Title

stata.com

spregress — Spatial autoregressive models

Description

`spregress` is the equivalent of `regress` for spatial data. `spregress` fits spatial autoregressive (SAR) models, also known as simultaneous autoregressive models. If you have not read [SP] **Intro 1**–[SP] **Intro 8**, you should do so before using `spregress`.

To use `spregress`, your data must be Sp data. See [SP] **Intro 3** for instructions on how to prepare your data.

To specify spatial lags, you will need to have one or more spatial weighting matrices. See [SP] **Intro 2** and [SP] **spmatrix** for an explanation of the types of weighting matrices and how to create them.

Quick start

Spatial autoregressive model of `y` on `x1` and `x2` with a spatial lag of `y` specified by the spatial weighting matrix `W` using the GS2SLS estimator

```
spregress y x1 x2, gs2sls dvarlag(W)
```

Add a spatially lagged error term also specified by `W`

```
spregress y x1 x2, gs2sls dvarlag(W) errorlag(W)
```

Add spatial lags of covariates `x1` and `x2`

```
spregress y x1 x2, gs2sls dvarlag(W) errorlag(W) ivarlag(W: x1 x2)
```

Add a higher-order spatial lag of `y` specified by another weighting matrix `M`

```
spregress y x1 x2, gs2sls dvarlag(W) errorlag(W) ivarlag(W: x1 x2) ///
     dvarlag(M)
```

Use the ML estimator and include spatial lags of `y`, `x1`, `x2` and the error term specified by `W`

```
spregress y x1 x2, ml dvarlag(W) errorlag(W) ivarlag(W: x1 x2)
```

Add an additional spatial lag of the covariates specified by the matrix `M`

```
spregress y x1 x2, ml dvarlag(W) errorlag(W) ivarlag(W: x1 x2)     ///
     ivarlag(M: x1 x2)
```

Same model fit by GS2SLS

```
spregress y x1 x2, gs2sls dvarlag(W) errorlag(W) ivarlag(W: x1 x2)  ///
     ivarlag(M: x1 x2)
```

Model fit by GS2SLS with spatial lags of `y` and of the error term and treating the errors as heteroskedastic

```
spregress y x1 x2, gs2sls heteroskedastic dvarlag(W) errorlag(W)
```

Menu

Statistics > Spatial autoregressive models

Syntax

Generalized spatial two-stage least squares

```
spregress depvar [indepvars] [if] [in], gs2sls [gs2sls_options]
```

Maximum likelihood

```
spregress depvar [indepvars] [if] [in], ml [ml_options]
```

gs2sls_options	Description
Model	
* gs2sls	use generalized spatial two-stage least-squares estimator
dvarlag(*spmatname*)	spatially lagged dependent variable; repeatable
errorlag(*spmatname*)	spatially lagged errors; repeatable
ivarlag(*spmatname* : *varlist*)	spatially lagged independent variables; repeatable
noconstant	suppress constant term
heteroskedastic	treat errors as heteroskedastic
force	allow estimation when estimation sample is a subset of the sample used to create the spatial weighting matrix
impower(#)	order of instrumental-variable approximation
Reporting	
level(#)	set confidence level; default is level(95)
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
optimization_options	control the optimization process; seldom used
coeflegend	display legend instead of statistics

ml_options	Description
Model	
* `ml`	use maximum likelihood estimator
`dvarlag(`*spmatname*`)`	spatially lagged dependent variable; not repeatable
`errorlag(`*spmatname*`)`	spatially lagged errors; not repeatable
`ivarlag(`*spmatname* `:` *varlist*`)`	spatially lagged independent variables; repeatable
`noconstant`	suppress constant term
`constraints(`*constraints*`)`	apply specified linear constraints
`force`	allow estimation when estimation sample is a subset of the sample used to create the spatial weighting matrix
`gridsearch(#)`	resolution of the initial-value search grid; seldom used
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim` or `robust`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
maximize_options	control the maximization process; seldom used
`coeflegend`	display legend instead of statistics

* You must specify either `gs2sls` or `ml`.

indepvars and *varlist* specified in `ivarlag()` may contain factor variables; see [U] **11.4.3 Factor variables**.

`collect` is allowed; see [U] **11.1.10 Prefix commands**.

`coeflegend` does not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options for spregress, gs2sls

Model

`gs2sls` requests that the generalized spatial two-stage least-squares estimator be used.

`dvarlag(`*spmatname*`)` specifies a spatial weighting matrix that defines a spatial lag of the dependent variable. This option is repeatable to allow higher-order models. By default, no spatial lags of the dependent variable are included.

`errorlag(`*spmatname*`)` specifies a spatial weighting matrix that defines a spatially lagged error. This option is repeatable to allow higher-order models. By default, no spatially lagged errors are included.

`ivarlag(`*spmatname* `:` *varlist*`)` specifies a spatial weighting matrix and a list of independent variables that define spatial lags of the variables. This option is repeatable to allow spatial lags created from different matrices. By default, no spatial lags of the independent variables are included.

`noconstant`; see [R] **Estimation options**.

`heteroskedastic` specifies that the estimator treat the errors as heteroskedastic instead of homoskedastic, which is the default; see *Methods and formulas*.

`force` requests that estimation be done when the estimation sample is a proper subset of the sample used to create the spatial weighting matrices. The default is to refuse to fit the model. Weighting matrices potentially connect all the spatial units. When the estimation sample is a subset of this space, the spatial connections differ and spillover effects can be altered. In addition, the normalization of the weighting matrix differs from what it would have been had the matrix been normalized over the estimation sample. The better alternative to `force` is first to understand the spatial space of the estimation sample and, if it is sensible, then create new weighting matrices for it. See [SP] **spmatrix** and *Missing values, dropped observations, and the W matrix* in [SP] **Intro 2**.

`impower(#)` specifies the order of an instrumental-variable approximation used in fitting the model. The derivation of the estimator involves a product of # matrices. Increasing # may improve the precision of the estimation and will not cause harm, but will require more computer time. The default is `impower(2)`. See *Methods and formulas* for additional details on `impower(#)`.

Reporting

`level(#)`; see [R] **Estimation options**.

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(%`*fmt*`)`, `pformat(%`*fmt*`)`, `sformat(%`*fmt*`)`, and `nolstretch`; see [R] **Estimation options**.

Optimization

optimization_options: `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, and `nonrtolerance`; see [M-5] **optimize( )**.

The following option is available with `spregress, gs2sls` but is not shown in the dialog box:

`coeflegend`; see [R] **Estimation options**.

Options for spregress, ml

Model

`ml` requests that the maximum likelihood estimator be used.

`dvarlag(`*spmatname*`)` specifies a spatial weighting matrix that defines a spatial lag of the dependent variable. Only one `dvarlag()` option may be specified. By default, no spatial lags of the dependent variable are included.

`errorlag(`*spmatname*`)` specifies a spatial weighting matrix that defines a spatially lagged error. Only one `errorlag()` option may be specified. By default, no spatially lagged errors are included.

