

# 1

---

## Γεωγραφικά Δεδομένα στην Python

---

### 1.1 Εισαγωγή

Αυτό το κεφάλαιο σκιαγραφεί δύο βασικά μοντέλα γεωγραφικών δεδομένων (διανυσματικά και πλεγματικά/ψηφιδωτά) και παρουσιάζει πακέτα της Python για την επεξεργασία αυτών των δεδομένων. Προτού να επιδειχθεί η υλοποίησή τους σε Python, θα παρουσιάσουμε τη θεωρία πίσω από κάθε μοντέλο δεδομένων και τους κλάδους στους οποίους αυτά τα μοντέλα κυριαρχούν.

Το μοντέλο διανυσματικών δεδομένων (Ενότητα 1.2) αναπαριστά γεωγραφικές οντότητες με σημεία, γραμμές και πολύγωνα. Αυτές οι οντότητες έχουν διακριτά, καλά καθορισμένα όρια, υπό την έννοια ότι τα διανυσματικά σύνολα δεδομένων συνήθως έχουν υψηλό επίπεδο ορθότητας (accuracy) (αλλά όχι απαραίτητα ακριβείας (precision)). Το μοντέλο πλεγματικών δεδομένων (Ενότητα 1.3), από την άλλη, διαιρεί τη συνολική επιφάνεια σε κελιά σταθερού μεγέθους. Τα πλεγματικά σύνολα δεδομένων αποτελούν τη βάση των εικόνων φόντου που χρησιμοποιούνται στους διαδικτυακούς χάρτες και έχουν αποτελέσει ζωτικής σημασίας πηγές γεωγραφικών δεδομένων από τις απαρχές της αεροφωτογραφίας και των δορυφορικών συσκευών τηλεπισκόπησης. Τα πλεγματικά δεδομένα συναθροίζουν χωρικά εξαρτημένα στοιχεία σε μια δεδομένη ανάλυση, γεγονός που σημαίνει ότι είναι συνεχή στο χώρο και επεκτάσιμα σε διαφορετικές κλίμακες, παρέχοντας πολλά διαθέσιμα πλεγματικά σύνολα δεδομένων σε παγκόσμιο επίπεδο.

Ποιο να χρησιμοποιήσετε; Η απάντηση ενδεχομένως εξαρτάται από τον τομέα εφαρμογής και από τα σύνολα δεδομένων στα οποία έχετε πρόσβαση:

- Τα διανυσματικά σύνολα δεδομένων και οι αντίστοιχες μέθοδοι κυριαρχούν

στις κοινωνικές επιστήμες, επειδή οι ανθρώπινοι οικισμοί και οι διαδικασίες (π.χ., υποδομές μεταφορών) τείνουν να έχουν διακριτά όρια.

- Τα πλεγματικά σύνολα δεδομένων και οι αντίστοιχες μέθοδοι κυριαρχούν σε πολλές περιβαλλοντικές επιστήμες λόγω της εξάρτησής τους από δεδομένα τηλεπισκόπησης.

Η Python παρέχει ισχυρή υποστήριξη και για τα δύο μοντέλα δεδομένων. Θα επικεντρωθούμε στα (πακέτα) **shapely** και **geopandas** για την επεξεργασία διανυσματικών γεωγραφικών δεδομένων και στο **rasterio** για την επεξεργασία πλεγματικών δεδομένων.

Το **shapely** είναι ένα πακέτο «χαμηλού επιπέδου» για την επεξεργασία μεμονωμένων αντικειμένων διανυσματικής γεωμετρίας. Το **geopandas** είναι ένα πακέτο «υψηλού επιπέδου» για εργασία με στήλες γεωμετρίας (αντικείμενα `GeoSeries`), οι οποίες εσωτερικά περιέχουν γεωμετρίες του **shapely**, καθώς και για την εργασία με διανυσματικά επιθέματα (αντικείμενα `GeoDataFrame`). Το οικοσύστημα **geopandas** παρέχει μια ολοκληρωμένη προσέγγιση για την εργασία με διανυσματικά επιθέματα στην Python, με πολλά πακέτα να βασίζονται σε αυτό.

Υπάρχουν διάφορα μερικώς επικαλυπτόμενα πακέτα για την εργασία με πλεγματικά δεδομένα, το καθένα με τα δικά του πλεονεκτήματα και μειονεκτήματα. Σε αυτό το βιβλίο, εστιάζουμε στο πιο γνωστό από αυτά: το **rasterio**, το οποίο αναπαριστά «απλά» πλεγματικά σύνολα δεδομένων με έναν συνδυασμό ενός πίνακα της **numpy** και ενός αντικειμένου μεταδεδομένων (`dict`) που παρέχει γεωγραφικά μεταδεδομένα, όπως το σύστημα συντεταγμένων. Το **xarray** είναι μια αξιοσημείωτη εναλλακτική λύση του **rasterio** που δεν καλύπτεται σε αυτό το βιβλίο, το οποίο χρησιμοποιεί τις εγγενείς κλάσεις `xarray.Dataset` και `xarray.DataArray` για την αποτελεσματική αναπαράσταση σύνθετων πλεγματικών συνόλων δεδομένων, όπως αρχεία `NetCDF` με πολλαπλά κανάλια και μεταδεδομένα.

Σε ορισμένα πεδία υπάρχει μεγάλη αλληλοεπικάλυψη, και τα πλεγματικά και διανυσματικά σύνολα δεδομένων μπορούν να χρησιμοποιηθούν από κοινού: οι οικολόγοι και οι δημογράφοι, για παράδειγμα, χρησιμοποιούν συνήθως αμφότερα τα διανυσματικά και τα πλεγματικά δεδομένα. Επιπλέον, είναι δυνατή η μετατροπή μεταξύ των δύο μορφών (βλ. Κεφάλαιο 5). Είτε η εργασία σας περιλαμβάνει τη χρήση διανυσματικών είτε πλεγματικών συνόλων δεδομένων, αξίζει να κατανοήσετε τα υποκείμενα μοντέλα δεδομένων πριν τα χρησιμοποιήσετε, όπως περιγράφεται και σε επόμενα κεφάλαια.

