GeoPandas : : CHEAT SHEET

Geometric Confirmation

```
gs = geopandas.GeoSeries()
```

gs.contains(other, align=True) Returns a `Series` of `dytpe('bool')` with value `True` for each aligned geometry that contains other.

gs.covered_by(other, align=True) Returns a `Series` of `dytpe('bool')` with value `True` for each aligned geometry that is entirely covered by other.

gs.covers(other, align=True) Returns a `Series` of `dytpe('bool')` with value `True` for each aligned geometry that is entirely covering other.

gs.crosses(other, align=True) Returns a `Series` of `dytpe('bool')` with value `True` for each aligned geometry that crosses other.

gs.disjoint(other, align=True) Returns a `Series` of `dytpe('bool')` with value `True` for each aligned geometry disjoint to other.

gs.geom_equals(other, align=True) Returns a `Series` of `dytpe('bool')` with value `True` for each aligned geometry is approximately equal to other.

gs.intersects(other, align=True) Returns a `Series` of `dytpe('bool')` with value `True` for each aligned geometry that intersects other.

gs.touches(other, align=True) Returns a `Series` of `dytpe('bool')` with value `True` for each aligned geometry that touches other.

gs.within(other, align=True) Returns a `Series` of `dytpe('bool')` with value `True` for each aligned geometry that is within other.

Geometric Operations

```
From shapely.ops import linemerge,
polygonize
```

gs.boundary Returns a `GeoSeries` of lower dimensional objects representing each geometry's set-theoretic boundary.

gs.buffer(distance, resolution=16) Returns a `GeoSeries` of geometries representing all points within a given distance of each geometric object.

gs.centroid Returns a `GeoSeries` of points representing the centroid of each geometry.

gs.convex_hull() Returns a `GeoSeries` of geometries representing the convex hull of each geometry.

linemerge(lines) Returns a `LineString` or `MultiLineString` representing the merger of all contiguous elements of lines.

gs.representative_point() Returns a `GeoSeries` of (cheaply computed) points that are guaranteed to be within each geometry.

polygonize(lines) Returns an iterator over polygons constructed from the input `lines`.

gs.simplify(*args, **kwargs) Returns a `GeoSeries` containing a simplified representation of each geometry.

Geometric Creation (shapely)

```
From shapely.ops import triangulate,
voronoi_diagram

From shapely.geometry import Point,
MultiPoint, LineString,
MultiLineString, MultiPolygon
```

triangulate(geom, tolerance=0.0, edges=False) Returns a Delaunay triangulation of the vertices of geom.

voronoi_diagram(geom, envelope=None) Constructs a Voronoi diagram from the vertices of `geom`.

Point(coordinates) Creates a Point object from coordinate values or point tuple parameters.

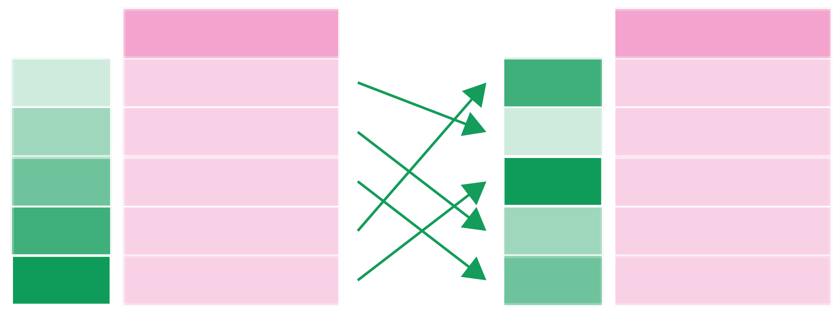
MultiPoint(points) Creates a MultiPoint object from a list of point tuples: `points`.

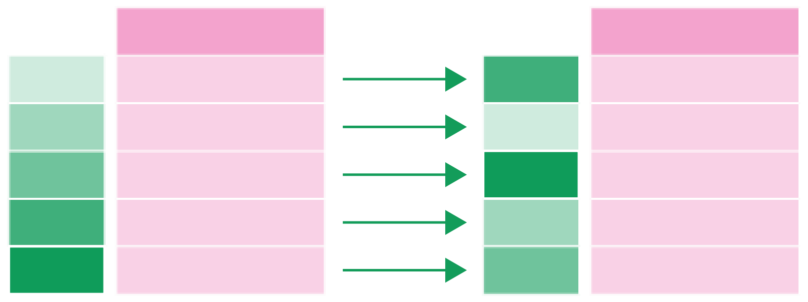
LineString(coordinates) Creates a `LineString` object from an ordered list of 2 or more point tuples: `coordinates`.