`ivarlag(`*spmatname* `:` *varlist*`)` specifies a spatial weighting matrix and a list of independent variables that define spatial lags of the variables. This option is repeatable to allow spatial lags created from different matrices. By default, no spatial lags of the independent variables are included.

`noconstant`, `constraints(`*constraints*`)`; see [R] **Estimation options**.

`force` requests that estimation be done when the estimation sample is a proper subset of the sample used to create the spatial weighting matrices. The default is to refuse to fit the model. This is the same `force` option described for use with `spregress, gs2sls`.

`gridsearch(#)` specifies the resolution of the initial-value search grid. The default is `gridsearch(0.1)`. You may specify any number between 0.001 and 0.1 inclusive.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`) and that are robust to nonnormal independent and identically distributed (i.i.d.) disturbance (`robust`). See [R] ***vce_option***.

Reporting

`level(#)`, `nocnsreport`; see [R] **Estimation options**.

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(%`*fmt*`)`, `pformat(%`*fmt*`)`, `sformat(%`*fmt*`)`, and `nolstretch`; see [R] **Estimation options**.

Maximization

maximize_options: `difficult`, `technique(`*algorithm_spec*`)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, and `nonrtolerance`; see [R] **Maximize**.

The following option is available with `spregress, ml` but is not shown in the dialog box:

`coeflegend`; see [R] **Estimation options**.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Introduction

See [SP] **Intro 1**–[SP] **Intro 8** for an overview of SAR models. The introductions also describe, in detail and with examples, how to prepare your data for analysis with `spregress` and the other Sp estimation commands.

Datasets for SAR models contain observations on geographical areas or other units; all that is required is that there be some measure of distance that distinguishes which units are close to each other. The `spregress` command models cross-sectional data. It requires each observation to represent one unique spatial unit. For data with multiple observations on each unit—namely, panel data—see [SP] **spxtregress**.

To fit models with endogenous regressors for cross-sectional data, see [SP] **spivregress**.

`spregress, gs2sls` uses a generalized method of moments estimator known as generalized spatial two-stage least squares (GS2SLS). `spregress, ml` uses a maximum likelihood (ML) estimator. For normally distributed data, `ml` is theoretically more efficient than `gs2sls`, but when data are i.i.d., `spregress, gs2sls` produces results that are not appreciably different from those of `spregress, ml`. See *Methods and formulas*.

The vce(robust) variance estimator can be used with spregress, ml to produce standard errors that are robust to nonnormal i.i.d. errors; see [R] *vce_option*. spregress, ml can produce inconsistent estimates with data that are not identically distributed.

spregress, gs2sls has a heteroskedastic option that relaxes the assumption that errors are i.i.d. With the heteroskedastic option, errors only need to be independent; see example 2.

Choosing weighting matrices and their normalization

Weighting matrices

It is important to understand that the choice of weighting matrices is part of your SAR model specification.

The choice of weighting matrix should be based on the formulation of your research question. Does it make sense to define spatial lags based on only neighboring areas? Or do you want to model effects across distances that decrease with increasing distance? Or do you want to model spatial lags based on some measure in your data, for example, the value of imports and exports between countries?

The Sp system has the spmatrix create command, which can create contiguity matrices and inverse-distance matrices. For instance, typing

```
spmatrix create contiguity W
```

creates a symmetric weighting matrix, W, that has the same positive weight for contiguous spatial units and, by default, a zero weight for all other units, with an option to include nonzero weights for second-order neighbors (neighbors of neighbors). There are also Sp commands for creating custom weighting matrices. See [SP] **Intro 2** and [SP] **spmatrix** for details.

Both spregress, gs2sls and spregress, ml can fit models with multiple spatial lags of the independent variables. You can specify multiple ivarlag() options with different spatial weighting matrices for the same or different variables.

With the gs2sls estimator, you can also include dependent-variable spatial lags and autoregressive error terms specified by two or more spatial weighting matrices. You do this by specifying multiple dvarlag() options or multiple errorlag() options. Multiple weighting matrices can be viewed as providing a "higher-order" approximation to the true dependent variable or error spatial dependence, and they allow testing of the formulation of the spatial lag.

With the ml estimator, you can include only one dvarlag() and one errorlag(), but each can have its own, possibly different, spatial weighting matrix.

Normalization of weighting matrices

spmatrix create by default normalizes the weighting matrix it creates by dividing the entries by the absolute value of the largest eigenvalue of the matrix; this is the normalize(spectral) option. The normalize(minmax) option scales the matrix using either the maximum of column sums or the maximum of the row sums, whichever is smaller. The normalize(row) option scales each row of the matrix by its row sum, so that each row sums to one.

You may have also created your own weighting matrix with good properties for the estimator. In this case, you may want to leave the matrix unnormalized using the normalize(none) option.

What are the differences among the three normalizations?

There are two reasons to normalize: interpretability of the spatial lag coefficients and estimability.

`normalize(spectral)` and `normalize(minmax)` produce matrices that differ from the original only by a scalar multiple. This not true for `normalize(row)`, so let's discuss it first.

Row normalization, `normalize(row)`, has a long history and is popular in applied work. Row normalization can potentially multiply different rows by different scalars, and if it does so, that changes the model specification given by the weighting matrix. For example, if you start with a contiguity matrix, and the first row has two 1s and the second row has four 1s, then after row normalization, the first row contains two halves and the second four quarters. This amounts to spreading the potential spillover effects of each spatial unit equally across its neighbors, whereas the original unnormalized contiguity matrix modeled equal potential spillover effects for each neighbor regardless of the number of neighbors. `normalize(row)` also transforms a symmetric contiguity matrix into an asymmetric matrix. Row normalization should be used when the spatial lags it specifies are appropriate for your research question and when the lags of the original matrix are not.

When the unnormalized matrix has been formulated to match your research question, there is the choice of `normalize(spectral)`, `normalize(minmax)`, or `normalize(none)`. The choice affects the interpretation of the spatial lag coefficients.

Because dependent-variable spatial lags enter the model as $\lambda\mathbf{Wy}$, covariate lags enter as $\gamma\mathbf{Wx}$, and the autoregressive errors are modeled using $\rho\mathbf{We}$, we would expect the spatial lag coefficient estimates to scale inversely by the scale of $\mathbf{W}$. If $\mathbf{W}$ is scaled by c to become $\mathbf{W}/c$, then $\widehat{\lambda}$ becomes $c\widehat{\lambda}$, $\widehat{\gamma}$ becomes $c\widehat{\gamma}$, and $\widehat{\rho}$ becomes $c\widehat{\rho}$.

For example, if an unnormalized matrix results in an estimation of $\widehat{\rho}_{\text{unnorm}} = 0.1$, and if the matrix is then scaled by $c = 5$, the estimation using the normalized matrix would yield $\widehat{\rho}_{\text{norm}} = 0.5$. So what we want for the interpretation of the parameter estimate is a scaling where $\widehat{\rho}_{\text{norm}}$ is typically in the range -1 to 1. Recall from the discussion in [SP] **Intro 2** and [SP] **Intro 7** that ρ is not a true correlation, only something like a correlation. There is no guarantee that the estimate for it will be between -1 and 1. In an explosive model, the estimate will be outside this range.

