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

Unit 05.02: Listing data

Dr. Bitterman

Today's schedule

- Open discussion
- Discussion and exercises
- For next class

Open discussion

Why do we automate our geospatial analysis with Python (or other tools?)

Batch processing

- Running the same process/task over multiple datasets
- But the first step? What datasets do we have available?
- Useful to list data

Working with a list

- Iterating over a list requires a loop, typically a *for loop*
- ArcPy has many list functions
  - `ListFields()`
  - `ListIndexes()`
  - `ListDatasets()`
  - `ListFeatureClasses()`
  - `ListFiles()`
  - `ListRasters()`
  - `ListTables()`
  - `ListWorkspaces()`
  - `ListVersions()`

Listing functions are generally similar

- Either work directly for your workspace, OR
- Take an argument for an input dataset
- We've used a few

`ListFeatureClasses()` syntax:

```
arcpy.ListFeatureClasses({wild_card}, {feature_type}, {feature_dataset})
```

- How many parameters?
- Which are required? Which are optional?

Listing all feature classes in a workspace

```
import arcpy
arcpy.env.workspace = "C:/Data"
fclist = arcpy.ListFeatureClasses()
print(fclist)
```

Output:

```
['floodzone.shp', 'roads.shp', 'streams.shp', 'wetlands.shp',
'zipcodes.shp']
```

But we can do things a bit more intelligently

```
arcpy.ListFeatureClasses({wild_card}, {feature_type}, {feature_dataset})
```

- The `{wild_card}` parameter limts the list by name!
- for example, `fclist = arcpy.ListFeatureClasses("w*")`

- The `{feature_type}` parameter limits by type
- `fclist = arcpy.ListFeatureClasses("", "point")`

And we can also very simply list the fields

- First, what's a "field"?

Syntax:

```
arcpy.ListFields(dataset, {wild_card}, {field_type})
```

- what do you think these parameters refer to?

Let's try something different (in-class exercise)

- Create a new project (or open an existing one) that has ALL of the week03inclass data in it. Shapefiles AND GeoDatabase

Task 1: try listing the fields of dataset in your workspace

Task(s) 2: List feature classes in workspace (we've done some of this)

1. List the feature classes in the workspace
2. List the feature classes that start with "S"
3. List the feature classes of type "point"

But how do we deal with nested locations?

Walk the file system!

Try this:

```
mywalk = arcpy.da.Walk("C:/the_path_to_your_data")
for dirpath, dirnames, filenames in mywalk:
    print(dirpath, dirnames, filenames)
```

Let's break it down. What is the code doing?

Try it with your week03data directory (or one that has both sahpefiles and a geodatabase)

An opportunity for practice

Tasks:

1. Clip all features in the geodatabase to Lancaster County BUT NOT Lancaster County itself. Append "_lc" to the end so you know which feature classes are the output
2. Once you have those outputs, buffer ONLY the files of type "point". Append "_buff" to the feature class name

For next class:

- Read Chapters 6 & 7 for this week
- Project proposal is due Friday!
- Friday <- workday