Integrating Open Source Statistical Packages with ArcGIS

Mark V. Janikas

Shaun Walbridge

Technical Workshop

Outline

- Introduction to Spatial Data Analysis in ArcGIS
  - Spatial Statistics, Geostatistics and Spatial Analyst
  - Python: Directly and Indirectly Extendable
  - Collaborative Motivation
- Direct
  - SciPy (Scientific Python)
  - PANDAS (Python Data Analysis Library)
  - PySAL (Python Spatial Analysis Library)
  - R (via IPython and RPy2 or Python Win Extensions)
- Indirect
  - R (matlab, SPSS, SAS)

Spatial Analytics in ArcGIS: Past and Present

- **Traditional Spatial Analysis**
  - Core tools continue to evolve
- **Spatial Analyst**
  - Raster
  - Map Algebra
- **Geostatistics**
  - Raster and Vector
  - Continuous Data
- **Spatial Statistics**
  - Vector
  - Exhaustive Data
  - Python

Spatial Analytics in ArcGIS: Moving Forward

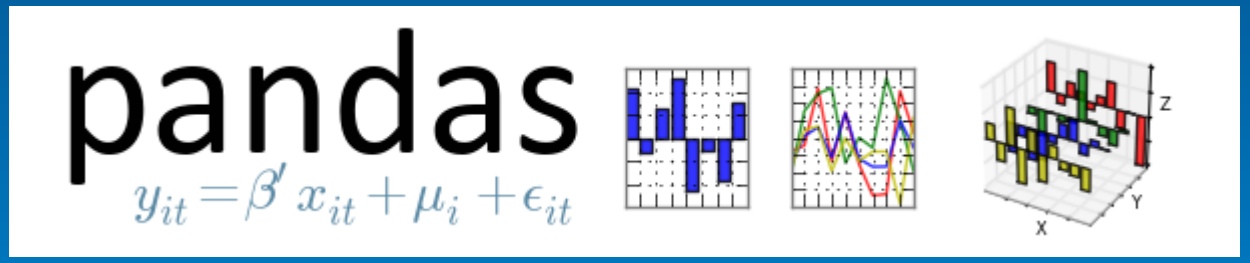
- Python
  - Spatial Analyst
    - Raster  NumPy
    - SciPy
  - Spatial Statistics and Geostatistics
    - Data Access Module
    - Vector  NumPy
    - Spatial Statistics Data Object and Utilities
    - Matplotlib, NetCDF4-Python
  - Effort to Support Scientific Community
    - SciPy, PANDAS, PySAL

The Great and Extendable Python

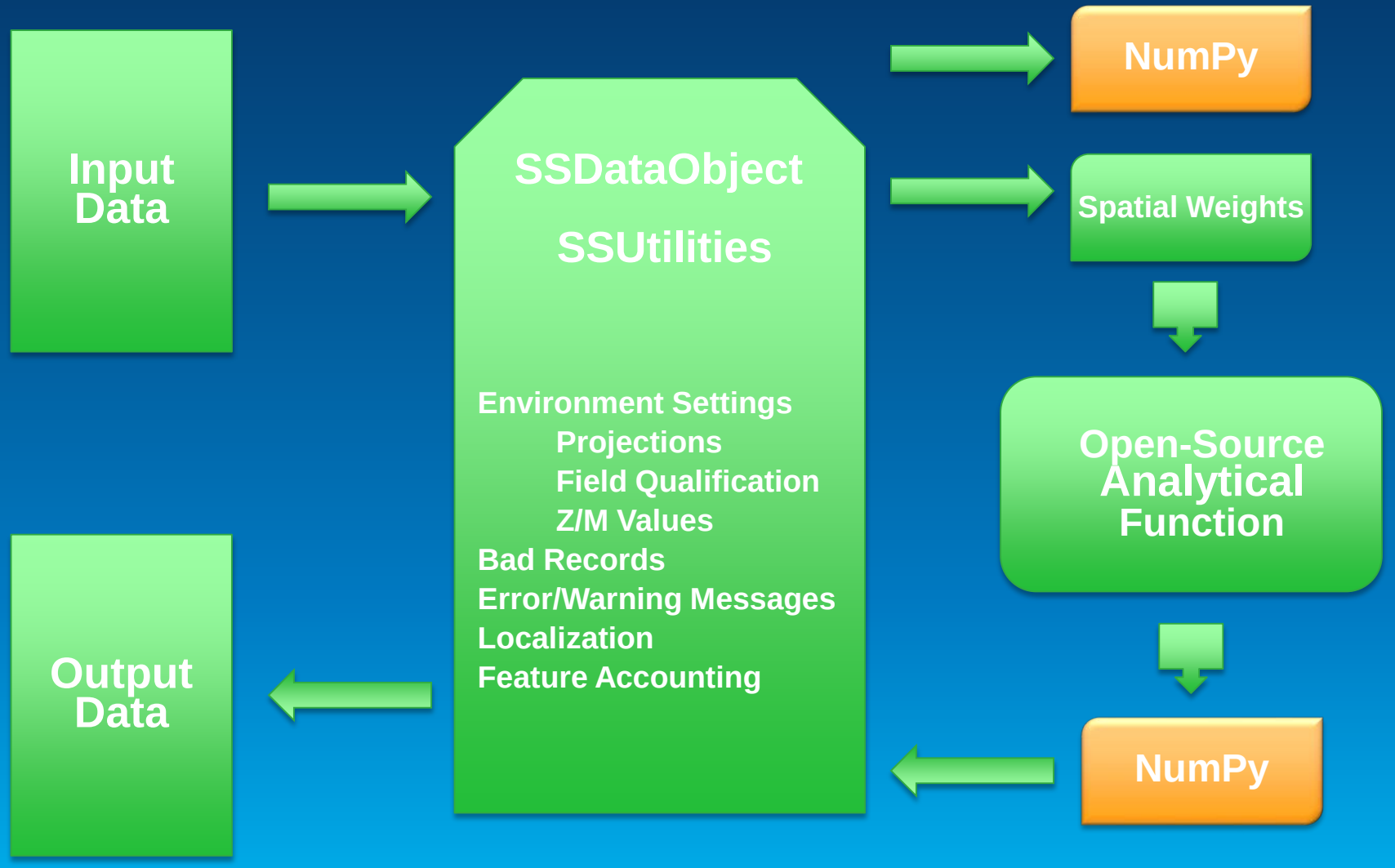
- Direct
  - Numeric/Scientific Python Modules
  - http://wiki.python.org/moin/NumericAndScientific
  - +60 Modules Listed
  - Check Compatibility… Then Plug and Play
    - pip, github, easy_install, svn
    - Unofficial Windows Binaries for Python Extensions – Christoph Gohlke, UC Irvine
      - http://www.lfd.uci.edu/~gohlke/pythonlibs/
- Indirect
  - Alternative Languages
  - No Python Hooks or Module
  - Python Serves as Active Script and OS
  - Out of Process
  - Using R in ArcGIS (Version Independent)
    - https://github.com/Esri/R-toolbox-py

