# Introduction to Spatial Econometrics in PySAL

Luc Anselin


http://spatial.uchicago.edu

- antecedents
- functionality
- interface
- workflow
- additional resources

# Antecedents

- Definition of Spatial Econometrics
  (e.g. , Anselin 1988, 2006)

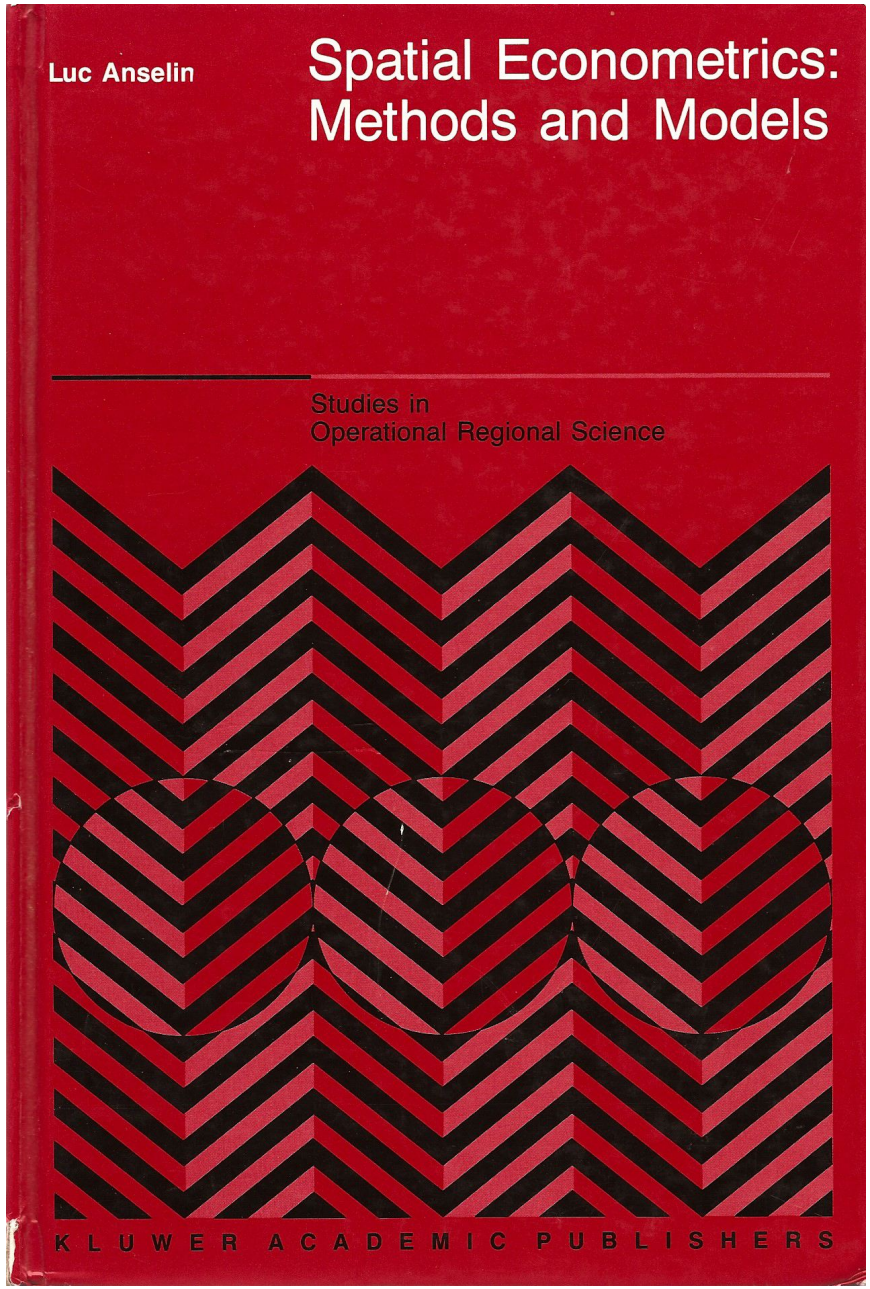
  - a subset of econometric methods concerned with spatial aspects present in cross-sectional and space-time observations

  - explicit treatment of location, distance and arrangement in model specification, estimation, diagnostics and prediction

