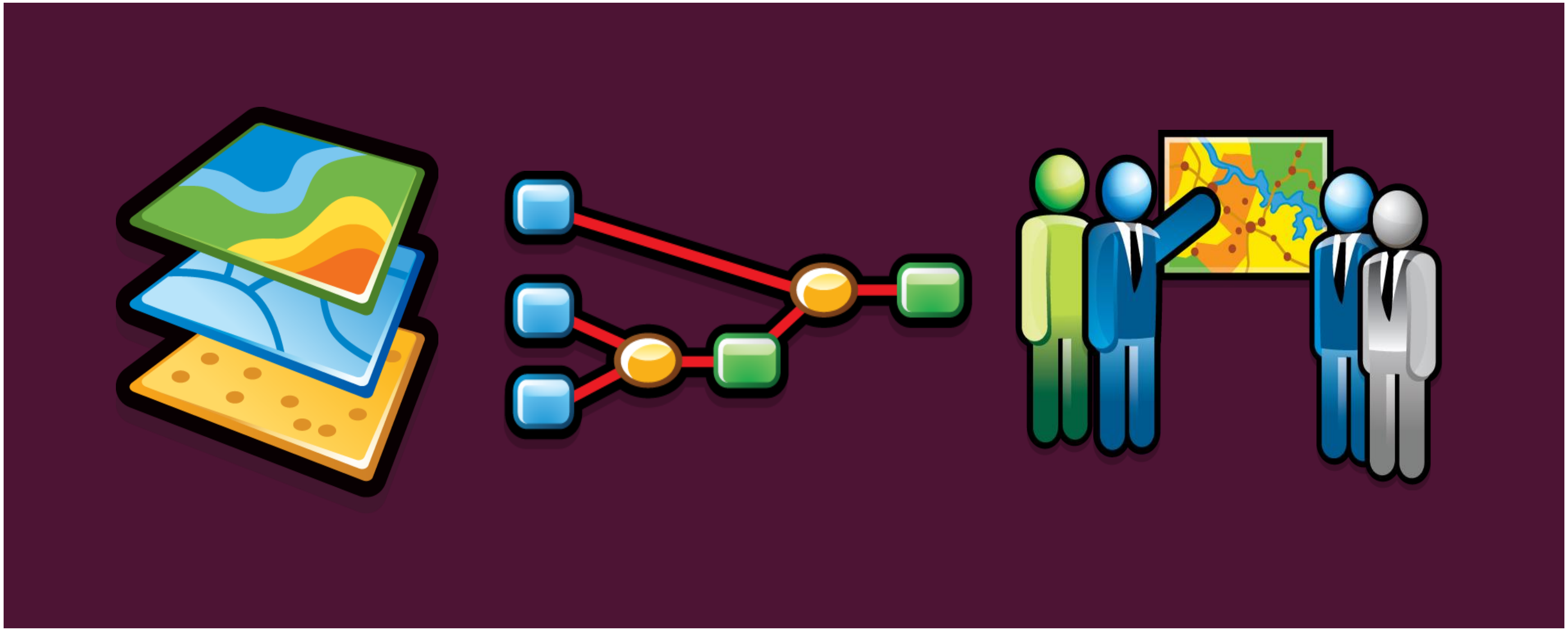
Spatial Data Analysis with Python

Song Gao

Email: sgao@geog.ucsb.edu



UCSB BROOM CENTER

Goals of Workshop

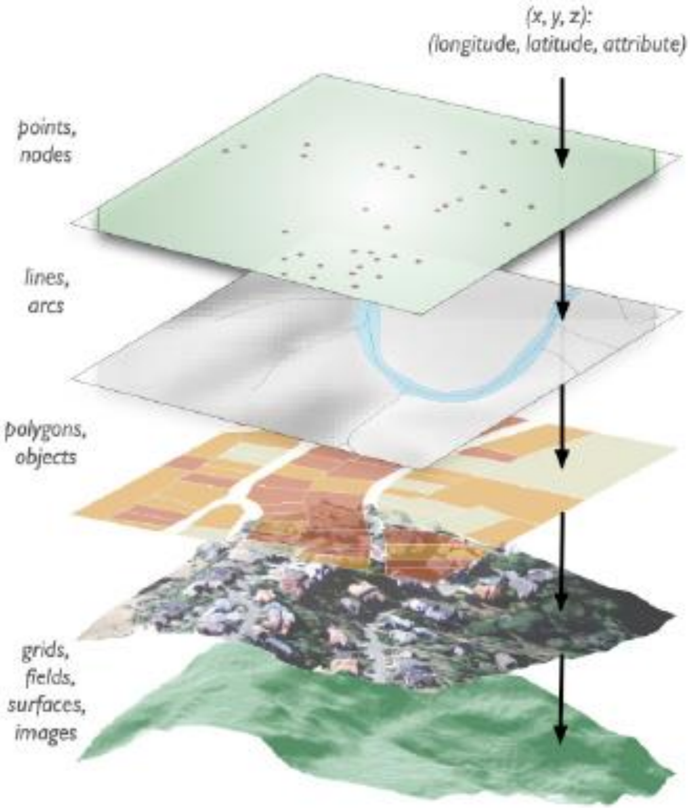
1. Introduction to the batch processing in ArcGIS;
2. Introduce the Python scripting language and its application in ArcGIS;
3. Become familiar with several methods for writing, and running geoprocessing scripts using Python;
4. Apply Python scripts to automate a GIS workflow;
5. Solve your own domain problem using Python.

1. Introduction

- Primary Data Types
- *vector*: point, line, polygon
- *raster*: continuous (e.g. elevation) or
- discrete surfaces (e.g. land use type)

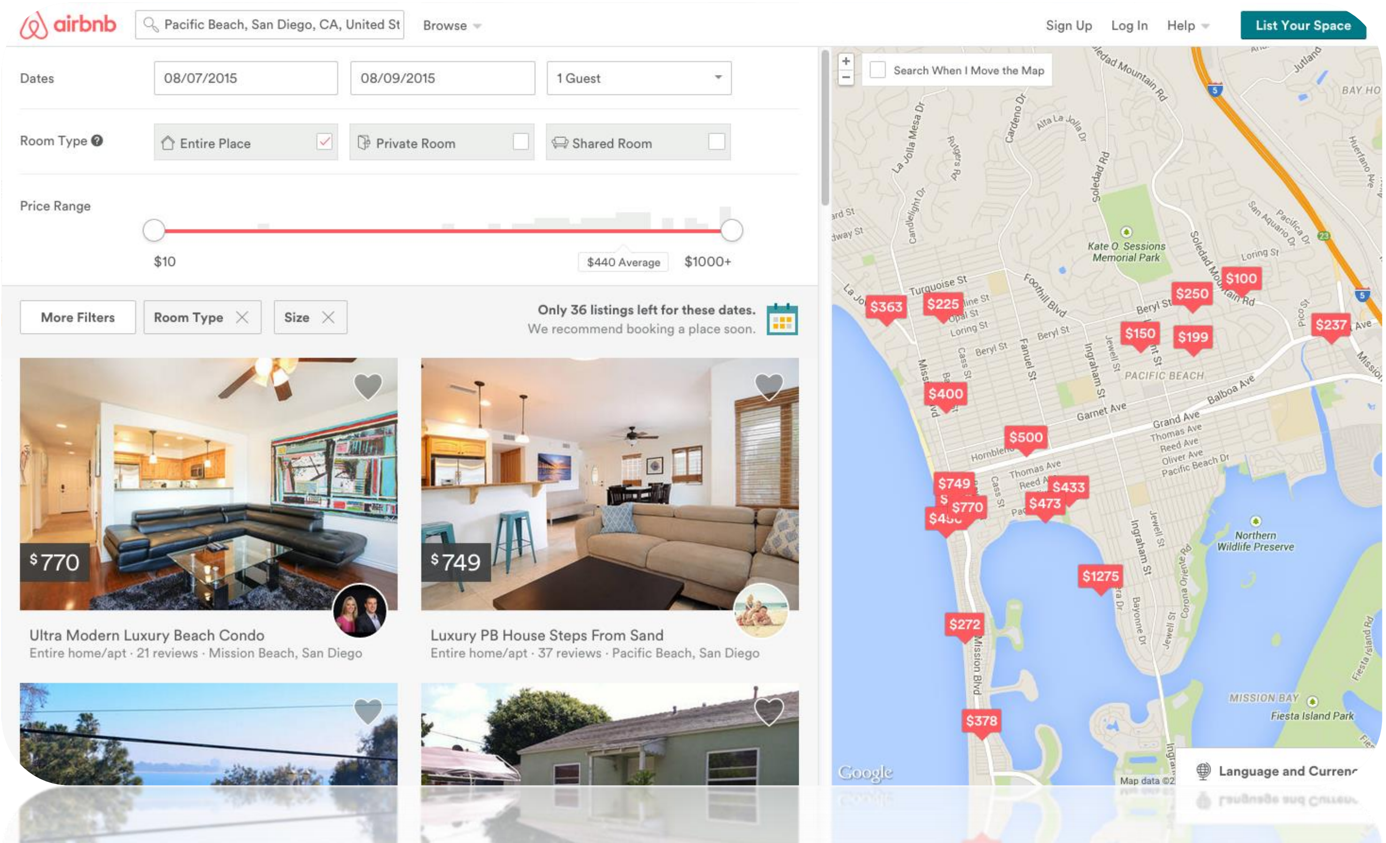
Common Data Storage Formats

- *vector*: shapefile, geodatabase feature
- tables (.dbf, .xlsx), KML, GeoJSON
- *raster*: ASCII, GeoTIFF, JPEG2000



Why Spatial?