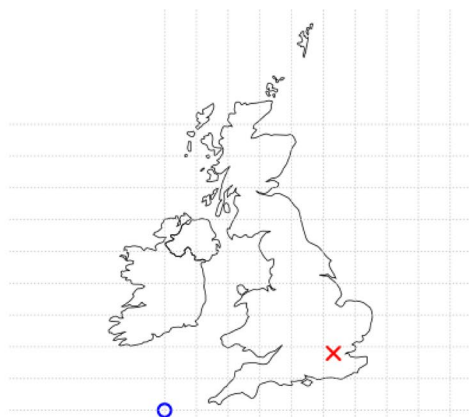
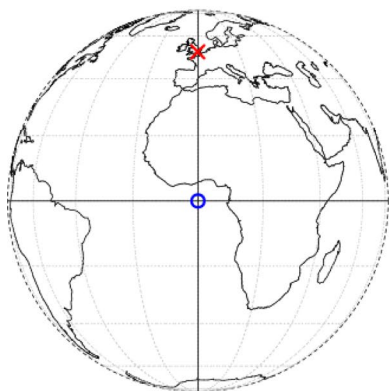
---

## 1.2 Διανυσματικά Δεδομένα

Το μοντέλο γεωγραφικών διανυσματικών δεδομένων βασίζεται σε σημεία που βρίσκονται εντός ενός συστήματος αναφοράς συντεταγμένων (ΣΑΣ – CRS). Τα σημεία μπορούν να αντιπροσωπεύουν αυτόνομα στοιχεία (π.χ., την τοποθεσία μιας στάσης λεωφορείου), ή μπορούν να συνδεθούν μεταξύ τους για να σχηματίσουν πιο σύνθετες γεωμετρίες, όπως γραμμές και πολύγωνα. Οι περισσότερες γεωμετρίες σημείων περιέχουν μόνο δύο διαστάσεις (τα τρισδιάστατα ΣΑΣ μπορεί να περιέχουν μια επιπλέον τιμή  $z$ , η οποία συνήθως αντιπροσωπεύει το ύψος πάνω από τη στάθμη της θάλασσας).

Σε αυτό το σύστημα, το Λονδίνο, για παράδειγμα, μπορεί να αναπαρασταθεί με τις συντεταγμένες:  $(-0,1, 51,5)$ . Αυτό σημαίνει ότι η θέση του βρίσκεται  $-0,1$  μοίρες ανατολικά και  $51,5$  μοίρες βόρεια της αρχής των συντεταγμένων. Η αρχή των συντεταγμένων, σε αυτή την περίπτωση, βρίσκεται στο γεωγραφικό μήκος  $0$  μοιρών (ένας πρώτος μεσημβρινός που διέρχεται από το Γκρίνουιτς) και στο γεωγραφικό πλάτος  $0$  μοιρών (ο Ισημερινός) σε ένα γεωγραφικό («lon/lat») ΣΑΣ (Εικόνα 1.1, αριστερό πλαίσιο). Το ίδιο σημείο θα μπορούσε επίσης να προσεγγιστεί σε ένα προβολικό ΣΑΣ με τιμές «Easting/Northing» ( $530000, 180000$ ) στο Βρετανικό Εθνικό Δίκτυο, κάτι που σημαίνει ότι το Λονδίνο βρίσκεται  $530\text{ km}$  ανατολικά και  $180\text{ km}$  βόρεια της αρχής του ΣΑΣ (Εικόνα 1.1, δεξί πλαίσιο). Η τοποθέτηση της αρχής του Εθνικού Δικτύου, στη θάλασσα πέρα από τη Νοτιοδυτική Χερσόνησο, εξασφαλίζει ότι οι περισσότερες τοποθεσίες στο Ηνωμένο Βασίλειο έχουν θετικές τιμές Easting και Northing.

Υπάρχουν και άλλα στοιχεία σχετικά με τα ΣΑΣ, τα οποία περιγράφονται στην Ενότητα 1.4 και στο Κεφάλαιο 6, αλλά, για τους σκοπούς της παρούσας ενότητας, αρκεί να γνωρίζουμε ότι οι συντεταγμένες αποτελούνται από δύο αριθμούς οι οποίοι αντιπροσωπεύουν την απόσταση από μια αρχή, συνήθως στις διαστάσεις  $x$  και  $y$ .



**Εικόνα 1.1:** Απεικόνιση διανυσματικών (σημειακών) δεδομένων, όπου η τοποθεσία του Λονδίνου (το κόκκινο X) αναπαρίσταται σε σχέση με μια αρχή συντεταγμένων (μπλε κύκλος). Η αριστερή γραφική απεικόνιση αναπαριστά ένα γεωγραφικό ΣΑΣ με αρχή συντεταγμένων γεωγραφικού μήκους και γεωγραφικού πλάτους  $0^\circ$ . Η δεξιά γραφική απεικόνιση διάγραμμα αναπαριστά ένα προβολικό ΣΑΣ με αρχή συντεταγμένων η οποία βρίσκεται στη θάλασσα δυτικά της Νοτιοδυτικής Χερσονήσου.

Το πακέτο **geopandas** (Bossche et al. 2023) παρέχει κλάσεις για γεωγραφικά διανυσματικά δεδομένα και ένα σταθερό περιβάλλον γραμμής εντολών για αναπαραγωγή ανάλυση γεωγραφικών δεδομένων σε Python. Παρέχει επίσης μια διεπαφή με τρεις ώριμες βιβλιοθήκες για γεωυπολογισμούς, μια ισχυρή βάση πάνω στην οποία αναπτύσσονται πολλές γεωγραφικές εφαρμογές:

- Τη GDAL, για την ανάγνωση, εγγραφή και τη διαχείριση ενός ευρέος φάσματος μορφότυπων γεωγραφικών δεδομένων, η οποία περιγράφεται στο Κεφάλαιο 7.
- Την PROJ, μια ισχυρή βιβλιοθήκη για μετασχηματισμούς συστημάτων συντεταγμένων, η οποία αποτελεί τη βάση του περιεχομένου που καλύπτεται στο Κεφάλαιο 6.
- Την GEOS, μια μηχανή επίπεδης γεωμετρίας για λειτουργίες όπως ο υπολογισμός ζωνών επιρροής (buffers) και κεντροειδών (centroids) σε δεδομένα με προβολικό ΣΑΣ, η οποία περιγράφεται στο Κεφάλαιο 4.

Η στενή προσαρμογή αυτών των γεωγραφικών βιβλιοθηκών καθιστά εφικτή την αναπαραγωγικότητα των γεω-υπολογισμών: ένα πλεονέκτημα της χρήσης μιας γλώσσας υψηλότερου επιπέδου (higher-level) όπως η Python για την πρόσβαση σε αυτές τις βιβλιοθήκες είναι ότι δεν χρειάζεται να γνωρίζετε τις λεπτομέρειες των χαμηλού

επιπέδου (low-level) συστατικών στοιχείων, επιτρέποντάς σας να εστιάσετε στις μεθόδους και όχι στην υλοποίηση.