The scaling factor c from `normalize(spectral)` is always less than or equal to the scaling factor from `normalize(minmax)`. So for the same model run with different normalizations, `minmax` will result in an estimate $\widehat{\rho}_{\text{minmax}}$ that is larger than $\widehat{\rho}_{\text{spectral}}$, the estimate resulting from using `spectral`. So the spectral normalization is more likely to produce estimates of ρ in the range -1 to 1.

The second reason for normalization is estimability. The scaling from `normalize(spectral)` guarantees nonsingularity of certain terms in the model estimation; see *Methods and formulas*. The bigger scaling of `normalize(minmax)`, of course, also guarantees nonsingularity, but it is a bigger scaling than necessary.

Row normalization also guarantees nonsingularity, but because it is not a scalar multiple of the unnormalized matrix, we cannot in general say how it will change the spatial lag coefficient estimates relative to the estimates produced using the unnormalized matrix. Row normalization, as we said earlier, results in a different model specification.

You may have created your own weighting matrix, and you know that based on its properties and the form of the estimator that it will not yield singularities. In this case, you need not normalize. If an unnormalized matrix, however, causes a singularity in the estimation, you may get "wrong" estimation results, that is, ones differing by other than a scale factor from those using a spectral or min–max normalization.

`spmatrix create` and other Sp matrix commands use spectral normalization by default because it is the smallest scaling that in general guarantees nonsingularity without changing the model specification of the original matrix. However, `normalize(spectral)` is computationally expensive. It can take

a long time for large matrices. If this is a consideration, `normalize(minmax)` is faster to compute and will yield results that are close to those of `normalize(spectral)`.

Direct and indirect effects and normalization

Direct and indirect, also called spillover, effects were discussed in [SP] **Intro 1** and [SP] **Intro 2**. In example 1 below, we show how to get these estimates using the `estat impact` command.

The scaling property between the spectral and min–max normalizations and the spatial lag coefficient estimates that we described in the previous section implies that the estimates of the direct and indirect effects should be scale invariant. `spregress, ml` has this scaling property and gives scale-invariant effects. When there is no autoregressive error term, `spregress, gs2sls` also has this scaling property and gives scale-invariant effects. When there is an autoregressive error term, however, the GS2SLS estimator is only asymptotically scale invariant.

Practically speaking, this means when you use `estat impact` to look at the direct and indirect effects of the covariates after `spregress, ml` in all cases, or `spregress, gs2sls` with no `errorlag()`, you will get results differing only by numerical precision whether you used `normalize(spectral)`, `normalize(minmax)`, or an unnormalized matrix with sound numerical properties.

The GS2SLS estimator, however, is a nonlinear function of the weighting matrix when an autoregressive error term is included. For this nonlinear GS2SLS estimator, models are well defined only if the coefficient on the spatial lag of the dependent variable and the coefficient on the spatially lagged error lie within certain intervals. Normalizing the weighting matrix by the spectral normalization or the row normalization puts the estimates in these intervals when there are no higher-order lags. Because min–max normalization is a close approximation to spectral normalization, the resulting estimates should be close.

Again, practically speaking, this means that even though `normalize(spectral)` and `normalize(minmax)` both simply multiply the original matrix by a scalar, and the scalars are similar in size, `estat impact` may give slightly different estimates depending on the normalization for the GS2SLS estimator with an autoregressive error term. This is especially the case in small samples, and the differences will decrease as the sample size increases.

Of course, the `normalize(row)` normalization will yield different estimates of effects compared with the other normalizations or with no normalization because row normalization results in a different model specification.

In higher-order models with GS2SLS and autoregressive error terms, the estimator is a nonlinear function of multiple weighting matrices. The sets of spatial lag coefficients for which the models are well defined are multidimensional regions, but the same normalizations are used, and the tradeoffs mentioned above still apply.

Examples

▷ Example 1: A spatial autoregressive model

We want to model the homicide rate in counties in southern states of the United States. `homicide1990.dta` contains `hrate`, the county-level homicide rate per year per 100,000 persons; `ln_population`, the logarithm of the county population; `ln_pdensity`, the logarithm of the population density; and `gini`, the Gini coefficient for the county, a measure of income inequality where larger values represent more inequality (Gini 1909). The data are an extract of the data originally used by Messner et al. (2000); see Britt (1994) for a literature review of the topic.

We used spshape2dta to create homicide1990.dta and homicide1990_shp.dta. The latter file contains the boundary coordinates for U.S. southern counties. See [SP] **Intro 4**, [SP] **Intro 7**, [SP] **spshape2dta**, and [SP] **spset**.

Because the analysis dataset and the Stata-formatted shapefile must be in our working directory to spset the data, we first save both homicide1990.dta and homicide1990_shp.dta to our working directory by using the copy command. We then load the data and type spset to see the Sp settings.

```
. copy https://www.stata-press.com/data/r17/homicide1990.dta .
. copy https://www.stata-press.com/data/r17/homicide1990_shp.dta .
. use homicide1990
(S.Messner et al.(2000), U.S southern county homicide rates in 1990)
. spset
      Sp dataset: homicide1990.dta
Linked shapefile: homicide1990_shp.dta
            Data: Cross sectional
 Spatial-unit ID: _ID
     Coordinates: _CX, _CY (planar)
```

We plot the homicide rate on a map of the counties by using the grmap command; see [SP] **grmap**. Figure 1 is the result.

```
. grmap hrate
```

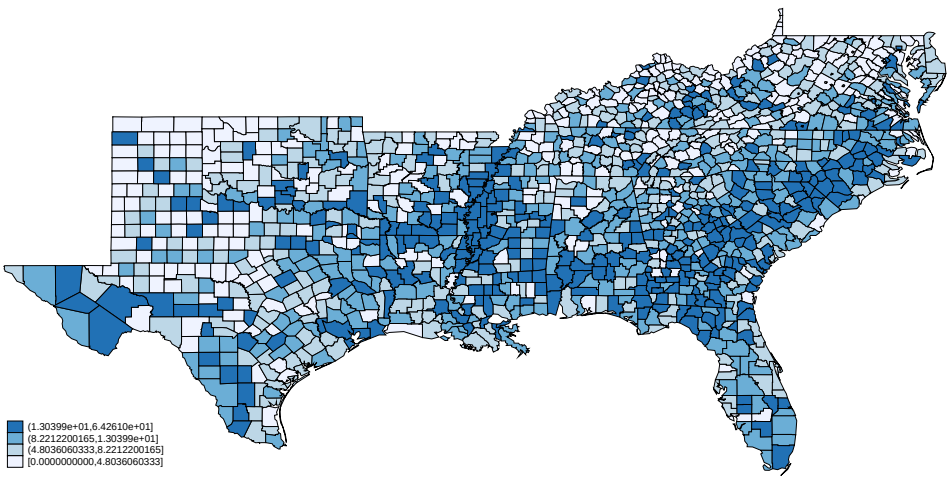

Figure 1: Homicide rate in 1990 for southern U.S. counties

The homicide rate appears to be spatially dependent because the high homicide-rate counties appear to be clustered together. As described in [SP] **Intro 7**, we can fit an ordinary linear regression and test whether the errors are spatially correlated using the Moran test.

