TMTplus Introduction to Scientific Programming

Mahdi Farnaghi, Mahdi Khodadadzadeh & Robert Ohuru

March 2021

Chapter 10

Gdal raster operations

10.1 *Gdal* exercises

Ex 10.1

```
from osgeo import gdal
import os

dataDirectory=r’PATH_TO_YOUR_FOLDER’

# change to the data directory
os.chdir(dataDirectory)

#  open dataset
raster = gdal.Open("2014.tif")

# getting driver name
dtype2 = raster.GetDriver().LongName
print("driver␣name:", dtype2)
print()

# getting raster size
x = raster.RasterXSize
y = raster.RasterYSize
print("x␣size:␣", x, "␣y␣size␣", y)
print()

# getting geotransform info
g = raster.GetGeoTransform()
if g is not None:
    print("top-left␣x:", g[0], "top␣left␣y:", g[3])
    print("pixel-size␣w-e:", g[1], "pixel-size␣n-s:", g[5])
    print("rotation␣x:", g[2], "rotation␣y:", g[4])
    print()
```

```
# getting bands
count = raster.RasterCount
print("There␣are␣" + str(count) + "␣bands")
print()
```

Observe how linear (sequential) this code is. No loops, no functions defined, just rather straightforward from A to Z code.

Ex 10.2

```
from osgeo import gdal
import os

dataDirectory=r’PATH_TO_YOUR_FOLDER’

# change to the data directory
os.chdir(dataDirectory)

#  open dataset
raster = gdal.Open("2014.tif")
band = raster.GetRasterBand(1)

min = band.GetMinimum()
max = band.GetMaximum()
print("min␣value:", min, "max␣value", max)
stats = band.GetStatistics(False, True)
print("min␣=␣%.2f␣max␣=␣%.2f␣mean␣=␣%.2f␣std␣=␣%.2f"
             % (stats[0], stats[1], stats[2], stats[3]))
print("no␣data␣value:", band.GetNoDataValue())
print("number␣of␣overviews:", band.GetOverviewCount())
print()
```

Ex 10.3

```
from osgeo import gdal
from osgeo import gdal_array as gdarr
import os

dataDirectory=r’PATH_TO_YOUR_FOLDER’

# change to the data directory
os.chdir(dataDirectory)
#  open dataset
raster = gdal.Open("2014.tif")
band = raster.GetRasterBand(1)
xoff = 200
yoff = 137
# use 1-1 to extract a single pixel,
# otherwise extract multiple pixels
win_xsize = 1
win_ysize = 1
```

```
px = gdarr.BandReadAsArray(band, xoff, yoff, win_xsize, win_ysize)
print(’pixelvalue’, px[0,0])
print()
```

Ex 10.4

```
from osgeo import gdal
from osgeo import gdal_array as gdarr
import os

dataDirectory=r’PATH_TO_YOUR_FOLDER’

# change to the data directory
os.chdir(dataDirectory)
# open dataset
raster = gdal.Open("2014.tif")
band = raster.GetRasterBand(1)
xoff = 0
yoff = 0
win_xsize = 200
win_ysize = 200
# read a single band as a two-dim array
px = gdarr.BandReadAsArray(band, xoff, yoff, win_xsize, win_ysize)

driver = gdal.GetDriverByName(’GTiff’)

outraster = driver.Create(’2014_amsterdam.tif’,
        px.shape[1], px.shape[0], 1,  gdal.GDT_Float32)
# define the new raster dataset projection and geotrasform
prj = raster.GetProjection()
outraster.SetProjection(prj)
gt = raster.GetGeoTransform()
outraster.SetGeoTransform(gt)

# create a band
outband = outraster.GetRasterBand(1)

# write the array to the band
outband.WriteArray(px)

# set a pixel nodata value
outband.SetNoDataValue(band.GetNoDataValue())
# flush the cache and clean memory
if outband is not None:
    outband.FlushCache()
    outband = None
if outraster is not None:
    outraster = None
if band is not None:
    band = None
if raster is not None:
    raster = None
print("file␣closed!")
```

