# Automating Geoprocessing tasks With ModelBuilder and Python

## Small Area Health Insurance Estimates Project (SAHIE) 2014


**Mehrshad Nourani**
**Senior GIS Analyst/Developer**
**merhrshad.nourani@census.gov**

U.S. Census Bureau, 4600 Silver Hill Road,
Washington DC, 20233

The Coleman Group, Inc.
Corporate Headquarters
10210 Greenbelt Road, Suite 720
Greenbelt, Maryland 20706

Atlanta Office
1170 peachtree street, suite 1200
Atlanta, Georgia 30309


The Coleman Group, Inc.
*Inspired Thinking... Intelligent Solutions.*

United States Census Bureau
*Measuring America*


ESRI International User Conference
July 14-18 | San Diego

# Small Area Health Insurance Estimates (SAHIE)

**Automate the process by which the mapping services are consumed by SAHIE interactive mapping tool (API)**

# Who we are and what we do

**Census Bureau**

- **Demographic Internet Staff (DIS)**
- **Dissemination of Public Products**

**The Coleman Group, Inc.**

- **IT/Cyber Security Service provider to Federal Agencies**

# Background

## Our Goal

To automate GIS workflow in order to process data, create mapping documents, and publish mapping services for SAHIE interactive web mapping tool (API)

## Our (geo)Process

The process of creating map services had been in place for a few years. It involves data edits and transformation, and several geoprocessing tasks, including creating map services. All tasks are currently performed manually, and last nearly 2 days to complete. The lengthy and repetitive nature of geoprocessing tasks compelled us to automate our  workflow so that we can increase process efficiency and reduce human errors.

## Our Approach

1. Use ModelBuilder to test and document our workflow; and
2. Use Python scripting to complete automating the process.

# Work Environment

1. SAS native data format
2. Excel(tab delimited .txt) – MS 2010
3. ArcGIS for desktop 10.1
4. ArcGIS for server 10.1
5. Python 2.7.2
6. Citrix XenApps
7. MS Windows 7

# Tools

1. ArcGIS 10.1, ModelBuilder
2. Python scripting

# Why ModelBuilder, or Python script?

**ModelBuilder**

1. Process simplification – No guess work!
2. Visualizing workflow
3. Standardizing workflow
4. Increase efficiency and reduce human error
5. Proof of concept/Test hypotheses
6. Maintaining institutional memory
7. Interacting with our clients – showing ModelBuilder instead of codes
8. Starting a script with a model – need editing
9. Documentation: the ModelBuilder can be used to document and publish our process

# Why ModelBuilder, or Python script?

**Python Script**

- Scripting is more efficient
  1. No need to run ArcGIS: use stand-alone script
  2. Simple edits
  3. Easy QA/QC
  4. Do more with scripting
  5. Support community

