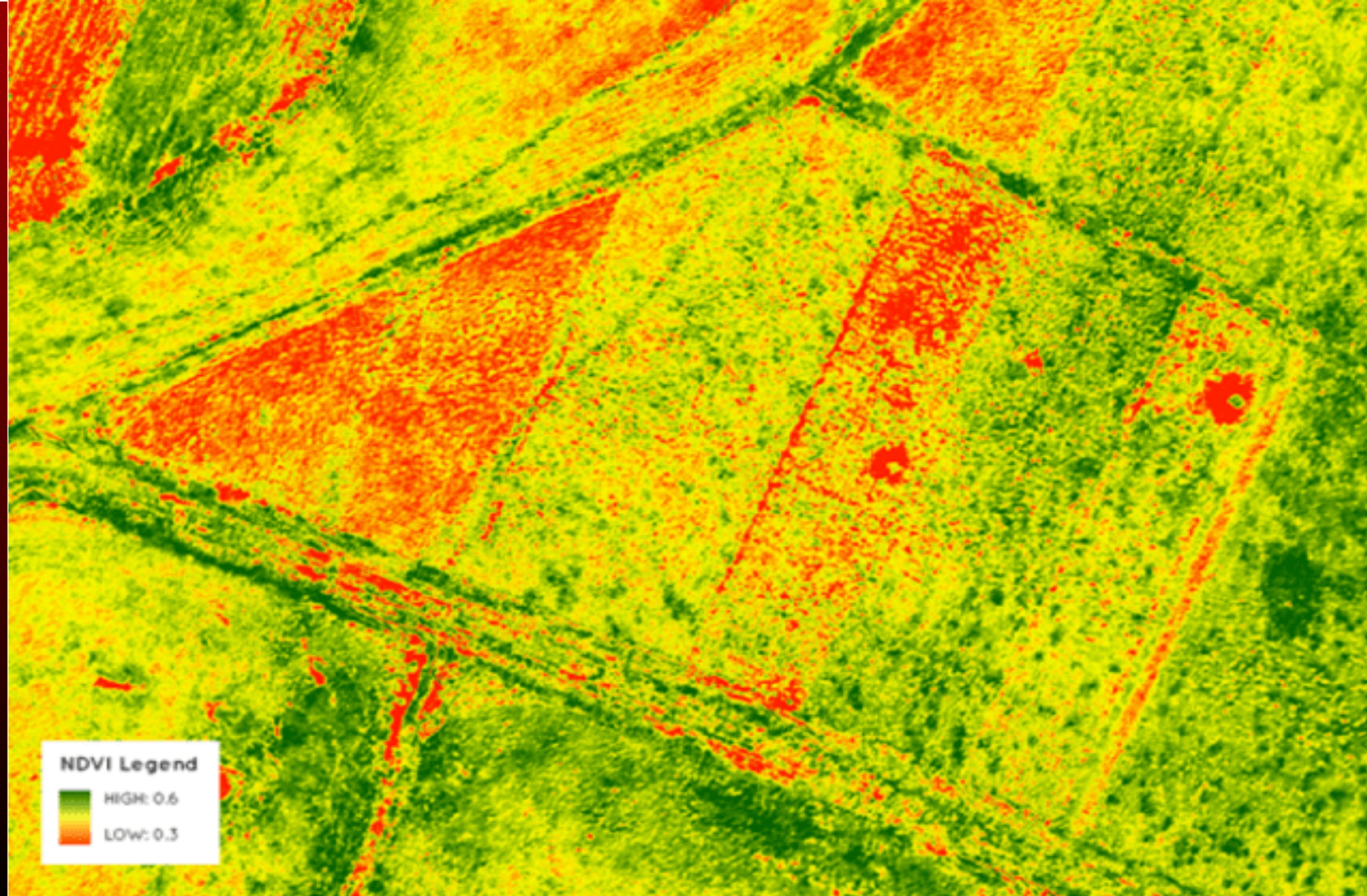


Kariera IT, 10.03.2018, Gdańsk

Automatyzacja przetwarzania danych
obserwacji Ziemi w Pythonie – studium
przypadku

Agenda:

