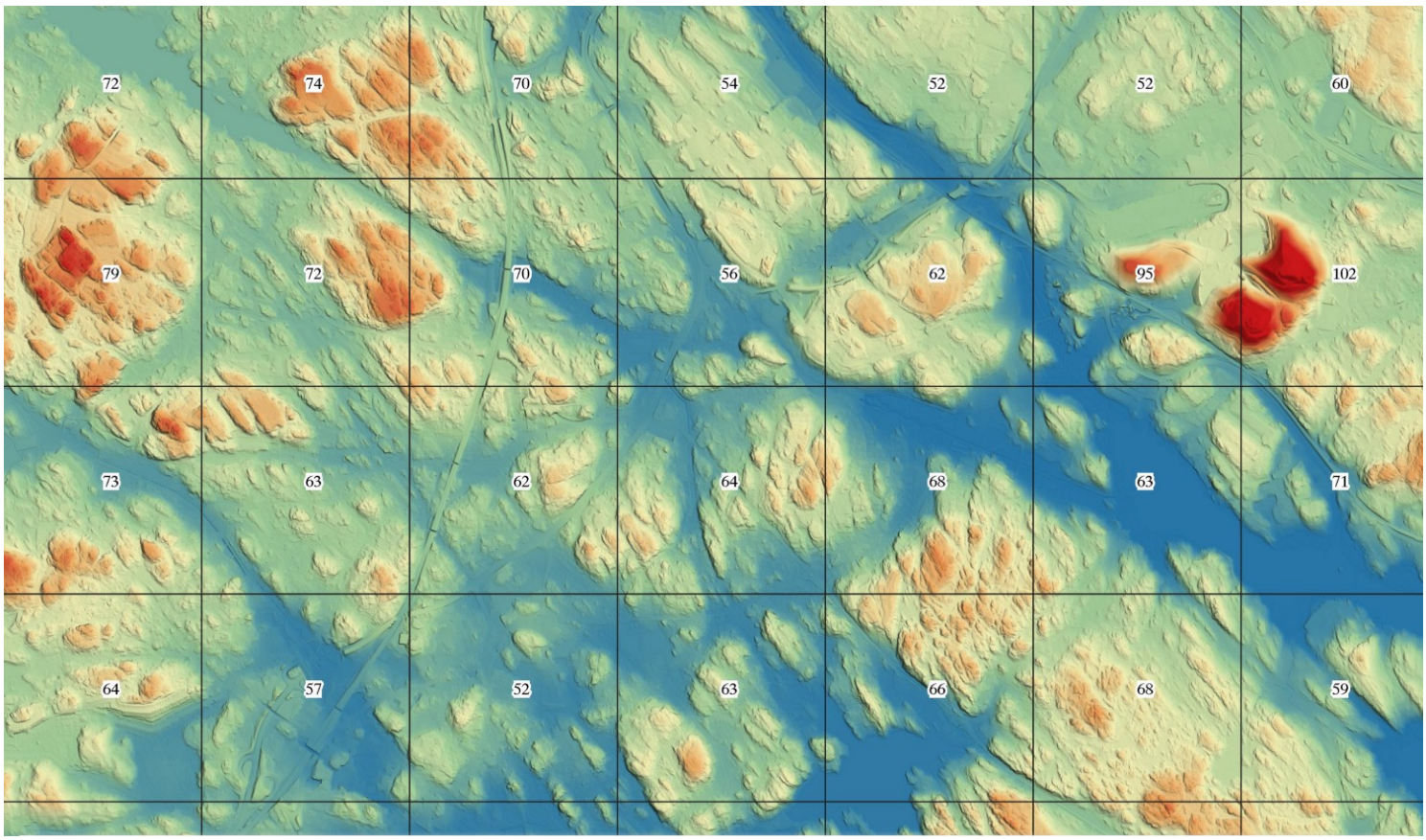
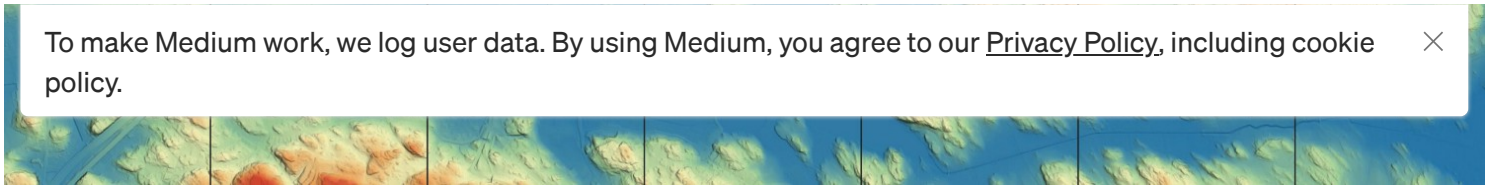
How To Compute Satellite Image Statistics And use It In Pandas

Derive zonal statistics from Sentinel 2 images and merge with your Pandas data frame. A walkthrough on calculating NDWI water index for flooded areas.

Abdishakur Nov 13, 2019 · 5 min read ★





Example of Zonal Statistics — Elevation DEM overlayed on derived statistics (Max elevation in each grid shown)

Satellite data is dense and uses cells to store values. In many cases, however, you want only a summary of the satellite (raster) image converted into a tabular format — CSV or Pandas data frame.