# Our Workflow

# Workflow Model

File Geodatabase

1. Processing data extracted from SAS/Excel and exported to fGDB

2. Verifying and change detection – QA/QC

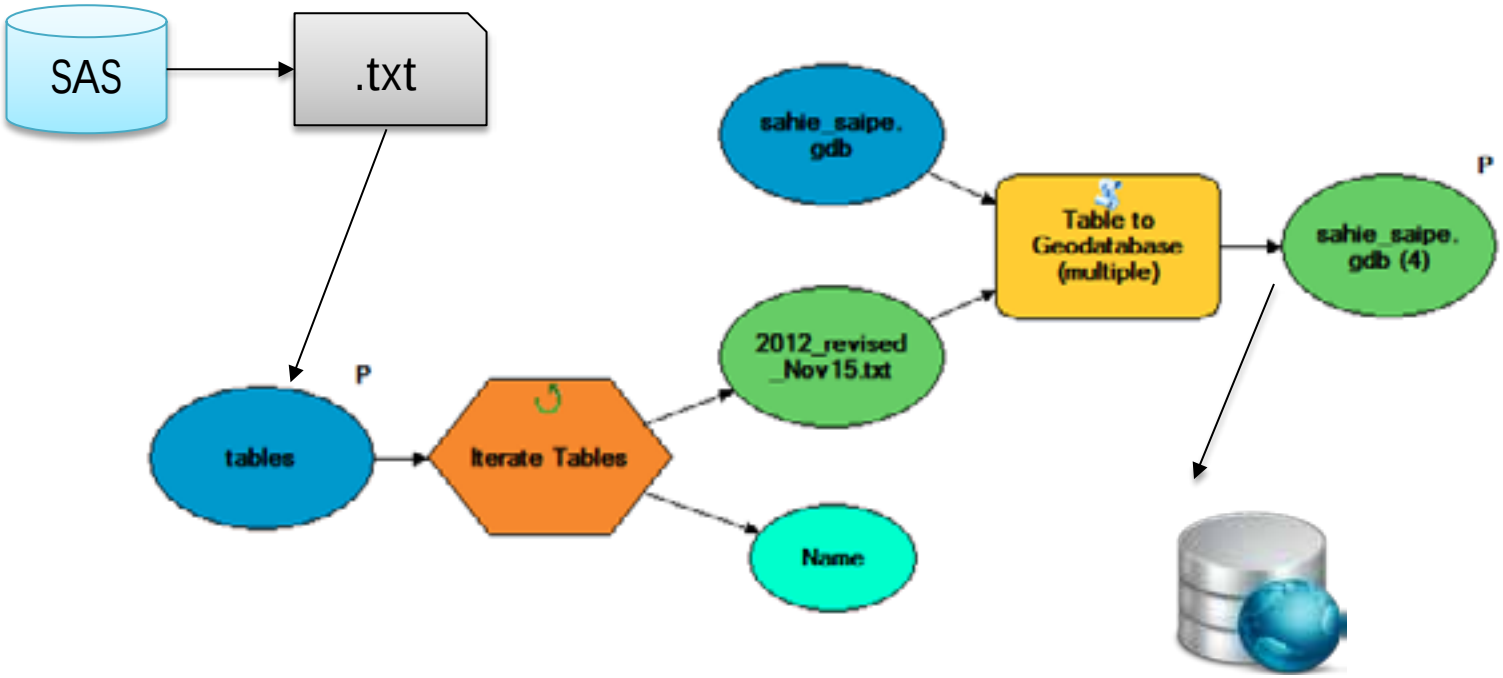
3. Geoprocessing feature classes

4. Creating, analyzing, and publishing 7 map services to ArcGIS Server à Flex API



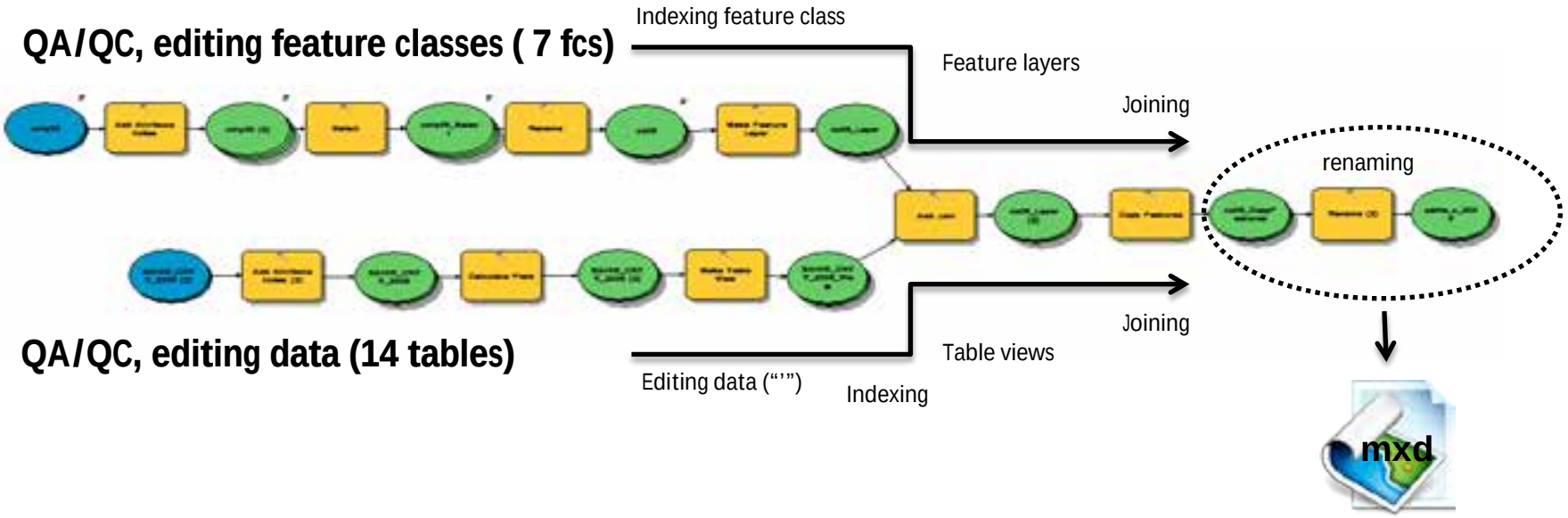

# 1. Extracting, Transforming, Loading (ETL)

1. Importing tabular data (SAS à .txt à fGDB)
2. QA/QC: verifying number and fields, rows, cells; text vs. integer;
3. Exporting to fGDB



