

```
1  # -*- coding: utf-8 -*-
2  """
3  created on: 2024-03-15
4  @author:    Jasper Heuer
5  use:        1) reproject input-DEM to fit Landsat data grid using gdal.Warp
6              2) crop DEM to glacier extent
7  """
8
9  # import packages =====
10
11  import os
12  from osgeo import gdal
13  import numpy as np
14
15  # import data =====
16
17  base_path = "C:/Jasper/Master/Thesis/Data/"
18  os.chdir(base_path)
19
20  # warp raster with gdal =====
21
22  # define meta data:
23  dst_crs = "EPSG:32624" # define destination CRS
24  res = 30 # define resolution in meters
25
26  fn_in = "./Arctic_DEM/merged_DEM.tif"
27  fn_out = "./Arctic_DEM/DEM_reproj.tif"
28  fn_mask = "./Masks/mask.tif"
29
30  # get corner coordinates for mask:
31  mask = gdal.Open(fn_mask)
32  xmin, ymax = mask.GetGeoTransform()[0], mask.GetGeoTransform()[3] # get top left coordinates
33
34  mask = None # set to none
35
36  ds = gdal.Open(fn_in) # read dataset
37  # inspect projection by typing ds.GetProjection() in the console- NOT the editor
38
39  # reproject to common grid:
40  ds_reproj = gdal.Warp(fn_out, ds, dstSRS=dst_crs,
41                        xRes=res, yRes=-res, # specify resolution
42                        cutlineDSName="./Masks/mask_UTM-24N.shp", # cut by extend of mask
43                        cropToCutline=True, dstNodata=np.nan,
44                        resampleAlg="average")
45  # if error with mask, try mask_WGS84.shp instead - projection issues not fully understood yet
46
47  # set data to none:
48  ds = None
49  ds_reproj = None
50
51  # read reprojected data and adjust GeoTransform to exactly align with common grid:
52  ds2 = gdal.Open(fn_out)
53  print(ds2.GetGeoTransform())
54
55  ds2.SetGeoTransform([xmin, res, 0.0, ymax, 0.0, -res]) # change top left coordinates
56  print(ds2.GetGeoTransform())
57
58  # write to disk:
59  driver = gdal.GetDriverByName("GTiff")
60  moved = driver.CreateCopy("./Arctic_DEM/DEM_full_extent.tif", ds2)
61
62  # set data to none:
63  ds2 = None
64  moved = None
65
66  # crop DEM =====
67
68  dem_in = "./Arctic_DEM/DEM_full_extent.tif"
69  dem_out = "./Arctic_DEM/DEM_crop.tif"
70
71  dem_ds = gdal.Open(dem_in)
```

```
72
73 dem_crop = gdal.Warp(dem_out, dem_ds,
74                       cutlineDSName="./Masks/mittivakkat_outline.shp",
75                       cropToCutline=True, dstNodata=np.nan)
76
77 dem_ds = None
78 dem_crop = None
79
80 # clean drive =====
81
82 os.remove("./Arctic_DEM/DEM_reproj.tif")
83 os.remove("./Arctic_DEM/merged_DEM.tif")
84 os.remove("./Arctic_DEM/DEM_reproj.tif.aux.xml")
85
86 # old version with rasterio =====
87
88 """
89 dst_crs = "EPSG:32624"
90
91 with rasterio.open("./Arctic_DEM/14_44_1_1_2m_v4.1/14_44_1_1_2m_v4.1_browse.tif") as src:
92     transform, width, height = calculate_default_transform(
93         src.crs, dst_crs, src.width, src.height, *src.bounds)
94     kwargs = src.meta.copy()
95     kwargs.update({
96         "crs": dst_crs,
97         "transform": transform,
98         "width": width,
99         "height": height
100     })
101
102 with rasterio.open("./Arctic_DEM/14_44_1_1_2m_v4.1/14_44_1_1_2m_v4.1_browse_resample.tif",
103                   "w", **kwargs) as dst:
104     for i in range(1, src.count+1):
105         reproject(
106             source=rasterio.band(src, i),
107             destination=rasterio.band(dst, i),
108             src_transform=src.transform,
109             src_crs=src.crs,
110             dst_transform=transform,
111             dst_crs=dst_crs,
112             resampling=Resampling.nearest)
113 """
114
```