Let us say, for example; you have a Digital Elevation Model (DEM). The DEM image gives a clear representation of the elevation and topography of the area. Now, what if you want to get elevation values and integrate tabular data you have, for example, buildings, to get the elevation of each building.

This process of deriving table outputs (Summary statistics) from raster images is called **Zonal Statistics.**

In this tutorial, We learn how to extract values from raster data and store these values in Tabular format (Pandas Dataframe).

The dataset and the code for this tutorial are available in Github. Let us start by exploring the data.

Data Exploration

The dataset for this tutorial is sentinel images taken on 1 November 2019 in Beledweyne, Somalia. The area is flooded during this period, and we calculate the NDWI to measure the level of water stress. We use Google Colab Notebook, and we download the data in the notebook directly from URL.

Let us import the libraries we use for this tutorial.

```
import pandas as pd
import numpy as np
import geopandas as gpd
import rasterio as rio
from rasterio.plot import show
from rasterio.mask import mask
import matplotlib.pyplot as plt
```

With Rasterio you can read the different bands of Sentinel 2 images. In this case, we read bands 8 (NIR), 4 (Red), 3 (Green), and 2 (Blue).

```
b8 = rio.open("/content/Data/20191101/B08-20191101.tif")
b4 = rio.open("/content/Data/20191101/B04-20191101.tif")
b3 = rio.open("/content/Data/20191101/B03-20191101.tif")
b2 = rio.open("/content/Data/20191101/B02-20191101.tif")
```

Let us look at the width and height of the images. I am only using b4, but you can check if all bands have the same weight and length.

```
b4.width,
b4.height
```

We plot the data to see and explore the satellite images we have. Here I am only plotting Band 3.

```
fig, ax = plt.subplots(1, figsize=(12, 10))
show(b3, ax=ax)
plt.show()
```

If you would like to see how to make RGB images with Rasterio, I have a tutorial you might want to see here.

Satellite imagery access and analysis in Python & Jupyter notebooks

Access, preprocess, analyse and visualize satellite images in Jupyter notebooks with Python.

towardsdatascience.com

The Sentinel 2 image of the area(only Band 3) is shown below.

Let us also read the buildings table which we will use to store the statistical summaries derived from the satellite image. Please know that you can use other polygons, like districts, rectangular grids instead of the building polygons for this example.

We use Geopandas to read the data.

```
buildings = buildings[["osm_id","building", "geometry"]]
buildings.head()
```

And here are the first five rows of the table. We have the Geometry column of each building, osm_id and building columns.

We can also plot the buildings on top of the Sentinel 2 images.

```
fig, ax = plt.subplots(figsize=(12, 10))
show(b4, ax=ax)
buildings.plot(ax=ax, color="white", alpha=.50)
plt.show();
```

The buildings are marked as white and overlayed in the image, as shown below. The picture shows the extent of the city (residential areas) with meandering Shabelle River.

Let us now calculate the NDWI values from Sentinel 2 images.

Calculate Normalized Difference Water Index (NDWI)

To calculate the NDWI values, we use this formula:

$$(Band3 — Band8)/(Band3 + Band8)$$

So, let us calculate using this formula in Rasterio.

```
green = b3.read()
nir = b8.read()
ndwi = (nir.astype(float)-green.astype(float))/(nir+green)
```

The NDWI arrays can be plotted with Rasterio, as shown below.

```
fig, ax = plt.subplots(1, figsize=(12, 10))
show(ndwi, ax=ax, cmap="coolwarm_r")
plt.show()
```

We can save the NDWI arrays as a raster image so that we can use it later.

```
meta = b4.meta
meta.update(driver='GTiff')
meta.update(dtype=rio.float32)
with rio.open('NDWI.tif', 'w', **meta) as dst:
     dst.write(ndvi.astype(rio.float32))

# Read the saved
ndwi_raster = rio.open("NDWI.tif")
```

to get the cell values from the NDWI raster image.

The following is a small function that masks the cell values to our data frame table.

```
def derive_stats(geom, data=ndwi_raster, **mask_kw):
    masked, mask_transform = mask(dataset=data,
          shapes=geom,)crop=True, all_touched=True, filled=True)
    return masked
```

We can derive now the values we want like this. Let us say we are interested in getting the mean values of NDWI for each building. We create a column for that “mean_ndwi” and pass our function to apply the building’s geometry and also apply to mean from using Numpy.

```
buildings[‘mean_ndwi’] =
buildings.geometry.apply(derive_stats).apply(np.mean)
```

Or, get the maximum NDWI values for each building.

```
buildings[‘max_ndwi’] =
buildings.geometry.apply(derive_stats).apply(np.max)
```

Our table has two new columns now, mean_ndwi and max_ndwi, where we store the mean and max NDWI values for each building.

area. In the “Data” folder, you have “20191002” folder. Try to calculate the mean and max NDWI values using these images. Compare this with statistics derived from images we have calculated for the date 20191101 (flooding period).

Conclusion

In this tutorial, we have seen how to calculate NDWI values from Sentinel 2 images and derive summary statistics from them. The code and Google Colab Notebooks are available in this Github repository.

shakasom/zonalstatistics

Python — Rasterio and Geopandas to calculate Zonal Statistics Satellite data is dense and uses cells to store values...

github.com

Or directly in this Google Collaboratory Notebook

shakasom/zonalstatistics

You can’t perform that action at this time. You signed in with another tab or window. You signed out in another tab or...

github.com