To conduct the test, we need a spatial weighting matrix. We will create one that puts the same positive weight on contiguous counties and a zero weight on all other counties—a matrix known as a contiguity matrix. We will use the default spectral normalization for the matrix. See [SP] **Intro 2**, [SP] **spmatrix create**, and *Choosing weighting matrices and their normalization* above for details. We type

```
. spmatrix create contiguity W
```

To create W, spmatrix used the coordinate data in homicide1990_shp.dta behind the scenes.

Now, we run regress and then estat moran.

```
. regress hrate
```

Source	SS	df	MS	Number of obs	=	1,412
				F(0, 1411)	=	0.00
Model	0	0	.	Prob > F	=	.
Residual	69908.59	1,411	49.5454217	R-squared	=	0.0000
				Adj R-squared	=	0.0000
Total	69908.59	1,411	49.5454217	Root MSE	=	7.0389

hrate	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
_cons	9.549293	.1873201	50.98	0.000	9.181837	9.916749

```
. estat moran, errorlag(W)

Moran test for spatial dependence
        H0: Error terms are i.i.d.
        Errorlags:  W

        chi2(1)      =   265.84
        Prob > chi2  =   0.0000
```

The test reports that we can reject that the errors are i.i.d. and confirms our visual appraisal of the data.

To model the homicide rate hrate, we will use the GS2SLS estimator and specify the option dvarlag(W) to fit a model with a spatial lag of hrate based on W.

```
. spregress hrate ln_population ln_pdensity gini, gs2sls dvarlag(W)
  (1412 observations)
  (1412 observations (places) used)
  (weighting matrix defines 1412 places)

Spatial autoregressive model                    Number of obs =  1,412
GS2SLS estimates                                Wald chi2(4)  = 328.40
                                                Prob > chi2   = 0.0000
                                                Pseudo R2     = 0.1754
```

hrate	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
hrate						
ln_populat~n	.195714	.2654999	0.74	0.461	-.3246563	.7160843
ln_pdensity	1.060728	.2303736	4.60	0.000	.6092043	1.512252
gini	77.10293	5.330446	14.46	0.000	66.65544	87.55041
_cons	-28.79865	2.945944	-9.78	0.000	-34.57259	-23.02471
W						
hrate	.2270154	.0607158	3.74	0.000	.1080146	.3460161

```
Wald test of spatial terms:          chi2(1) = 13.98      Prob > chi2 = 0.0002
```

The estimated coefficient on the spatial lag of hrate is 0.23, indicating positive correlation between the homicide rate in one county and the homicide rate in a neighboring county.

As we discussed in [SP] **Intro 7** the coefficients cannot be interpreted as they are in standard regression models. We can use estat impact to interpret results, but first we will illustrate how to fit other SAR models.

We now include a spatial autoregressive error term by adding `errorlag(W)`.

```
. spregress hrate ln_population ln_pdensity gini, gs2sls dvarlag(W) errorlag(W)
  (1412 observations)
  (1412 observations (places) used)
  (weighting matrix defines 1412 places)
Estimating rho using 2SLS residuals:
initial:       GMM criterion =  16.837319
alternative:   GMM criterion =  10.842722
rescale:       GMM criterion =  1.1522691
Iteration 0:   GMM criterion =  1.1522691
Iteration 1:   GMM criterion =  1.1386586
Iteration 2:   GMM criterion =  1.1386578
Iteration 3:   GMM criterion =  1.1386578
Estimating rho using GS2SLS residuals:
Iteration 0:   GMM criterion =  .02771702
Iteration 1:   GMM criterion =   .0262056
Iteration 2:   GMM criterion =  .02606375
Iteration 3:   GMM criterion =  .02601873
Iteration 4:   GMM criterion =  .02601004
Iteration 5:   GMM criterion =  .02600789
Iteration 6:   GMM criterion =  .02600742
Iteration 7:   GMM criterion =  .02600731
Iteration 8:   GMM criterion =  .02600729
Spatial autoregressive model                    Number of obs =  1,412
GS2SLS estimates                                Wald chi2(4)  = 276.72
                                                Prob > chi2   = 0.0000
                                                Pseudo R2     = 0.1736
```

hrate	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
hrate						
ln_populat~n	.1034997	.2810656	0.37	0.713	-.4473787	.6543781
ln_pdensity	1.081404	.2520505	4.29	0.000	.5873939	1.575414
gini	82.0687	5.658372	14.50	0.000	70.9785	93.1589
_cons	-29.63033	3.070332	-9.65	0.000	-35.64807	-23.61259
W						
hrate	.1937419	.0654322	2.96	0.003	.0654972	.3219867
e.hrate	.3555443	.0786465	4.52	0.000	.2014	.5096887

```
Wald test of spatial terms:          chi2(2) = 226.21     Prob > chi2 = 0.0000
. estimates store gs2sls_model
```

Note that when an autoregressive error term is included, the estimation procedure becomes an iterative generalized method of moments procedure.

We keep the SAR error term `e.hrate` in our model and add terms representing spatial lags of the independent variables by using `ivarlag(W: ...)`.

```
. spregress hrate ln_population ln_pdensity gini, gs2sls dvarlag(W) errorlag(W)
> ivarlag(W: ln_population ln_pdensity gini)
  (1412 observations)
  (1412 observations (places) used)
  (weighting matrix defines 1412 places)
```

(output omitted)

```
Spatial autoregressive model                         Number of obs =  1,412
GS2SLS estimates                                     Wald chi2(7)  = 394.61
                                                     Prob > chi2   = 0.0000
                                                     Pseudo R2     = 0.1866
```

hrate	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
hrate						
ln_populat~n	-.3489221	.3050009	-1.14	0.253	-.9467129	.2488687
ln_pdensity	1.210485	.3015442	4.01	0.000	.6194695	1.801501
gini	89.17773	6.454876	13.82	0.000	76.5264	101.8291
_cons	-28.80191	3.178656	-9.06	0.000	-35.03196	-22.57186
W						
ln_populat~n	1.918436	.4598247	4.17	0.000	1.017196	2.819676
ln_pdensity	-1.260725	.5326521	-2.37	0.018	-2.304704	-.2167459
gini	-43.4606	8.607378	-5.05	0.000	-60.33075	-26.59045
hrate	.5071798	.1139532	4.45	0.000	.2838356	.730524
e.hrate	-.3135187	.1396411	-2.25	0.025	-.5872103	-.0398271

```
Wald test of spatial terms:          chi2(5) = 61.81      Prob > chi2 = 0.0000
```

The coefficients for the lagged variables and the autoregressive error term are significant.

