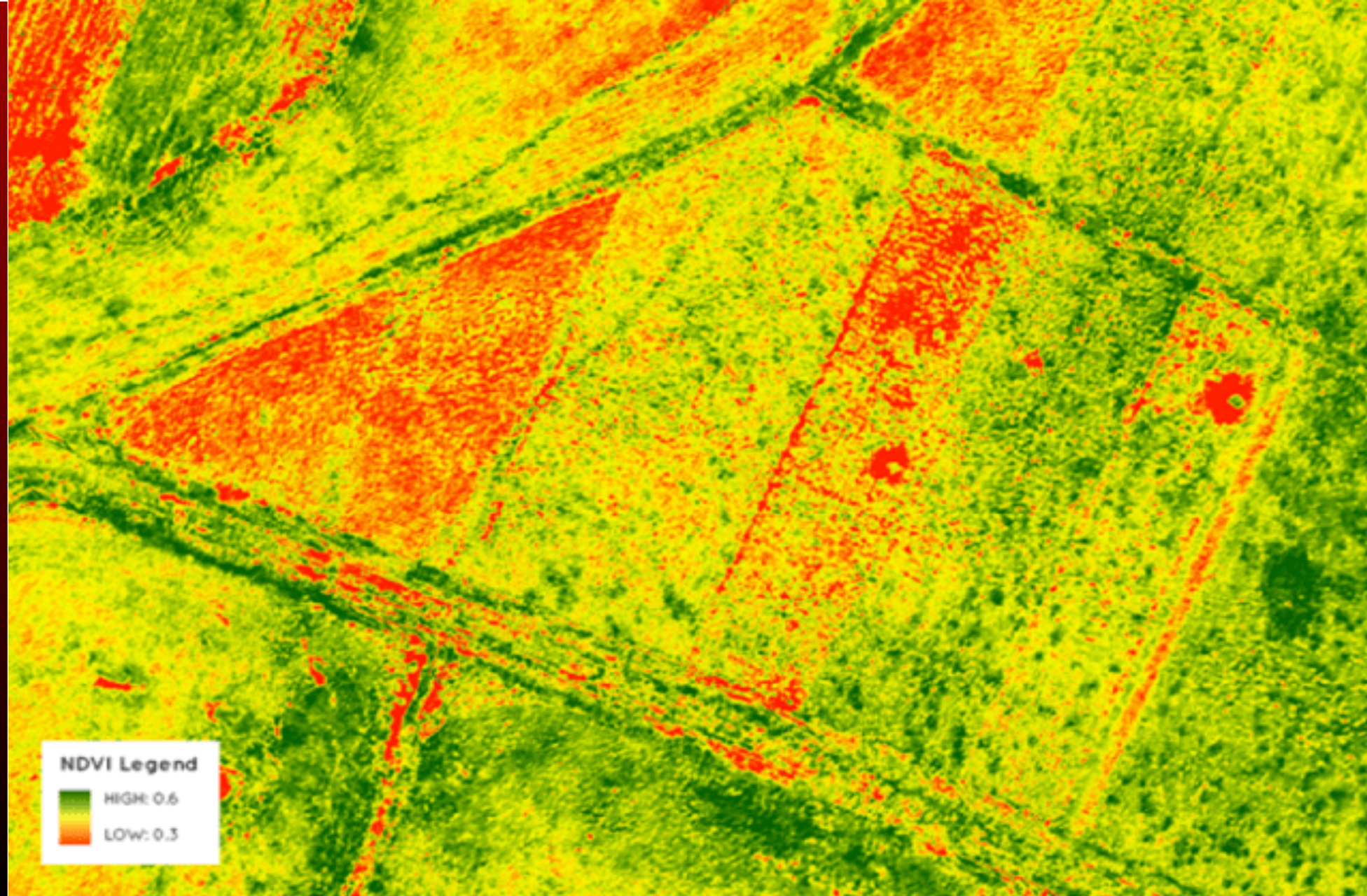


## PyGDA #32

Obrabianie zdjęć satelitarnych po  
Pythonowemu, czyli wprowadzenie do  
biblioteki rasterio

# Agenda:

- 1) -> Zdjęcia satelitarne (EO – Earth Observation), po co je obrabiać?
- 2) -> Sposoby przetwarzania zdjęć EO (przy wykorzystaniu Pythona)
- 3) -> Biblioteka rasterio
- 4) -> Przykład 1: wczytanie i wyświetlenie zdjęcia
- 5) -> Przykład 2: operowanie na warstwach
- 6) -> Przykład 3: automatyzacja obliczeń wskaźnika NDVI