Ex 10.5

```
from osgeo import gdal
from osgeo import gdal_array as gdarr
import os
dataDirectory=r’PATH_TO_YOUR_FOLDER’

#change to the data directory
os.chdir(dataDirectory)

#  open dataset
raster = gdal.Open("2014.tif")
band = raster.GetRasterBand(1)
xoff = 0
yoff = 0
win_xsize = 200
win_ysize = 200

# read a all bands as a 3D array
px = gdarr.DatasetReadAsArray(raster, xoff, yoff, win_xsize, win_ysize)
print(’shape’, px.shape)

# the order is px[Days,Y,X]
print(’line␣1’, px[0,0,0])
print(’line␣2’, px[30,100,100])
print(’line␣3’, px[58,150,78])
print(’line␣4’, px[89,180,186])
print(’line␣5’, px[119,160,20])
print(’line␣6’, px[150,100,100])
print(’line␣7’, px[364,150,78])
```

Days	Pixel position	Temperature
1	0, 0	-9999.0
31	100, 100	5.84388
59	78, 150	-9999.0
90	186, 180	19.8644
120	20, 160	-9999.0
151	100, 100	16.8981
365	78, 150	-9999.0

Note: Indices of numpy arrays start at 0; this important to obtain the correct daily information.

Ex 10.6

```
from osgeo import gdal
from osgeo import gdal_array as gdarr
import os
```

```
dataDirectory=r'PATH_TO_YOUR_FOLDER'

#change to the data directory
os.chdir(dataDirectory)

#  open dataset
raster = gdal.Open("2014.tif")
band = raster.GetRasterBand(1)
xoff = 100
yoff = 40
win_xsize = 200
win_ysize = 200

# read a single band as a two-dim array
px = gdarr.BandReadAsArray(band, xoff, yoff, win_xsize, win_ysize)
driver = gdal.GetDriverByName('GTiff')
outraster = driver.Create('2014_enschede.tif',
        px.shape[1], px.shape[0], 1,  gdal.GDT_Float32)

# define the new raster dataset projection and geotransform
prj = raster.GetProjection()
outraster.SetProjection(prj)
gt = raster.GetGeoTransform()

# getting new top left value
newTL=gdal.ApplyGeoTransform(gt, 100, 150)
outraster.SetGeoTransform([newTL[0], 1000, 0, newTL[1], 0, -1000])

# create a band
outband = outraster.GetRasterBand(1)

# write the array to the band
outband.WriteArray(px)

# set a pixel nodata value
outband.SetNoDataValue(band.GetNoDataValue())

# flush the cache and clean memory
if outband is not None:
    outband.FlushCache()
    outband = None
if outraster is not None:
    outraster = None
if band is not None:
    band = None
if raster is not None:
    raster = None
print("file␣closed!")
```

11.2 The use of .Translate() and .Warp() methods

Ex 10.7

```
from osgeo import gdal
import os
directory=r'PATH_TO_YOUR_FOLDER'
os.chdir(directory)

for filename in os.listdir(directory):
        if filename.endswith(".csv"):
            filenameSimple=filename.split('.')[0]
            newDataset = gdal.Translate(filenameSimple+'.tif',
                    filename, format="GTiff",noData='-9999',
                    outputSRS='EPSG:28992')
            continue
        else:
            continue
print('Finished')
```

Ex 10.8

```
from osgeo import gdal
from osgeo import gdal_array as gdarr
import matplotlib.pyplot as plt

raster = gdal.Open(r'PATH_TO_YOUR_CSV_FOLDER/2014.tif')

    # Exercise 4
AmsDataset = gdal.Translate("2014_amsterdam2.tif",raster,
        format="GTiff",srcWin=[0,0,200,200])
    # Show the raster image to confirm
newBand=AmsDataset.GetRasterBand(1)
px=gdarr.BandReadAsArray(newBand, 0, 0, AmsDataset.RasterXSize,
        AmsDataset.RasterYSize)
px[px == -9999] = None
plt.imshow(px)
plt.show()

    #Exercise 6
EnsDataset = gdal.Translate("2014_enschede2.tif",raster,
        format="GTiff",srcWin=[100,40,200,200])
    # Show the raster image to confirm
newBand=EnsDataset.GetRasterBand(1)
px=gdarr.BandReadAsArray(newBand, 0, 0,
        EnsDataset.RasterXSize, EnsDataset.RasterYSize)
px[px == -9999] = None

plt.imshow(px)
plt.show()
print('Finished')
```

Ex 10.9

```
from osgeo import gdal
import os

directory = r’PATH_TO_YOUR_CSV_FOLDER/2014_csv’

os.chdir(directory)
for filename in os.listdir(directory):
    if filename.endswith(".tif"):
        raster = gdal.Open(filename)
        newDataset = gdal.Warp("", raster, format="Mem",
            dstSRS = ’EPSG:4326’)
    else:
        continue
print(’Finished’)
```