Python for ArcGIS

Day 2 - Focus on ArcGIS

Presented by
Matthew Collier
president of
Attribute Q, LLC

Agenda – Day 2

- Python Review
- Modelbuilder Review
- Basic Geoprocessing Scripts with arcpy
- Debug/Handling Errors from Python and ArcGIS
- Using Python to Work with Geo Files and Data
- Field Calculations with Python in ArcGIS 10
- Python on the ArcGIS 10 Commandline
- Creating a Python-based ArcTool

Review: Python 1/5

- Class Discussion
  - Variables (naming? types? dynamic?)
  - Control Structures (loops, conditionals)
  - Anything from previous day?
  - http://www.python.org/dev/peps/pep-0008/
- Requested code walk-throughs?

Review: Model Builder 2/5

- Worked Example
  - Find the area in Oklahoma that is more than X kilometers from a mesonet site.
- Build a new toolbox
  - R-click free space in ArcToolbox
  - Choose "Add Toolbox..."
  - Choose a meaningful path
  - Now, choose "New Toolbox" icon
  - Name it "z-pyWorkshop" and open it

Review: Model Builder 3/5

- Create a new model
  - R-click your new toolbox, choose "New Model..."
  - R-click on new model ***in the toolbox TOC*** and open the properties.
  - Under "General" tab, fill in the name and label, and select "Store relative path names"
  - Under "Environments" tab, select workspace, then current and scratch workspaces.
  - Hit the "Values..." button and set them appropriately.

Review: Model Builder 4/5

- Build the model
  - Drag and drop mesonet from TOC to model
  - Drag and drop buffer tool from ArcToolbox into the model
  - Select the "Connect" icon and connect the data to the tool as input features.
  - Double click tool in the model to fill in parameters: buffer of 30km, dissolve ALL.
  - Click OK

Review: Model Builder 5/5

- Continue building the model
  - Drag and drop Oklahoma, and the erase tool into the model
  - Connect Oklahoma as input feature, and the buffer we just set up as an erase feature
  - Double click output and rename meaningfully
- Run Model
  - Choose "Model | Run Entire Model..."
  - Check OK when complete, R-click the output and "Add to Display"

Search for “model builder” at http://desktop.arcgis.com/en/desktop/

BREAK

Basic GeoScripts 1/2

- Create an idea of the structure *before* you start coding. The more explicit your ideas, the better!
  - Research existing codebase: web, books, gurus, institutional code, ArcToolbox code
  - Write out complex algorithms
  - Write pseudocode
  - Visually organize in Model Builder
  - "An hour now is worth 10 hours later"

Basic GeoScripts 2/2

- Let's develop the mesonet buffer script from scratch based on:
  - The visualisation from our model, and
  - pseudocode named mesoBuffer.py that I've provided in the class folder
- Return to the model
  - export it as mesoModel.py
  - we'll compare and dissect
  - take notes on these pages, or add comments to the scripts for future reference

Search for “creating new python script” at http://desktop.arcgis.com/en/desktop/

BREAK

Debugging 1/2

- What types of syntax errors have we seen?
- What types of runtime errors can we expect?
- Semantic errors?
- Simple techniques:
  - Comments
  - Print statements
  - Make intermediate results
  - Search internet for particular error message
  - Fresh set of eyes, or a walk
- Finally, read the source below for more ideas

openbookproject.net/thinkcs/python/english2e/app_a.html

Handling Runtime Errors 2/2

- Important to maintain control of script
  - A crash on a single automated operation at 2am could leave much work undone at 8am!
  - Recovering from error allows script to continue
  - Use the Python try-except-else-finally structure
- Best to keep errors for investigation
  - Use traceback module to capture
  - Use print statements and log files to report
- We head back to the mesonet buffer idea and handle potential runtime errors by using hints in the mesoBuffWithError.py script

BREAK

Files: Listing Them 1/3

- arcpy.ListFeatureClasses( )
- Arguments
  - Wild card
  - Feature type
  - Feature dataset
- As name suggests, a Python list is returned
- Let's experiment in the ArcGIS commandline

Search for “ListFeatureClasses()” at http://desktop.arcgis.com/en/desktop/

Files: Describing Them 2/3

- arcpy.Describe( )
- The function's single argument is context sensitive. See help listed at end of this slide for much more information.
- Let's try passing the function a shapefile from the ArcGIS commandline
- Given your experiences, how would you find the spatial reference information?

Search for “Describe()” at http://desktop.arcgis.com/en/desktop/

Files: Working with Them 3/3

- Worked Class Example: Shapefiles to GDB
- We will "copy" code snippets from ArcGIS online help into a single script, and then whittle it down to a working geoprocess.
- We will need the following functions:
  - arcpy.CreateFileGDB_management( )
  - arcpy.CreateFeatureDataset_management( )
  - arcpy.CopyFeatures_management( )
- A starting script is ready for us to begin at cp2GDB.py in the class folder

Search terms for help at http://desktop.arcgis.com/en/desktop/

BREAK

Data: Query 1/4

- Some useful arcpy members:
  - MakeFeatureLayer_management( )
  - MakeTableView_management( )
  - SelectLayerByAttribute_management( )
  - SelectLayerByLocation_management( )
  - GetCount_management( )
  - Exists_management( )
  - Delete_management( )
  - CopyFeatures_management( )
  - CopyRows_management( )
  - arcpy.env.overwriteoutput = True

Data: Query 2/4

- Basic SQL dialect for file GDB:
  - Fields are double quoted
  - Strings are single quoted,
  - Case sensitive
  - Wildcard is % character
  - Comparisons include: >, <, >=, <=, =, <>
  - Expressions with: *, /, +, -
  - Combination: AND, OR
  - Others: NOT, LIKE, IS NULL
- See below for more discussion, especially in considering different dialects.