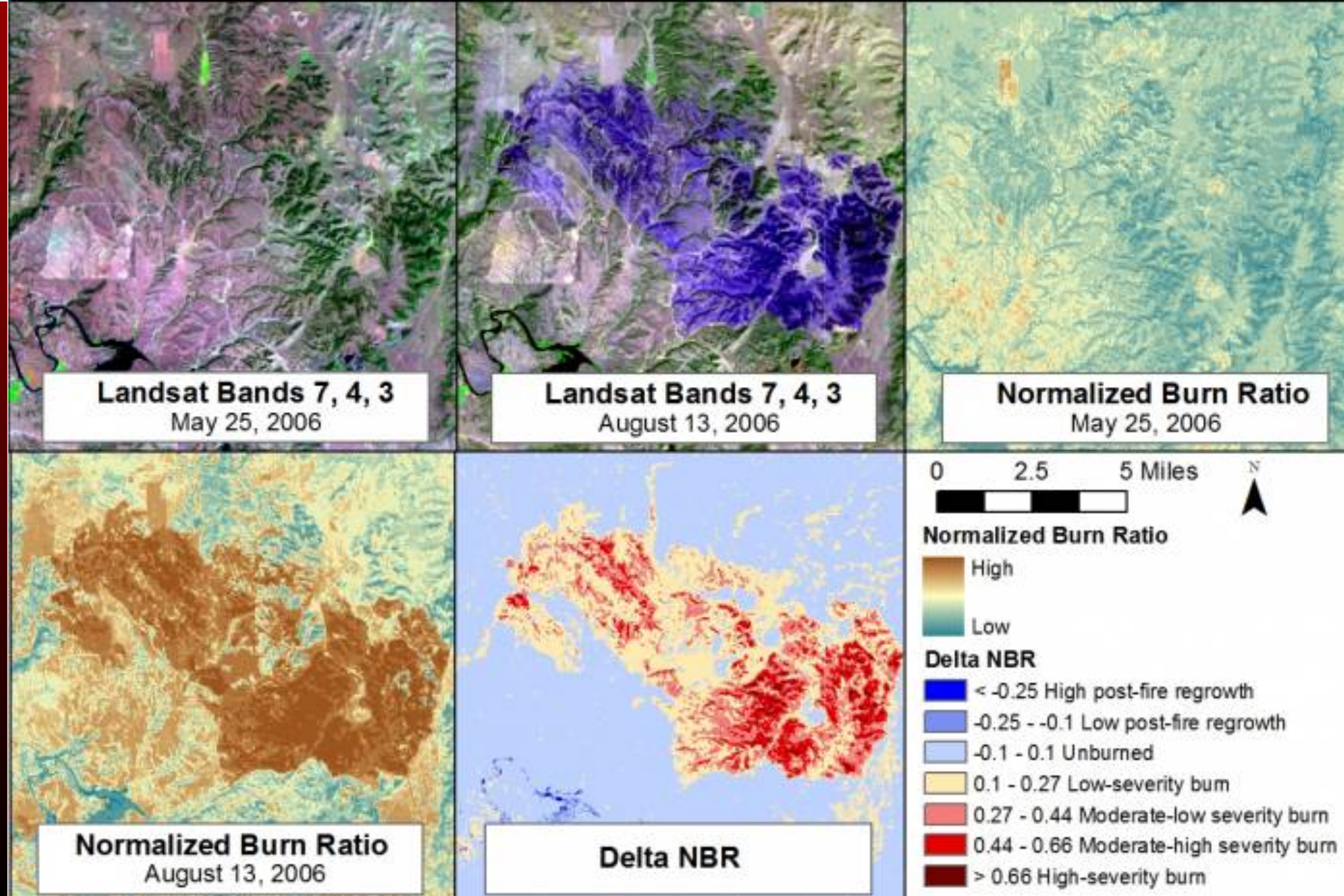
- 1) -> Zdjęcia satelitarne (EO – Earth Observation), po co je obrabiać?
- 2) -> Sposoby przetwarzania zdjęć EO (przy wykorzystaniu Pythona)
- 3) -> Przykład 1: Sceny z satelity Sentinel 2 jako źródło informacji o stanie i czystości wody
- 4) -> Przykład 2: kompletny algorytm do przetwarzania i analizy serii czasowych z satelity Terra (instrument MODIS) - zmiany temperatury powierzchni na przestrzeni lat w polskich miastach



