pygmt.org/latest/tutorials/advanced/3d_perspective_image.html
3. https://www.generic-mapping-tools.org/egu22pygmt/mars_maps.html

Ejemplos con integración con otras bibliotecas de Python

- Scikit-learn – Manipulación de grillas con la biblioteca Xarray clasificación no supervisada del fondo mediante el GMM

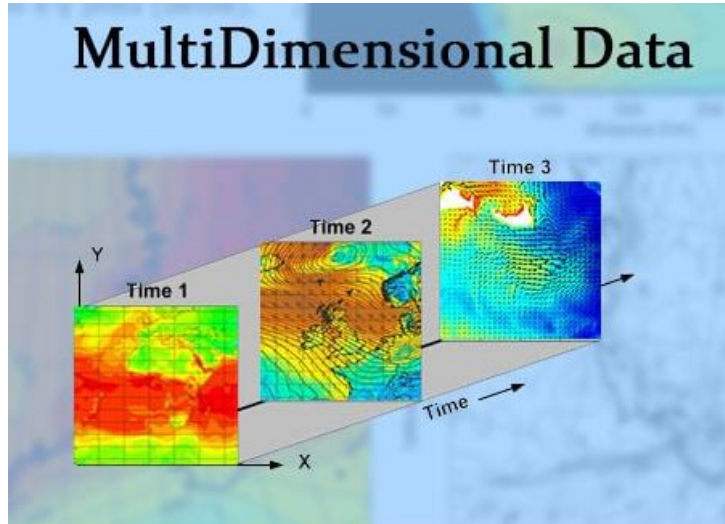
Retrodispersión



Ejemplos con integración con otras bibliotecas de Python

- Creación de animaciones con ciclos for y librería Open CV

Datos multidimensionales NETCDF



```
#Obtengo todas las fechas individuales
fechas=model_csv.time.unique()

#Genero un mapa por cada fecha
for i in range(0,len(fechas)):

    #Filtro por fechas los vectores de velocidad de uo y vo

    vector_df=model_csv[model_csv["time"]==fechas[i]][["Longitude", "Latitude","uo","vo"]]

    #Filtro por fecha los datos de magnitud de velocidad de corriente (ws)
    model_df=model_csv[model_csv["time"]==fechas[i]][["Longitude", "Latitude","ws"]]

    #Para ws necesito una grilla, primero paso los datos a dataArray para que los lea PyGMT
    idx=pd.MultiIndex.from_arrays(arrays=[model_df.latitude,model_df.longitude],names=['Y',
    'X'])
    s= pd.Series(data=model_df.ws.values, index=idx)
    model_data_array=xr.DataArray.from_series(s)

    #Grillo los datos de magnitud de ws

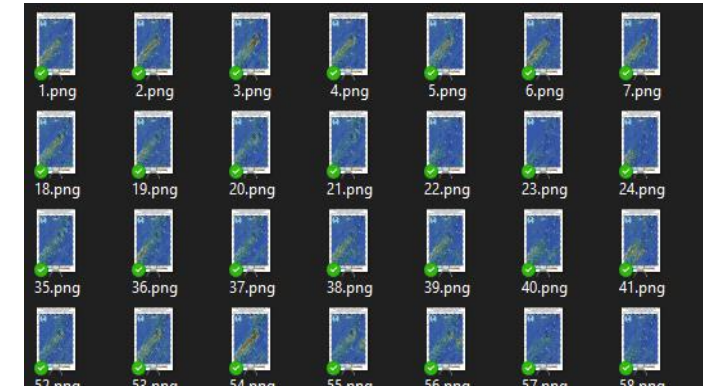
    model_sampled=pygmt.grdsample(model_data_array,spacing="0.3",interpolation="L")

    #genero figura
    fig = pygmt.Figure()
    pygmt.config(MAP_FRAME_AXES='WesN')

    #Grafico la magnitud de velocidad de la corriente ws

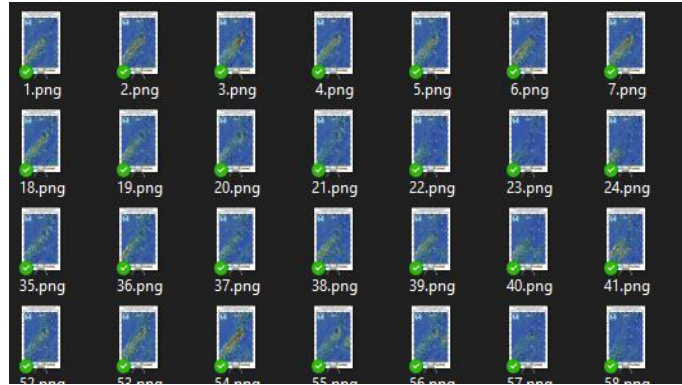
    pygmt.makecpt(cmap="roma",
                  series=(0,20,0.01),
                  reverse=True
                  )

    fig.grdimage(
        grid=model_sampled,
        projection="M14c",
        cmap=True,
        dpi=720,
        frame=True,
        region=subset_region,
        transparency="20",
        nan_transparent=True)
    fig.savefig(f'{fechas[i].strftime("%Y-%m-%d")}.png')
```