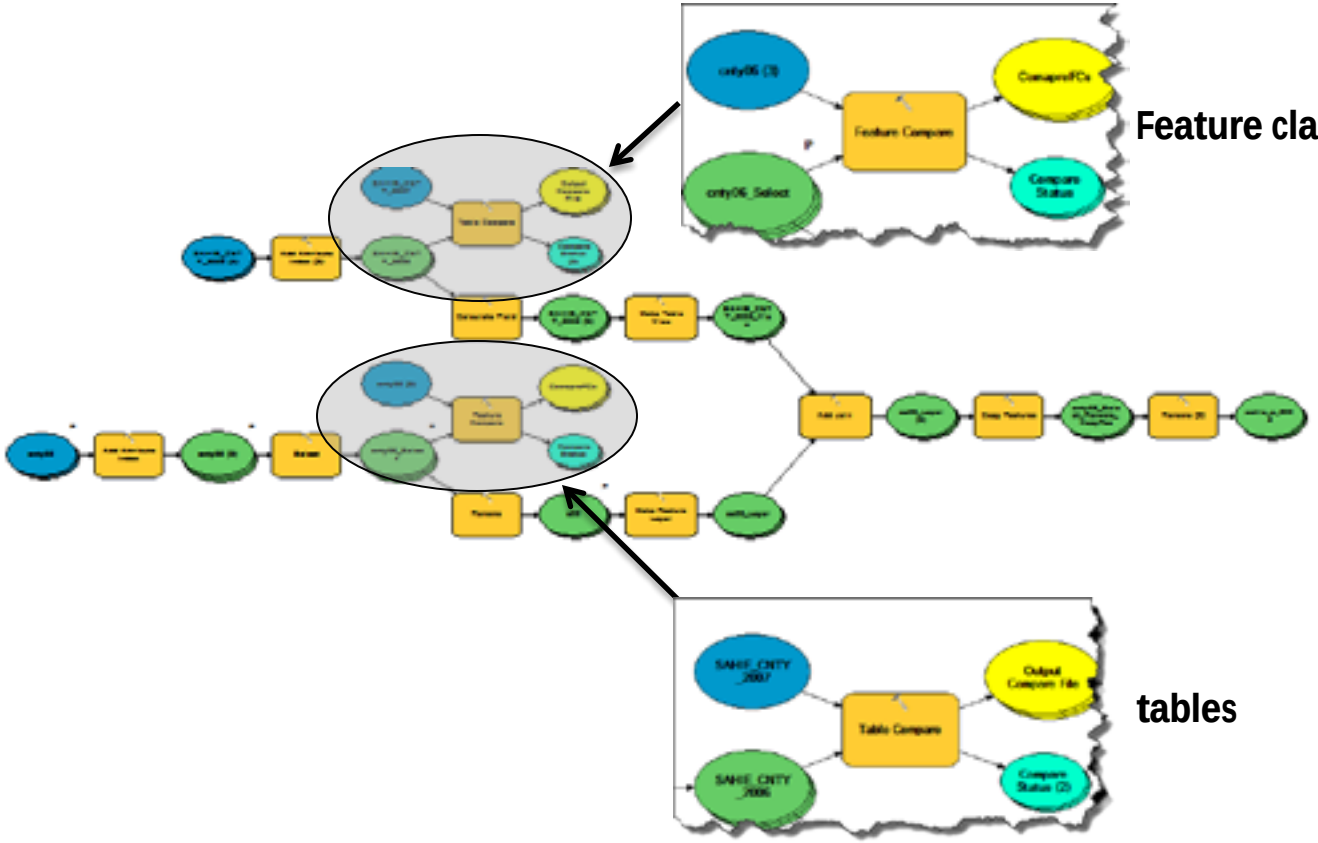
# 2. Editing data and Feature Classes

1. **Reformatting fields in tables : SAS à .txt à fGDB (text to integer!!)=> add “quotation marks”**
2. **Indexing tables and feature classes for faster processing**
3. **Creating feature layers and table views**
4. **AddJoin à create new fcs**
5. **Rename fcs = must match Flex API config**
6. **Creating mxds**