### 1.2.1 Κλάσεις Διανυσματικών Δεδομένων

Οι κύριες κλάσεις για την εργασία με γεωγραφικά διανυσματικά δεδομένα στην Python είναι ιεραρχικές, δηλαδή η κλάση «διανυσματικό επίθεμα (vector layer)» αποτελείται από απλούστερα συστατικά στοιχεία «στηλών γεωμετρίας» και μεμονωμένα συστατικά στοιχεία «γεωμετρίας». Σε αυτή την ενότητα παρουσιάζονται με τη σειρά, ξεκινώντας από την κλάση ανώτερου επιπέδου. Για πολλές εφαρμογές, το μόνο που χρειάζεται είναι η κλάση διανυσματικό επίθεμα, ένα πλαίσιο δεδομένων (data frame) με στήλες γεωμετρίας. Ωστόσο, είναι σημαντικό να εξοικειωθούμε με τη δομή των διανυσματικών γεωγραφικών αντικειμένων και τα συστατικά τους για ορισμένες εφαρμογές και για μια περισσότερο εις βάθος κατανόηση. Οι τρεις κύριες κλάσεις διανυσματικών γεωγραφικών δεδομένων στην Python είναι οι ακόλουθες:

- `GeoDataFrame`, μια κλάση που αντιπροσωπεύει διανυσματικά επιθέματα, με μια στήλη γεωμετρίας (κλάση `GeoSeries`) ως μία από τις στήλες.
- `GeoSeries`, μια κλάση που χρησιμοποιείται για την αναπαράσταση της στήλης γεωμετρίας σε αντικείμενα `GeoDataFrame`.
- Αντικείμενα γεωμετρίας **shapely**, τα οποία αντιπροσωπεύουν μεμονωμένες γεωμετρίες, όπως ένα σημείο ή ένα πολύγωνο σε αντικείμενα `GeoSeries`.

Οι δύο πρώτες κλάσεις (`GeoDataFrame` και `GeoSeries`) ορίζονται στο **geopandas**. Η τρίτη κλάση ορίζεται στο πακέτο **shapely**, το οποίο διαχειρίζεται μεμονωμένες γεωμετρίες, και αποτελεί βασική εξάρτηση του πακέτου **geopandas**.

### 1.2.2 Διανυσματικά Επιθέματα

Η πιο κοινώς χρησιμοποιούμενη δομή γεωγραφικών διανυσματικών δεδομένων είναι το διανυσματικό επίθεμα. Υπάρχουν διάφορες προσεγγίσεις για την εργασία με διανυσματικά επιθέματα στην Python, που κυμαίνονται από πακέτα χαμηλού επιπέδου (π.χ. **osgeo**, **fiona**) έως το σχετικά υψηλού επιπέδου πακέτο **geopandas**, στο οποίο εστιάζει αυτή η ενότητα. Πριν γράψουμε και εκτελέσουμε κώδικα για τη δημιουργία γεωγραφικών διανυσματικών αντικειμένων και την εργασία με αυτά, πρέπει να εισαγάγουμε (`import`) το **geopandas** (κατά σύμβαση ως `gpd` για πιο συνοπτικό κώδικα) και το **shapely**.

```
import pandas as pd
import shapely
import geopandas as gpd
```

Επίσης, περιορίζουμε το μέγιστο αριθμό σειρών που εκτυπώνονται σε έξι, για εξοικονόμηση χώρου, χρησιμοποιώντας την επιλογή `'display.max_rows'` του **pandas**.

```
pd.set_option('display.max_rows', 6)
```

Τα έργα (projects) ξεκινούν συχνά με την εισαγωγή ενός υπάρχοντος διανυσματικού επιθέματος το οποίο έχει αποθηκευτεί ως αρχείο GeoPackage (`.gpkg`), ως αρχείο ESRI Shapefile (`.shp`) ή ως αρχείο άλλου γεωγραφικού μορφότυπου. Η συνάρτηση `gpd.read_file` εισάγει ένα αρχείο GeoPackage με το όνομα `world.gpkg`, το οποίο βρίσκεται στον κατάλογο `data` του τρέχοντος καταλόγου εργασίας της Python, σε ένα `GeoDataFrame` με το όνομα `gdf`.

```
gdf = gpd.read_file('data/world.gpkg')
```

Το αποτέλεσμα είναι ένα αντικείμενο τύπου (κλάσης) `GeoDataFrame` με 177 γραμμές (στοιχεία) και 11 στήλες, όπως φαίνεται στην έξοδο του παρακάτω κώδικα:

```
type(gdf)
geopandas.geodataframe.GeoDataFrame
gdf.shape

(177, 11)
```

Η κλάση `GeoDataFrame` αποτελεί επέκταση της κλάσης `DataFrame` από το δημοφιλές πακέτο **pandas** (McKinney 2010). Αυτό σημαίνει ότι μπορούμε να αντιμετωπίσουμε τα μη-χωρικά χαρακτηριστικά ενός διανυσματικού επιθέματος ως πίνακα, και να τα επεξεργαζόμαστε χρησιμοποιώντας τις συνήθειες, δηλαδή τις μη-χωρικές, καθιερωμένες μεθόδους συναρτήσεων. Για παράδειγμα, μπορούν να χρησιμοποιηθούν οι τυπικές μέθοδοι δημιουργίας υποσυνόλων πλαισίων δεδομένων. Ο παρακάτω κώδικας δημιουργεί ένα υποσύνολο του συνόλου δεδομένων `gdf` το οποίο περιέχει μόνο το όνομα της χώρας και τη γεωμετρία.

```
gdf = gdf[['name_long', 'geometry']]
gdf
```

|     | name_long           | geometry                           |
|-----|---------------------|------------------------------------|
| 0   | Fiji                | MULTIPOLYGON (((-180 -16.55522,... |
| 1   | Tanzania            | MULTIPOLYGON (((33.90371 -0.95,... |
| 2   | Western Sahara      | MULTIPOLYGON (((-8.66559 27.656... |
| ... | ...                 | ...                                |
| 174 | Kosovo              | MULTIPOLYGON (((20.59025 41.855... |
| 175 | Trinidad and Tobago | MULTIPOLYGON (((-61.68 10.76, -... |
| 176 | South Sudan         | MULTIPOLYGON (((30.83385 3.5091... |