- Discussion: What kinds of spatial variables can you think of for determining the house prices in cities?



Pacific Beach, San Diego, CA, United St

Dates: 08/07/2015 — 08/09/2015 — 1 Guest

Room Type: Entire Place ☑ Private Room □ Shared Room □

Price Range: $10 — $440 Average — $1000+

Only 36 listings left for these dates.
We recommend booking a place soon.

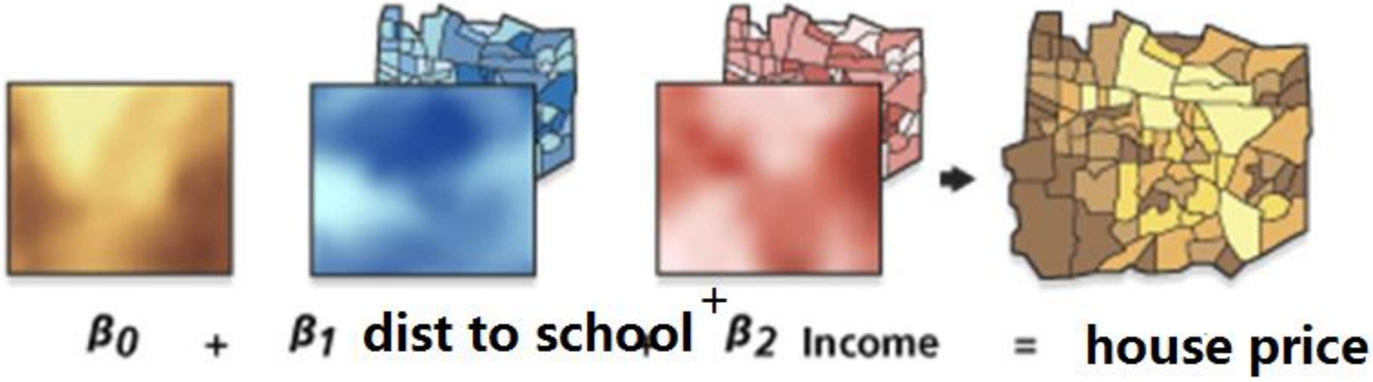
$770 Ultra Modern Luxury Beach Condo
Entire home/apt · 21 reviews · Mission Beach, San Diego

$749 Luxury PB House Steps From Sand
Entire home/apt · 37 reviews · Pacific Beach, San Diego

Geographically Weighted Regression (GWR)

- Discussion: What kinds of spatial variables can you think of for determining the house prices in cities?

A local form of linear regression used to model spatially varying relationships
Fotheringham, Stewart A., Chris Brunsdon, and Martin Charlton. *Geographically Weighted Regression: the analysis of spatially varying relationships*. John Wiley & Sons, 2002.



Geographically Weighted Regression (GWR)

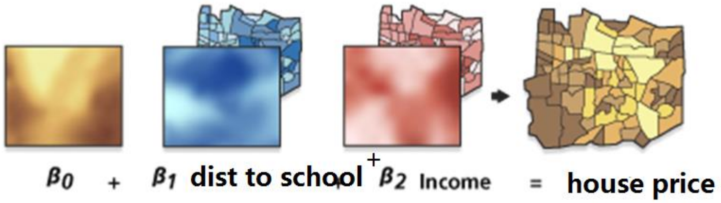
Global model

$$y = \alpha + \beta X + \varepsilon$$

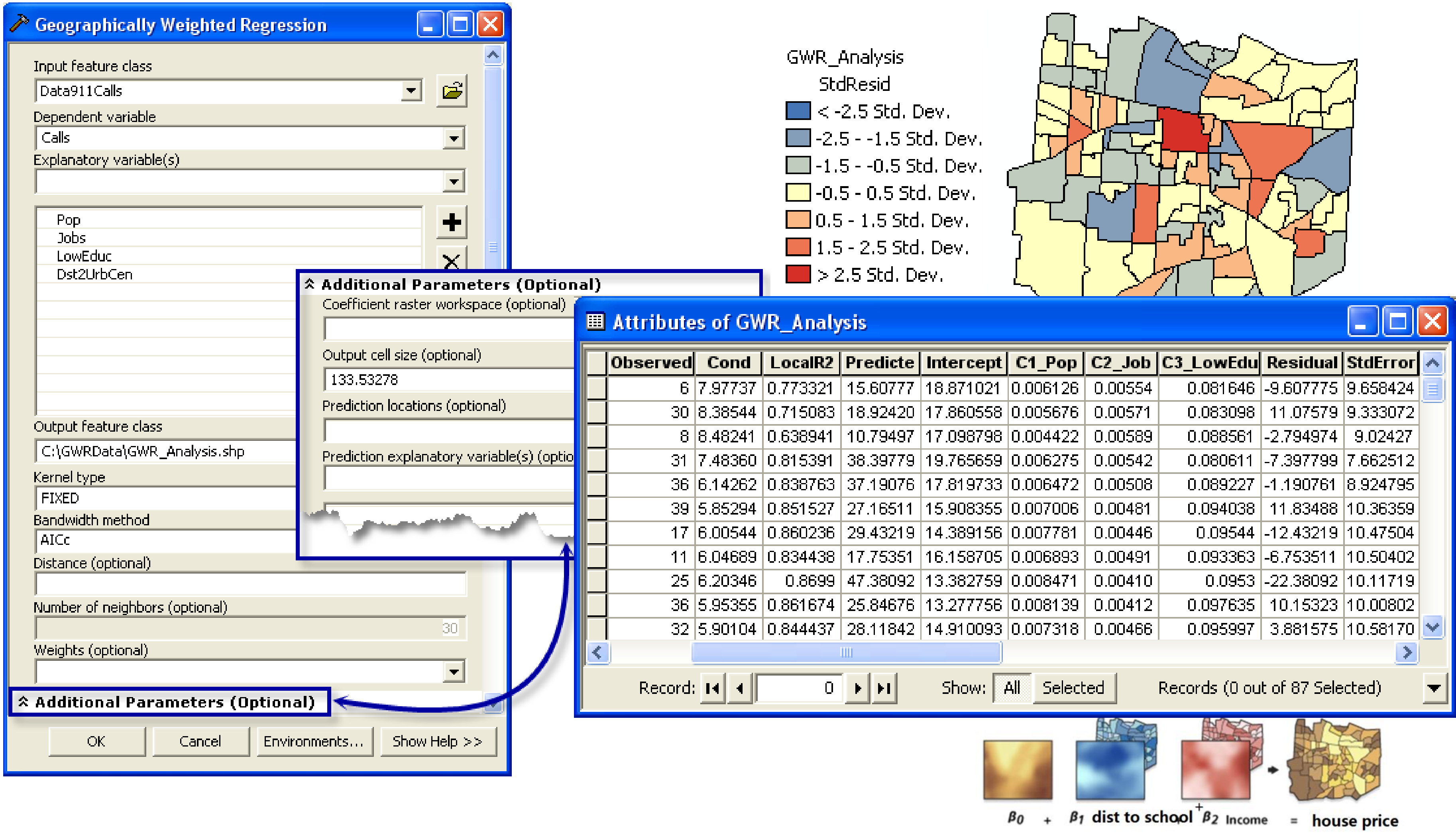
becomes

$$y_i = \alpha_i + \beta_i X + \varepsilon_i$$

Where *i* indicates that there is a set of coefficients estimated for every observation in our data set



Geographically Weighted Regression (GWR)



Geographically Weighted Regression

Input feature class
Data911Calls

Dependent variable
Calls

Explanatory variable(s)

Pop
Jobs
LowEduc
Dst2UrbCen

Output feature class
C:\GWRData\GWR_Analysis.shp

Kernel type
FIXED

Bandwidth method
AICc

Distance (optional)

Number of neighbors (optional)
30

Weights (optional)

Additional Parameters (Optional)

Coefficient raster workspace (optional)

Output cell size (optional)
133.53278

Prediction locations (optional)

Prediction explanatory variable(s) (optio

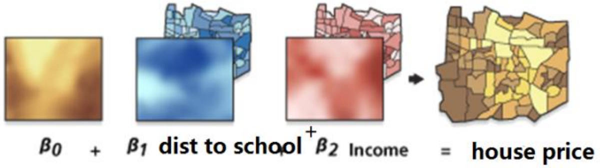
Attributes of GWR_Analysis

Observed	Cond	LocalR2	Predicte	Intercept	C1_Pop	C2_Job	C3_LowEdu	Residual	StdError
6	7.97737	0.773321	15.60777	18.871021	0.006126	0.00554	0.081646	-9.607775	9.658424
30	8.38544	0.715083	18.92420	17.860558	0.005676	0.00571	0.083098	11.07579	9.333072
8	8.48241	0.638941	10.79497	17.098798	0.004422	0.00589	0.088561	-2.794974	9.02427
31	7.48360	0.815391	38.39779	19.765659	0.006275	0.00542	0.080611	-7.397799	7.662512
36	6.14262	0.838763	37.19076	17.819733	0.006472	0.00508	0.089227	-1.190761	8.924795
39	5.85294	0.851527	27.16511	15.908355	0.007006	0.00481	0.094038	11.83488	10.36359
17	6.00544	0.860236	29.43219	14.389156	0.007781	0.00446	0.09544	-12.43219	10.47504
11	6.04689	0.834438	17.75351	16.158705	0.006893	0.00491	0.093363	-6.753511	10.50402
25	6.20346	0.8699	47.38092	13.382759	0.008471	0.00410	0.0953	-22.38092	10.11719
36	5.95355	0.861674	25.84676	13.277756	0.008139	0.00412	0.097635	10.15323	10.00802
32	5.90104	0.844437	28.11842	14.910093	0.007318	0.00466	0.095997	3.881575	10.58170

