GEOG 432/832: Programming, Scripting, and Automation for GIS

Unit 07.01: Spatial queries

Dr. Bitterman

Today's schedule

- Open discussion
- Slides, discussion and exercises
- For next class

Open discussion

How's lab 3 going?

Spatial queries

What's a spatial query?

Implementation

```
arcpy.management.SelectLayerByLocation(in_layer, {overlap_type}, {select_features}, {search_distance},
{selection_type}, {invert_spatial_relationship})
```

- **in_layer**: what to select from
- **overlap_type**: spatial relationship to be evaluated
- **select_features**: the features in in_layer will be selected based on their relationship to the features from *this* layer or feature class.
- **search_distance**: the specified distance that will be searched
- **selection_type**: how the selection will be applied to the input
- **invert_spatial_relationship**: whether the spatial relationship evaluation result will be used or the opposite result

Supported spatial query types (an excerpt)

- INTERSECT
- WITHIN_A_DISTANCE
- CONTAINS
- COMPLETELY_CONTAINS
- BOUNDARY_TOUCHES
- HAVE_THEIR_CENTER_IN
- And more (see https://pro.arcgis.com/en/pro-app/latest/tool-reference/data-management/select-layer-by-location.htm)

Let's give it a shot

Make a new project and get `week07inclass.zip` from GitHub

Some new syntax and control structures for today

- The `try` block lets you test a block of code for errors.
- The `except` block lets you handle the error.
- The `else` block lets you execute code when there is no error.
- The `finally` block lets you execute code, regardless of the result of the try- and except blocks.

Example to attempt

What do you expect to happen?

```
try:
  print(x)

except:
  print("An exception occurred")
```

What happened?

Our first goal: find the 303d streams in Lancaster County

Let's break it down line by line

```
workspace_string = "C:\\Users\\pjbitterman\\Dropbox\\GEOG432\\week07\\week07data"
arcpy.env.workspace = workspace_string

# Get the streams and lancaster County feature classes
streams = "Streams_303_d_.shp"
lancaster = "lancaster_county.shp"

try:
    # Apply a selection to the streams layer
    intersectLayer = arcpy.management.SelectLayerByLocation(streams, 'INTERSECT', lancaster)

except:
    print(arcpy.GetMessages())
```

What happened?

Let's do some work with the selection

```
arcpy.env.workspace = "C:\\Users\\pjbitterman\\Dropbox\\GEOG432\\week07\\week07data"

# Get the streams and lancaster County feature classes
streams = "Streams_303_d_.shp"
lancaster = "lancaster_county.shp"
nameField = "Waterbody_"

try:
    # Apply a selection to the streams layer
    intersectLayer = arcpy.management.SelectLayerByLocation(streams, 'INTERSECT', lancaster)

        # Open a search cursor on the US States layer
    with arcpy.da.SearchCursor(intersectLayer, (nameField)) as cursor:
        for row in cursor:
            # Print the name of all the streams in the selection
            print (row[0])

except:
    print(arcpy.GetMessages())

finally:
    # Clean up feature layers and cursor
    arcpy.Delete_management(intersectLayer)
    del cursor
```

What happened?

Let's talk GI*Science*

- Why not just use a *clip*?
- How is a *clip* different from our *instersection* spatial selection?

Let's get more complicated

Attribute query + a spatial query

Do we have a volunteer to walk us through the code?

```python
# Get the streams and lancaster County feature classes
streams = "Streams_303_d_.shp"
neCounties = "County_Boundaries-_Census"
nameField = "Waterbody_"
countyFieldName = "NAME10"
countyName = "Lancaster"

try:
    # build SQL clause and do an attribute query
    whereClause = countyFieldName + " = '" + countyName + "'"
    selectionCountyLayer = arcpy.SelectLayerByAttribute_management(neCounties, 'NEW_SELECTION', whereClause)

    # Apply a selection to the streams layer
    intersectLayer = arcpy.management.SelectLayerByLocation(streams, 'INTERSECT', selectionCountyLayer)

    # Open a search cursor on the US States layer
    with arcpy.da.SearchCursor(intersectLayer, (nameField)) as cursor:
        for row in cursor:
            # Print the name of all the states in the selection
            print (row[0])

except:
    print (arcpy.GetMessages())

finally:
    # Clean up feature layers and cursor
    arcpy.Delete_management(intersectLayer)
    arcpy.Delete_management(selectionCountyLayer)
    del cursor
```

A new volunteer to the board!

As a class, your task is as follows:

For all state parks within 2 miles of a municipal boundary, figure out the size (in acres) of the parks and what county they are in. Print the output like:

```
Arbor Lodge SHP is 56.788745 acres and in Otoe County
```

Optional bonus points (only try once you've completed above):

- (easier) Format the park size such that it only prints 2 decimal places
- (harder) Include what muncipality they are nearest to
- (even harder) Print whether the state parks are within 10 miles of a 303d stream

For next class

- Read Chapter 9 this week