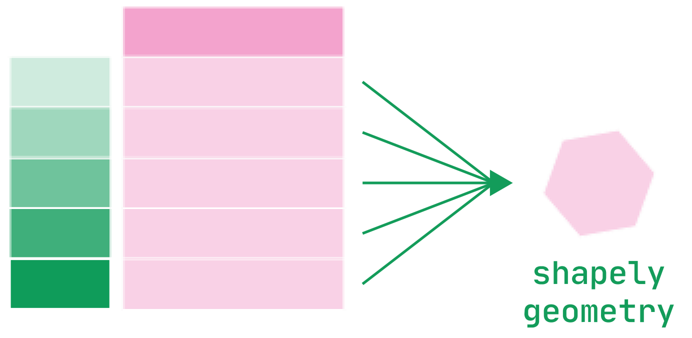
MultiLineString(lines) Creates a `MultiLineString` object from a sequence of line-like sequences or objects.

Polygon(shell, holes=None) Creates a `Polygon` object from an ordered sequence of point tuples.

MultiPolygon(polygons) Creates a `MultiPolygon` object from an unordered sequence of Polygon instances.


align=True


align=False



GeoPandas : : CHEAT SHEET

Geometry Operations

```
gs = geopandas.GeoSeries()

from shapely.ops import snap
```

gs.difference(other, align=True) Returns a `GeoSeries` of the points in each aligned geometry that are not in `other`.

gs.intersection(other, align=True) Returns a `GeoSeries` of the intersection of points in each aligned geometry with `other`.

gs.symmetric_difference(other, align=True) Returns a `GeoSeries` of the symmetric difference of points in each aligned geometry with `other`.

gs.union(other, align=True) Returns a `GeoSeries` of the union of points in each aligned geometry with `other`.

snap(geom1, geom2, tolerance) Snaps vertices in `geom1` to vertices in `geom2`, returning a copy. Input geometries are not modified.

Geometric Measurement

gs.area Returns a `Series` containing the area of each geometry in the `GeoSeries` expressed in the units of the CRS.

gs.distance(other, align=True) Returns a `Series` containing the distance to each aligned geometry in `other`.

gs.length Returns a `Series` containing the length of each geometry expressed in the units of the CRS.

Misc. Operations

```
import geopandas as gpd
```

gpd.GeoDataFrame(data=None, *args, geometry=None, crs=None, **kwargs) Creates a `GeoDataFrame` object from a `pandas.DataFrame` like object.

gs.astype(dtype, …) Cast a pandas object to a specified dtype dtype.

gs.crs The Coordinate Reference System (CRS) represented as a `pyproj.CRS` object.

GeoDataFrame.sjoin(gdf, how='inner', …)
gpd.sjoin(left_df, right_df, how='inner', …) Spatial join of two `GeoDataFrames`.

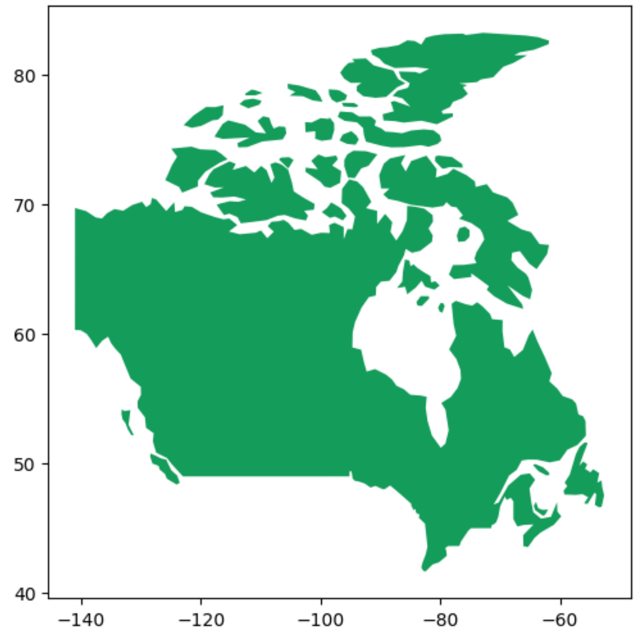
gs.sindex.nearest(geometry, return_all=True) Return the nearest geometry in the tree for each input geometry in `geometry`.

gpd.read_file(filename, bbox=None, mask=None, rows=None) Returns a `GeoDataFrame` from a file or URL.

