10) Spatial Econometrics with PySAL

Vitor Kamada

September 2019

Reference

Tables, Graphics, and Figures from:

Rey and Arribas-Bel (2018). **Geographic Data Science with PySAL**

http://darribas.org/gds_scipy16/

Texas Counties from the Census Bureau

```
import pysal as ps
import pandas as pd
import numpy as np
from pysal.contrib.viz import mapping as maps
shp_path = 'C:/Users/Vitor/Desktop/ECO 7110 Ec
data = ps.pdio.read_files(shp_path)
```

data.head()

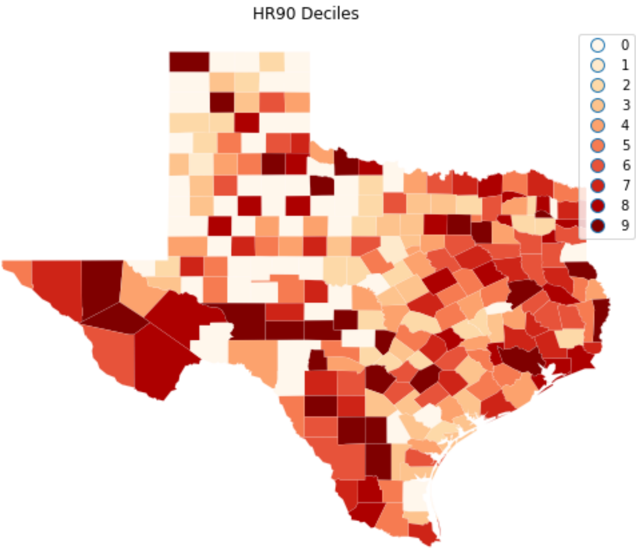
	NAME	STATE_NAME	STATE_FIPS	CNTY_FIPS	FIPS	STFIPS	COFIPS	FIPSNO
0	Lipscomb	Texas	48	295	48295	48	295	48295
1	Sherman	Texas	48	421	48421	48	421	48421
2	Dallam	Texas	48	111	48111	48	111	48111
3	Hansford	Texas	48	195	48195	48	195	48195
4	Ochiltree	Texas	48	357	48357	48	357	48357

	FH90	geometry
0	6.093580	<pysal.cg.shapes.Polygon object at 0x000020C5...
1	3.869407	<pysal.cg.shapes.Polygon object at 0x000020C5...
2	14.231738	<pysal.cg.shapes.Polygon object at 0x000020C5...
3	7.125457	<pysal.cg.shapes.Polygon object at 0x000020C5...
4	9.159159	<pysal.cg.shapes.Polygon object at 0x000020C5...

Map Pattern

```
import matplotlib.pyplot as plt
import geopandas as gpd
tx = gpd.read_file(shp_path)
hr10 = ps.Quantiles(data.HR90, k=10)
f, ax = plt.subplots(1, figsize=(9, 9))
tx.assign(cl=hr10.yb).plot(column='cl',
        categorical=True, k=10, cmap='OrRd',
        linewidth=0.1, ax=ax,
        edgecolor='white', legend=True)
ax.set_axis_off()
plt.title("HR90 Deciles")
plt.show()
```

County Homicide Rates in 1990



Spatial Weights

Queen Contiguity: adjacency relationships as a binary indicator variable denoting whether or not a polygon shares an **edge or a vertex** with another polygon