# 3. Verifying and change-detection

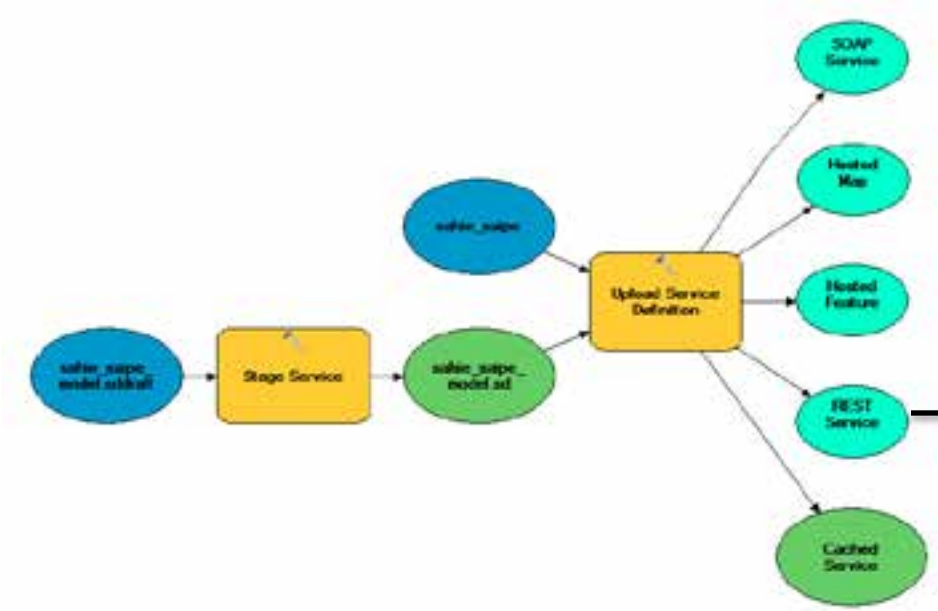
1. Comparing feature classes by year and determining changes in geography
2. Comparing annual data tables and determining changes in number of fields, etc.

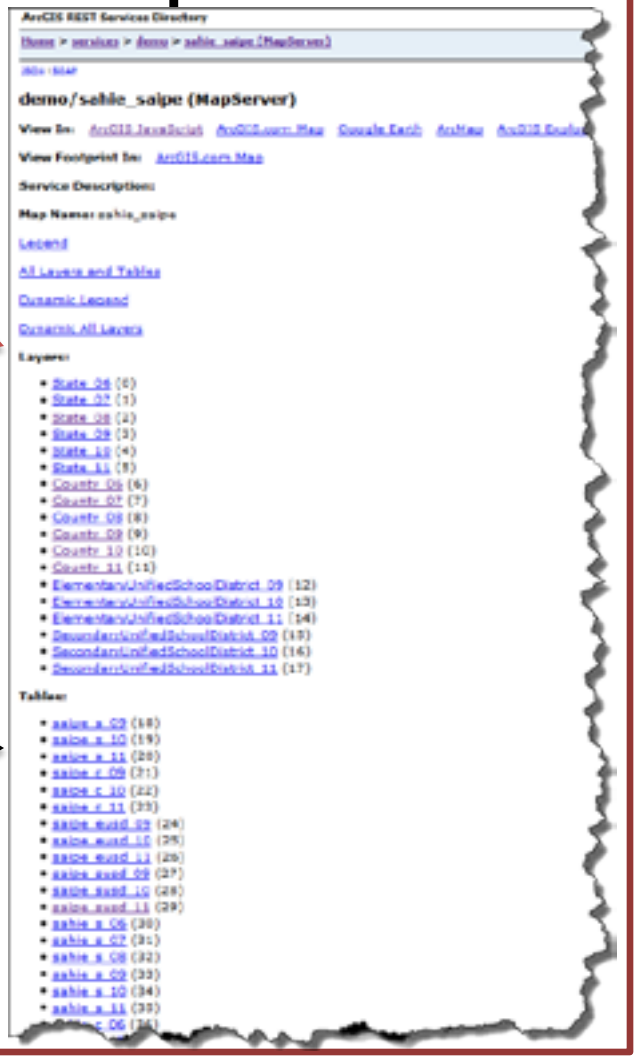


# 4. Creating and publishing map services

Publishing mxds as map services for consumption by Flex API through REST services