Record: 0 Show: All Selected Records (0 out of 87 Selected)

GWR using Python

- GeographicallyWeightedRegression Example (Python Window)
- The following Python Window script demonstrates how to use the GeographicallyWeightedRegression tool.
- import arcpy arcpy.env.workspace = "c:/data" ***arcpy.GeographicallyWeightedRegression_stats***("CallData.shp", "Calls","BUS_COUNT;RENTROCC00;NoHSDip", "CallsGWR.shp", "ADAPTIVE", "BANDWIDTH PARAMETER", "#", "25", "#","CoefRasters", "135", "PredictionPoints", "#", "GWRCallPredictions.shp")



Geographically Weighted Regression (GWR)

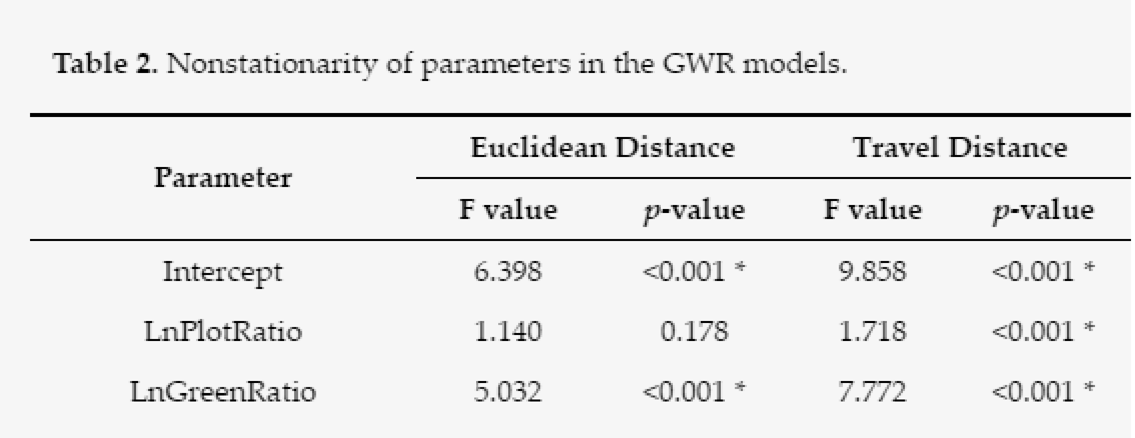
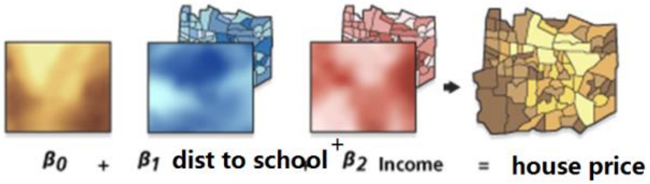


Table 2. Nonstationarity of parameters in the GWR models.

Parameter	Euclidean Distance		Travel Distance	
	F value	*p*-value	F value	*p*-value
Intercept	6.398	<0.001 *	9.858	<0.001 *
LnPlotRatio	1.140	0.178	1.718	<0.001 *
LnGreenRatio	5.032	<0.001 *	7.772	<0.001 *
LnFloorArea				
LnPropertyFee				
$LnEucD_{PriSchool}$				
$LnEucD_{ShoppingMall}$				
Age				

* The statistically significa



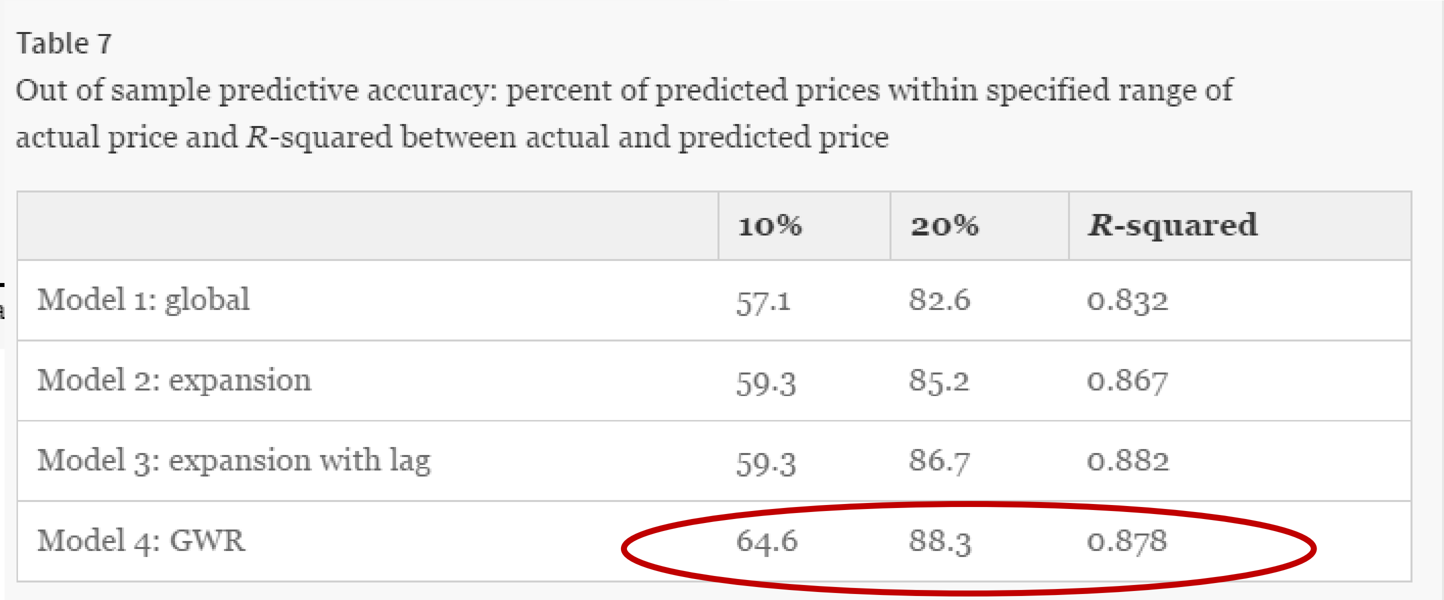


Table 7

Out of sample predictive accuracy: percent of predicted prices within specified range of actual price and *R*-squared between actual and predicted price

	10%	20%	*R*-squared
Model 1: global	57.1	82.6	0.832
Model 2: expansion	59.3	85.2	0.867
Model 3: expansion with lag	59.3	86.7	0.882
Model 4: GWR	64.6	88.3	0.878

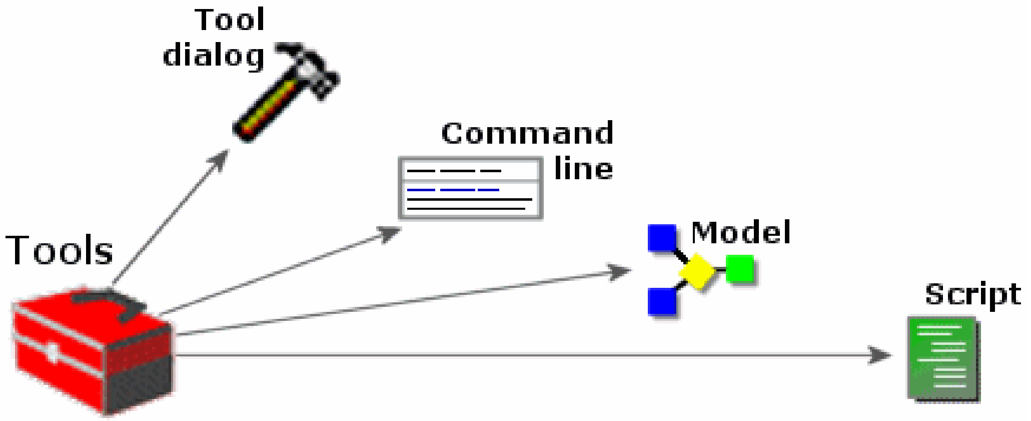
Types of Models in GIS (by function)

- Descriptive models – patterns
- Change models – before and after
- Impact models – what happens
- Explanatory models – process influence
- Predictive models – what will be like

