More OGR

More OGR

Overview

1. Geometries
2. Spatial References and Transformations
3. OGR Examples

Geometries

Geometries

- OGR handles geometries quite similarly to arcpy
- Geometry types
- Geometry methods
- How do you create a geometry in OGR?

Spatial References and Transformations

Spatial References and Transformations

- The OSR part of GDAL handles spatial references and transformations
- Based on the Proj.4 library for coordinate system and datum transformations
- For the purposes of this class, we will assume it is doing what we need it to do
    - **This is not a safe assumption in production**

How to access OSR:

```
from osgeo import osr
```

Spatial References and Transformations

- Spatial References are like in arcpy:
  - Objects
  - We can create them a lot of different ways
  - We can export them a lot of different ways

To create a spatial reference instance:

```
sr = osr.SpatialReference()
```

Spatial References and Transformations

- To assign the specific spatial reference to sr, we could use a number of different methods:
  - `sr.ImportFromWKT(wktstring)`
  - `sr.ImportFromProj4(proj4string)`
  - `sr.ImportFromEPSG(epsgID)`
  - `sr.ImportFromESRI(string_from_prj_file)`
  - `sr.ImportFromURL(url_to_a_spatial_reference)`
  - ...and more: see the docs

Spatial References and Transformations

- We can get the spatial reference out of sr with a number of different methods:
    - `sr.ExportToWkt()`
    - `sr.ExportToPrettyWkt()`
    - `sr.ExportToExportToPCI()`
    - `sr.ExportToUSGS()`
    - `sr.ExportToXML()`

Spatial References and Transformations

- Once we have two spatial reference instances, we can create a transformation between them
- For this we have the OSR CoordinateTransformation class

Spatial References and Transformations

```
wkt = 'GEOGCS["WGS 84",DATUM["WGS_1984",' + \
      'SPHEROID["WGS 84",6378137,298.257223563,' + \
      'AUTHORITY["EPSG","7030"]],AUTHORITY["EPSG","6326"]],' + \
      'PRIMEM["Greenwich",0,AUTHORITY["EPSG","8901"]],' + \
      'UNIT["degree",0.01745329251994328,AUTHORITY["EPSG","9122"]],' + \
      'AUTHORITY["EPSG","4326"]]'

insr = osr.SpatialReference()
insr.ImportFromWkt(wkt)

outsr = osr.SpatialReference()
outsr.ImportFromEPSG(2913)

transform = osr.CoordinateTransformation(insr, outsr)

# we can use the transformation with a geometry
geometry.Transform(transform)

# note: sometimes we can get away with not using a transform:
geometry.TransformTo(outsr)
# this is effectively just like geometry.projectAs() in arcpy

# transformations can also transform point coords
new_coords = transform.TransformPoint(-121.6743, 45.5345, [z is optional])
```

Spatial References and Transformations

- Using spatial references between ESRI and Proj.4 is not straightforward
- WKT to ESRI is not always WKT to OGR
- `spatialreference.MorphToESRI()` and `.MorphFromESRI()` supposed to handle this
  - Operation not always as one would expect
- Example:
  - 4326 WKT:
    ```
    GEOGCS["WGS 84",DATUM["WGS_1984", SPHEROID["WGS
    84",6378137,298.257223563,AUTHORITY["EPSG","7030"]],
    AUTHORITY["EPSG","6326"]],PRIMEM["Greenwich",0,AUTHORITY["EPSG","8901"]],
    UNIT["degree",0.01745329251994328,AUTHORITY["EPSG","9122"]],
    AUTHORITY["EPSG","4326"]]
    ```
  - 4326 ESRI WKT/.prj:
    ```
    GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",
    SPHEROID["WGS_1984",6378137,298.257223563]],
    PRIMEM["Greenwich",0],UNIT["Degree",0.01745329519943295]]
    ```