**Publishing mxds as map services (SD): all layers must be in the order the Flex API reads and accesses data**

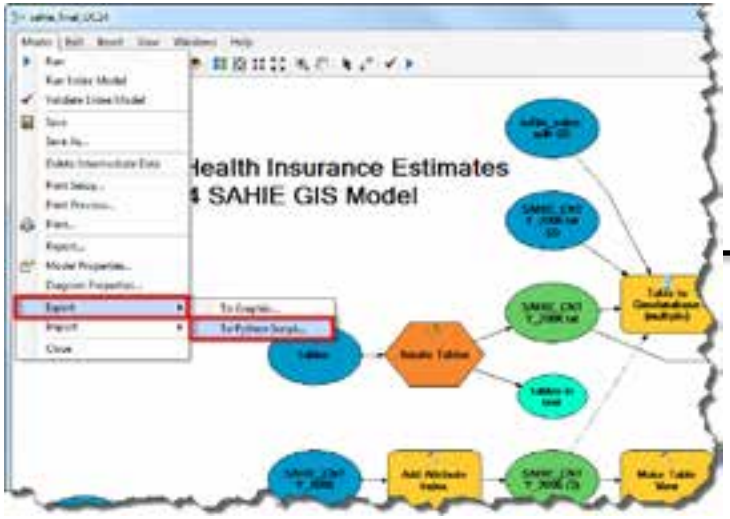


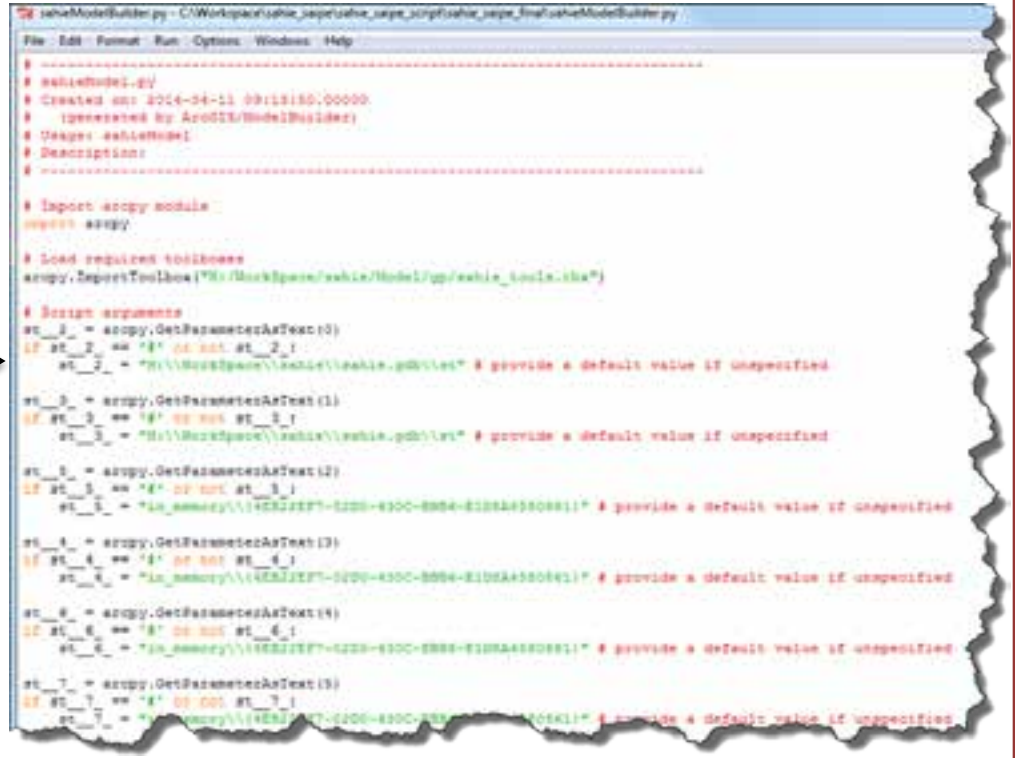



# The First Step

# From ModelBuilder to Python

1. Export your model from ModelBuilder to Python script
2. Review Python script in text editor
3. Can you run this script?