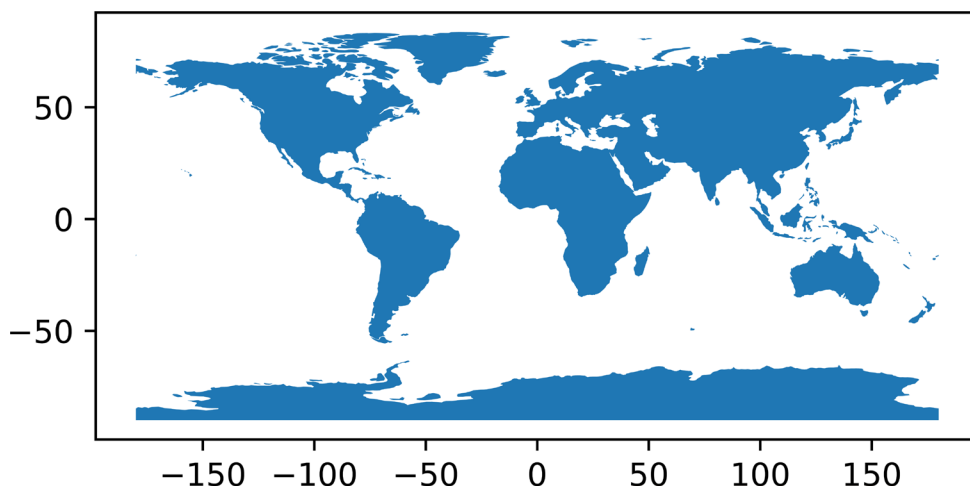
Η παρακάτω έκφραση δημιουργεί ένα υποσύνολο δεδομένων με βάση μια συνθήκη, όπως η ισότητα της τιμής στη στήλη 'name\_long' με τη συμβολοσειρά 'Egypt'.

```
gdf[gdf['name_long'] == 'Egypt']
```

|     | name_long | geometry                           |
|-----|-----------|------------------------------------|
| 163 | Egypt     | MULTIPOLYGON (((36.86623 22, 36... |

Τέλος, για να αποκτήσουμε μια αίσθηση της χωρικής διάστασης του διανυσματικού επιθέματος, μπορούμε να το απεικονίσουμε γραφικά χρησιμοποιώντας τη μέθοδο `.plot` (Εικόνα 1.2).

```
gdf.plot();
```



**Εικόνα 1.2:** Βασική γραφική απεικόνιση ενός GeoDataFrame

Μπορούν να δημιουργηθούν διαδραστικοί χάρτες αντικειμένων GeoDataFrame με τη μέθοδο `.explore`, όπως ο χάρτης που απεικονίζεται στην Εικόνα 1.3 ο οποίος δημιουργήθηκε με την ακόλουθη εντολή:

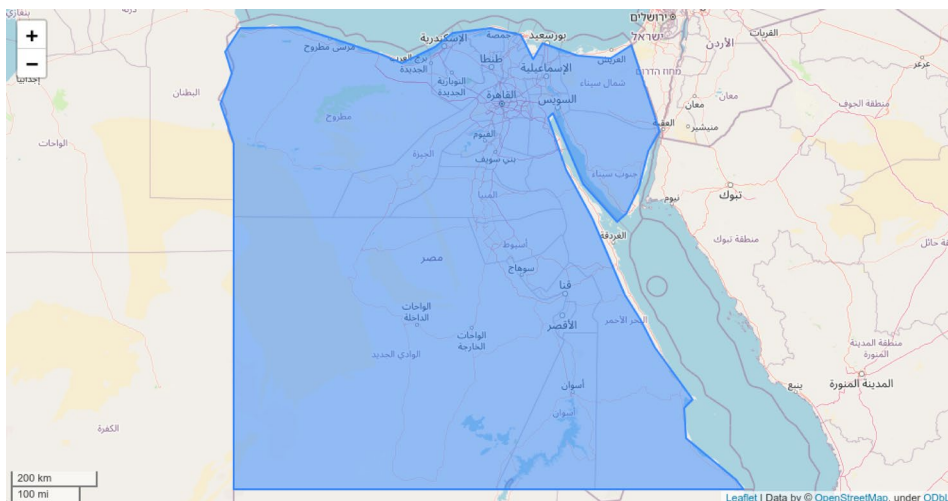
```
gdf.explore()
```



**Εικόνα 1.3:** Βασικός διαδραστικός χάρτης που προκύπτει από την `.explore`.

Ένα υποσύνολο των δεδομένων μπορεί επίσης να απεικονιστεί γραφικά με παρόμοιο τρόπο.

```
gdf[gdf['name_long'] == 'Egypt'].explore()
```



**Εικόνα 1.4:** Διαδραστικός χάρτης ενός υποσυνόλου δεδομένων GeoDataFrame.

### 1.2.3 Στήλες Γεωμετρίας

Η στήλη γεωμετρίας της κλάσης `GeoSeries` αποτελεί ουσιώδη στήλη σε ένα `GeoDataFrame`. Περιέχει το γεωμετρικό μέρος του διανυσματικού επιθέματος και αποτελεί τη βάση για όλες τις χωρικές λειτουργίες. Η στήλη αυτή μπορεί να προσπελαστεί μέσω του ονόματός της, το οποίο συνήθως (π.χ., κατά την ανάγνωση από αρχείο) είναι `'geometry'`, όπως στην έκφραση `gdf['geometry']`. Ωστόσο, συνιστάται η χρήση της σταθερής ιδιότητας `.geometry`, η οποία αναφέρεται στη στήλη γεωμετρίας ανεξάρτητα από το αν το όνομά της είναι `'geometry'` ή όχι. Στην περίπτωση του αντικειμένου `gdf`, η στήλη γεωμετρίας περιέχει `'MultiPolygon'`, τα οποία συνδέονται με κάθε χώρα.

```
gdf.geometry
```

```
0    MULTIPOLYGON (((-180 -16.55522,...
1    MULTIPOLYGON (((33.90371 -0.95,...
2    MULTIPOLYGON (((-8.66559 27.656...

...

174  MULTIPOLYGON (((20.59025 41.855...
175  MULTIPOLYGON (((-61.68 10.76, -...
176  MULTIPOLYGON (((30.83385 3.5091...
Name: geometry, Length: 177, dtype: geometry
```

Η στήλη της γεωμετρίας περιέχει επίσης τις πληροφορίες της χωρικής αναφοράς, εφόσον υπάρχουν (επίσης προσπελάσιμες μέσω της συντόμευσης `gdf.crs`).

```
gdf.geometry.crs
```

```
<Geographic 2D CRS: EPSG:4326>
Name: WGS 84
Axis Info [ellipsoidal]:
- Lat[north]: Geodetic latitude (degree)
- Lon[east]: Geodetic longitude (degree)
Area of Use:
- name: World.
- bounds: (-180.0, -90.0, 180.0, 90.0)
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

Πολλές λειτουργίες γεωμετρίας, όπως ο υπολογισμός του κεντροειδούς, των ζωνών επιρροής, ή του πλαισίου οριοθέτησης (bounding box) κάθε στοιχείου, αφορούν αποκλειστικά τη γεωμετρία. Η εφαρμογή αυτού του είδους λειτουργιών σε ένα GeoDataFrame επομένως, είναι ουσιαστικά μια συντόμευση για την εφαρμογή τους στο αντικείμενο GeoSeries στη στήλη γεωμετρίας. Για παράδειγμα, οι δύο ακόλουθες εντολές επιστρέφουν ακριβώς το ίδιο αποτέλεσμα, ένα GeoSeries που περιέχει πολύγωνα πλαισίων οριοθέτησης (χρησιμοποιώντας τη μέθοδο `.envelope`).