  - variables at other locations

  - spatial effects: dependence, heterogeneity

- Spatial Effects
  - spatial dependence
    - two-dimensional and multidirectional cross-sectional dependence
    - not a direct extension of time series methods
  - spatial heterogeneity
    - structural change, varying coefficients
    - standard methods apply
  - identification problems




spatial econometrics (1988)

# *SpaceStat* TUTORIAL

## A Workbook for Using *SpaceStat* in the Analysis of Spatial Data

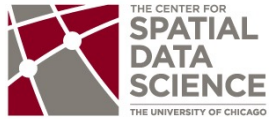
by

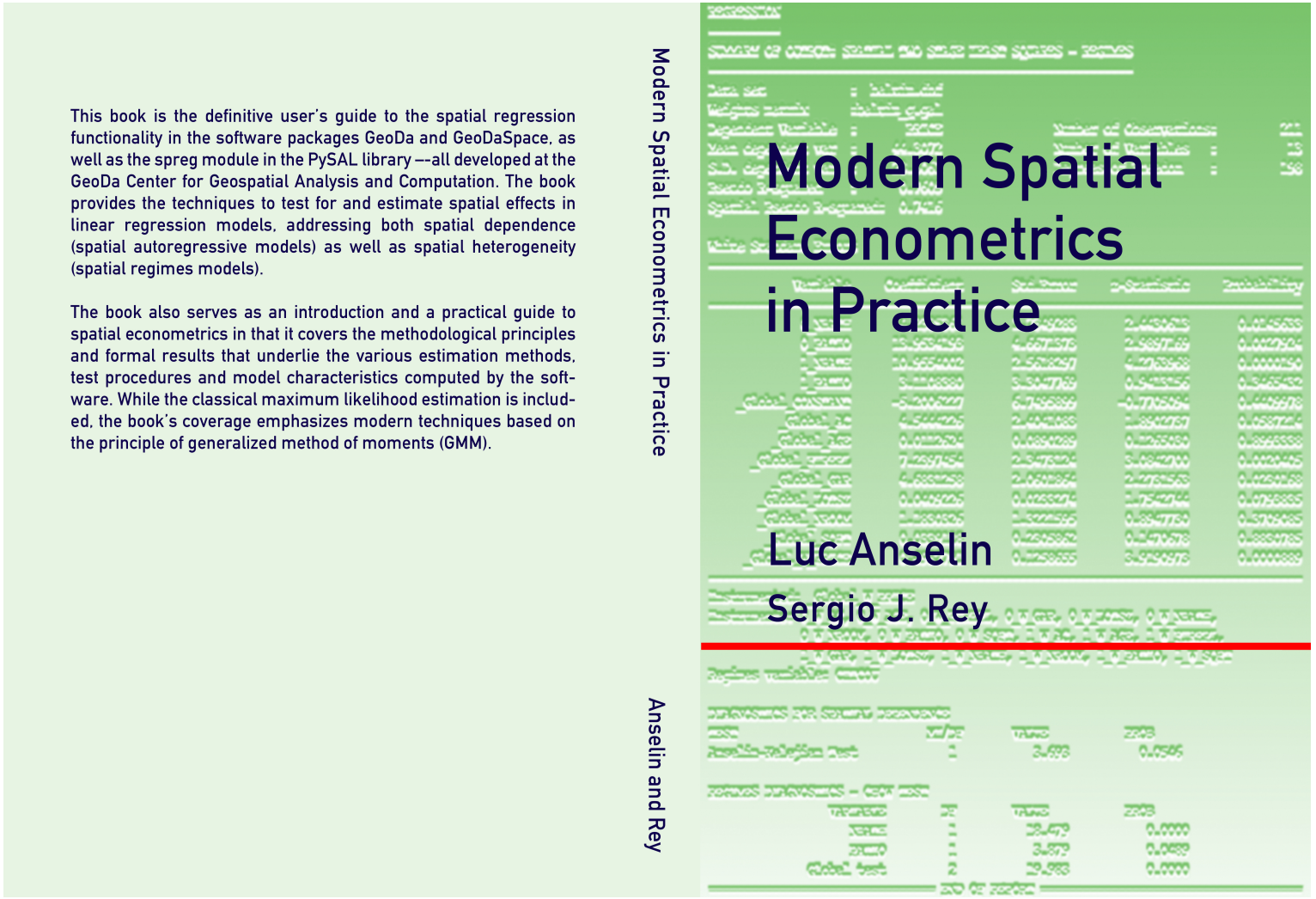
Luc Anselin

University of Illinois, Urbana-Champaign
Urbana, IL 61801
**luc@spacestat.com**



The original version of this Tutorial was distributed as Technical Report S-92-1 of the National Center for Geographic Information and Analysis, University of California, Santa Barbara, CA.