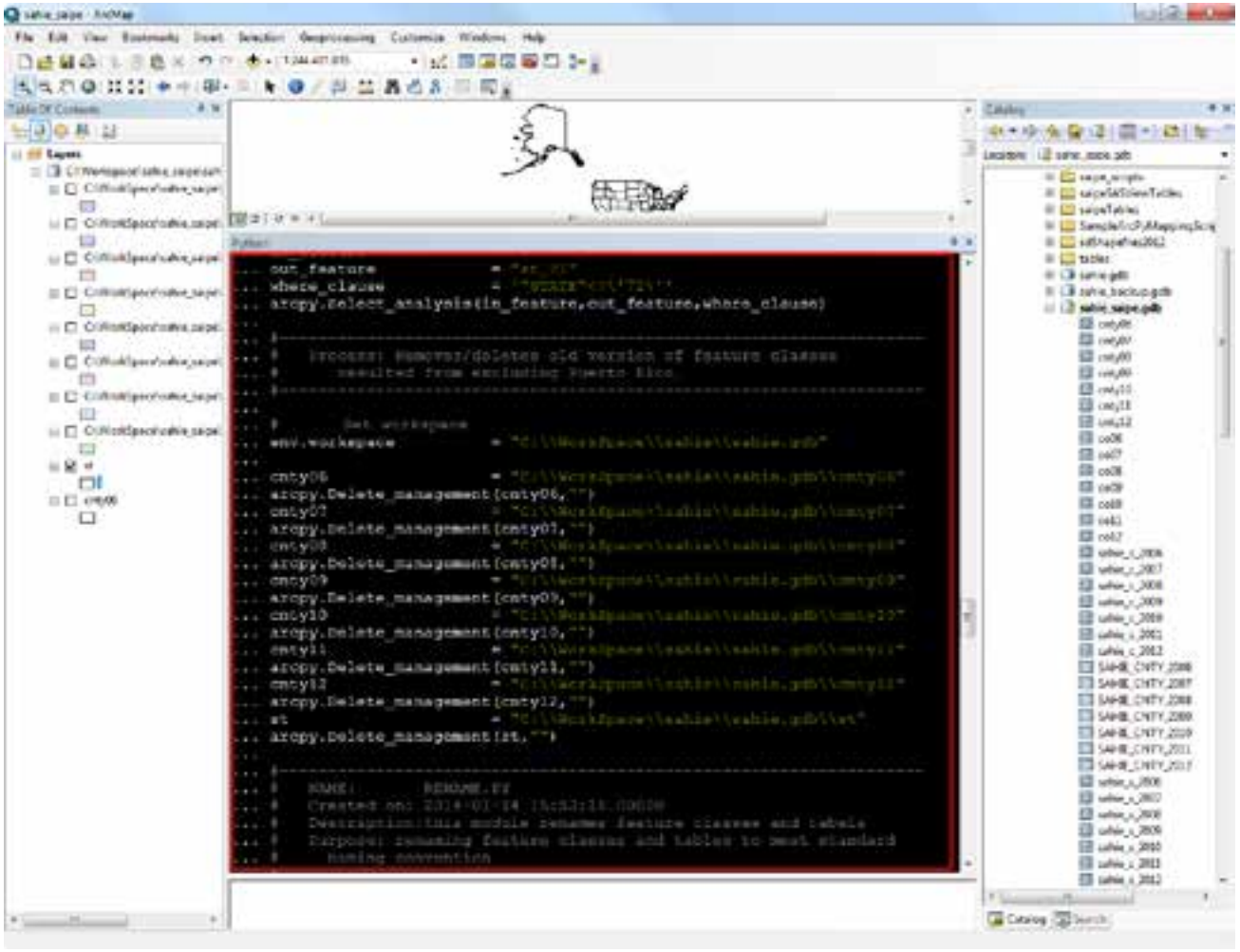



# Review your script in text editor…

**Before being able to run your script, you may have to make some modifications to the script that is generated through ModelBuilder. For example:**

1. If you are processing a series of fcs using several tools, the ModelBuilder repeats the same script individually for all the fcs – 450 lines of code were generated by the ModelBuilder. (runs)
2. "iterators" do not work as Python script (from ModelBuilder) (modify)
3. If you have imported tools from Toolbox, the script may have difficulty locating and running it! (modify)
4. Change the gp name to default toolbox name
5. Check the paths to your workspace (gdb, etc.)