KNN: distance to k nearest neighbors

Kernel: neighbors defined by bandwidth

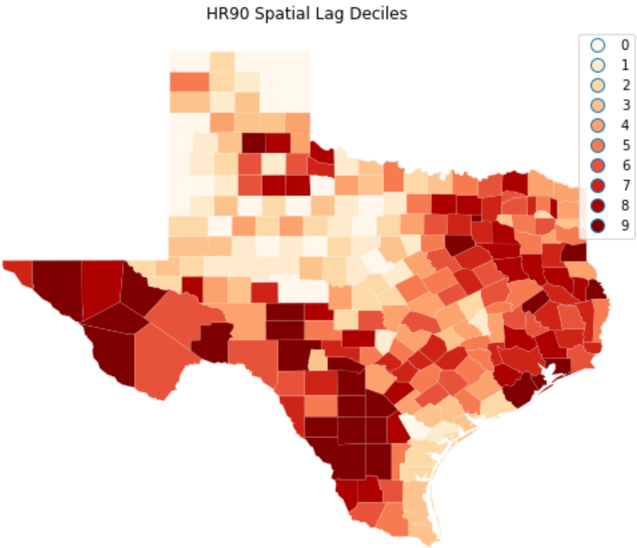
Spatial Lag: $\sum_j w_{i,j} HR90_j$

```
W = ps.queen_from_shapefile(shp_path)
W.transform = 'r'

HR90Lag = ps.lag_spatial(W, data.HR90)
HR90LagQ10 = ps.Quantiles(HR90Lag, k=10)

f, ax = plt.subplots(1, figsize=(9, 9))
tx.assign(cl=HR90LagQ10.yb).plot(column='cl',
              categorical=True, k=10, cmap='OrRd',
              linewidth=0.1, ax=ax,
              edgecolor='white', legend=True)
ax.set_axis_off()
plt.title("HR90 Spatial Lag Deciles")
plt.show()
```

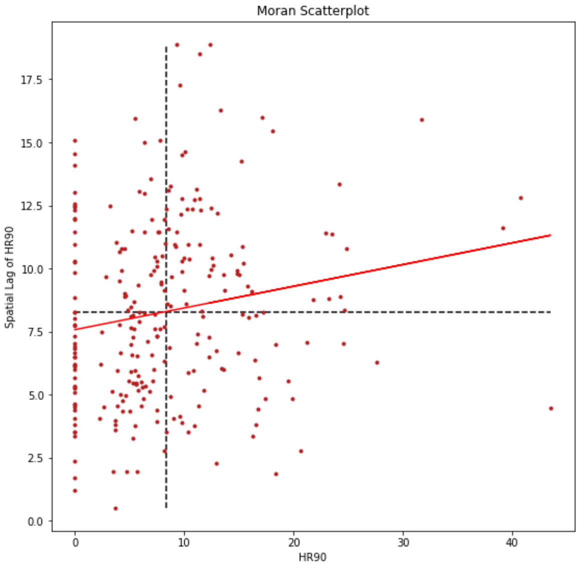
HR90 Spatial Lag Deciles



Moran Scatterplot

```
HR90 = data.HR90
b,a = np.polyfit(HR90, HR90Lag, 1)
f, ax = plt.subplots(1, figsize=(9, 9))
plt.plot(HR90, HR90Lag, '.', color='firebrick')
 # dashed vert at mean of the last year's PCI
plt.vlines(HR90.mean(), HR90Lag.min(), HR90Lag.max(),
           linestyle='--')
 # dashed horizontal at mean of lagged PCI
plt.hlines(HR90Lag.mean(), HR90.min(), HR90.max(),
           linestyle='--')
# red line of best fit using global I as slope
plt.plot(HR90, a + b*HR90, 'r')
plt.title('Moran Scatterplot')
plt.ylabel('Spatial Lag of HR90')
plt.xlabel('HR90')
plt.show()
```

$\sum_j w_{i,j} HR90_j$ vs HR90



Moran's Statistic (I)

I_HR90 = ps.Moran(data.HR90.values, W)

I_HR90.I, I_HR90.p_sim

(0.08597664031388977, 0.01)

b

0.0859766403138895

Austin Properties Listed in AirBnb

http://insideairbnb.com/austin/index.html

```
from pysal.model import spreg
from pysal.lib import weights
from pysal.explore import esda
from scipy import stats
import statsmodels.formula.api as sm
import numpy as np
import pandas as pd
import geopandas
import matplotlib.pyplot as plt
```

```
abb_link = 'https://github.com/VitorKamada/ECO6100/raw/master/Data/Texas/listings.csv.gz'
lst = pd.read_csv(abb_link)
```

```
x = ['host_listings_count', 'bathrooms',
     'bedrooms', 'beds', 'guests_included']
```

