

GeoPandas (geoprocessing): Clip

Dr. Cornelis Stal

April 28, 2022

1 Vectordata uitknippen (Clip) met GeoPandas

In deze sectie zal toegelicht worden hoe vector data geknipt kunnen worden in functie van de grenzen van een gegeven polygoon-geometrie. Opnieuw zal hier **GeoPandas** gebruikt worden.

In het onderstaande voorbeeld wordt geïllustreerd hoe een verzameling vector-geometriën geknipt kunnen worden aan de hand van de ruimtelijke begrenzing van een ander vectorobject. Beide datasets worden eerst geopend met **Geopandas** als twee afzonderlijke **GeoDataFrames**. Om de **clip**-functie uit te kunnen voeren dienen beide datasets in hetzelfde coördinaatreferentiesysteem (CRS) te staan.

Voor dit voorbeeld zullen we de polygonen uit het Vlaamse gewestplan (GWP, grondvlakken) uitknippen aan de hand van de gemeentegrenzen van de stad Gent. Het resultaat zal een nieuwe **GeoDataFrame** zijn.

Opmerking: de te knippen dataset zal uitgeknipt worden aan de hand van de totale begrenzing van het knip-object. Indien er meerdere polygonen aanwezig zijn in het knipobject zal de operatie daadwerkelijk uitgevoerd worden op alle geometriën in het knipobject.

1.1 Bibliotheken importeren

Uiteraard starten we met het importeren van de benodigde bibliotheken.

```
[ ]: import matplotlib.pyplot as plt
import geopandas as gpd

# Instelling om figuren wat groter af te beelden
plt.rcParams['figure.figsize'] = [7, 10]
```

1.2 Data importeren

Vervolgens worden de data geïmporteerd, bestaande uit de gemeentegrenzen van de stad Gent en de grondvlakken van het Vlaamse gewestplan. In het eerste geval wordt een shapefile geïmporteerd, waarna een subset aangemaakt wordt voor het object (of objecten) waarvoor het veld **NAAM** overeenkomt met de **string** **Gent**.

De data van het gewestplan worden geïmporteerd vanuit een WFS. Hiervoor wordt de basis-url opgegeven als **string**, alsook een **dictionary** met een aantal parameters die meegegeven zullen

worden met de `request`. De data worden tot slot opgevraagd met de `get`-methode uit de `requests`-bibliotheek. Volledigheidshalve definiëren we ook het CRS nog eens voor de resulterende data.

Opmerking: voor het bevragen van de data via de WFS wordt in de `request` al een begrenzing gedefiniëerd (BBOX, met respectievelijk `xmin`, `ymin`, `xmax` en `ymax`). Deze waarden worden uit het `bb`-object gehaald. We hadden ook zelf een polygoon kunnen aanmaken met de volgende code:

```
from shapely.geometry import Polygon
polygon = Polygon([(100000, 190000), (100000, 195000), (105000, 195000), (105000, 190000), (100000, 190000)])
poly_gdf = geopandas.GeoDataFrame([1], geometry=[polygon], crs="epsg:31370")
```

```
[ ]: import requests

# Gent uit de dataset met gemeentegrenzen
refgem = gpd.read_file("data/refgem_2018.shp", crs="epsg:31370")
gent = refgem[refgem["NAAM"] == "Gent"]

url = "https://www.mercator.vlaanderen.be/raadpleegdienstenmercatorpubliek/wfs"
bb = gent.total_bounds
params = {
    "service": "WFS",
    "version": "2.0.0",
    "request": "GetFeature",
    "typeName": "lu:lu_gwp_gv",
    "bbox": "%s,%s,%s,%s" % (bb[0], bb[1], bb[2], bb[3]),
    "srsName": "EPSG:31370",
    "outputFormat": "application/json",
}

r = requests.get(url, params=params)
gwp = gpd.read_file(r.text, crs=gent.crs)
gwp = gwp.set_crs(gent.crs, allow_override=True)
gwp.plot()
```

1.3 Originele data visualiseren

```
[ ]: fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 20))
refgem.plot(ax=ax1)
gent.boundary.plot(ax=ax1, color="red")
gent.boundary.plot(ax=ax2, color="green")
gwp.plot(ax=ax2, column='legende', cmap='tab20b')
ax1.set_title("Gent in Vlaanderen", fontsize=20)
ax2.set_title("Alle ongeknipte zones voor de b", fontsize=20)
ax1.set_axis_off()
ax2.set_axis_off()
plt.show()
```

1.4 Data knippen

In eerdere versies van **GeoPandas** werd de `clip`-methode uitgevoerd op de **GeoDataFrame** waar de te knippen objecten zich in bevinden. Als attribuut werd aan deze methode dan de knip-geometrie meegegeven. Bij de huidige versie van **GeoPandas** (ten minste versie 0.8.1) zal de `clip`-methode als bibliotheek-methode aangeroepen worden. We dienen dan 2 argumenten mee te geven, namelijk de te knippen objecten en de knipper. De geretourneerde output is een verzameling objecten in een nieuwe **GeoDataFrame**. Alle attributen uit de originele dataset zijn toegekend aan deze objecten.

Opmerking: de `clip`-methode zal slechts werken wanneer beide datasets in een en hetzelfde CRS staan. Mocht dit niet het geval zijn, moet de `GeoDataFrame.to_crs`-methode gebruikt worden uit de **GeoPandas** bibliotheek. Enkele andere methoden kunnen bekeken worden in een eerder behandelde Jupyter Notebook over [Overlays](#).

Hulp en uitleg over de `clip`-methode kunnen eenvoudig opgevraagd worden:

```
[ ]: help(gpd.clip)
```

We bekijken ook de inhoud van de dataset met grondvlakken van het Vlaamse gewestplan nog een keer.

```
[ ]: print(gwp)
```

Vervolgens voeren we de `clip`-operatie uit en en visualiseren we het resultaat.

```
[ ]: gwp_clipped = gpd.clip(gwp, gent)
fig, ax = plt.subplots(figsize=(12, 8))
gwp_clipped.plot(ax=ax, column='legende', cmap='tab20b')
gent.boundary.plot(ax=ax, color="red")
ax.set_title("Gewestplan Gent", fontsize=20)
ax.set_axis_off()
plt.show()
```

Opdracht: [OpenStreetMap \(OSM\)](#) biedt een zeer uitgebreide dataset met ruimtelijke objecten van overal op aarde. De dataset wordt continu aangevuld door een zeer actieve ‘community’ en is geheel ‘open-source’. Verschillende desktop GIS-programma’s, zoals [QGIS](#) laten het downloaden van OSM-data toe. Via [GeoFabrik](#) is het ook mogelijk om specifieke continenten, landen of regio’s te downloaden als ESRI Shapefile. De complete dataset van België heeft tijdens het voorbereiden van deze ‘notebook’ een grootte van 840 Mb en bevat een groot aantal lagen en objecten.

Vermits de gehele dataset van België niet altijd erg performant is om te gebruiken bij studies met betrekking tot kleinere studiegebieden, valt het aan te bevelen om de dataset uit te knippen aan de hand van een ruimtelijke begrenzing. De opdracht is daarom om de dataset van België te downloaden als ESRI Shapefile, en vervolgens alle data uit te knippen voor de stad Gent.

```
[ ]: ## UW CODE HIER ##
```