Directly Extendable Via Python

- IPython
  - http://ipython.org/
  - Notebook (HTML Option)
- SciPy
- PANDAS
- PySAL
- R (Rpy Revisited)



Direct Python – ArcGIS Interaction Model



SSDataObject NumPy Arrays to PANDAS DataFrame

In [8]:
```
ssdo = SSDO.SSDataObject(inputFC)
years = NUM.arange(1975, 2015, 5)
fieldNames = ['PCR' + str(i) for i in years]
fieldNamesAll = fieldNames + ['NEW_NAME', 'SOCAL']
ssdo.obtainData("MYID", fieldNamesAll)
ids = [ssdo.order2Master[i] for i in xrange(ssdo.numObs)]
convertDictDF = {}
for fieldName, fieldObject in ssdo.fields.iteritems():
    convertDictDF[fieldName] = fieldObject.data
df = PANDA.DataFrame(convertDictDF, index = ids)
print df[0:5]
```

```
     NEW_NAME   PCR1975   PCR1980   PCR1985   PCR1990   PCR1995   PCR2000   PCR2005  \
158   Alameda  1.169255  1.195712  1.200988  1.165406  1.158115  1.307115  1.248997
159    Alpine  0.844546  0.906803  0.855655  0.924508  0.820581  0.949886  0.930033
160    Amador  0.991467  0.963228  0.921839  0.823639  0.815521  0.814954  0.864324
161     Butte  0.910668  0.898385  0.817796  0.794387  0.773955  0.763665  0.790418
162  Calaveras  0.941372  0.875469  0.891595  0.870938  0.806776  0.867385  0.880388
```

Analysis Using PANDAS, SSDataObject Makes Output Easy

Example: Calculating the Trend of Rolling Means

In [11]:
```
pcr = df.ix[:,1:9]
rollMeans = NUM.apply_along_axis(PANDA.rolling_mean, 1, pcr, 4)
timeInts = NUM.arange(0, 5)
outArray = NUM.empty((ssdo.numObs, 5), float)
for i in xrange(ssdo.numObs):
    outArray[i] = SCIPY.stats.linregress(timeInts, rollMeans[i,3:])
```

Write to Output (Same as Always...)