Ejemplos con integración con otras bibliotecas de Python

- Creación de animaciones con ciclos for y librería Open CV



```
#Genero la animación

image_folder = path_gif
video_name = 'MCAB_bottom_currents.avi'

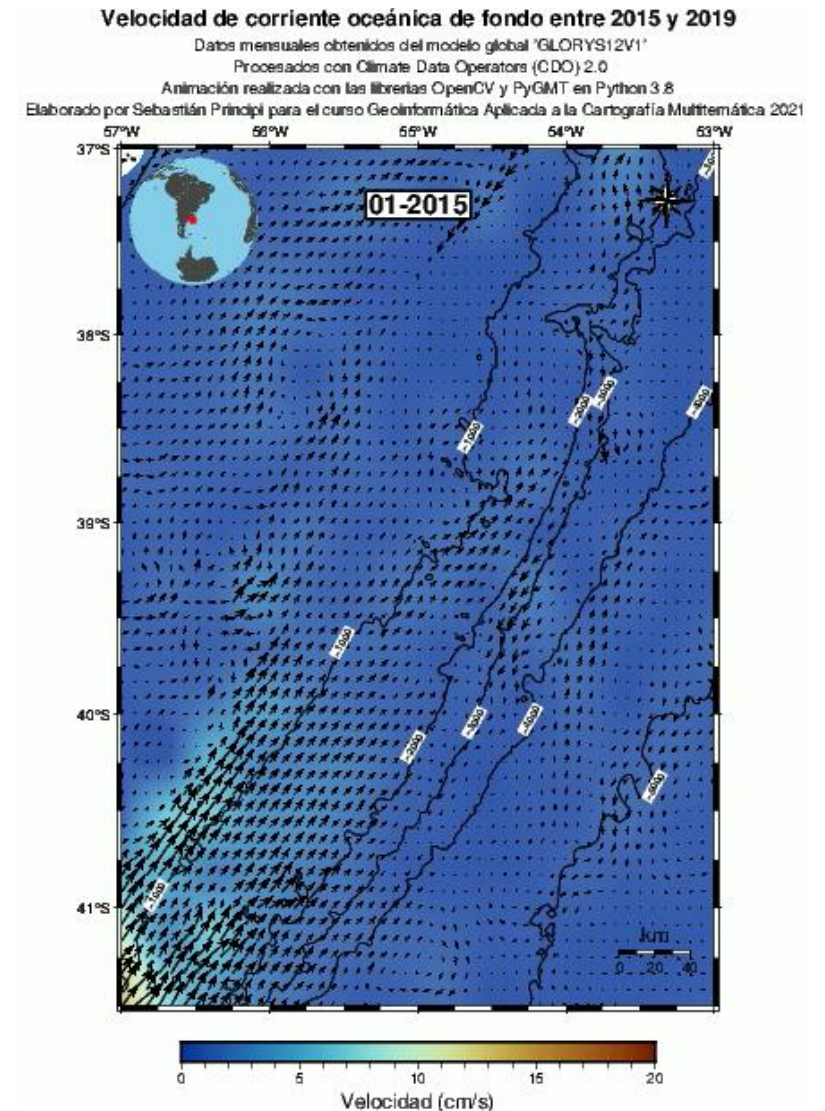
images = [img for img in os.listdir(image_folder) if img.endswith(".png")]
images = sorted(images, key=lambda x: int(os.path.splitext(x)[0]))

images=images[0:len(images)-1]
frame = cv2.imread(os.path.join(image_folder, images[0]))
height, width, layers = frame.shape

video = cv2.VideoWriter(video_name, 0, 5, (width,height))

for image in images:
    video.write(cv2.imread(os.path.join(image_folder, image)))

cv2.destroyAllWindows()
video.release()
```



Recursos y links utiles:

Anaconda Python:

<https://www.anaconda.com/products/distribution>

Instalación pygmt:

<https://www.pygmt.org/latest/install.html>

(Prestar atención al tutorial en la parte de instalación de dependencias)

Documentación oficial:

<https://www.pygmt.org/latest/api/index.html>

<https://www.pygmt.org/latest/gallery/>

Webinar con tutoriales:

<https://www.generic-mapping-tools.org/egu22pygmt/intro.html>

https://www.youtube.com/watch?v=Dgf6ijduNoE&list=PL3GHXjKa-p6VBA_MlUP7T_ByCFYQZ5uDG

Pagina con varios tutoriales:

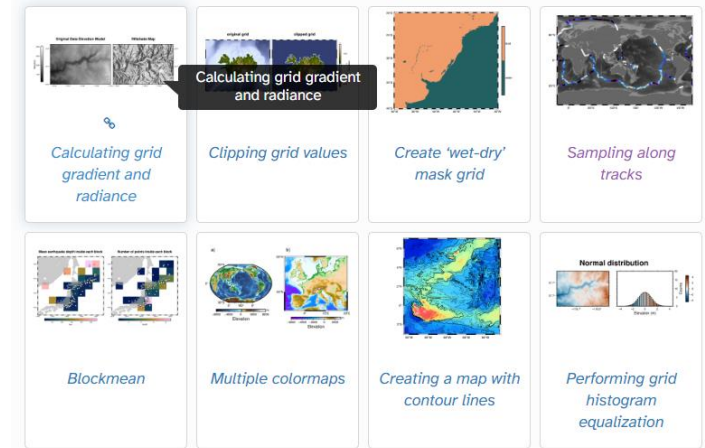
<https://www.earthinversion.com/utilities/pygmt-high-resolution-topographic-map-in-python/>

» API Reference » pygmt.Figure.grdimage [Improve this page](#)

pygmt.Figure.grdimage



Figure.grdimage(grid, *, img_out=None, frame=None, cmap=None, img_in=None, dpi=None, bit_color=None, shading=None, projection=None, monochrome=None, no_clip=None, nan_transparent=None, region=None, timestamp=None, verbose=None, xshift=None, yshift=None, interpolation=None, panel=None, coltypes=None, perspective=None, transparency=None, cores=None, **kwargs)

Examples using `pygmt.Figure.grdimage`



Crafting beautiful maps with PyGMT

Welcome to the EGU22 PyGMT short course

This Jupyter book  contains PyGMT tutorials for producing maps  and doing geospatial data processing 