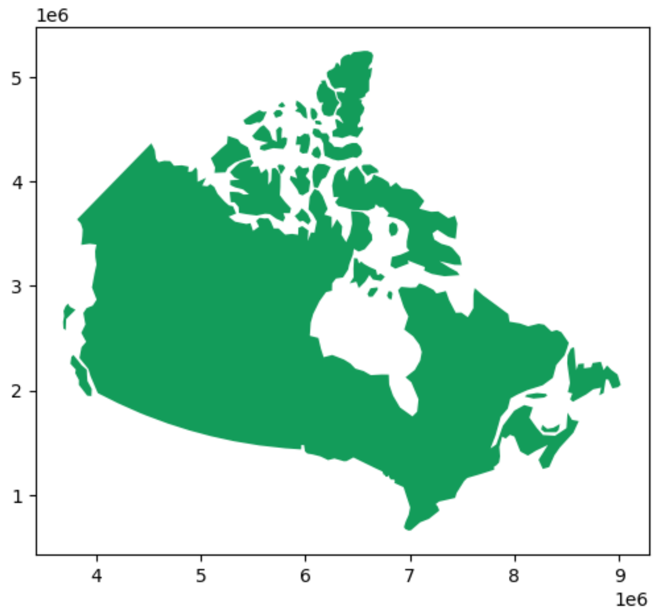
gs.to_crs(crs=None) Returns a `GeoSeries` with all geometries transformed to a new coordinate reference system.

CRS Examples

```
canada = gpd.read_file("canada.shp")
print(canada.crs)
# epsg:4326
canada.plot()
```



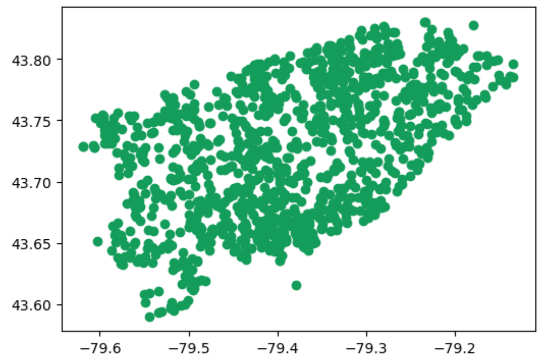
```
canada.to_crs("3347").plot()
```



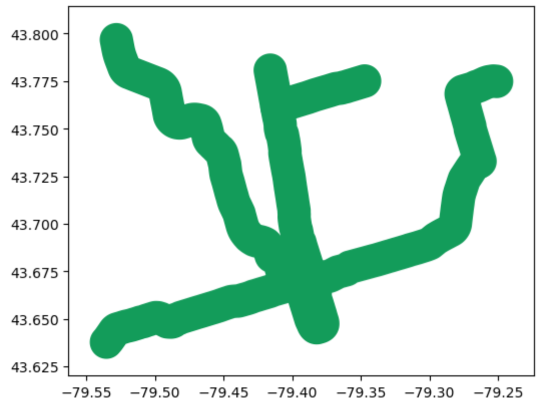
```
schools.plot()
```

```
subway_mask = subway.to_crs("3347") \
.buffer(1000).to_crs("4326")
subway_mask.plot()
```

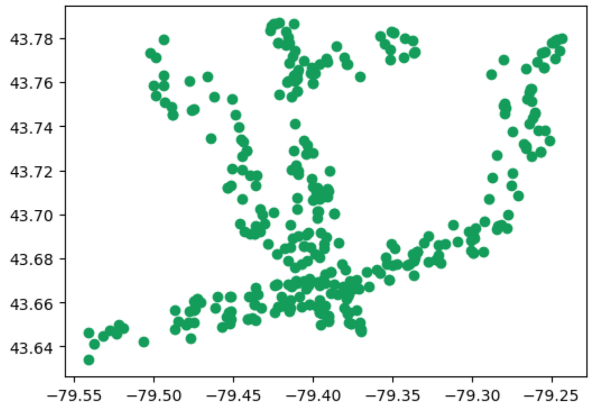
```
schools \
.intersection(subway_mask.unary_union) \
.plot()
```



43.80
43.75
43.70
43.65
43.60
−79.6
−79.5
−79.4
−79.3
−79.2



43.800
43.775
43.750
43.725
43.700
43.675
43.650
43.625
−79.55
−79.50
−79.45
−79.40
−79.35
−79.30
−79.25



43.78
43.76
43.74
43.72
43.70
43.68
43.66
43.64
−79.55
−79.50
−79.45
−79.40
−79.35
−79.30
−79.25