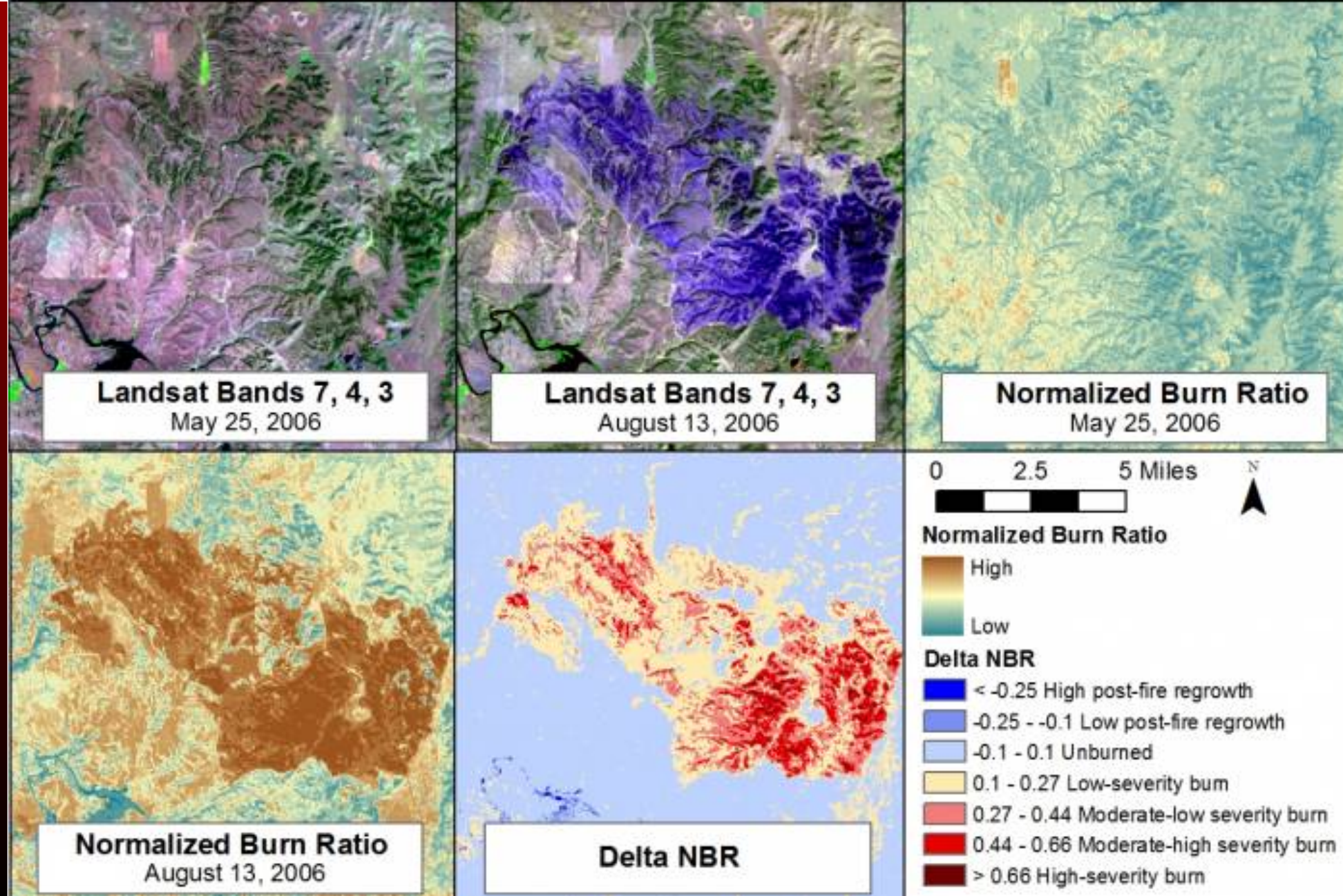
<http://geoawesomeness.com/understanding-aerial-data-normalized-difference-vegetation-index-ndvi/>  
(GEO Awesomeness)





<https://earthobservatory.nasa.gov/NaturalHazards/view.php?id=83697> (NASA)