# ...run your script in ArcGIS/Python



# Do you know Python?
# Rewrite/edit ModelBuilder script

Script imported from ModelBuilder ~ 43 lines

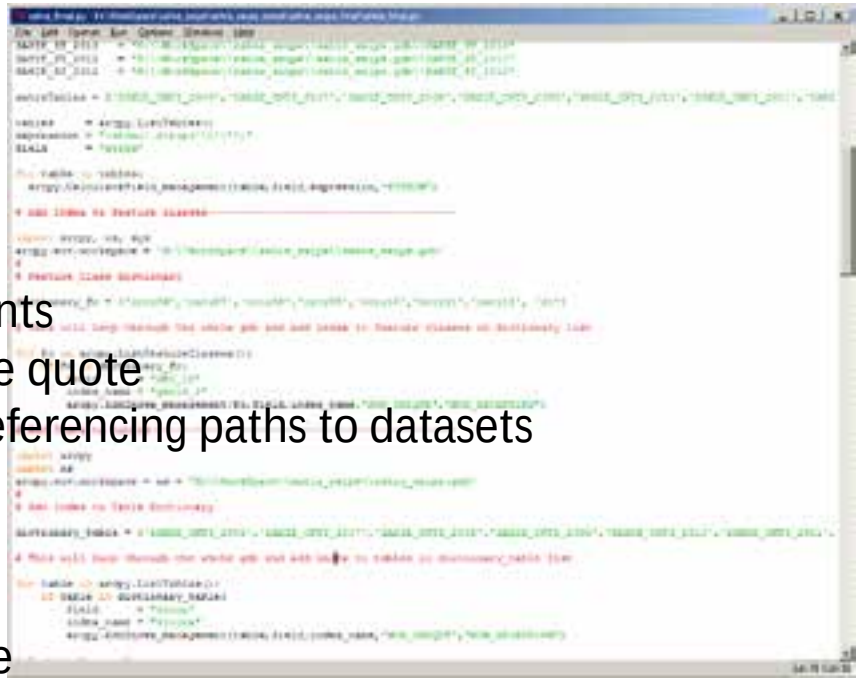
Edited in Python ~ 10 lines

```
import arcpy
import os
from arcpy import env
env.workspace = u'C:\\WorkSpace\\sable_saipe\\tables'
tables = arcpy.ListTables()
outLocation = u'C:\\WorkSpace\\sable_saipe\\sable_saipe.gdb'
try:
   print"Importing tables to gdb:  " + outLocation
   arcpy.TableToGeodatabase_conversion(tables,outLocation)
except:
   print arcpy.GetMessages()
```