```
gdf.envelope
```

```
0      POLYGON ((-180 -18.28799, 179.9...
1      POLYGON ((29.34 -11.72094, 40.3...
2      POLYGON ((-17.06342 20.99975, -...
...
174     POLYGON ((20.0707 41.84711, 21....
175     POLYGON ((-61.95 10, -60.895 10...
176     POLYGON ((23.88698 3.50917, 35....
Length: 177, dtype: geometry
```

```
gdf.geometry.envelope
```

```
0      POLYGON ((-180 -18.28799, 179.9...
1      POLYGON ((29.34 -11.72094, 40.3...
2      POLYGON ((-17.06342 20.99975, -...
...
174     POLYGON ((20.0707 41.84711, 21....
175     POLYGON ((-61.95 10, -60.895 10...
176     POLYGON ((23.88698 3.50917, 35....
Length: 177, dtype: geometry
```

Σημειώστε ότι η `.envelope`, καθώς και άλλες παρόμοιες λειτουργίες στο **geopandas**, όπως η `.centroid` (Ενότητα 4.2.2), η `.buffer` (Ενότητα 4.2.3) ή η `.convex_hull`, επιστρέφουν μόνο τη γεωμετρία (δηλ. ένα GeoSeries) και όχι ένα GeoDataFrame με τα αρχικά δεδομένα των περιγραφικών χαρακτηριστικών. Σε περίπτωση που θέλουμε το τελευταίο, μπορούμε να δημιουργήσουμε ένα αντίγραφο του GeoDataFrame και στη συνέχεια να «αντικαταστήσουμε» τη γεωμετρία του (ή, μπορούμε να αντικαταστήσουμε τις γεωμετρίες απευθείας σε περίπτωση που δεν

χρησιμοποιήστε τις αρχικές, όπως στην έκφραση `gdf.geometry=gdf.envelope`).

```
gdf2 = gdf.copy()
gdf2.geometry = gdf.envelope
gdf2
```

|     | name_long           | geometry                           |
|-----|---------------------|------------------------------------|
| 0   | Fiji                | POLYGON ((-180 -18.28799, 179.9... |
| 1   | Tanzania            | POLYGON ((29.34 -11.72094, 40.3... |
| 2   | Western Sahara      | POLYGON ((-17.06342 20.99975, -... |
| ... | ...                 | ...                                |
| 174 | Kosovo              | POLYGON ((20.0707 41.84711, 21.... |
| 175 | Trinidad and Tobago | POLYGON ((-61.95 10, -60.895 10... |
| 176 | South Sudan         | POLYGON ((23.88698 3.50917, 35.... |

Μια άλλη χρήσιμη ιδιότητα της στήλης γεωμετρίας είναι ο τύπος γεωμετρίας, όπως φαίνεται στον παρακάτω κώδικα. Σημειώστε ότι οι τύποι γεωμετριών που περιέχονται σε μια στήλη γεωμετρίας (και, συνεπώς, σε ένα διανυσματικό επίθεμα) δεν είναι απαραίτητα οι ίδιοι για κάθε γραμμή. Είναι δυνατό να υπάρχουν πολλαπλοί τύποι γεωμετριών σε ένα μόνο GeoSeries. Κατά συνέπεια, η ιδιότητα `.type` επιστρέφει ένα Series (με τιμές τύπου `str`, δηλαδή, συμβολοσειρές), αντί για μια μεμονωμένη τιμή (το ίδιο μπορεί να γίνει με τη συντόμευση `gdf.geom_type`).

```
gdf.geometry.type
0      MultiPolygon
1      MultiPolygon
2      MultiPolygon
...
174     MultiPolygon
175     MultiPolygon
176     MultiPolygon
Length: 177, dtype: object
```

Για να συνοψίσουμε την παρουσία διαφορετικών τύπων γεωμετριών σε μια στήλη γεωμετρίας, μπορούμε να χρησιμοποιήσουμε τη μέθοδο `.value_counts` του **pandas**. Σε αυτή την περίπτωση, παρατηρούμε ότι το επίθεμα `gdf` περιέχει μόνο γεωμετρίες τύπου `'MultiPolygon'`.

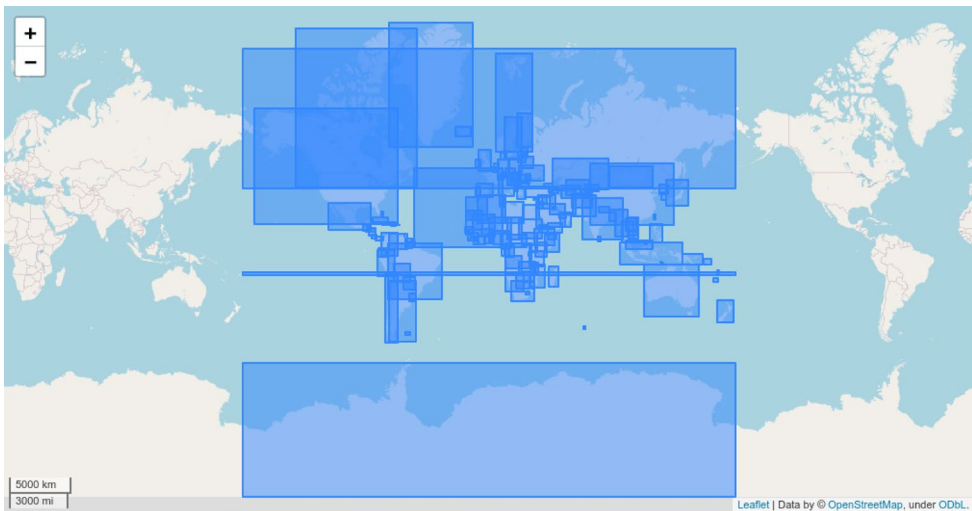
```
gdf.geometry.type.value_counts()
```

```
MultiPolygon 177
```

```
Name: count, dtype: int64
```

Ένα GeoDataFrame επίσης μπορεί να περιέχει πολλές στήλες GeoSeries, όπως φαίνεται στο παρακάτω απόσπασμα κώδικα.