GeoDa, GeoDaSpace and PySAL implementation (2014)

# Functionality

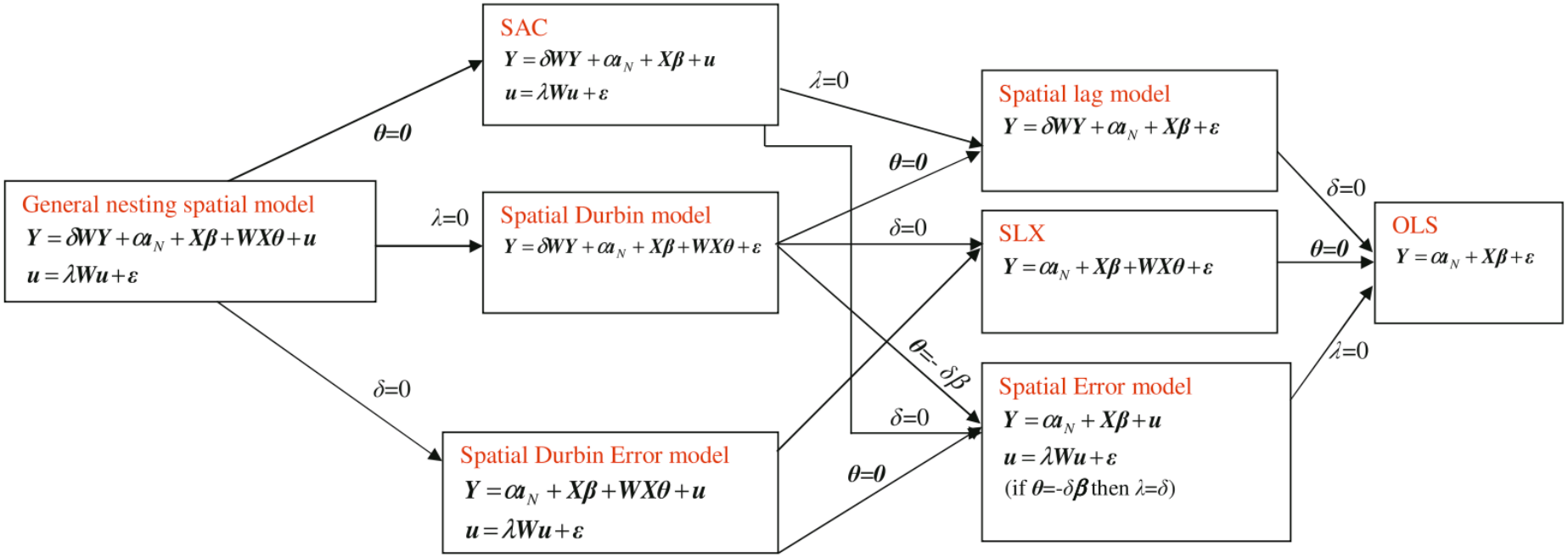



**Fig. 2.1** The relationships between different spatial dependence models for cross-section data (*source* Halleck Vega and Elhorst 2012)

- Estimation
  - classic regression models (OLS, 2SLS)
  - spatial lag, spatial error, linear and nonlinear SLX
  - Spatial Durbin, SLX-Error, GNS
  - spatial regimes - exogenous and endogenous
  - spatial SUR

- Regression Diagnostics
  - LM tests on OLS residuals
  - AK test on 2SLS residuals
  - Durbin-Wu-Hausman test for endogeneity
  - tests on probit residuals
  - test on coefficient heterogeneity

- Specification Search
  - forward search (specific to general - STGE)
    - based on LM tests
  - backward search (general to specific - GETS)
    - based on testing coefficient constraints
  - hybrid
    - start with spatial Durbin model

- Helper Functions
  - data generating process (dgp)
    - simulate dependent variable for a range of spatial models
  - model impacts
    - three different methods (simple, full, power)
  - location-specific spatial multipliers
    - direct, effect of neighbors, effect on neighbors

# Interface

- Four Main Functions/Classes + Regimes
  - OLS
  - TSLS
  - GM_Lag and ML_Lag
  - GMM_Error and ML_Error
  - xxx_Regimes