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

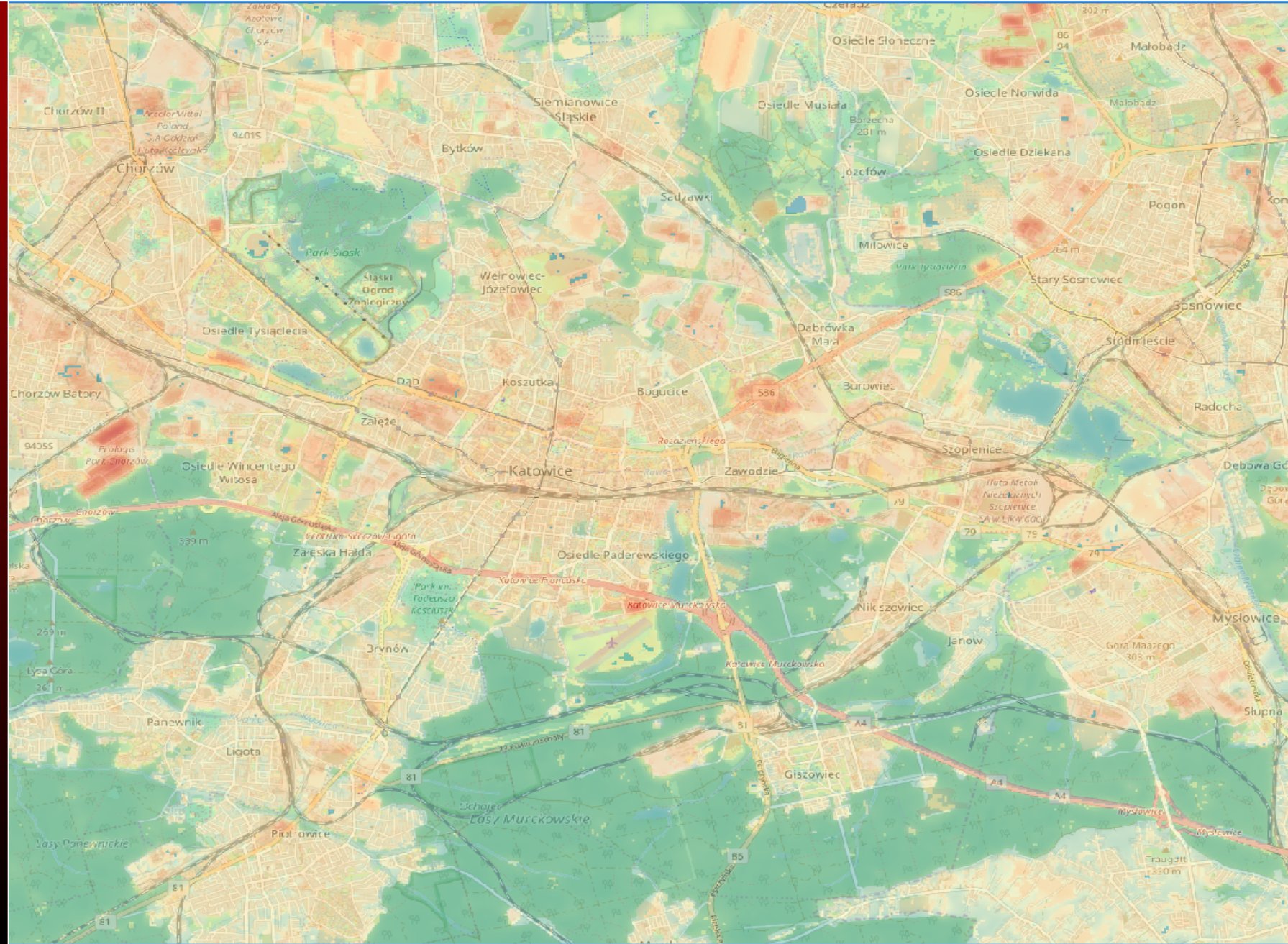


<https://earthobservatory.nasa.gov/NaturalHazards/view.php?id=83697> (NASA)



http://wiki.landscapetoolbox.org/doku.php/remote_sensing_methods:normalized_burn_ratio

(The Landscape Toolbox)



Miejskie wyspy ciepła w okolicy Katowic (Data Lions)

Jak przetwarzać zdjęcia satelitarne?

- ✓ Aktualnie najbardziej „Pythonowy” sposób to wykorzystanie biblioteki rasterio do plików rastrowych -> <https://github.com/mapbox/rasterio> i biblioteki fiona do plików wektorowych -> <https://github.com/Toblerity/Fiona>
- ✓ 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. Za to najlepszy wybór przy pisaniu skryptów “konsolowych”.
- ✓ 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, ale nowa wersja QGIS obsługuje Pythona 3.0 (wcześniejsza tylko Pythona 2.7).

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 dotyczy 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!