Cleaning Data

```
def has_pool(a):
    if 'Pool' in a:
        return 1
    else:
        return 0

lst['pool'] = lst['amenities'].apply(has_pool)

yxs = lst.loc[:, x + ['pool', 'price']].dropna()
y = np.log(yxs['price'].apply(lambda x:
      float(x.strip('$').replace(',', '')))
      + 0.000001)
```

8 nearest neighbors

```
w = weights.KNN.from_array(lst.loc[yxs.index,
       ['longitude', 'latitude']].values)
w.transform = 'R'
```

```
m1 = spreg.OLS(y.values[:, None],
        yxs.drop('price', axis=1).values,
        w=w, spat_diag=True,
                name_x=yxs.drop('price',
                axis=1).columns.tolist(),
                 name_y='ln(price)')
print(m1.summary)
```

$ln(P) = \alpha + \beta X + \epsilon$

Variable	Coefficient	Std.Error	t-Statistic	Probability
CONSTANT	4.0976886	0.0223530	183.3171506	0.0000000
host_listings_count	-0.0000130	0.0001790	-0.0726772	0.9420655
bathrooms	0.2947079	0.0194817	15.1273879	0.0000000
bedrooms	0.3274226	0.0159666	20.5067654	0.0000000
beds	0.0245741	0.0097379	2.5235601	0.0116440
guests_included	0.0075119	0.0060551	1.2406028	0.2148030
pool	0.0888039	0.0221903	4.0019209	0.0000636

DIAGNOSTICS FOR SPATIAL DEPENDENCE

TEST	MI/DF	VALUE	PROB
Lagrange Multiplier (lag)	1	255.796	0.0000
Robust LM (lag)	1	13.039	0.0003
Lagrange Multiplier (error)	1	278.752	0.0000
Robust LM (error)	1	35.995	0.0000
Lagrange Multiplier (SARMA)	2	291.791	0.0000

Spatially Lagged Exogenous Regressors

$$ln(P_i) = \alpha + \beta X_i + \delta \sum_j w_{ij} X_i' + \epsilon_i$$

```
w_pool = weights.KNN.from_array(lst.loc[yxs.index,
                               ['longitude', 'latitude']].values)
```

```
yxs_w = yxs.assign(w_pool=weights.lag_spatial(w_pool,
                                           yxs['pool'].values))
```

```
m2 = spreg.OLS(y.values[:, None],
          yxs_w.drop('price', axis=1).values,
  w=w, spat_diag=True,
  name_x=yxs_w.drop('price', axis=1).columns.tolist(),
  name_y='ln(price)')

print(m2.summary)
```

print(m2.summary)

```
Sum squared residual:    3070.363          F-statistic            :     558.6139
Sigma-square        :       0.533          Prob(F-statistic)      :            0
S.E. of regression  :       0.730          Log likelihood         :    -6365.387
Sigma-square ML     :       0.532          Akaike info criterion  :    12746.773
S.E of regression ML:      0.7297          Schwarz criterion      :    12800.053
```

Variable	Coefficient	Std.Error	t-Statistic	Probability
CONSTANT	4.0906444	0.0230571	177.4134022	0.0000000
host_listings_count	-0.0000108	0.0001790	-0.0603617	0.9518697
bathrooms	0.2948787	0.0194813	15.1365024	0.0000000
bedrooms	0.3277450	0.0159679	20.5252404	0.0000000
beds	0.0246650	0.0097377	2.5329419	0.0113373
guests_included	0.0076894	0.0060564	1.2696250	0.2042695
pool	0.0725756	0.0257356	2.8200486	0.0048181
w_pool	0.0188875	0.0151729	1.2448141	0.2132508

Spatially Lagged Endogenous Regressors

$$ln(P_i) = \alpha + \lambda \sum_j w_{ij} ln(P_i) + \beta X_i + \epsilon_i$$

```
m3 = spreg.GM_Lag(y.values[:, None],
    yxs.drop('price', axis=1).values,
    w=w, spat_diag=True,
    name_x=yxs.drop('price',
                axis=1).columns.tolist(),
    name_y='ln(price)')

print(m3.summary)
```

Spatial 2SLS

```
Dependent Variable   :    ln(price)          Number of Observations:      5767
Mean dependent var   :       5.1952          Number of Variables    :        8
S.D. dependent var   :       0.9455          Degrees of Freedom     :     5759
Pseudo R-squared     :       0.4224
Spatial Pseudo R-squared:    0.4056
```