We are often unsure which spatial weighting matrix should be used to compute spatial lags. Many researchers use a spatial weighting matrix whose (i, j)th element is the inverse of the distance between units i and j. This inverse-distance matrix has many nice properties and a long history in spatial analysis; see [SP] **spmatrix** and *Choosing weighting matrices and their normalization* above.

With the GS2SLS estimator, we can include spatial lags using two spatial weighting matrices, in which case we might view them as together providing a "higher-order" approximation to the true spatial process. We had in our model a spatial lag of the dependent variable using a contiguity matrix alone. Now, we will include that and another lag of the dependent variable using an inverse-distance matrix.

We create the inverse-distance matrix `M` with the default spectral normalization and use `spmatrix dir` to list our Sp matrices.

```
. spmatrix create idistance M
. spmatrix dir
```

Weighting matrix name	N x N	Type	Normalization
M	1412 x 1412	idistance	spectral
W	1412 x 1412	contiguity	spectral

Now, we add dvarlag(M) to our model.

```
. spregress hrate ln_population ln_pdensity gini, gs2sls dvarlag(W) errorlag(W)
> ivarlag(W: ln_population ln_pdensity gini) dvarlag(M)
  (1412 observations)
  (1412 observations (places) used)
  (weighting matrices define 1412 places)
```

(*output omitted*)

```
Spatial autoregressive model                    Number of obs =   1,412
GS2SLS estimates                                Wald chi2(8)  = 1323.43
                                                Prob > chi2   =  0.0000
                                                Pseudo R2     =  0.1121
```

hrate	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
hrate						
ln_populat~n	-.6245265	.2830848	-2.21	0.027	-1.179362	-.0696905
ln_pdensity	1.266527	.2831372	4.47	0.000	.7115887	1.821466
gini	69.30289	5.64501	12.28	0.000	58.23887	80.36691
_cons	-19.77152	2.753498	-7.18	0.000	-25.16828	-14.37476
W						
ln_populat~n	2.590823	.3806543	6.81	0.000	1.844754	3.336892
ln_pdensity	-2.63202	.4261689	-6.18	0.000	-3.467296	-1.796745
gini	-59.75958	6.438899	-9.28	0.000	-72.37959	-47.13957
hrate	.926941	.0492867	18.81	0.000	.830341	1.023541
e.hrate	-.853115	.0914652	-9.33	0.000	-1.032384	-.6738464
M						
hrate	.2289788	.0755038	3.03	0.002	.0809941	.3769635

```
Wald test of spatial terms:          chi2(6) = 676.93     Prob > chi2 = 0.0000
```

The hrate lag specified by M is significant in addition to the hrate lag specified by W. We may well want to include both in our final model.

We could repeat the process, fitting a model with errorlag(M) in addition to errorlag(W), and another model with ivarlag(M: ...) in addition to ivarlag(W: ...). One issue is that we have "only" $N = 1412$ spatial units (observations) in this example. To fit higher-order lags, one needs lots of spatial units, so we need to exercise judgment just as in any other model-building process. In our final model, we keep a single weighting matrix for each term. We use W for dvarlag() and ivarlag(), but M for errorlag().

```
. spregress hrate ln_population ln_pdensity gini, gs2sls dvarlag(W) errorlag(M)
> ivarlag(W: ln_population ln_pdensity gini)
  (1412 observations)
  (1412 observations (places) used)
  (weighting matrices define 1412 places)
```

(*output omitted*)

```
Spatial autoregressive model                    Number of obs =  1,412
GS2SLS estimates                                Wald chi2(7)  = 357.06
                                                Prob > chi2   = 0.0000
                                                Pseudo R2     = 0.1241
```

hrate	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
hrate						
ln_populat~n	-.0475582	.3295548	-0.14	0.885	-.6934737	.5983573
ln_pdensity	.8989538	.3211524	2.80	0.005	.2695066	1.528401
gini	89.91969	6.409286	14.03	0.000	77.35772	102.4817
_cons	-32.21599	3.590014	-8.97	0.000	-39.25229	-25.17969
W						
ln_populat~n	2.679931	.5218152	5.14	0.000	1.657192	3.702669
ln_pdensity	-2.468953	.6209688	-3.98	0.000	-3.686029	-1.251876
gini	-57.38302	9.418108	-6.09	0.000	-75.84217	-38.92387
hrate	.6818566	.1141573	5.97	0.000	.4581125	.9056007
M						
e.hrate	.9533048	.1324392	7.20	0.000	.6937289	1.212881

```
Wald test of spatial terms:          chi2(5) = 169.23     Prob > chi2 = 0.0000
. estimates store model_ex1_last
```

In [SP] **Intro 7**, we cautioned that interpreting covariate effects based on their coefficient estimates is difficult when there is a dependent-variable lag or an independent-variable lag in the model.

The spatial lag of `hrate` modifies the covariate effects. A change in `gini` in a county changes the conditional mean of `hrate` in that county, and that change in `hrate` changes the conditional mean of `hrate` in all contiguous counties. The change in `hrate` in these counties then affects `hrate` in all counties contiguous to them, and so on, until all counties linked by a chain of contiguous counties are affected.

The effects of a covariate vary over the counties because of how the spatial lag of `hrate` modifies the covariate effects. There are as many effects of a covariate as there are spatial units. As discussed by LeSage and Pace (2009, sec. 2.7), we define the average of these spatial unit-level effects to be the covariate effect.

The effect of `gini` on the conditional mean of `hrate` in other counties is called an indirect, or spillover, effect.

Because a spatial lag of `gini` is included in the model, there is a second indirect effect. The equation for `hrate` includes a term for `gini` in neighboring counties, so a change in `gini` in one county changes the conditional mean of `hrate` in neighboring counties.

The effect of `gini` on the conditional mean of `hrate` in the same county is called a direct, or own, effect. The sum of the direct and indirect effects is called the total effect.

We use estat impact to estimate the magnitude of these effects.

```
. estat impact
progress   : 33%  67% 100%
Average impacts                                 Number of obs     =      1,412
```

	Delta-Method dy/dx	std. err.	z	P>|z|	[95% conf.	interval]
direct						
ln_populat~n	.3149608	.3545409	0.89	0.374	-.3799266	1.009848
ln_pdensity	.6448149	.3426066	1.88	0.060	-.0266817	1.316311
gini	90.45773	6.380729	14.18	0.000	77.95173	102.9637
indirect						
ln_populat~n	5.856241	2.256561	2.60	0.009	1.433463	10.27902
ln_pdensity	-4.105437	1.883462	-2.18	0.029	-7.796956	-.413919
gini	8.691593	19.58268	0.44	0.657	-29.68975	47.07294
total						
ln_populat~n	6.171202	2.411894	2.56	0.011	1.443976	10.89843
ln_pdensity	-3.460622	2.029163	-1.71	0.088	-7.437708	.5164636
gini	99.14932	21.03394	4.71	0.000	57.92356	140.3751

See the percentages at the top of the output? For large datasets, calculating standard errors of the effects can be time consuming, so estat impact reports its progress as it does the computations.