Geoprocessing in GIS

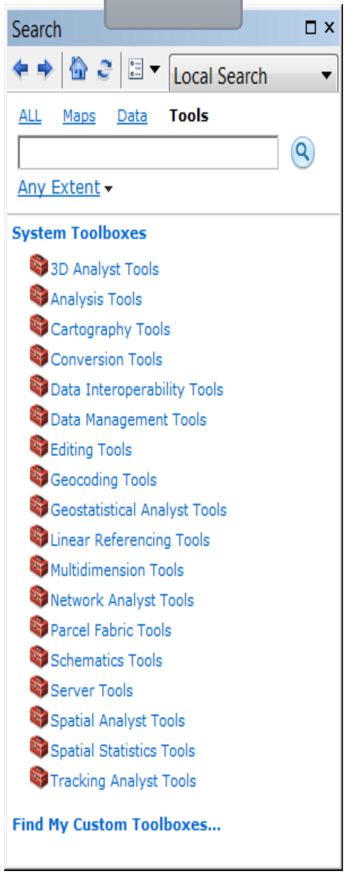
- building data processing chains in GIS:

data -> operations -> output



Geoprocessing in GIS

Search

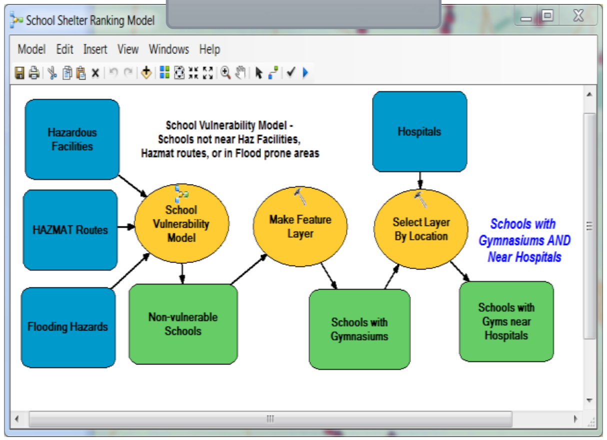


System Toolboxes

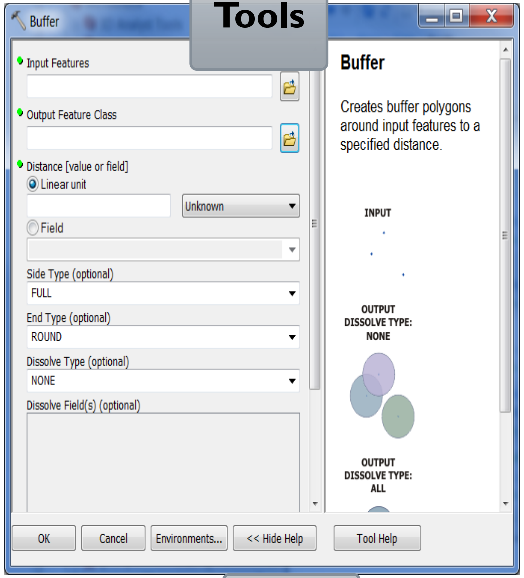
- 3D Analyst Tools
- Analysis Tools
- Cartography Tools
- Conversion Tools
- Data Interoperability Tools
- Data Management Tools
- Editing Tools
- Geocoding Tools
- Geostatistical Analyst Tools
- Linear Referencing Tools
- Multidimension Tools
- Network Analyst Tools
- Parcel Fabric Tools
- Schematics Tools
- Server Tools
- Spatial Analyst Tools
- Spatial Statistics Tools
- Tracking Analyst Tools

Find My Custom Toolboxes...

ModelBuilder


School Shelter Ranking Model
Model
Edit
Insert
View
Windows
Help
Hazardous Facilities
HAZMAT Routes
Flooding Hazards
School Vulnerability Model - Schools not near Haz Facilities, Hazmat routes, or in Flood prone areas
School Vulnerability Model
Non-vulnerable Schools
Make Feature Layer
Schools with Gymnasiums
Hospitals
Select Layer By Location
Schools with Gyms near Hospitals
Schools with Gymnasiums AND Near Hospitals

Tools



Input Features

Output Feature Class

Distance [value or field]

● Linear unit

Unknown

○ Field

Side Type (optional)

FULL

End Type (optional)

ROUND

Dissolve Type (optional)

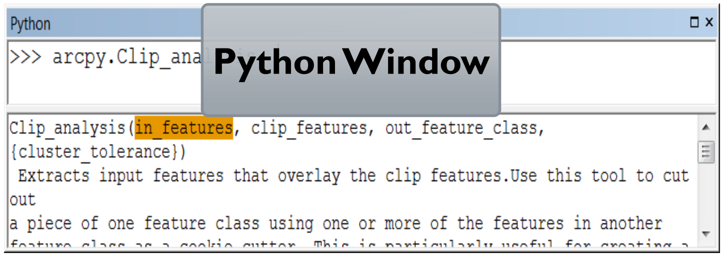
NONE

Dissolve Field(s) (optional)

Buffer

Creates buffer polygons around input features to a specified distance.

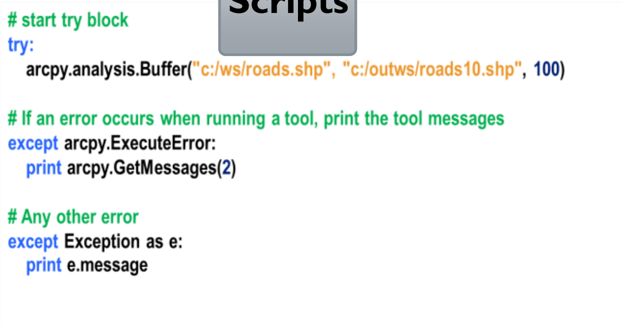
Python Window



```
>>> arcpy.Clip_ana
```

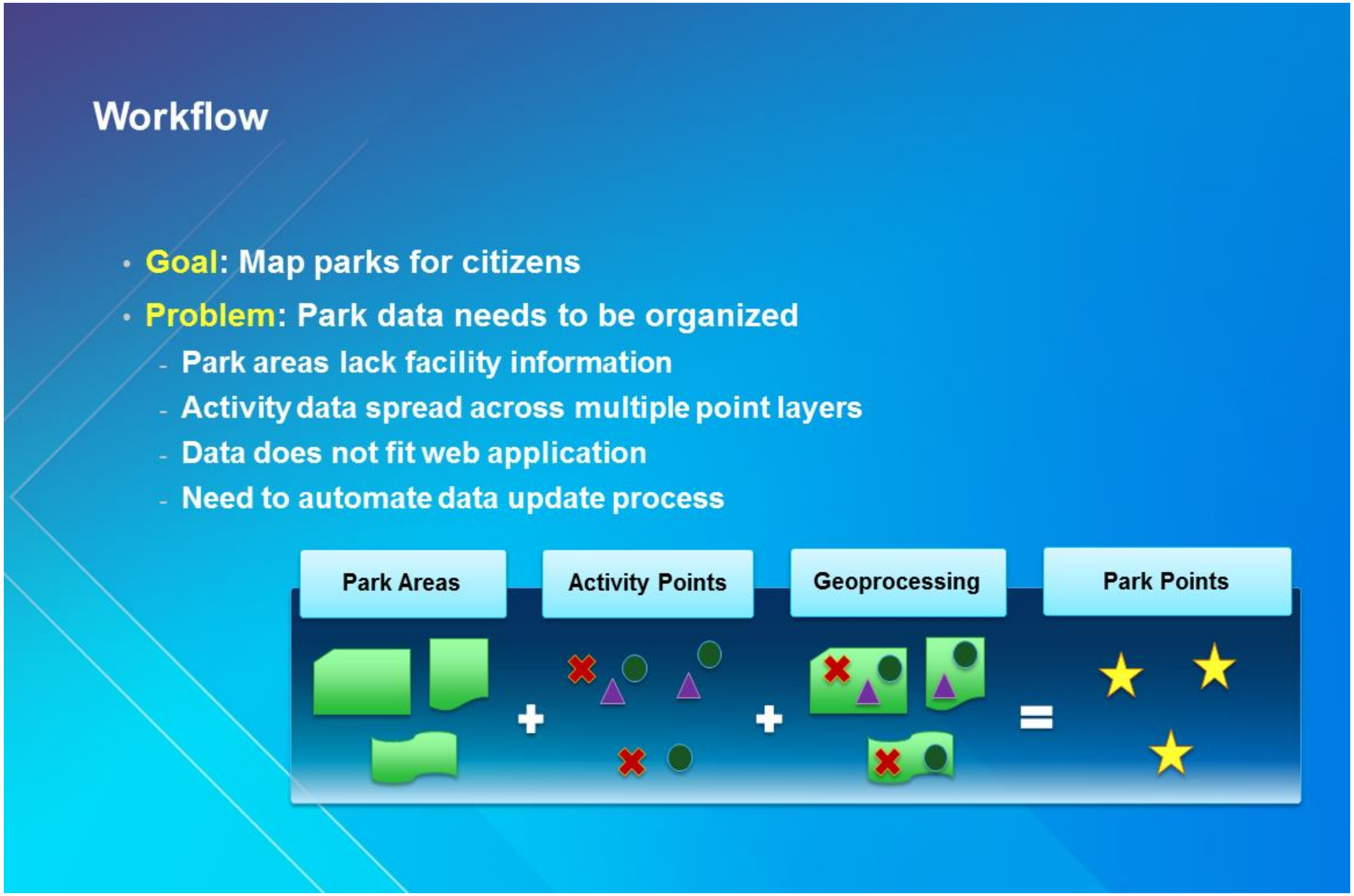
```
Clip_analysis(in_features, clip_features, out_feature_class,
{cluster_tolerance})
 Extracts input features that overlay the clip features.Use this tool to cut
out
a piece of one feature class using one or more of the features in another
feature class as a cookie cutter. This is particularly useful for creating a
```

Scripts



```
# start try block
try:
    arcpy.analysis.Buffer("c:/ws/roads.shp", "c:/outws/roads10.shp", 100)

# If an error occurs when running a tool, print the tool messages
except arcpy.ExecuteError:
    print arcpy.GetMessages(2)

# Any other error
except Exception as e:
    print e.message
```