```
gdf['bbox'] = gdf.envelope  
gdf['polygon'] = gdf.geometry  
gdf
```



**Εικόνα 1.5:** Μετάβαση στη στήλη γεωμετρίας του 'bbox' στο επίθεμα world και γραφική απεικόνισή της.

Μόνο μία στήλη γεωμετρίας είναι «ενεργή» κάθε φορά, υπό την έννοια ότι προσπελαύνεται από λειτουργίες που αφορούν τις γεωμετρίες (όπως `.centroid`, `.crs`, κ.λπ.). Για να αλλάξουμε την ενεργή στήλη γεωμετρίας από τη μία στήλη GeoSeries σε μια άλλη, χρησιμοποιούμε το `.set_geometry`. Στο Σχήμα 1.5 και στο Σχήμα 1.6 παρουσιάζονται διαδραστικοί χάρτες του επιθέματος gdf με ενεργοποιημένες τις στήλες γεωμετρίας 'bbox' και 'polygon', αντίστοιχα.



**Εικόνα 1.6:** Μετάβαση στη στήλη γεωμετρίας του ‘polygons’ στο επίθεμα world και γραφική απεικόνισή της.

```
gdf = gdf.set_geometry('bbox')  
gdf.explore()
```

```
gdf = gdf.set_geometry('polygon')  
gdf.explore()
```

#### 1.2.4 Το Πρότυπο Simple Features (Απλών Στοιχείων)

Οι γεωμετρίες αποτελούν τα βασικά δομικά στοιχεία των διανυσματικών επιθεμάτων. Αν και το πρότυπο Simple Features ορίζει περίπου 20 τύπους γεωμετριών, θα επικεντρωθούμε στους επτά πιο συχνά χρησιμοποιούμενους: POINT, LINESTRING, POLYGON, MULTIPOINT, MULTILINESTRING, MULTIPOLYGON και GEOMETRYCOLLECTION. Μια χρήσιμη λίστα με τους πιθανούς τύπους γεωμετριών μπορεί να βρεθεί στην τεκμηρίωση του πακέτου **sf** της R.<sup>1</sup>

Οι γεωμετρίες απλών στοιχείων μπορούν να αναπαρασταθούν από τις κωδικοποιήσεις well-known binary (WKB) και well-known text (WKT). Οι αναπαραστάσεις WKB είναι συνήθως δεκαεξαδικές συμβολοσειρές που διαβάζονται εύκολα από τους υπολογιστές, και αυτός είναι ο λόγος για τον οποίο τα λογισμικά ΣΓΠ και οι χωρικές

<sup>1</sup> <https://r-spatial.github.io/sf/articles/sf1.html#simple-feature-geometry-types>

βάσεις δεδομένων χρησιμοποιούν την WKB για τη μεταφορά και την αποθήκευση γεωμετρικών αντικειμένων. Η WKT, από την άλλη πλευρά, είναι μια αναγνώσιμη από τον άνθρωπο περιγραφή σήμανσης κειμένου των Simple Features. Αν και οι δύο μορφότυποι είναι εναλλάξιμοι, όταν χρειάζεται να παρουσιάσουμε έναν εξ αυτών, επιλέγουμε φυσικά την αναπαράσταση WKT.

Η βάση κάθε τύπου γεωμετρίας είναι το σημείο (point). Ένα σημείο είναι απλώς ένα ζεύγος συντεταγμένων σε δισδιάστατο, τρισδιάστατο ή τετραδιάστατο χώρο, όπως φαίνεται στο Σχήμα 1.7.

```
POINT (5 2)
```

Μια γραμμή (linestring) είναι μια ακολουθία σημείων που συνδέονται μεταξύ τους με μια ευθεία γραμμή (Εικόνα 1.8).

```
LINESTRING (1 5, 4 4, 4 1, 2 2, 3 2)
```

Ένα πολύγωνο είναι μια ακολουθία σημείων που σχηματίζουν έναν κλειστό, μη τεμνόμενο δακτύλιο. Κλειστό σημαίνει ότι το πρώτο και το τελευταίο σημείο ενός πολυγώνου έχουν τις ίδιες συντεταγμένες (Εικόνα 1.9).

```
POLYGON ((1 5, 2 2, 4 1, 4 4, 1 5))
```

Μέχρι τώρα έχουμε δημιουργήσει γεωμετρίες με μία μόνο γεωμετρική οντότητα ανά στοιχείο. Ωστόσο, το πρότυπο Simple Features επιτρέπει την ύπαρξη πολλαπλών γεωμετριών εντός ενός μεμονωμένου στοιχείου, χρησιμοποιώντας ‘πολλαπλές’ εκδόσεις κάθε τύπου γεωμετρίας, όπως απεικονίζεται στις Εικόνες 1.10, 1.11 και 1.12.

```
MULTIPOINT (5 2, 1 3, 3 4, 3 2)
```

```
MULTILINESTRING ((1 5, 4 4, 4 1, 2 2, 3 2), (1 2, 2 4))
```

```
MULTIPOLYGON (((1 5, 2 2, 4 1, 4 4, 1 5), (0 2, 1 2, 1 3, 0 3, 0 2)))
```

Τέλος, μια συλλογή γεωμετριών (geometry collection) μπορεί να περιέχει οποιονδήποτε συνδυασμό γεωμετριών από τους άλλους έξι τύπους, όπως ο συνδυασμός ενός πολυσημείου (multipoint) και μιας γραμμής, όπως φαίνεται παρακάτω (Εικόνα 1.13).

```
GEOMETRYCOLLECTION (MULTIPOINT (5 2, 1 3, 3 4, 3 2),  
LINESTRING (1 5, 4 4, 4 1, 2 2, 3 2))
```

### 1.2.5 Γεωμετρίες

Κάθε στοιχείο στη στήλη γεωμετρίας (GeoSeries) είναι ένα αντικείμενο γεωμετρίας

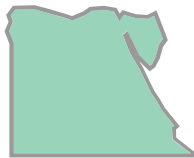
της κλάσης **shapely** (Gillies et al. 2007--). Για παράδειγμα, εδώ είναι μια συγκεκριμένη γεωμετρία που επιλέγεται με έμμεσο δείκτη (Καναδάς, το 4<sup>ο</sup> στοιχείο στη στήλη γεωμετρίας του `gdf`).