In [12]:
```
outputFC = OS.path.abspath(r'../data/testMyRollingMeanInfo.shp')
outFields = [ "SLOPE", "INTERCEPT", "R_SQRAURED", "P_VALUE", "STD_ERR" ]
outDict = {}
for fieldInd, fieldName in enumerate(outFields):
    outDict[fieldName] = SSDO.CandidateField(fieldName, "DOUBLE", outArray[:,fieldInd])
ssdo.output2NewFC(outputFC, outDict, fieldOrder = outFields)
del ssdo
```

Advanced Example: Spatially Constrained Clustering Using PySAL

```
ssdo = SSDO.SSDataObject(inputFC)
ssdo.obtainData(ssdo.oidName, ['GROWTH', 'POP1970', 'PERCNOHS'])
w = PYSAL.weights.knnW(ssdo.xyCoords, k=5)
X = NUM.empty((ssdo.numObs,2), float)
X[:,0] = ssdo.fields['GROWTH'].data
X[:,1] = ssdo.fields['PERCNOHS'].data
floorVal = 1000000.0
floorVar = ssdo.fields['POP1970'].returnDouble()
maxp = PYSAL.region.Maxp(w, X, floorVal, floor_variable = floorVar)
outArray = NUM.empty((ssdo.numObs,), int)
for regionID, orderIDs in enumerate(maxp.regions):
    outArray[orderIDs] = regionID
    print regionID, orderIDs
```

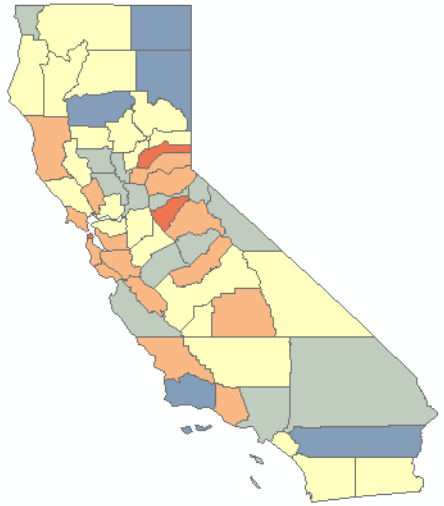
```
0 [7, 11, 52, 44, 22, 51, 17, 5, 24, 16, 3, 46, 8, 45, 10, 4, 54, 57, 50, 21, 9]
1 [1, 2, 33, 47, 56, 25, 13, 37, 27, 30, 28, 31]
2 [36, 32]
3 [41, 55, 29]
4 [15, 23, 53, 34, 14, 49, 19, 38]
5 [40, 0, 6, 42]
6 [26, 39, 43]
7 [18]
8 [20, 48]
9 [12, 35]
```

Directly Extendible

Using the IPython Notebook to Demonstrate How ArcGIS Can Leverage Python Modules

Using the ArcGIS Script Tool Interface to Wrap Advanced Spatial Data Analysis Functions

IP[y]: Notebook



Conclusions

- SciPy, PANDAS, PySAL
  - Advanced spatial analytic techniques
  - Combined with SSDataObject and Utilities
    - NumPy - Directly compatible
  - Python Harness Implementation
  - BSD
- R
  - Needs a collaborative effort to grow
    - New Tools on GitHub
  - Revisit In Proc Methodology
    - Installation Process is still a roadblock

Additional Resources

- This Presentation (Slides, Data, IPython Notebook)
  - Public GitHub Repository:
    - https://github.com/Esri/gis-stat-analysis-py-tutor
- ArcGIS – PySAL Toolbox
  - http://geodacenter.asu.edu/software
  - Keep checking for release version… Coming soon on GitHub!

- Mark Janikas, Ph. D.
  - mjanikas@esri.com
- Shaun Walbridge
  - swalbridge@esri.com

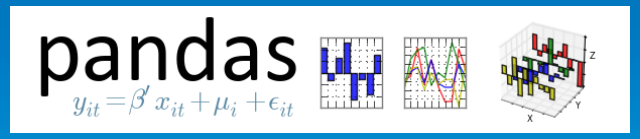
Additional Resources (Cont.)

- **Using R in ArcGIS (Version Independent – Out of Proc)**
  - https://github.com/Esri/R-toolbox-py
- **Spatial Statistics Resource Blog**
  - http://blogs.esri.com/esri/arcgis/2010/07/13/spatial-statistics-resources/

Book Title	Formats	Comments
GIS Tutorial for Python Scripting Esri Press, 2014	Paperback and e-book	Just released! Offers several hands-on tutorial exercises.
Python Scripting for ArcGIS Esri Press, 2013	Paperback and e-book	Good reference text

Software Links

- **PySAL**
  - https://geodacenter.asu.edu/pysal
  - http://code.google.com/p/pysal/
- **NumPy and SciPy**
  - http://www.numpy.org/
- **IPython**
  - http://ipython.org/
- **PANDAS**
  - http://pandas.pydata.org/
- **R**
  - http://www.r-project.org/index.html



Thank you…

- **Please fill out the session survey:**

First Offering ID: 193 / 1736

Online – www.esri.com/ucsessionsurveys

Paper – pick up and put in drop box

Understanding our world.