**Required Arguments** (depending on the function)

| | |
|---|---|
| y | numpy array or Pandas Series with observations on the dependent variable |
| x | numpy array or Pandas/GeoPandas dataframe/geo-dataframe with observations on the explanatory variables (no constant term!) |
| w | PySAL spatial weights or Graph object |
| yend | numpy array of Pandas/GeoPandas dataframe/geo-dataframe with observations on the endogenous variables |
| q | numpy array of Pandas/GeoPandas dataframe/geo-dataframe with observations on the instruments |
| coords | numpy array, Pandas/GeoPandas dataframe/geo-dataframe, or geo-dataframe geometry column with point coordinates to construct the sparse input weights - nonlinear SLX only |

## General Options

| | |
|---|---|
| name_y | string with name of dependent variable (not needed when using Pandas Series) |
| name_x | list of strings with names of explanatory variables (not needed when using Pandas dataframe) – no constant term |
| name_ds | string with name for data set |
| name_w | string with name for spatial weights |
| name_gwk | string with name for kernel weights (for HAC only, not in SLX) |
| name_yend | list of strings with names for endogenous variables (not including Wy) – not needed when using Pandas dataframe |
| name_q | list of strings with names for instruments (not including WX) – not needed when using Pandas dataframe |
| robust | type of robust standard error - default None, “white” or “hac” (all but ML estimation) |
| spat_diag | spatial diagnostics (LM tests; common factor test) - default False, if True, requires w |
| latex | coefficient table output in Latex format - default False |
| method | Jacobian computation in ML estimation, default = “full”, others “ord” or “LU”; type of GMM estimation in GMM_Error model, default = “het”, others “hom” or “kp98” |
| vm | include full variance covariance matrix in output listing, default = False |

## Common Options

| | |
|---|---|
| nonspat_diag | non-spatial diagnostics (multicollinearity, heteroskedasticity; Durbin-Wu-Hausman test for 2SLS) - default True (OLS and TSLS) |
| spat_impacts | method for spatial multipliers computation, default = “simple”, other options are “full”, “power”, “all” or None (all models with Wy) |
| sig2n_k | use n-k as denominator in variance calculation, default = False (OLS and TSLS) |
| w_lags | order of spatial lags to use as instruments for Wy, default = 1 (all IV estimations) |
| lag_q | boolean, whether instruments should be lagged as well, default = True (all IV estimations of lag model with endogenous variables) |
| slx_lags | (inclusive) order of spatial lag SLX terms, default = 0 (should be 1 or higher for SLX model) |
| slx_vars | number of variables to apply spatial lag to, default = “All”, otherwise list of Booleans matching X variables indicating whether (True) or not (False) a spatial lag should be applied to that variable |
| add_wy | flag for inclusion of Wy in other than spatial lag specification, default = False |

*Classic Models*

- OLS
  - spreg.OLS(y, x)
    - options:
      - moran: default False
      - white_test: default False
      - vif: default False
- 2SLS
  - spreg.TSLS(y, x, yend, q)

*Spatial Lag*

- S2SLS spatial lag – exogenous only
  - spreg.GM_Lag(y, x, w)
- S2SLS spatial lag – exogenous and endogenous
  - spreg.GM_Lag(y, x, yend, q, w)
- ML spatial lag
  - spreg.ML_Lag(y, x, w)

*Spatial Error*

- GMM spatial
  - spreg.GMM_Error(y, x, w)
- GMM spatial error with endogenous variables
  - spreg.GMM_Error(y, x, w, yend, q)
- ML Error
  - spreg.ML_Error(y, x, w)

*SLX*

- SLX
  - spreg.OLS(y, x, slx_lags=1,slx_vars='All')
- 2SLS with SLX
  - spreg.TSLS(y, x, yend, q, slx_lags=1,slx_vars='All')
- Nonlinear SLX
  - spreg.NSLX(y,x,coords)
  - options:
    - params: list with tuples; default is [(10,np.inf,"exponential")]
      - k=10, number of nearest neighbors
      - distance_upper_bound = np.inf (adaptive bandwidth), a specific distance value for fixed bandwidth
      - model = "exponential" – "power" for inverse distance power
    - distance_metric – default "Euclidean", other "Arc"
    - var_flag: analytical standard errors, default = 1, 0 = numerical approximation
    - conv_flag: convergence summary listing, default = 1
    - verbose: full output listing (every iteration), default = False