The direct effect of gini is positive because the coefficient of gini is positive. The indirect effect of gini due to the spatial lag of hrate is positive because the coefficient of the dependent-variable lag is positive and the coefficient of gini is positive. The indirect effect of gini due to its spatial lag, however, is negative because the coefficient of its lag is negative. estat impact shows that the two indirect effects of gini sum to a net positive indirect effect, although the sum is not significantly different from 0.

Note that the normalization of W affects the size of the coefficient estimates for the lags of the covariates. For the GS2SLS estimator, the normalization of W (except for the case of row normalization) does not affect the asymptotic estimates of the covariate effects. In finite samples, this means that the normalization of W may have a small effect on the estimates produced by estat impact—small compared with the effect's standard error. For the ML estimator, the normalization does not affect the size of estimated effects shown by estat impact. See *Choosing weighting matrices and their normalization*.

Running estat impact after spregress is essential for proper interpretation of the model. The output of estat impact can be read directly as the change in the metric of the dependent variable per incremental change of the covariate averaged across all the spatial units (observations).

estat impact shows marginal (incremental change) effects. We might want to see the total effect of a discrete change in a covariate. The expectation of the dependent variable is linear in the covariates in this example. We did not fit polynomial or other nonlinear terms. We could just multiply the incremental change by the discrete change of the covariate. Or, we could use the margins command, which works for both linear and nonlinear terms; see [R] **margins**.

The median of gini is 0.39, its 25th percentile is 0.37, and its 75th percentile is 0.41. So it is reasonable to ask how a change of ± 0.02 in the Gini coefficient affects the homicide rate. Here's how to get the answer by using margins:

```
. margins, at(gini = generate(gini - 0.02)) at(gini = generate(gini))
> at(gini = generate(gini + 0.02))
Predictive margins                                      Number of obs = 1,412
Expression: Reduced-form mean, predict()
1._at: gini = gini - 0.02
2._at: gini =        gini
3._at: gini = gini + 0.02
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
_at						
1	2.550868	2.651383	0.96	0.336	-2.645746	7.747482
2	4.533855	2.584986	1.75	0.079	-.5326253	9.600334
3	6.516841	2.586198	2.52	0.012	1.447986	11.5857

A change of ± 0.02 in the Gini coefficient causes the homicide rate to change by roughly ± 2.0 per 100,000 persons per year.

The computations that `margins` must do to calculate standard errors can sometimes be time consuming. Time will depend on the complexity of the spatial model and the number of spatial units in the data. You may want to fit your model with a subsample of your data, run `margins`, and extrapolate to estimate the time required to run `margins` on the full sample. See [P] **timer** and [P] **rmsg**.

◁

▷ Example 2: spregress, gs2sls heteroskedastic

The `spregress, gs2sls` command has a `heteroskedastic` option that requires the errors to be independent but not necessarily identically distributed. Practically speaking, this option causes the estimates of the spatial autoregressive error correlations and the standard errors to change. In models without spatially autoregressive errors, only standard errors will change. See *Methods and formulas*.

If we add the `heteroskedastic` option to the last model we fit in example 1, we get

```
. spregress hrate ln_population ln_pdensity gini, gs2sls heteroskedastic
> dvarlag(W) errorlag(M) ivarlag(W: ln_population ln_pdensity gini)
  (1412 observations)
  (1412 observations (places) used)
  (weighting matrices define 1412 places)
  (output omitted)
```

```
Spatial autoregressive model                    Number of obs =  1,412
GS2SLS estimates                                Wald chi2(7)  = 248.74
                                                Prob > chi2   = 0.0000
                                                Pseudo R2     = 0.1241
```

hrate	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
hrate						
ln_populat~n	-.0475582	.3545931	-0.13	0.893	-.7425479	.6474315
ln_pdensity	.8989538	.4016155	2.24	0.025	.1118019	1.686106
gini	89.91969	10.71501	8.39	0.000	68.91866	110.9207
_cons	-32.21599	5.013344	-6.43	0.000	-42.04197	-22.39002
W						
ln_populat~n	2.679931	.5247129	5.11	0.000	1.651512	3.708349
ln_pdensity	-2.468953	.6786844	-3.64	0.000	-3.79915	-1.138756
gini	-57.38302	9.719208	-5.90	0.000	-76.43232	-38.33372
hrate	.6818566	.13258	5.14	0.000	.4220047	.9417085
M						
e.hrate	.9614507	.1554489	6.18	0.000	.6567764	1.266125

```
Wald test of spatial terms:          chi2(5) = 156.95     Prob > chi2 = 0.0000
. estimates store heterosk_model
```

We used `estimates store` to store the results of the earlier model, and we stored this model, too. We can now use `estimates table` to display coefficient estimates with their standard errors side by side. See [R] **estimates store** and [R] **estimates table**.

```
. estimates table model_ex1_last heterosk_model, b(%6.3f) se(%6.3f)
```

Variable	model~t	heter~l
hrate		
ln_populat~n	-0.048	-0.048
	0.330	0.355
ln_pdensity	0.899	0.899
	0.321	0.402
gini	89.920	89.920
	6.409	10.715
_cons	-32.216	-32.216
	3.590	5.013
W		
ln_populat~n	2.680	2.680
	0.522	0.525
ln_pdensity	-2.469	-2.469
	0.621	0.679
gini	-57.383	-57.383
	9.418	9.719
hrate	0.682	0.682
	0.114	0.133
M		
e.hrate	0.953	0.961
	0.132	0.155

Legend: b/se

We see that standard errors are larger, especially those for the direct-effect coefficients of the covariates. We also see that the estimate of ρ, the SAR error correlation labeled as `e.hrate`, differs between the two estimators.

◁

▷ Example 3: spregress, ml

SAR models can be fit using ML estimation. Here's the second model we fit in example 1 estimated using `ml` in place of `gs2sls`.

```
. spregress hrate ln_population ln_pdensity gini, ml dvarlag(W) errorlag(W)
  (1412 observations)
  (1412 observations (places) used)
  (weighting matrix defines 1412 places)
Performing grid search ... finished
Optimizing concentrated log likelihood:
Iteration 0:   log likelihood =  -4557.201
Iteration 1:   log likelihood =  -4556.763
Iteration 2:   log likelihood = -4556.7539
Iteration 3:   log likelihood = -4556.7539
Optimizing unconcentrated log likelihood:
Iteration 0:   log likelihood = -4556.7539
Iteration 1:   log likelihood = -4556.7539  (backed up)
```

```
Spatial autoregressive model                    Number of obs =  1,412
Maximum likelihood estimates                    Wald chi2(4)  =  240.21
                                                Prob > chi2   =  0.0000
Log likelihood = -4556.7539                     Pseudo R2     =  0.1590
```

hrate	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
hrate						
ln_populat~n	.5268247	.3038837	1.73	0.083	-.0687763	1.122426
ln_pdensity	.5269135	.3136226	1.68	0.093	-.0877755	1.141603
gini	91.44471	6.263932	14.60	0.000	79.16763	103.7218
_cons	-32.8348	3.205075	-10.24	0.000	-39.11663	-26.55297
W						
hrate	-.1850846	.1218453	-1.52	0.129	-.423897	.0537279
e.hrate	.6244211	.0897639	6.96	0.000	.4484871	.8003551
var(e.hrate)	34.79054	1.599235			31.79315	38.07052

```
Wald test of spatial terms:          chi2(2) = 227.84     Prob > chi2 = 0.0000
. estimates store ml_model
```

We stored the estimation results with `estimates store`, as we did with the same model fit with `gs2sls`, and now we use `estimates table` to compare coefficient estimates and their standard errors.