Search for “sql reference guide” at http://desktop.arcgis.com/en/desktop/

Data: Query 3/4

- Let's generate some example SQL statements as a class...
- Very useful to build them manually in ArcGIS in the wizards under the "Selection" menu
- We'll do these as a class:
  - query1 - a string query
  - query2 - a numerical comparison query
  - query3 - a categorical match query

Data: Query 4/4

- We'll now build a script to query and write a new shapefile on selected records as a class.
- Find SelectLayerByAttribute_management( ) by searching for it in the ArcGIS online help.
- After reviewing the help, copy the stand-alone script example to a new Python file. We'll clean it up as a class and run it.
- *NOTE: Be careful that all file locks are released between runs! Crashed or open, ArcGIS or Python, processes can leave locks on files.*

Search for “SelectLayerByAttribute()” at http://desktop.arcgis.com/en/desktop/

BREAK

Data: Direct 1/2

- Cursors (foiled again)
  - Create an object to hold a set of database records
  - Can take a SQL query and other parameters
- There are three types:
  - arcpy.SearchCursor( )
  - arcpy.InsertCursor( )
  - arcpy.UpdateCursor( )

Search terms for help at http://desktop.arcgis.com/en/desktop/

Data: Direct 2/2

- Three worked class exercises. For each:
  - Find the ArcGIS help page for the right cursor
  - Paste in the code samples and modify to achieve the following scenarios. Save each when done:

1. Search for incorrectly coded month strings in OK, and write records out to a CSV file.
2. Insert a new county into OK, named Mordor. Use arcpy.AddField_management( ) to create a nullable boolean field named "Walkable".
3. Update the "Walkable" attribute to be True for all but Mordor.

BREAK

ArcGIS: Field Calculations 1/1

- Introduction to the data sets
  - Goofy data in the attribute tables
  - With built-in errors from the field techs
- LOTS of Field Calculations
  - R-click on the attribute table field name
  - Choose "Field Calculator..."
  - Check the Python box
  - Check the "Show Codeblock" box
  - Follow along, and "Save..." each function separately as a ".cal" file.

BREAK

ArcGIS: The Commandline 1/4

- You can access the commandline by:
  - Clicking "Geoprocessing | Python" on main menu
  - Or, by clicking the "Python window" icon on the standard tool bar.
  - All of Python is available at the prompt
  - Geoprocessing tools become methods to the arcpy module.
- Now, explore the help pane, and the R-click menu with me...

ArcGIS: The Commandline 2/4

- Executing an arpy method:
  - Start typing "arcpy." to see a HUGE list
  - Either scroll with mouse, or keep typing letters to filter the list
  - We'll type in ListFeatureClasses
  - Note the appearance of help to guide you
  - From the help we see that nothing is required, so we open and close with parentheses and hit enter to execute it in the most general sense.
- Anything? It's likely an empty list.

ArcGIS: The Commandline 3/4

- Let's set the workspace and try again:
  - arcpy.env.workspace = "your working folder"
  - Hit the up arrow to retrieve our previous call to arcpy.ListFeatureClasses( ), and hit enter
  - Now we should have a list of strings representing the available feature classes, or shapefiles.
- The commandline is that simple and can be used to test arcpy methodology before placing it in an edited script.
- ALL the tools are there!

ArcGIS: The Commandline 4/4

- Running a pre-written script:
  - R-click the command pane and select "Load..."
  - Load the provided helloWorld.py script.
  - The code should populate the commandlines.
  - Hit enter.
  - A new greetings.txt file should appear in the same class folder as the script.
- We'll see more commandline, and scripts tomorrow!

Search for “python window” at http://desktop.arcgis.com/en/desktop/

BREAK

Making Tools: Prepare Script 1/2

- There are two ways to pass information from ArcGIS to our script:
- arcpy.GetParameterAsText( )
  - the zero-th element is the first parameter
  - no limit to length of input parameter
  - Ex: workspace = arcpy.GetParameterAsText(0)
- sys.argv
  - One-th element is the first parameter
  - 1024 character limit to parameter length
  - Ex: workspace = sys.argv[1]

Making Tools: Create a GUI 2/2

- Find our new toolbox that we created in ArcToolbox, then follow along:
  - R-click toolbox, select "Add | Script..."
  - Name it, label it
  - Give it path, and filename for new tool
  - Add "Display Name" and "Data Type" for the values we pass
  - Click finish, and try it out!
- See link below for much more detailed information

Search for “creating tools” at http://desktop.arcgis.com/en/desktop/

Review

Questions?

Comments