*Spatial Durbin*

- Spatial Durbin
  - spreg.GM_Lag(y, x, w, slx_lags=1, slx_vars='All')
- Spatial Durbin with endogenous variables
  - spreg.GM_Lag(y, x, yend, q, w, slx_lags=1, slx_vars='All')
- ML spatial Durbin
  - spreg.ML_Lag(y, x, w, slx_lags=1,slx_vars='All')

*SLX-Error*

- SLX Error
  - spreg.GMM_Error(y, x, w, slx_lags=1,slx_vars='All')
- SLX Error with endogenous variables
  - spreg.GMM_Error(y, x, w, yend, q, slx_lags=1,slx_vars='All')
- ML SLX Error
  - spreg.ML_Error(y, x, w, slx_lags=1,slx_vars='All')

*SAR-Error*

- Combo model (SAR-Error)
  - spreg.GMM_Error(y, x, w, add_wy=True)
- Combo model (SAR-Error) with endogenous
  - spreg.GMM_Error(y, x, w, yend, q, add_wy=True)

*GNS*

- Spatial Durbin Error = GNS = Combo with SLX
  - spreg.GMM_Error(y, x, w, slx_lags=1, add_wy=True)
- Spatial Durbin Error with endogenous = GNS = Combo with SLX
  - spreg.GMM_Error(y, x, w, yend, q, slx_lags=1, add_wy=True)

- Spatial Regimes
  - _Regimes for all cross-sectional methods
  - exogenous regimes
    - regime classification specified as an argument
  - endogenous regimes
    - skater-reg

- Spatial SUR
  - spatial seemingly unrelated regression with time-period specific coefficients
  - test on homogeneity of coefficients
  - Lag and Error models

# Workflow

- read in Data (pandas, geopandas)
  - into Pandas dataframe or geodataframe
- read/create Spatial Weights (libpysal)
  - read from GAL/GWT files
  - create from geodataframe/shape file
- specify Variable Names (columns in dataframe)
  - lists for y, x, yend, q
- optional: specify Names for Data and Weights

