spreg Cheat Sheet (version 1.7)

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 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

Estimation

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

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')

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')

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)