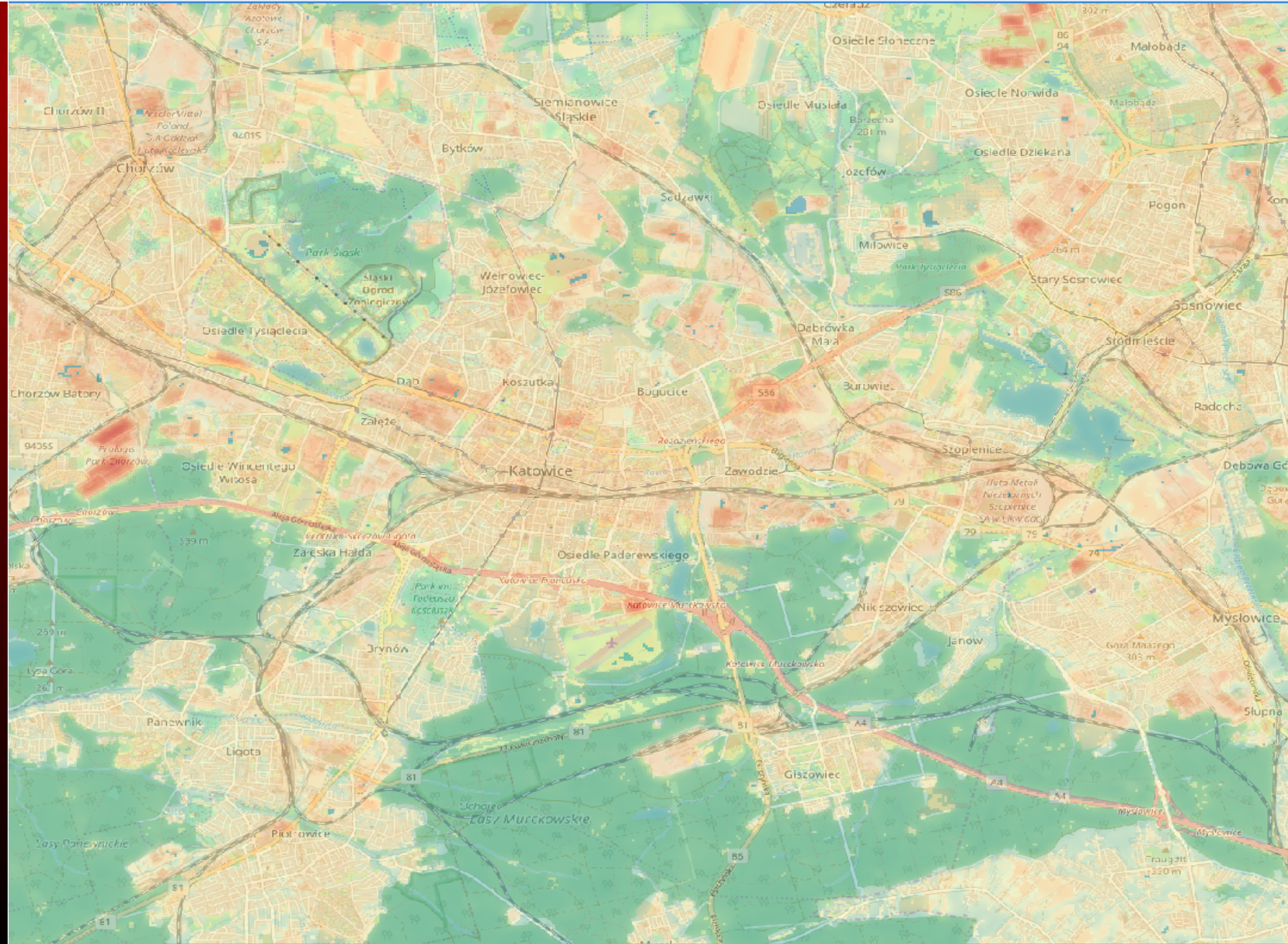




[http://wiki.landscapetoolbox.org/doku.php/remote\\_sensing\\_methods:normalized\\_burn\\_ratio](http://wiki.landscapetoolbox.org/doku.php/remote_sensing_methods:normalized_burn_ratio)

(The Landscape Toolbox)





(Data Lions)

# Jak przetwarzać zdjęcia satelitarne?

- ✓ Aktualnie najbardziej „Pythonowy” sposób to wykorzystanie biblioteki rasterio -> <https://github.com/mapbox/rasterio>
- ✓ Mniej Pythonowy sposób, ale wciąż oparty na Open Source, to wykorzystanie bibliotek **GDAL/OGR**. Są to potężne narzędzia, wyposażone w wiele algorytmów do przetwarzania danych satelitarnych i danych wektorowych. Minusy: trzeba pamiętać o alokacji oraz dealokacji pamięci, biblioteki raczej pod C/C++ niż Pythona.
- ✓ Wciąż mniej Pythonowy sposób to wykorzystanie zewnętrznych źródeł takich jak **GRASS GIS, SAGA GIS, QGIS**. Być może bardziej wskazany na etapie prototypowania skomplikowanych algorytmów! Trudno zintegrować z Anacondą, narzucony Python 2.7.
- ✓ A już w ogóle niepythonowe jest wykorzystywanie takich kombajnów jak **ArcGIS** ;) (Gdzie pisać można pythonowe skrypty).

# Kilka słów o różnicach między GDAL a rasterio

Biblioteka powstała ze względu na problemy z biblioteką GDAL, która była daleka od ZEN znanego z Pythona. Najważniejszy problem tyczył się alokacji pamięci i zarządzania pamięcią – co w przypadku Pythona powinno odbywać się automatycznie.

```
-----  
import gdal  
Dataset = gdal.Open('test.tif')
```

```
(...)
```

```
dataset = None
```

```
-----  
import rasterio  
with rasterio.open('test.tif') as src:  
    img = src.read()  
(...)
```



**Czas na kodowanie!**