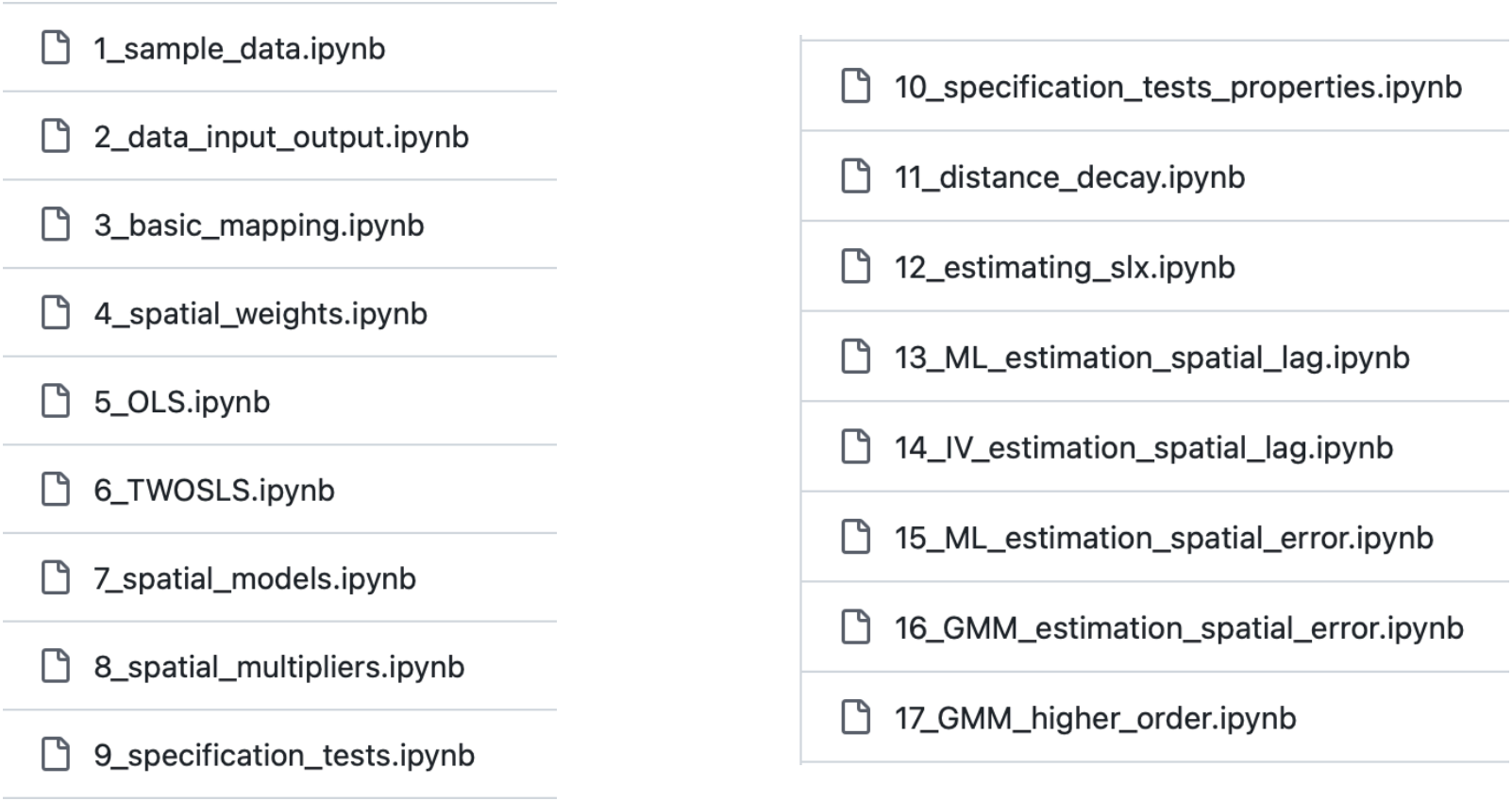
- Regression
  - output is an object with attributes and methods
  - output listing as: print(reg_object.summary)

```
['__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__',
'__format__', '__ge__', '__getattribute__', '__gt__', '__hash__', '__init__',
'__init_subclass__', '__le__', '__lt__', '__module__', '__ne__', '__new__',
'__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__',
'__str__', '__subclasshook__', '__summary', '__weakref__', '_cache', 'ar2',
'betas', 'k', 'mean_y', 'n', 'name_ds', 'name_gwk', 'name_w', 'name_x',
'name_y', 'other_mid', 'other_top', 'output', 'predy', 'r2', 'robust', 'sig2',
'sig2n', 'sig2n_k', 'std_err', 'std_y', 'summary', 't_stat', 'title', 'u',
'utu', 'vm', 'x', 'xtx', 'xtxi', 'y']
```

OLS basic regression object

- Spatial Diagnostics
  - obtained with spat_diag = True
  - part of output listing
  - or called as separate functions

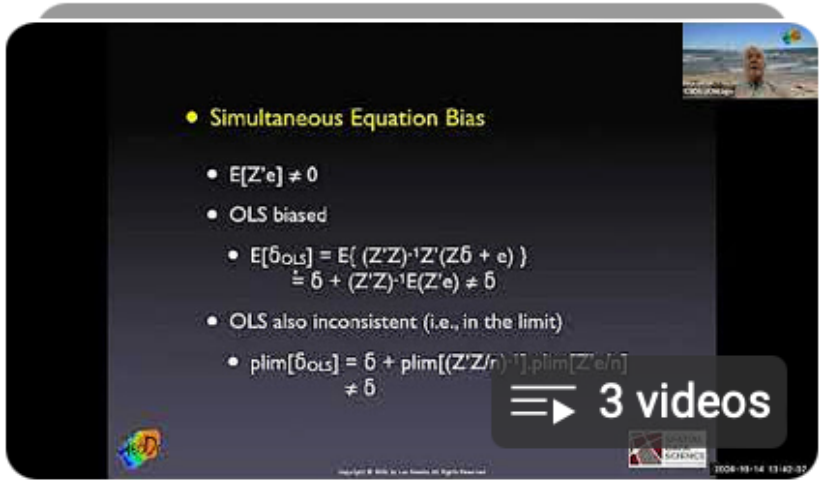
# Additional Resources




# Jupyter Notebooks

https://github.com/lanselin/spatial_regression_notebooks


**Applied Spatial Regression Analysis - Notebooks**




**Applied Spatial Regression Analysis - Lectures**

narrated lectures and labs

https://www.youtube.com/@GeoDaCenter/playlists

# Enjoy!