

Geopandas: Ruimtelijke selecties

Dr. Cornelis Stal

April 28, 2022

```
[ ]: import geopandas as gpd
```

1 Ruimtelijke selecties

In de vorige tutorial hebben we gezien hoe we data kunnen selecteren

Zoals bij attribuutselecties laten ruimtelijke selecties toe om subsets aan te maken door elementen uit een dataset te halen op basis van een bepaalde expressie. In plaats van waarden in een tabel te vergelijken in een bepaalde voorwaarde, zal in deze tutorial gewerkt worden met ruimtelijke predicaten. In QGIS en ERSS ArcGIS vinden we deze tools terug via ‘Select by Location’. De basissyntax voor een ruimtelijke selectie is telkens hetzelfde:

```
<INPUTLAAG> <RUIMTELIJK PREDICAAT> <VERGELIJKINGSLAAG>
```

Vanuit de GUI van QGIS zijn de volgende ruimtelijke predicaten beschikbaar: - intersect - contain - disjoin - equal - touch - overlap - are within - cross

In deze tutorial zullen we kijken welke landen doorsneden worden door de Rijn. We maken hiervoor gebruik van een dataset met landen en rivieren, die beschikbaar is via <https://www.naturalearthdata.com/>. Laten we starten met het importeren van deze data als twee afzonderlijke GeoDataFrames met behulp van de `from_file()`-methode:

```
[ ]: countries = gpd.GeoDataFrame.from_file('data/ne_10m_admin_0_countries/
    ↪ne_10m_admin_0_countries.shp')
rivers = gpd.GeoDataFrame.from_file('data/
    ↪ne_10m_rivers_lake_centerlines_scale_rank/
    ↪ne_10m_rivers_lake_centerlines_scale_rank.shp')
```

Op basis van een attribuutbevraging kunnen we een extractie maken van de `rivers`-GeoDataFrame:

```
[ ]: river = rivers[rivers['name_nl'] == 'Rijn']
river.plot()
```

Bij het selecteren van objecten op basis van attributen hebben we gezien dat `pandas` rij per rij een bepaalde kolom zal evalueren. Voor een ruimtelijke selectie lijken desktop-GIS lijken de verschillende selectietools hetzelfde te doen, vermits we telkens een gehele datalaag als input- en vergelijkingslaag meegeven.

`GeoPandas` ondersteunt tijdens het schrijven van deze tutorial (voor zover de auteur heeft kunnen achterhalen) geen ruimtelijke selectiealgorithmen die rechtstreeks uitgevoerd kunnen worden op een

`GeoDataFrame`. In plaats daarvan zullen we voor iedere unieke combinatie van objecten uit zowel de input- als de vergelijkingslaag evalueren of een bepaalde ruimtelijke relatie gerespecteerd wordt.

Op deze manier selecteren we dus alle landen waar de Rijn doorheen stroomt:

```
[ ]: countryIds = []
for iC, fC in countries.iterrows():
    for iR, fR in river.iterrows():
        if fC['geometry'].intersects(fR['geometry']):
            countryIds.append(iC)
riverCountries = countries.loc[countryIds]
riverCountries.head()
```

De selectie van de rivier de Rijn uit de corresponderende dataset resulteert in een verzameling lijnelementen. Aangezien voor ieder segment een selectie wordt gemaakt van de snijdende landen, zullen verschillende meermaals voorkomen. Volledigheidshalve verwijderen we daarom alle duplicaten uit het resultaat:

```
[ ]: riverCountries.drop_duplicates()
```

En tot slot plotten we de geselecteerde entiteiten:

```
[ ]: riverCountries.plot(edgecolor='k')
```

In plaats van deze selectie uit te voeren op alle individuele lijnstukken, kunnen we de verschillende segmenten ook eerst samenvoegen met behulp van de `unary_union`-eigenschap:

```
[ ]: rijn = river.unary_union
countryIds = []
for iC, fC in countries.iterrows():
    if fC['geometry'].intersects(rijn):
        countryIds.append(iC)
riverCountries = countries.loc[countryIds]
riverCountries.head()
```

Te onthouden: - `GeoPandas` ondersteunt verschillende type ruimtelijke bevragingen;
- Om ruimtelijke bevragingen uit te voeren itereren we over een gewenste inputlaag;
- De vergelijkingslaag overlopen we ofwel via een iterator, ofwel als samengevoegd object via de `unary_union`-eigenschap van de `GeoSeries`;