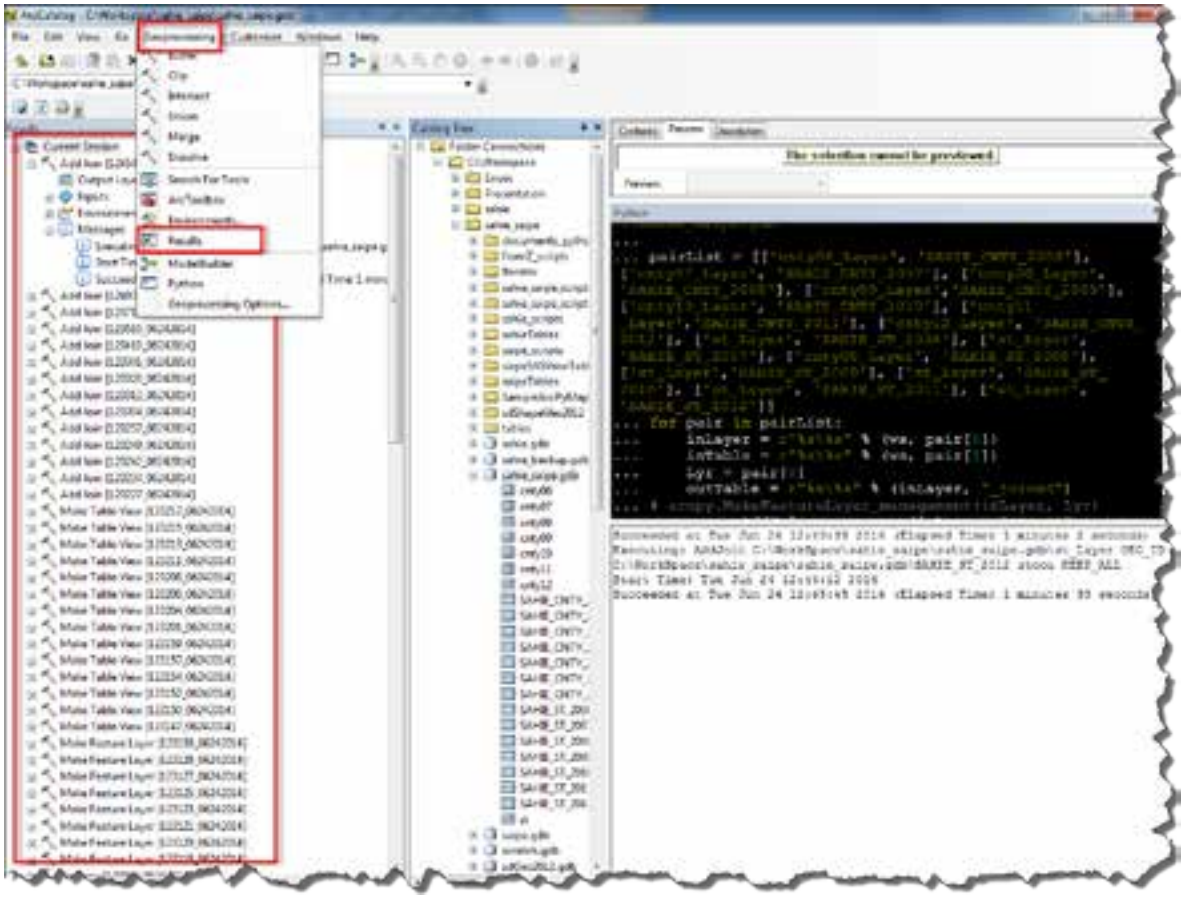
Example: importing multiple data tables

# Test your Python script

1. Verify your tasks were completed
2. Use geoprocessing results (only if run from ArcGIS environment)
3. Look for empty fcs
4. Task may be completed but no results generated
5. Look for errors
    a. Python error
        I. Typos
        II. Python is case sensitive
        III.Indentation
        IV.Colons for compound statements
        V. Mixing single quote and double quote
        VI.Using single backslash when referencing paths to datasets
    b. ArcGIS error
        I. fc/table does not exist
        II. Incorrect paths
        III. Previous task did not complete



# Review/test ModelBuilder or script results in Geoprocessing



# ModelBuilder tips that worked for me

1. Set your work environment from geoprocessing menu (ArcMap/Catalog)
2. Use relative path
3. Start with smaller models
4. Integrate smaller models to create, or expand to a larger model
5. Use Geoprocessing menu to review and monitor your model
6. Become active in the community and ask questions in forums

# Python tips that worked for me

1. Use ModelBuilder to create your first Python script
2. Rewrite Python by using Toolbox examples in tool's Help:
   a. Iterating over a list or a string
   b. for statement
   c. List
   d. Dictionary à

```
>>> import arcpy
import os
arcpy.env.workspace = ws = 'C:\\WorkSpace\\sahie_saipe\\sahie_saipe.gdb'
# Add Index to Table Dictionary
dictionary_table = ['SAHIE_CNTY_2006','SAHIE_CNTY_2007','SAHIE_CNTY_2008','
SAHIE_CNTY_2009','SAHIE_CNTY_2010','SAHIE_CNTY_2011','SAHIE_CNTY_2012','SAH
IE_ST_2006','SAHIE_ST_2007','SAHIE_ST_2008','SAHIE_ST_2009','SAHIE_ST_2010'
,'SAHIE_ST_2011','SAHIE_ST_2012']
for table in arcpy.ListTables():
    if table in dictionary_table:
        field      = "stcou"
        index_name = "stcou65"
        arcpy.AddIndex_management(table,field,index_name,"NON_UNIQUE","NON_
ASCENDING")
```

3. Use online Python resources: ArcGIS Resources, GIS StackExchange… enhance your script
4. Join user communities/forums: ArcGIS Resource, GIS StackExchange, Python…

# Lessons learned
# ModelBuilder and Python

1. ModelBuilder = capable – limited efficiency
2. ModelBuilder + Python = capable, efficient, and powerful
3. Best resource: your system tool box = most likely, there is already a script tool for your need
4. Modify existing tools to create new tools for your needs
5. Build and integrate smaller models to create a larger model
   a. If computing resources are limited, you can run smaller models to complete your tasks
6. Convert your model to an interactive Python tool for future use

# Work environment challenges

**Limitations of building a complex model in Desktop Virtual Environment** - Using a network location to save and process = slow processing time

1. System latency (hours vs minutes)
2. Writing to a common area folder = conflict with other users
3. Desktop alternative

# ArcGIS/Python Learning Resources

1. Learn Python: abundantly free online resources
2. ArcGIS Tools Help examples and modify for your needs
3. ArcGIS Resources: videos, forums, blogs
4. Python programming YouTube videos
5. Python.org

# Questions

**MEHRSHAD NOURANI**
**SENIOR GIS ANALYST/DEVELOPER**
**THE COLEMAN GROUP, INC.**
**U.S. CENSUS BUREAU, WASHINGTON DC**
**merhrshad.nourani@census.gov**