Geoprocessing in GIS

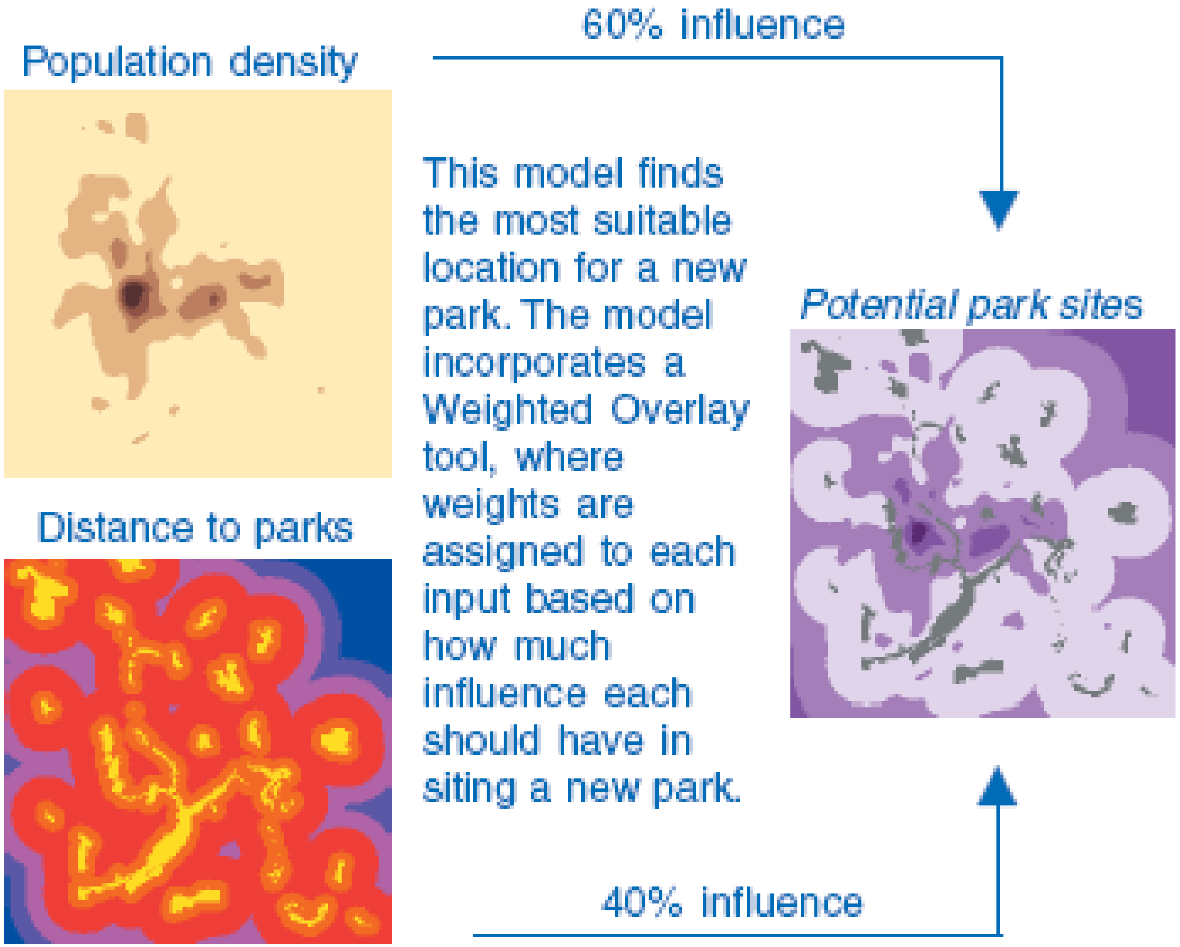


Workflow

- **Goal**: Map parks for citizens
- **Problem**: Park data needs to be organized
  - Park areas lack facility information
  - Activity data spread across multiple point layers
  - Data does not fit web application
  - Need to automate data update process

https://blogs.esri.com/esri/arcgis/2011/07/01/locating-a-park-through-suitability-analysis/

Geoprocessing in GIS



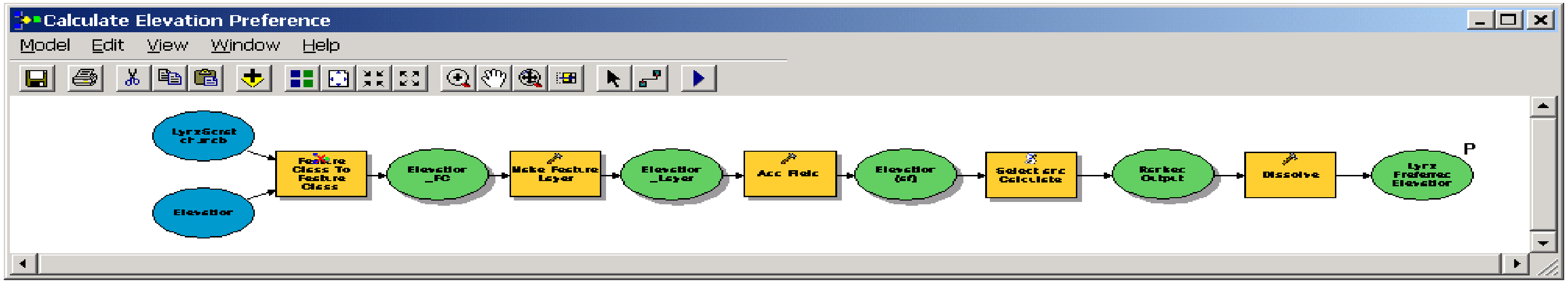
This model finds the most suitable location for a new park. The model incorporates a Weighted Overlay tool, where weights are assigned to each input based on how much influence each should have in siting a new park.

“Euclidean Distance”

```
import arcpy from arcpy
import env from arcpy.sa
import * env.workspace = "C:/sapyexamples/data"

outEucDistance = EucDistance("rec_sites.shp", 5000,
5, "c:/sapyexamples/output/EucDirOut")
outEucDistance.save("C:/sapyexamples/output/eucdist")
```

The Need For GIS Automation

- Automation makes work **easier**. You don't have to remember which tools to use or the proper sequence in which they should be run.
- Automation makes work **faster**.
- Automation makes work **more standarlized.**



Let us solve a problem!

- Find the population living within the 500 meter of HW 101 in Santa Barbara

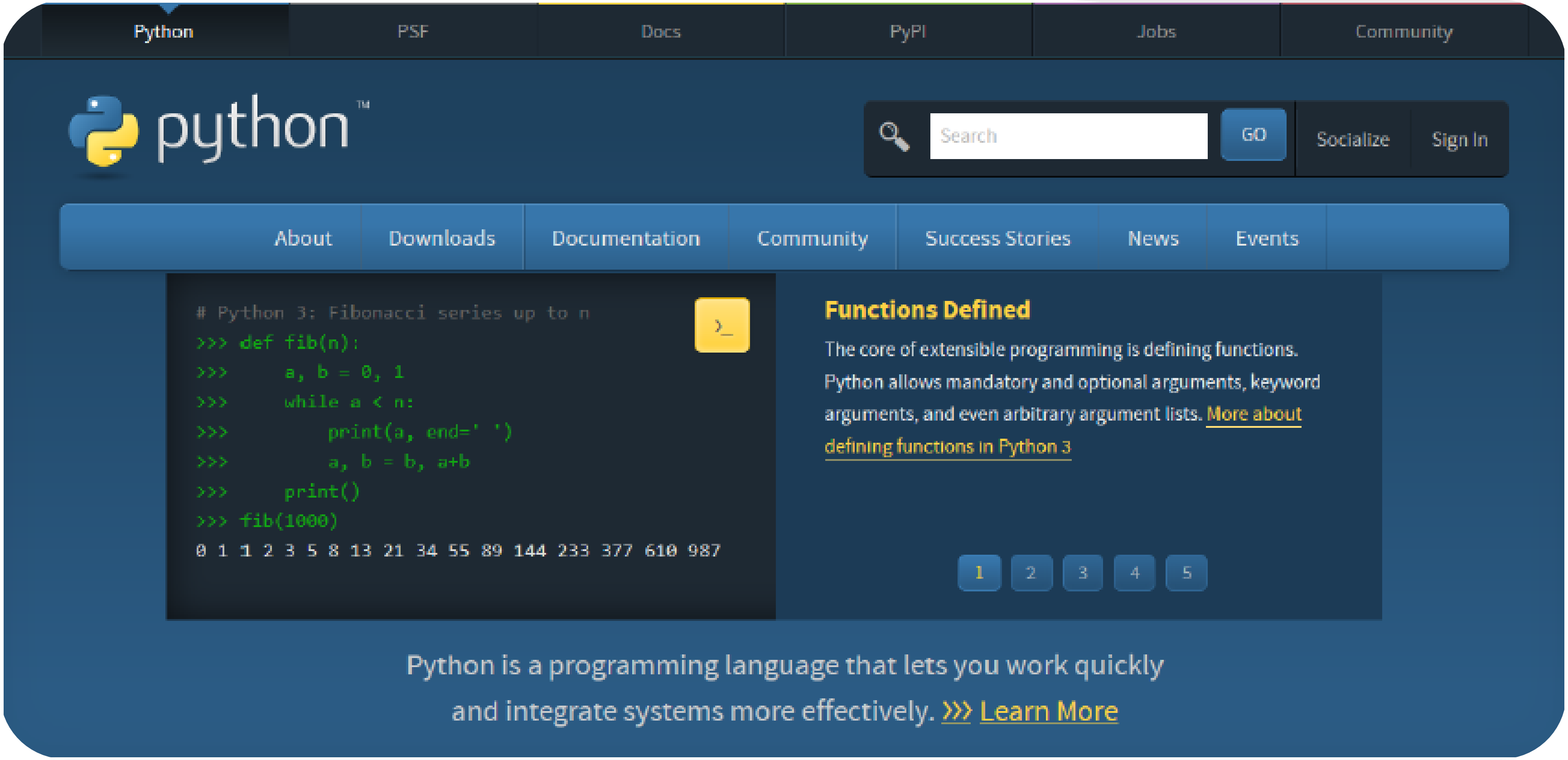
Population Data:

https://www.census.gov/cgi-bin/geo/shapefiles/index.php

Roads:

http://geog.ucsb.edu/~sgao/data/santabarbara.zip

2. What is Python?



```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. More about defining functions in Python 3

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

https://www.python.org/

Applications of Python

1. Automate workflows;
2. Batch process data;
3. Manipulate data tables, geometry, and map docs;
4. Use functions accessible only by scripts.

Advantages of Python

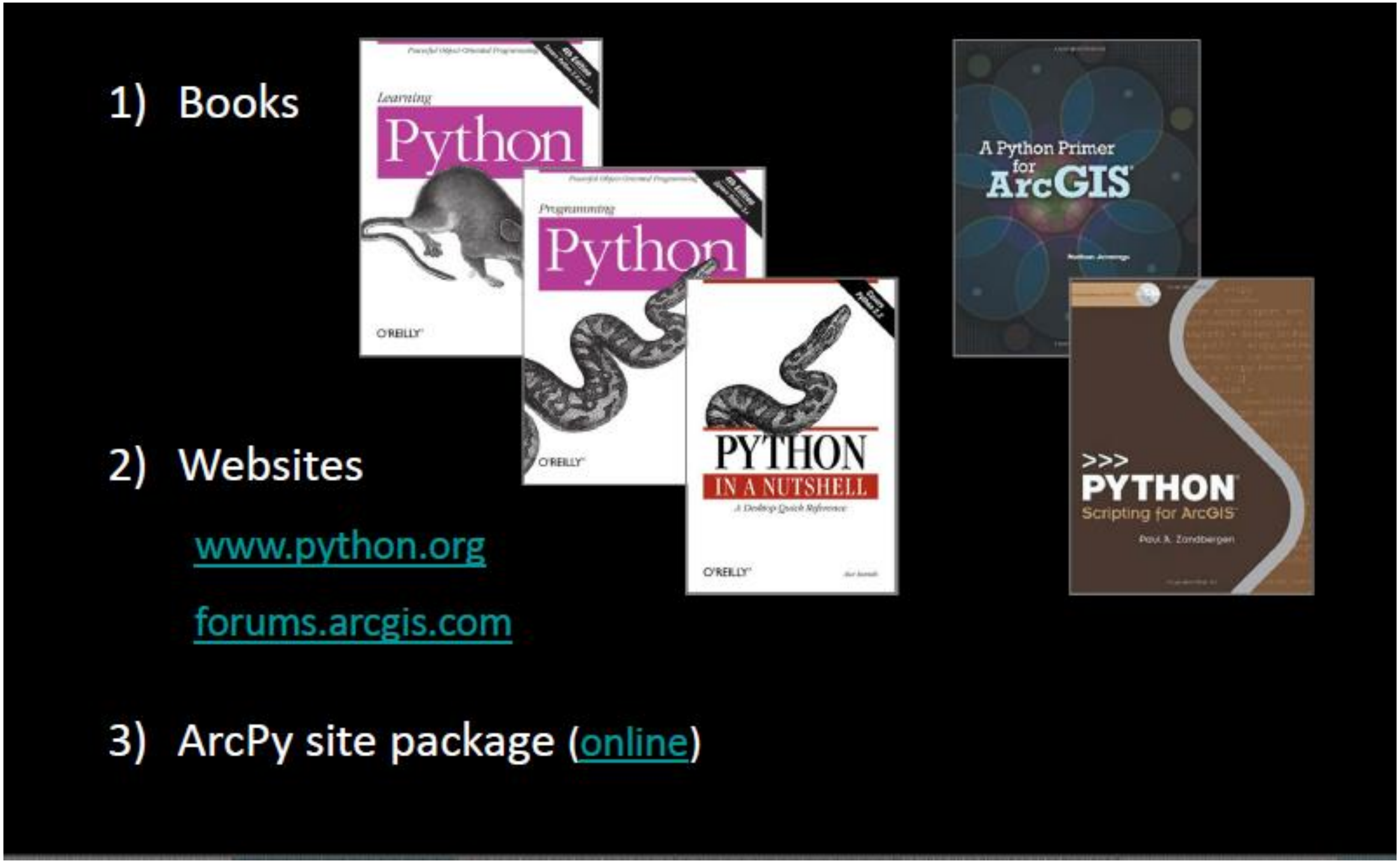
1. Less restricted data types;
2. Open source support;
3. Cross-platform;
4. Object-orientated (A data structure that combines data with a set of methods for accessing and managing those data).

Python Editors

- Integrated Development Environment (IDE): A software application for programming and software development
- Source code editor: A text editor for software code, with features specially designed to simplify and speed up writing and editing of code
- Suggested Python editors:

1) IDLE
2) PythonWin
3) IPython
4) Others (wiki.python.org/moin/PythonEditors)

User Resources



1) Books

2) Websites

www.python.org

forums.arcgis.com

3) ArcPy site package ([online]())

Open Source Python Packages

WinPython

WinPython for Python 2.7:

- numpy 1.9
- scipy 0.15
- PySAL: not included
- pandas 0.15
- shapely: not included
- fiona: not included
- six 1.8
- Windows only

CONTINUUM ANALYTICS

Anaconda for Python 2.7:

- numpy 1.9
- scipy 0.14
- PySAL 1.6
- pandas 0.14
- shapely: not included
- fiona: not included
- six 1.8
- Virtual Machine images
- Windows, Mac, Linux

ENTHOUGHT SCIENTIFIC COMPUTING SOLUTIONS

Enthought Canopy for Python 2.7:

- numpy 1.8
- scipy 0.15
- PySAL 1.7 (in academic option)
- pandas 0.15
- shapely: 1.5.1 (in academic option)
- fiona: 1.4.8 (in academic option)
- six 1.9
- Windows, Mac, Linux

Other Python distribution options listed at: http://www.scipy.org/install.html

Useful Open Source Python Spatial Libraries

Data Handling:

- shapely
- GDAL/OGR
- pyQGIS
- pyshp
- Pyproj

Analysis:

- shapely
- numpy, scipy
- pandas, GeoPandas
- PySAL
- Rasterio
- scikit-learn