```
gdf.geometry.iloc[3]
```



Μπορούμε επίσης να επιλέξουμε μια συγκεκριμένη γεωμετρία με βάση το περιγραφικό χαρακτηριστικό `'name_long'` (δηλαδή, το πρώτο και μοναδικό στοιχείο στο υποσύνολο του `gdf` όπου το όνομα της χώρας ισούται με `Egypt`):

```
gdf[gdf['name_long'] == 'Egypt'].geometry.iloc[0]
```



Το πακέτο **shapely** είναι συμβατό με το πρότυπο Simple Features (Ενότητα 1.2.4). Συνεπώς, υποστηρίζονται επτά τύποι γεωμετριών. Η παρακάτω ενότητα παρουσιάζει τη δημιουργία μιας γεωμετρίας **shapely** κάθε τύπου από την αρχή. Στο πρώτο παράδειγμα (ένα `'Point'` («Σημείο»)) παρουσιάζουμε δύο τύπους εισόδων για τη δημιουργία μιας γεωμετρίας: μια λίστα συντεταγμένων ή μια συμβολοσειρά σε μορφότυπο WKT. Στα παραδείγματα για τις υπόλοιπες γεωμετρίες χρησιμοποιούμε την πρώτη προσέγγιση.

Για τη δημιουργία μιας γεωμετρίας τύπου `'Point'` από μια λίστα ζευγών συντεταγμένων χρησιμοποιείται η συνάρτηση `shapely.Point` στην παρακάτω έκφραση (Εικόνα 1.7).

```
point = shapely.Point([5, 2])
point
```



**Εικόνα 1.7:** Μια γεωμετρία τύπου `Point` (που δημιουργήθηκε είτε από λίστα είτε από WKT).

Εναλλακτικά, μπορούμε να χρησιμοποιήσουμε τη συνάρτηση `shapely.from_wkt` για να μετατρέψουμε μια συμβολοσειρά WKT σε ένα αντικείμενο γεωμετρίας

**shapely.** Ακολουθεί ένα παράδειγμα δημιουργίας της ίδιας γεωμετρίας 'Point' από WKT (Εικόνα 1.7).

```
point = shapely.from_wkt('POINT (5 2)')
point
```

Μια γεωμετρία τύπου 'LineString' μπορεί να δημιουργηθεί με βάση μια λίστα από πλειάδες (tuples) ζευγών συντεταγμένων ή λίστες (lists) (βλ. Σχήμα 1.8).

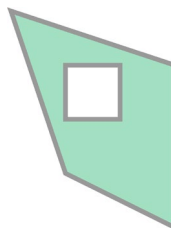
```
linestring = shapely.LineString([(1,5), (4,4), (4,1), (2,2), (3,2)])
linestring
```



**Εικόνα 1.8:** Μια γεωμετρία τύπου LineString.

Η δημιουργία μιας γεωμετρίας τύπου 'Polygon' είναι παρόμοια, αλλά το πρώτο και το τελευταίο ζεύγος συντεταγμένων μας πρέπει να είναι ίδιο, ώστε το πολύγωνο να είναι κλειστό. Σημειώστε ότι στο ακόλουθο παράδειγμα, υπάρχει μία λίστα ζευγών συντεταγμένων που ορίζει το εξωτερικό περίγραμμα του πολυγώνου, ακολουθούμενη από μια list από lists των ζευγών συντεταγμένων που ορίζουν τις οπές (αν υπάρχουν) στο πολύγωνο (Εικόνα 1.9).

```
polygon = shapely.Polygon(
    [(1,5), (2,2), (4,1), (4,4), (1,5)], ## Exterior
    [[(2,4), (3,4), (3,3), (2,3), (2,4)]] ## Hole(s)
)
polygon
```



**Εικόνα 1.9:** Μια γεωμετρία τύπου Polygon.

Μια γεωμετρία τύπου ‘MultiPoint’ δημιουργείται επίσης από μια λίστα ζευγών συντεταγμένων (Εικόνα 1.10), όπου κάθε στοιχείο αντιπροσωπεύει ένα μεμονωμένο σημείο.

```
multipoint = shapely.MultiPoint([(5,2), (1,3), (3,4), (3,2)])
multipoint
```



**Εικόνα 1.10:** Μια γεωμετρία τύπου MultiPoint.

Μια γεωμετρία τύπου ‘MultiLineString’, από την άλλη πλευρά, διαθέτει μία λίστα ζευγών συντεταγμένων για κάθε γραμμή της MultiLineString (Εικόνα 1.11).

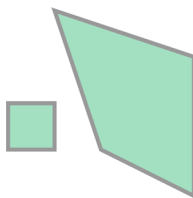
```
multilinestring = shapely.MultiLineString([
    [(1,5), (4,4), (4,1), (2,2), (3,2)], ## 1st sequence
    [(1,2), (2,4)] ## 2nd sequence, etc.
])
multilinestring
```



**Εικόνα 1.11:** Μια γεωμετρία τύπου MultiLineString.

Μια γεωμετρία τύπου ‘MultiPolygon’ (Εικόνα 1.12) δημιουργείται από μια list γεωμετριών τύπου Polygon. Για παράδειγμα, εδώ δημιουργούμε ένα ‘MultiPolygon’ με δύο τμήματα (parts), και τα δύο χωρίς οπές.

```
multipolygon = shapely.MultiPolygon([
    [(1,5), (2,2), (4,1), (4,4), (1,5)], [], ## Polygon 1
    [(0,2), (1,2), (1,3), (0,3), (0,2)], [] ## Polygon 2, etc.
])
multipolygon
```



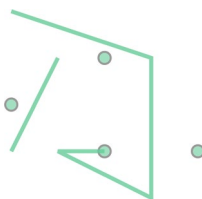
**Εικόνα 1.12:** Μια γεωμετρία τύπου `MultiPolygon`.

Με δεδομένο ότι τα απαιτούμενα δεδομένα εισόδου διαθέτουν τέσσερα ιεραρχικά επίπεδα, ίσως είναι πιο ξεκάθαρο το να δημιουργηθούν εκ των προτέρων οι μονο-τμηματικές (single-part) γεωμετρίες `'Polygon'`, χρησιμοποιώντας την αντίστοιχη συνάρτηση (`shapely.Polygon`), και στη συνέχεια να μεταβιβαστούν στη `shapely.MultiPolygon` (Εικόνα 1.12). (Η ίδια τεχνική μπορεί να χρησιμοποιηθεί και με τις άλλες συναρτήσεις `shapely.Multi*`.)

