

# raster\_analysis

June 17, 2024

## 1 Working with raster data in Python

### 1.1 Libraries and Settings

```
[ ]: # Libraries
import os
import pprint
import requests
import numpy as np
import pandas as pd

import fiona
import geopandas as gpd

import rasterio
from rasterio import plot
import rasterio.mask

from shapely.geometry import Polygon

import matplotlib.patches as patches
from matplotlib import pyplot as plt

# Ignore warnings
import warnings
warnings.filterwarnings("ignore")

print(os.getcwd())
```

c:\Users\dimit\Documents\applied\_data\_science\week\_04\spatial\_data\_analysis\06\_Python\_Raster\_Data

### 1.2 Download example raster (.geotiff) from SwissTopo

Source: <https://www.swisstopo.admin.ch/de/orthobilder-swissimage-10-cm>

```
[ ]: # Get example raster from swisstopo
url = "https://data.geo.admin.ch/ch.swisstopo.swissimage-dop10/
↳swissimage-dop10_2022_2693-1246/swissimage-dop10_2022_2693-1246_0.1_2056.tif"
```

```
response = requests.get(url)

with open('swissimage.tif', 'wb') as f:
    f.write(response.content)
```

```
[ ]: # Import raster
src = rasterio.open("swissimage.tif")

# Show raster
fig, ax = plt.subplots(figsize=(7,7))
rasterio.plot.show(src, ax=ax)

# Hide axes
ax.axis('off')

plt.show()
```



### 1.3 Show histogram of RGB colors

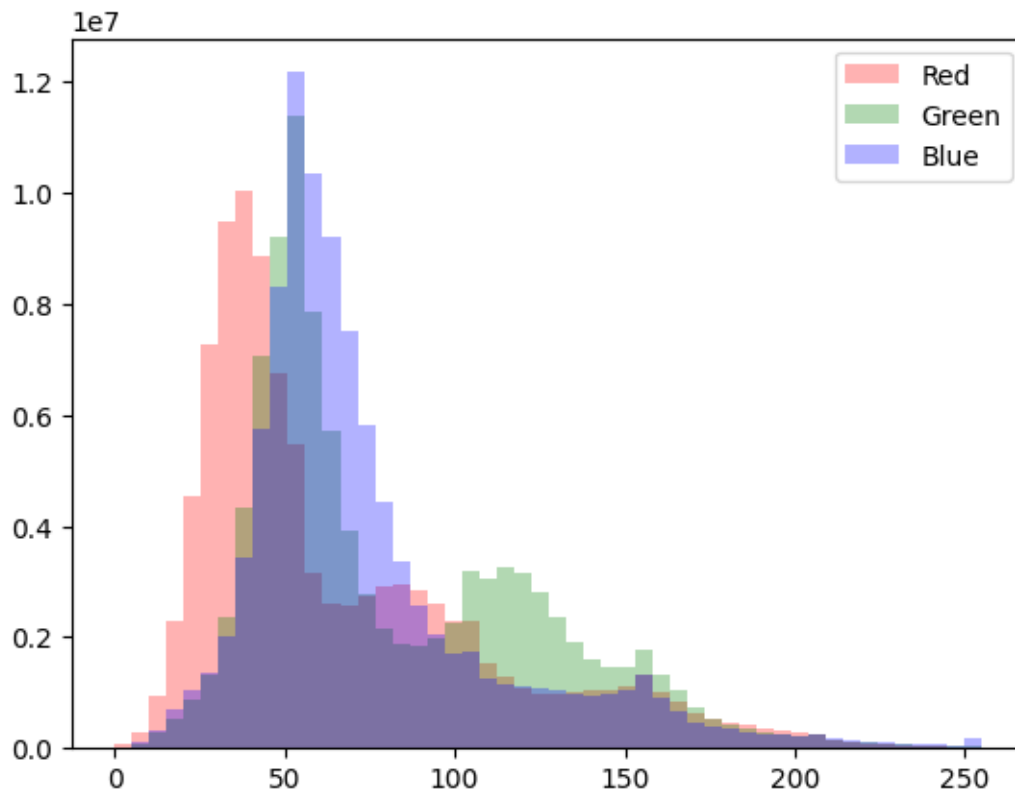
```
[ ]: # Open raster
with rasterio.open('swissimage.tif') as src:

    red_band = src.read(1)
    green_band = src.read(2)
    blue_band = src.read(3)

# Flatten the arrays
red = red_band.flatten()
green = green_band.flatten()
blue = blue_band.flatten()

# Plot a histogram
plt.hist(red, bins=50, color='red', alpha=0.3, label='Red')
plt.hist(green, bins=50, color='green', alpha=0.3, label='Green')
plt.hist(blue, bins=50, color='blue', alpha=0.3, label='Blue')

plt.legend(loc='upper right')
plt.show()
```



## 1.4 Masking a raster using a shapefile

### 1.4.1 Create a shapefile

```
[ ]: # Read raster
src = rasterio.open("swissimage.tif")

# Bottom left corner of square (LV95 coordinates)
x1, y1 = 2693300, 1246567

# Side lenght of square in meters
size = 250

# Create the other three corners
coordinates = [(x1, y1), (x1, y1 + size), (x1 + size, y1 + size), (x1 + size,
↪y1)]

# Create a shapely polygon from the coordinates
polygon = Polygon(coordinates)

# Create a GeoDataFrame
gdf = gpd.GeoDataFrame(index=[0], geometry=[polygon])

# Save the GeoDataFrame as a shapefile
gdf.to_file("municipality_part.shp")

# Plot the shapefile over the raster
fig, ax = plt.subplots(figsize=(7,7))
rasterio.plot.show(src, ax=ax)
patch = patches.Polygon(coordinates, fill=False, edgecolor='red', linewidth=2)
ax.add_patch(patch)
ax.axis('off')

plt.show()
```



#### 1.4.2 Masking the raster

```
[ ]: # Read shapefile
with fiona.open("municipality_part.shp", "r") as shapefile:
    shapes = [feature["geometry"] for feature in shapefile]

# Mask the raster with the shapefile
with rasterio.open("swissimage.tif") as src:
    out_image, out_transform = rasterio.mask.mask(src, shapes, crop=True)
    out_meta = src.meta

# Plot the masked raster
fig, ax = plt.subplots(figsize=(7,7))
rasterio.plot.show(out_image, ax=ax)
```



```
# Hide the axes
ax.axis('off')

plt.show()
```



## 1.5 Save masked raster as .geotiff

```
[ ]: # Update metadata
out_meta.update({
    "driver": "GTiff",
    "height": out_image.shape[1],
    "width": out_image.shape[2],
    "transform": out_transform
```

```

})

# Write the masked image to a new GeoTIFF file
with rasterio.open("masked_swissimage.tif", "w", **out_meta) as dest:
    dest.write(out_image)

# Print metadata
print('Meta data:')
pprint.pprint(out_meta)

```

```

Meta data:
{'count': 3,
 'crs': CRS.from_epsg(2056),
 'driver': 'GTiff',
 'dtype': 'uint8',
 'height': 2500,
 'nodata': None,
 'transform': Affine(0.1, 0.0, 2693300.0,
                    0.0, -0.1, 1246817.0),
 'width': 2500}

```