Plotting:

- matplotlib, prettyplotlib
- descartes
- cartopyit-image

3. Python Structure and Syntax

- Programs are composed of modules
- Modules contain statements
- Statements contain expressions
- Expressions create and process objects

3. Python Structure and Syntax

- Object: A piece of memory, with values and associated operations; also known as variables

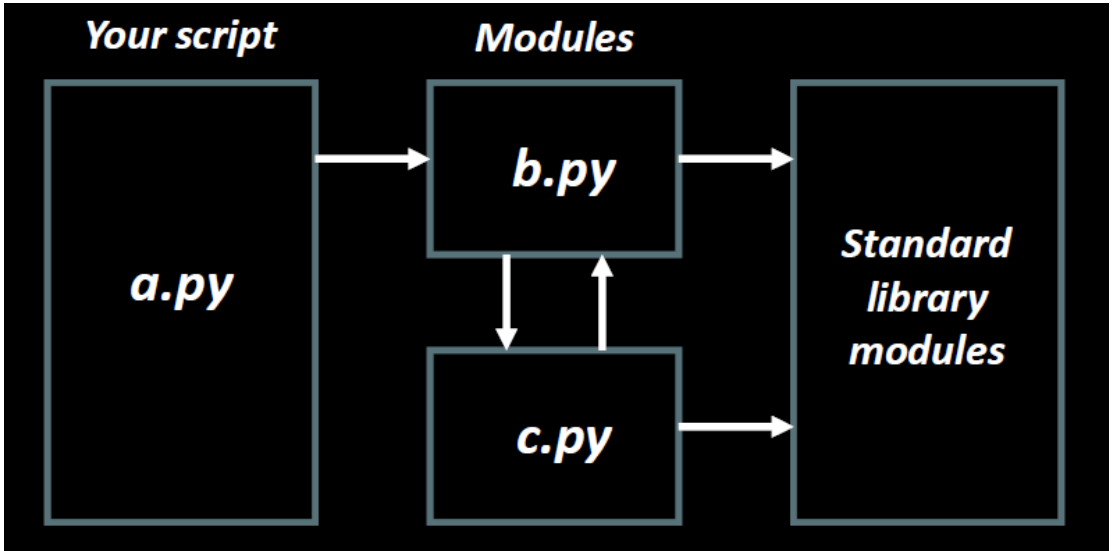
- Types of objects:
  - Numbers
  - Strings
  - Lists
  - Dictionaries
  - Files

3. Python Structure and Syntax

- Expression: Processes an object: **x * 2**
- Statement: Performs a task, via an expression: **y = x * 2**
- Types of statements:
  - *Assignment:* x=5
  - *Call:* open(‘DataFile.csv’)
  - import
  - print
  - if/elif/else
  - for, while

3. Python Structure and Syntax

- Module: A library of tools; permanent file of code, composed of statements.
- Types of modules:
  - Standard library modules: *os, sys, string* … (module index)
  - Specialized modules or site-packages: *arcpy* (site package)



3. Python Structure and Syntax

- Case sensitivity (DataFile ≠ datafile)
- Indentation (4, 6, 8…)
- File paths (/, \\ or r'string')
- Quotation marks (" , ')
- Commenting (#)

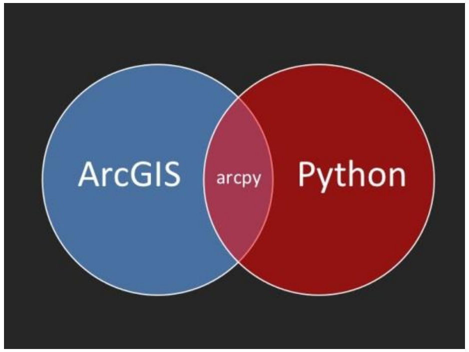
4. Running A Python Script In ArcGIS

- Provides Python access to all geoprocessing tools and extensions in ArcGIS

a. All geoprocessing tools in ArcMap are provided as functions in ArcPy

b. ArcPy also includes several functions not available as tools in ArcMap

- ArcPy has several sub-modules with related sets of functions (e.g., spatial analyst, mapping)



4. Running A Python Script In ArcGIS

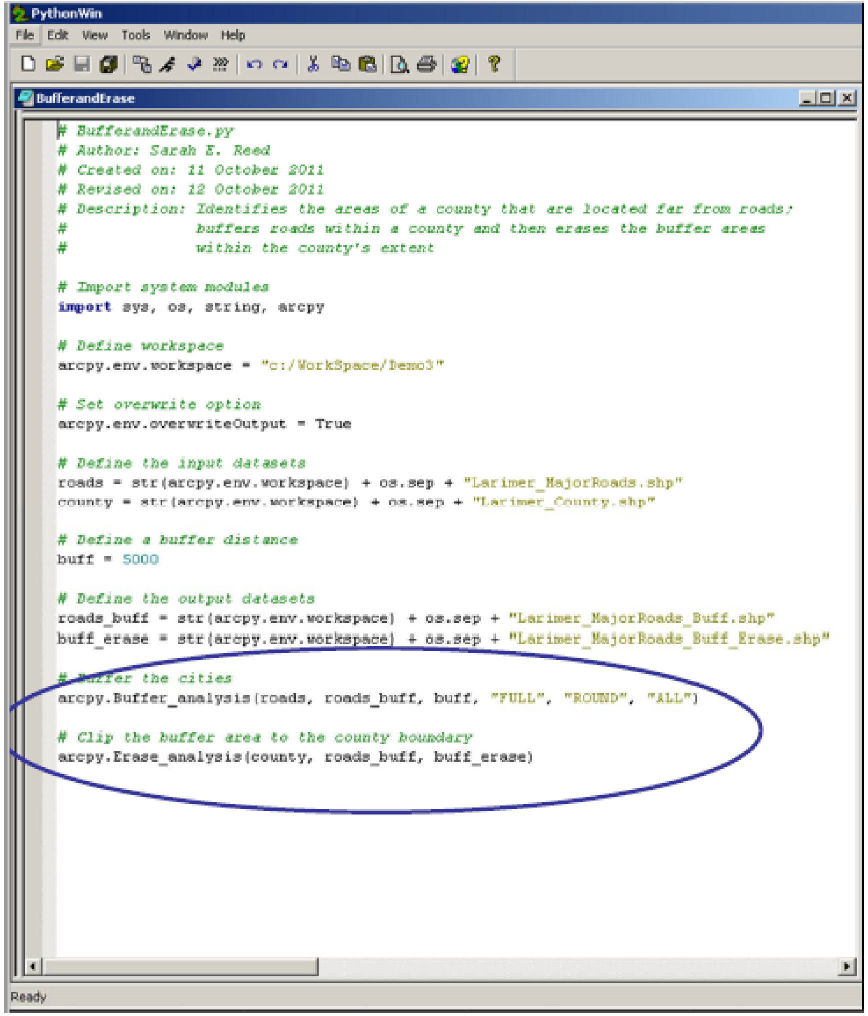
1) Include a header

2) Import modules

3) Specify environment settings

4) Define variables

5) Run geoprocessing tools



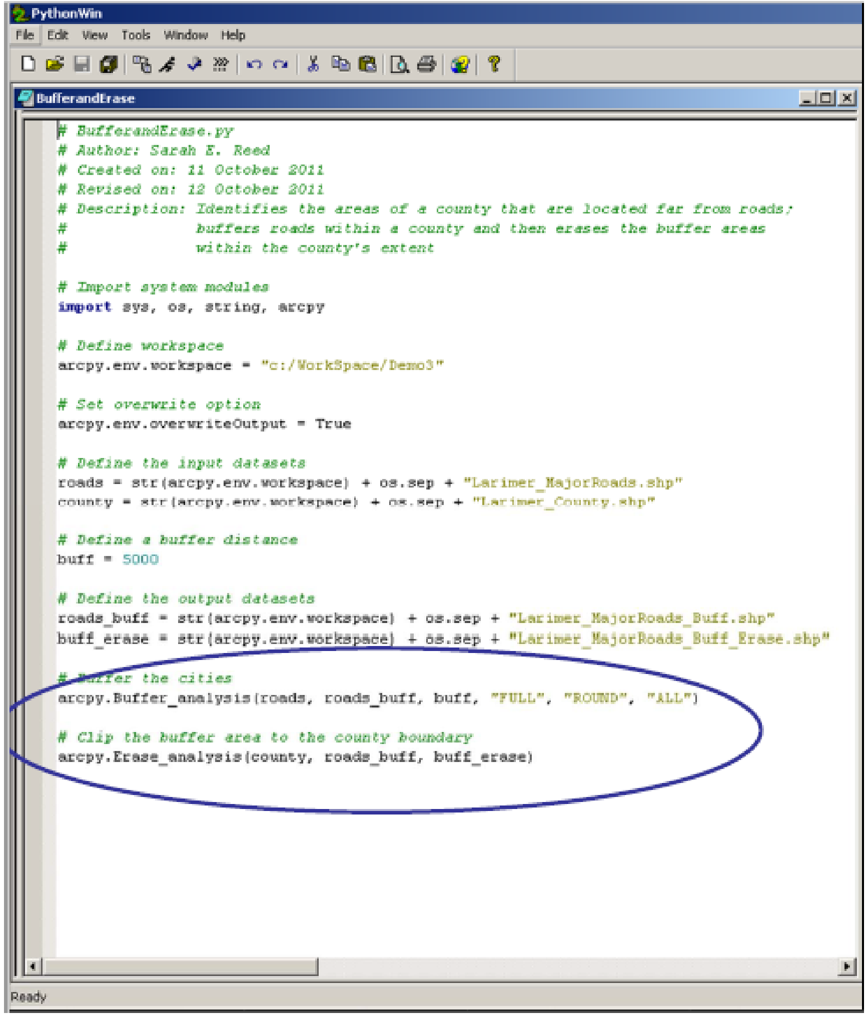
```
# BufferandErase.py
# Author: Sarah E. Reed
# Created on: 11 October 2011
# Revised on: 12 October 2011
# Description: Identifies the areas of a county that are located far from roads;
#              buffers roads within a county and then erases the buffer areas
#              within the county's extent