```
. estimates table gs2sls_model ml_model, b(%6.3f) se(%6.3f)
```

Variable	gs2sl~l	ml_mo~l
hrate		
ln_populat~n	0.103	0.527
	0.281	0.304
ln_pdensity	1.081	0.527
	0.252	0.314
gini	82.069	91.445
	5.658	6.264
_cons	-29.630	-32.835
	3.070	3.205
W		
hrate	0.194	-0.185
	0.065	0.122
e.hrate	0.356	0.624
	0.079	0.090
var(e.hrate)		34.791
		1.599

Legend: b/se

There are meaningful differences in the results. The coefficient of `ln_pdensity` was significant in the GS2SLS model but is nonsignificant in the ML model. The coefficient estimates for `gini`, however, are similar, as are their standard errors. The coefficient of the lag of `hrate` becomes negative in the ML model, and the SAR error correlation increases from $\rho = 0.36$ to $\rho = 0.62$.

We note that the ML estimator is not consistent under heteroskedasticity; for consistency, the error distribution needs to be i.i.d., although it need not be normal. Heteroskedasticity may be the reason why the estimates differ as they do. See Arraiz et al. (2010).

◁

Stored results

`spregress, gs2sls` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(k)`	number of parameters
`e(df_m)`	model degrees of freedom
`e(df_c)`	degrees of freedom for test of spatial terms
`e(iterations)`	number of generalized method of moments iterations
`e(iterations_2sls)`	number of two-stage least-squares iterations
`e(rank)`	rank of `e(V)`
`e(r2_p)`	pseudo-R^2
`e(chi2)`	χ^2
`e(chi2_c)`	χ^2 for test of spatial terms
`e(p)`	p-value for model test
`e(p_c)`	p-value for test of spatial terms
`e(converged)`	`1` if generalized method of moments converged, `0` otherwise
`e(converged_2sls)`	`1` if two-stage least-squares converged, `0` otherwise

Macros

`e(cmd)`	`spregress`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(indeps)`	names of independent variables
`e(idvar)`	name of ID variable
`e(estimator)`	`gs2sls`
`e(title)`	title in estimation output
`e(constant)`	`hasconstant` or `noconstant`
`e(exogr)`	exogenous regressors
`e(dlmat)`	names of spatial weighting matrices applied to *depvar*
`e(elmat)`	names of spatial weighting matrices applied to errors
`e(het)`	`heteroskedastic` or `homoskedastic`
`e(chi2type)`	`Wald`; type of model χ^2 test
`e(properties)`	`b V`
`e(estat_cmd)`	program used to implement `estat`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(marginsnotok)`	predictions disallowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(delta_2sls)`	two-stage least-squares estimates of coefficients in spatial lag equation
`e(rho_2sls)`	generalized method of moments estimates of coefficients in spatial error equation
`e(V)`	variance–covariance matrix of the estimators

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

`spregress, ml` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(k)`	number of parameters
`e(df_m)`	model degrees of freedom
`e(df_c)`	degrees of freedom for test of spatial terms
`e(ll)`	log likelihood
`e(iterations)`	number of maximum log-likelihood estimation iterations
`e(rank)`	rank of `e(V)`
`e(r2_p)`	pseudo-R^2
`e(chi2)`	χ^2
`e(chi2_c)`	χ^2 for test of spatial terms
`e(p)`	p-value for model test
`e(p_c)`	p-value for test of spatial terms
`e(converged)`	`1` if converged, `0` otherwise

Macros

`e(cmd)`	`spregress`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(indeps)`	names of independent variables
`e(idvar)`	name of ID variable
`e(estimator)`	`ml`
`e(title)`	title in estimation output
`e(constant)`	`hasconstant` or `noconstant`
`e(dlmat)`	name of spatial weighting matrix applied to *depvar*
`e(elmat)`	name of spatial weighting matrix applied to errors
`e(chi2type)`	`Wald`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(ml_method)`	type of `ml` method
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(estat_cmd)`	program used to implement `estat`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(marginsnotok)`	predictions disallowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`e(ilog)`	iteration log (up to 20 iterations)
`e(gradient)`	gradient vector
`e(Hessian)`	Hessian matrix
`e(V)`	variance–covariance matrix of the estimators

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

SAR models date back to the works of Whittle (1954) and Cliff and Ord (1973, 1981). Cressie (1993), LeSage and Pace (2009), and Waller and Gotway (2004) provide textbook introductions. Spatial models have been applied in a variety of disciplines, such as criminology, demography, economics, epidemiology, political science, and public health. See Darmofal (2015), Waller and Gotway (2004), Kelejian and Prucha (2010), Drukker, Egger, and Prucha (2013), and Lee, Liu, and Lin (2010) for examples in economics, social science, and public health, including examples of nongeographic models such as social interactions and social networks.

The GS2SLS estimator was derived by Kelejian and Prucha (1998, 1999, 2010) and extended by Arraiz et al. (2010) and Drukker, Egger, and Prucha (2013).

The formulas for the GS2SLS without higher-order spatial weighting matrices were published in Drukker, Prucha, and Raciborski (2013). For the higher-order models, `spregress, gs2sls` implements the estimator derived in Badinger and Egger (2011) and Prucha, Drukker, and Egger (2016).

The properties of the ML estimator were proven by Lee (2004), which also provides the formulas for the robust estimator of the VCE.

Methods and formulas are presented under the following headings:

Model

We consider a cross-sectional spatial autoregressive model with autoregressive disturbances (SARAR), allowing for higher-order spatial dependence in the dependent variable, exogenous independent variables, and spatial errors. The model is

$$\begin{aligned}\mathbf{y} &= \sum_{k=1}^{K}\beta_k\mathbf{x}_k + \sum_{p=1}^{P}\gamma_p\mathbf{W}_p\,\mathbf{x}_p + \sum_{r=1}^{R}\lambda_r\mathbf{W}_r\,\mathbf{y} + \mathbf{u}\\ \mathbf{u} &= \sum_{s=1}^{S}\rho_s\mathbf{M}_s\mathbf{u} + \boldsymbol{\epsilon}\end{aligned} \tag{1}$$