```
multipolygon = shapely.MultiPolygon([
    shapely.Polygon([(1,5), (2,2), (4,1), (4,4), (1,5)]), ## Polygon 1
    shapely.Polygon([(0,2), (1,2), (1,3), (0,3), (0,2)]) ## Polygon 2, etc.
])
multipolygon
```

Και, τέλος, μια γεωμετρία τύπου `'GeometryCollection'` είναι μια λίστα που περιέχει έναν ή περισσότερους από τους υπόλοιπους έξι τύπους γεωμετρίας (Εικόνα 1.13):

```
geometrycollection = shapely.GeometryCollection([multipoint, multilinestring])
geometrycollection
```

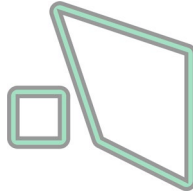


**Εικόνα 1.13:** Μια γεωμετρία τύπου `'GeometryCollection'`.

Οι γεωμετρίες **shapely** λειτουργούν ως ατομικές μονάδες διανυσματικών δεδομένων, πράγμα που σημαίνει ότι δεν υπάρχει η έννοια των συνόλων γεωμετριών: κάθε λειτουργία δέχεται μεμονωμένα αντικείμενα γεωμετρίας ως είσοδο και επιστρέφει μια μεμονωμένη γεωμετρία ως έξοδο. (Τα αντικείμενα `GeoSeries` και `GeoDataFrame`, που ορίζονται στο **geopandas**, χρησιμοποιούνται για τη συλλογική διαχείριση συνόλων γεωμετριών **shapely**). Για παράδειγμα, η ακόλουθη έκφραση υπολογίζει τη δια-

φορά (βλ. Ενότητα 4.2.5) μεταξύ του πολυγώνου με ζώνη επιρροής (βλ. Ενότητα 4.2.3) (χρησιμοποιώντας απόσταση 0,2) και του ίδιου του πολυγώνου (Εικόνα 1.14):

```
multipolygon.buffer(0.2).difference(multipolygon)
```



**Εικόνα 1.14:** Η διαφορά μεταξύ ενός MultiPolygon με ζώνη επιρροής και του ίδιου του MultiPolygon.

Όπως φαίνεται στα τελευταία σχήματα, ένα αντικείμενο γεωμετρίας **shapely** αποδίδεται αυτόματα ως μια μικρή εικόνα της γεωμετρίας (όταν χρησιμοποιείται μια διεπαφή που παρέχει τη δυνατότητα απόδοσής της, όπως το Jupyter Notebook). Για να εκτυπώσουμε τη συμβολοσειρά WKT αντ' αυτού, μπορούμε να χρησιμοποιήσουμε τη συνάρτηση `print`:

```
print(linestring)
```

```
LINESTRING (1 5, 4 4, 4 1, 2 2, 3 2)
```

Τέλος, είναι σημαντικό να σημειωθεί ότι οι πρωταρχικές (raw) συντεταγμένες των γεωμετριών **shapely** είναι προσβάσιμες μέσω ενός συνδυασμού των ιδιοτήτων `.coords`, `.geoms`, `.exterior`, και `.interiors` (ανάλογα με τον τύπο της γεωμετρίας). Αυτές οι μέθοδοι πρόσβασης είναι χρήσιμες όταν χρειάζεται να αναπτύξουμε τους δικούς μας χωρικούς τελεστές για συγκεκριμένες εργασίες. Για παράδειγμα, η ακόλουθη έκφραση επιστρέφει τη `list` όλων των συντεταγμένων του εξωτερικού της γεωμετρίας τύπου `polygon`:

```
list(polygon.exterior.coords)
```

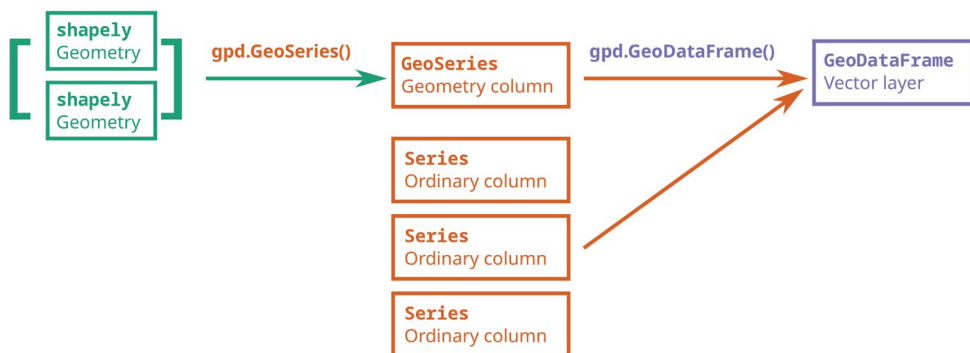
```
[(1.0, 5.0), (2.0, 2.0), (4.0, 1.0), (4.0, 4.0), (1.0, 5.0)]
```

Δείτε επίσης την ενότητα 4.2.8, όπου οι παράμετροι `coords`, `.geoms`, και `.exterior` χρησιμοποιούνται για να επιτελέσουν το μετασχηματισμό από έναν δοσμένο τύπο γεωμετρίας **shapely** σε έναν διαφορετικό τύπο (π.χ. από `'Polygon'` σε `'MultiPoint'`).

### 1.2.6 Διανυσματικό Επίθεμα από την Αρχή

Στις προηγούμενες ενότητες, ξεκινήσαμε με ένα διανυσματικό επίθεμα (GeoDataFrame), από ένα υπάρχον αρχείο GeoPackage, και το «αποσυνθέσαμε» για να εξαγάγουμε τη στήλη γεωμετρίας (GeoSeries, Ενότητα 1.2.3) και τις ξεχωριστές γεωμετρίες (**shapely**, βλ. Ενότητα 1.2.5). Σε αυτή την ενότητα, θα δείξουμε την αντίστροφη διαδικασία, δημιουργώντας ένα GeoDataFrame από γεωμετρίες **shapely**, συνδυασμένες σε ένα GeoSeries. Αυτό θα σας βοηθήσει να κατανοήσετε καλύτερα τη δομή ενός GeoDataFrame και μπορεί να αποδειχθεί χρήσιμο όταν χρειαστεί να δημιουργήσετε με προγραμματιστικό τρόπο απλά διανυσματικά επιθέματα, όπως μια γραμμή μεταξύ δύο δοσμένων σημείων.

Τα διανυσματικά επιθέματα αποτελούνται από δύο βασικά μέρη: γεωμετρίες και μη-γεωγραφικά χαρακτηριστικά. Το Σχήμα 1.15 δείχνει πώς δημιουργείται ένα αντικείμενο GeoDataFrame: οι γεωμετρίες προέρχονται από ένα αντικείμενο GeoSeries (το οποίο αποτελείται από γεωμετρίες τύπου **shapely**), ενώ τα περιγραφικά χαρακτηριστικά λαμβάνονται από αντικείμενα Series.



**Εικόνα 1.15:** Δημιουργία ενός GeoDataFrame από την αρχή.

Το τελικό αποτέλεσμα, ένα διανυσματικό επίθεμα (GeoDataFrame), είναι επομένως μια ιεραρχική δομή (Εικόνα 1.16), η οποία περιέχει τη στήλη γεωμετρίας (GeoSeries), η οποία με τη σειρά της περιέχει γεωμετρίες (**shapely**). Κάθε ένα από τα «εσωτερικά» συστατικά μπορεί να προσπελαστεί ή να «εξαχθεί», κάτι που είναι μερικές φορές απαραίτητο, όπως θα δούμε αργότερα.