# Import system modules
import sys, os, string, arcpy

# Define workspace
arcpy.env.workspace = "c:/WorkSpace/Demo3"

# Set overwrite option
arcpy.env.overwriteOutput = True

# Define the input datasets
roads = str(arcpy.env.workspace) + os.sep + "Larimer_MajorRoads.shp"
county = str(arcpy.env.workspace) + os.sep + "Larimer_County.shp"

# Define a buffer distance
buff = 5000

# Define the output datasets
roads_buff = str(arcpy.env.workspace) + os.sep + "Larimer_MajorRoads_Buff.shp"
buff_erase = str(arcpy.env.workspace) + os.sep + "Larimer_MajorRoads_Buff_Erase.shp"

# Buffer the cities
arcpy.Buffer_analysis(roads, roads_buff, buff, "FULL", "ROUND", "ALL")

# Clip the buffer area to the county boundary
arcpy.Erase_analysis(county, roads_buff, buff_erase)
```

Three Ways to Run a Python Script In ArcGIS

1) In a Python editor

2) In the Python window in ArcMap

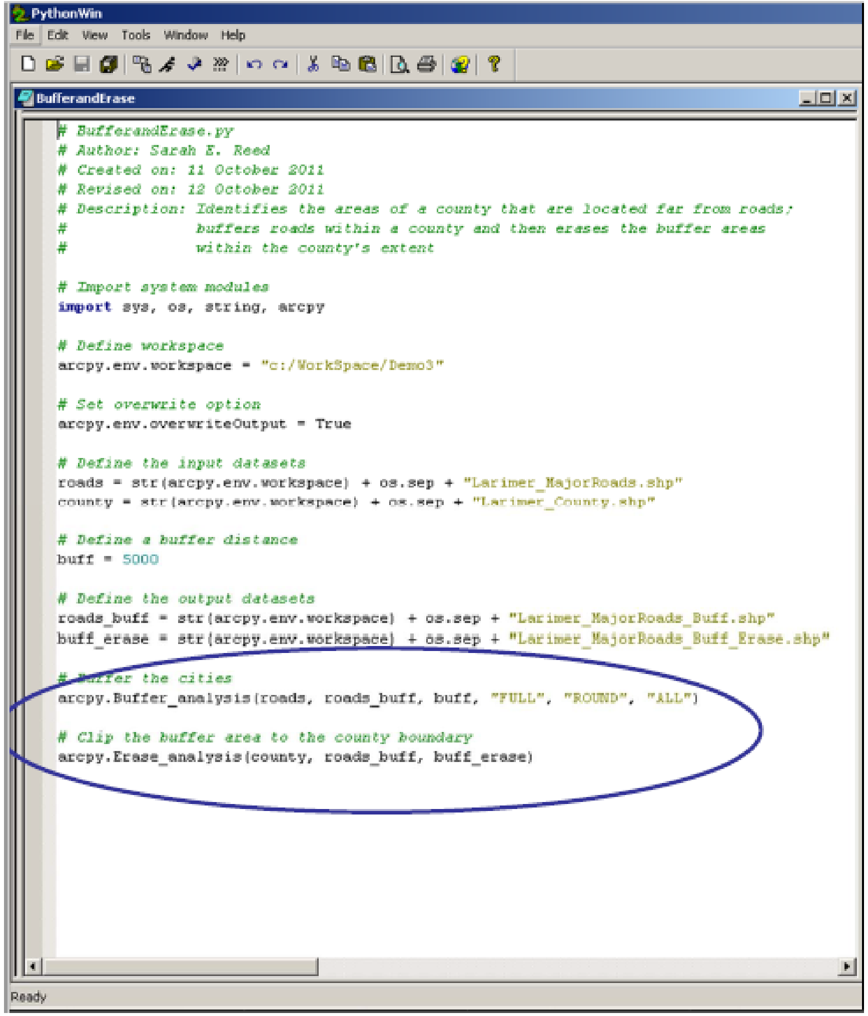
3) As a script tool in ArcToolbox



```
# BufferandErase.py
# Author: Sarah E. Reed
# Created on: 11 October 2011
# Revised on: 12 October 2011
# Description: Identifies the areas of a county that are located far from roads;
#              buffers roads within a county and then erases the buffer areas
#              within the county's extent

# Import system modules
import sys, os, string, arcpy

# Define workspace
arcpy.env.workspace = "c:/WorkSpace/Demo3"

# Set overwrite option
arcpy.env.overwriteOutput = True

# Define the input datasets
roads = str(arcpy.env.workspace) + os.sep + "Larimer_MajorRoads.shp"
county = str(arcpy.env.workspace) + os.sep + "Larimer_County.shp"

# Define a buffer distance
buff = 5000

# Define the output datasets
roads_buff = str(arcpy.env.workspace) + os.sep + "Larimer_MajorRoads_Buff.shp"
buff_erase = str(arcpy.env.workspace) + os.sep + "Larimer_MajorRoads_Buff_Erase.shp"

# Buffer the cities
arcpy.Buffer_analysis(roads, roads_buff, buff, "FULL", "ROUND", "ALL")

# Clip the buffer area to the county boundary
arcpy.Erase_analysis(county, roads_buff, buff_erase)
```

Three Ways to Write a Python Script In ArcGIS

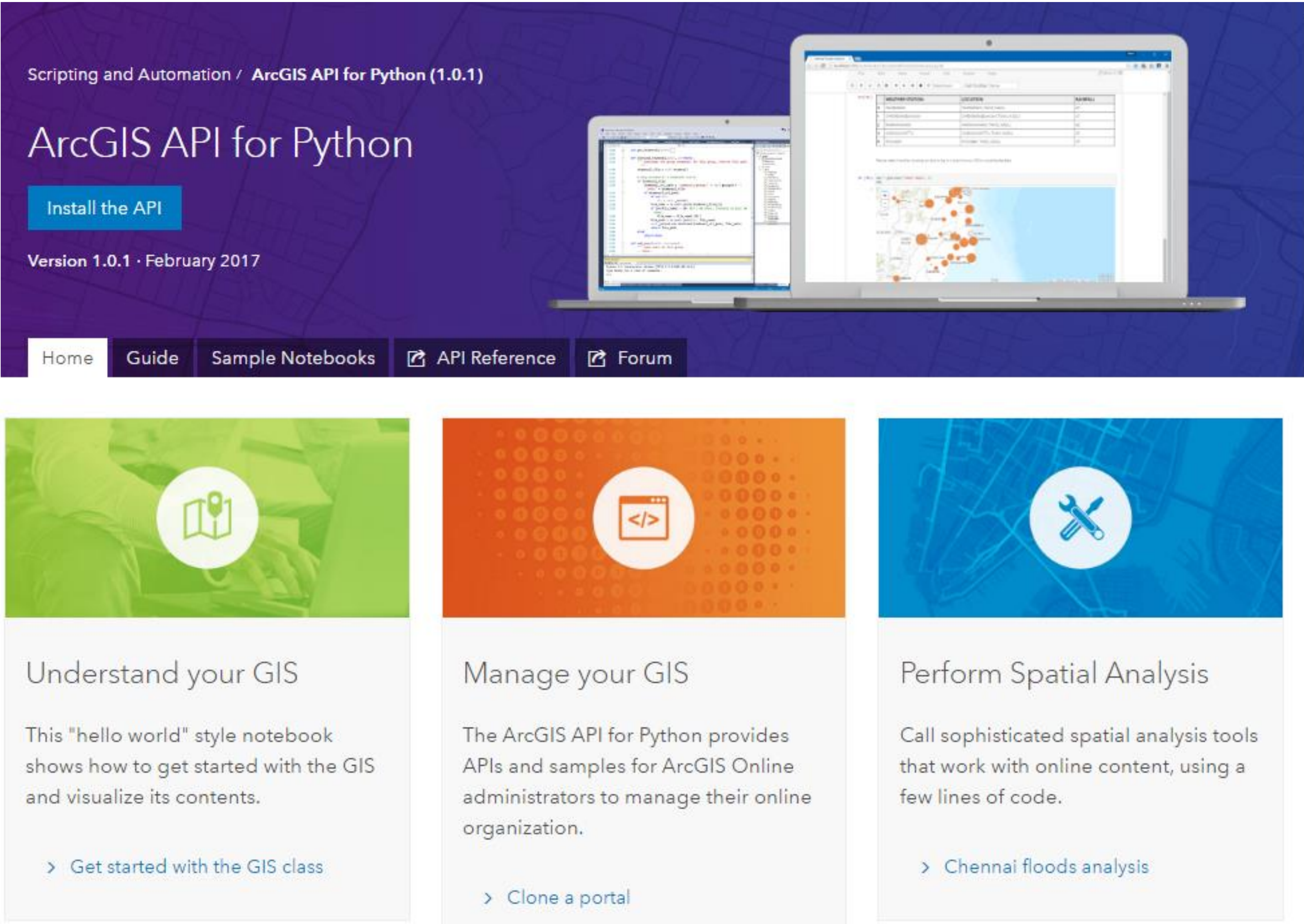
1) Edit an existing script
2) Export a script from ModelBuilder
3) Build a script in the Python window



```
# BufferandErase.py
# Author: Sarah E. Reed
# Created on: 11 October 2011
# Revised on: 12 October 2011
# Description: Identifies the areas of a county that are located far from roads;
#              buffers roads within a county and then erases the buffer areas
#              within the county's extent

# Import system modules
import sys, os, string, arcpy

# Define workspace
arcpy.env.workspace = "c:/WorkSpace/Demo3"

# Set overwrite option
arcpy.env.overwriteOutput = True

# Define the input datasets
roads = str(arcpy.env.workspace) + os.sep + "Larimer_MajorRoads.shp"
county = str(arcpy.env.workspace) + os.sep + "Larimer_County.shp"

# Define a buffer distance
buff = 5000

# Define the output datasets
roads_buff = str(arcpy.env.workspace) + os.sep + "Larimer_MajorRoads_Buff.shp"
buff_erase = str(arcpy.env.workspace) + os.sep + "Larimer_MajorRoads_Buff_Erase.shp"

# Buffer the cities
arcpy.Buffer_analysis(roads, roads_buff, buff, "FULL", "ROUND", "ALL")

# Clip the buffer area to the county boundary
arcpy.Erase_analysis(county, roads_buff, buff_erase)
```

Tips

1) Python is case sensitive;
2) Python is sensitive to indentation
3) Filepaths use single forward slash (/), double back slash (\\), or raw string supression (r"filepath")
4) May need to hard code filepaths (workspace + os.dir + "filename")
5) Save scripts with the *.py* file extension
6) Avoid schema lock: remove datasets from ArcMap



Scripting and Automation / **ArcGIS API for Python (1.0.1)**

ArcGIS API for Python

Version 1.0.1 · February 2017

Understand your GIS

This "hello world" style notebook shows how to get started with the GIS and visualize its contents.

› Get started with the GIS class

Manage your GIS

The ArcGIS API for Python provides APIs and samples for ArcGIS Online administrators to manage their online organization.

› Clone a portal

Perform Spatial Analysis

Call sophisticated spatial analysis tools that work with online content, using a few lines of code.

› Chennai floods analysis

https://developers.arcgis.com/python/