Variable	Coefficient	Std.Error	z-Statistic	Probability
CONSTANT	3.7085715	0.1075621	34.4784213	0.0000000
host_listings_count	-0.0000587	0.0001765	-0.3324585	0.7395430
bathrooms	0.2857932	0.0193237	14.7897969	0.0000000
bedrooms	0.3272598	0.0157132	20.8270544	0.0000000
beds	0.0239548	0.0095848	2.4992528	0.0124455
guests_included	0.0065147	0.0059651	1.0921407	0.2747713
pool	0.0891100	0.0218383	4.0804521	0.0000449
W_ln(price)	0.0785059	0.0212424	3.6957202	0.0002193

```
Instrumented: W_ln(price)
Instruments: W_bathrooms, W_bedrooms, W_beds, W_guests_included,
             W_host_listings_count, W_pool
```

Spatial Durbin Model (SDM)

$$(I_n - \rho W)y = X\beta + WX\theta + \epsilon$$

$$y = \sum_{r=1}^{k} S_r(W)x_r + V(W)\epsilon$$

$$V(W) = (I_n - \rho W)^{-1}$$

$$S_r(W) = V(W)(I_n\beta_r + W\theta_r)$$

Summary Measures of Impacts

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \sum_{r=1}^{k} \begin{pmatrix} S_r(W)_{11} & S_r(W)_{12} & \cdots & S_r(W)_{1n} \\ S_r(W)_{21} & S_r(W)_{22} & & \\ \vdots & \vdots & \ddots & \\ S_r(W)_{n1} & S_r(W)_{n2} & & S_r(W)_{nn} \end{pmatrix} \begin{pmatrix} x_{1r} \\ x_{2r} \\ \vdots \\ x_{nr} \end{pmatrix} + V(W)\epsilon$$

$$\frac{\partial y_i}{\partial x_{jr}} = S_r(W)_{ij} \textbf{ and } \frac{\partial y_i}{\partial x_{ir}} = S_r(W)_{ii}$$

$$\bar{M}(r)_{direct} = n^{-1} tr(S_r(W))$$

$$\bar{M}(r)_{total} = n^{-1} \iota_n'(S_r(W))\iota_n$$

$$\bar{M}(r)_{indirect} = \bar{M}(r)_{total} - \bar{M}(r)_{direct}$$

Spatial Autoregressive (SAR) Model

$$(I_n - \rho W)y = X\beta + \epsilon$$

$$y = \sum_{r=1}^{k} (I_n - \rho W)^{-1} I_n \beta_r x_r + (I_n - \rho W)^{-1}\epsilon$$

Total Impact

$$n^{-1}\iota_n'(I_n - \rho W)^{-1}\beta_r \iota_n = (1-\rho)^{-1}\beta_r$$

Indirect Impact: $\frac{\beta_r}{(1-\rho)} - \beta_r$

Code: Direct & Indirect Impacts

```
b = m3.betas[:-1]
b

rho = m3.betas[-1]
rho

btot = b / (1.0 - rho) #total impact
bind = btot - b #indirect impact

x_names = ['NROOM','NBATH','PATIO','FIREPL','AC','GAR','AGE',
           'LOTSZ','SQFT']
varnames = ["CONSTANT"] + x + ["pool"]
print("              Variable        Direct        Indirect   \
    Total" )
for i in range(len(varnames)):
    print("%20s %12.3f %12.3f %12.3f" % (varnames[i],b[i][0],
                                      bind[i][0],btot[i][0]))
```

Direct & Indirect Impacts

$$ln(P_i) = \alpha + \lambda \sum_j w_{ij} ln(P_i) + \beta X_i + \epsilon_i$$

Variable	Direct	Indirect	Total
CONSTANT	3.709	0.316	4.025
host_listings_count	-0.000	-0.000	-0.000
bathrooms	0.286	0.024	0.310
bedrooms	0.327	0.028	0.355
beds	0.024	0.002	0.026
guests_included	0.007	0.001	0.007
pool	0.089	0.008	0.097