where

$\mathbf{y}$ is an $n \times 1$ vector of observations on the dependent variable;

$\mathbf{x}_k$ is an $n \times 1$ vector of observations on the exogenous variable; β_k is the corresponding scalar parameter;

$\mathbf{W}_p$, $\mathbf{W}_r$, and $\mathbf{M}_s$ are $n \times n$ spatial weighting matrices with 0 diagonal elements;

$\mathbf{W}_p\,\mathbf{x}_p$, $\mathbf{W}_r\,\mathbf{y}$, and $\mathbf{M}_s\mathbf{u}$ are $n \times 1$ vectors typically referred to as spatial lags for the exogenous variable, dependent variable, and error term; γ_p, λ_r, and ρ_s are scalar parameters; and

$\boldsymbol{\epsilon}$ is an $n \times 1$ vector of innovations (i.i.d. disturbances).

The model in (1) is frequently referred to as a higher-order spatial autoregressive model with spatial autoregressive disturbances, or namely, a SARAR(R, S) model.

The spatial weighting matrices $\mathbf{W}_p$, $\mathbf{W}_r$, and $\mathbf{M}_s$ are assumed to be known and nonstochastic. See [SP] **Intro 2** and Darmofal (2015, chap. 2) for an introduction to spatial weighting matrices, and see Kelejian and Prucha (2010) for a technical discussion of how normalization affects parameter definition.

The scalar parameters γ_p and λ_r measure the degree to which the dependent variable depends on its neighboring covariate's values and outcomes. See example 1 and LeSage and Pace (2009, sec. 2.7) for discussions of effect estimation.

The innovations $\boldsymbol{\epsilon}$ are assumed to be i.i.d. or independent but heteroskedastically distributed, where the heteroskedasticity is of unknown form. The errors $\mathbf{u}$ are spatially autoregressive.

The GS2SLS estimator produces consistent estimates in both cases when the `heteroskedastic` option is specified. For the first-order SARAR model, see Kelejian and Prucha (1998, 1999, 2010), Arraiz et al. (2010), and Drukker, Egger, and Prucha (2013) for formal results and discussions; for the higher-order SARAR(R, S) model, see Badinger and Egger (2011) for formal results. The ML estimator is consistent in the i.i.d. case for the SARAR(1, 1) model but generally not consistent in the heteroskedastic case. See Lee (2004) for some results for the ML estimator; see Arraiz et al. (2010) for evidence that the ML estimator does not produce consistent estimates in the heteroskedastic case.

The GS2SLS estimator can fit the SARAR(R, S) model, whereas the ML estimator can only fit the SARAR(1, 1) model.

GS2SLS estimator

In this section, we give a detailed description of the computations performed by `spregress, gs2sls`. For the SARAR(1, 1) model, `spregress, gs2sls` implements the estimator described in Kelejian and Prucha (2010), Arraiz et al. (2010), and Drukker, Egger, and Prucha (2013); for the SARAR(R, S) model, `spregress, gs2sls` implements the estimator described in Badinger and Egger (2011). We will describe the GS2SLS estimator for the SARAR(R, S) model, which generalizes the first-order SARAR model.

Let's first rewrite (1) in a compact form:

$$\begin{aligned} \mathbf{y} &= \mathbf{X}\beta + \overline{\mathbf{X}}\gamma + \overline{\mathbf{Y}}\lambda + \mathbf{u} = \mathbf{Z}\delta + \mathbf{u} \\ \mathbf{u} &= \overline{\mathbf{U}}\rho + \boldsymbol{\epsilon} \end{aligned} \tag{2}$$

where

$\mathbf{X} = [\mathbf{x}_k]_{k=1,\ldots,K}$ is an $n \times K$ matrix of exogenous covariates;

$\overline{\mathbf{X}} = [\mathbf{W}_p\,\mathbf{x}_p]_{p=1,\ldots,P}$ is an $n \times P$ matrix of spatial lags for the exogenous covariates;

$\overline{\mathbf{Y}} = [\mathbf{W}_r\,\mathbf{y}]_{r=1,\ldots,R}$ is an $n \times R$ matrix of spatial lags for the dependent variables;

$\overline{\mathbf{U}} = [\mathbf{M}_s\mathbf{u}]_{s=1,\ldots,S}$ is an $n \times S$ matrix of spatial lags for the error term;

$\mathbf{Z} = \left[\mathbf{X}, \overline{\mathbf{X}}, \overline{\mathbf{Y}}\right]$ is an $n \times (K + P + R)$ matrix;

β, γ, λ, and ρ denote the $K \times 1$, $P \times 1$, $R \times 1$, and $S \times 1$ vectors of coefficients corresponding to $\mathbf{X}$, $\overline{\mathbf{X}}$, $\overline{\mathbf{Y}}$, and $\overline{\mathbf{U}}$, respectively; and

$\delta = (\beta', \gamma', \lambda')'$ is a $(K + P + R) \times 1$ vector of coefficients for $\mathbf{Z}$.

In the following, we review the two-stage least-squares (2SLS), generalized spatial two-stage least-squares (GS2SLS), and GMM estimation approaches as discussed in Badinger and Egger (2011).

2SLS estimator of δ

In the first step, we apply 2SLS to (2) using an instrument matrix $\mathbf{H}_1$ to estimate δ. The 2SLS estimator of δ—say, $\widetilde{\delta}$—is defined as

$$\widetilde{\delta} = \left(\widetilde{\mathbf{Z}}'\mathbf{Z}\right)^{-1}\widetilde{\mathbf{Z}}'\mathbf{y}$$

where $\widetilde{\mathbf{Z}} = \mathbf{P}_{\mathbf{H}_1}\mathbf{Z}$ and $\mathbf{P}_{\mathbf{H}_1} = \mathbf{H}_1\left(\mathbf{H}_1'\mathbf{H}_1\right)^{-1}\mathbf{H}_1'$. The 2SLS estimator $\widetilde{\delta}$ depends on the instrument matrix $\mathbf{H}_1$. Let $\mathbf{X}_\mathrm{f}$ denote all the exogenous regressors; that is, $\mathbf{X}_\mathrm{f} = \left[\mathbf{X}, \overline{\mathbf{X}}\right]$ in our case. The instrument matrix $\mathbf{H}_1$ contains the linearly independent columns in

$$\mathbf{H}_1 = \left[\mathbf{X}_\mathrm{f}, \mathbf{W}^1\mathbf{X}_\mathrm{f}, \ldots, \mathbf{W}^q\mathbf{X}_\mathrm{f}\right]$$

where $\mathbf{W}^1 \equiv \{W_r\}_{r=1,\ldots,R}$ denotes all the spatial weighting matrices applied to the dependent variable, and $\mathbf{W}^q \equiv \left\{\mathbf{W}_{j_1}\mathbf{W}_{j_2}\ldots\mathbf{W}_{j_q}\right\}_{j_1,j_2,\ldots,j_q=1,\ldots,R}$ denotes the product of q matrices from W^1 in any possible permutation order.

The `impower(#)` option specifies q, the number of the power in $\mathbf{W}^q$. The default is `impower(2)`. Increasing q may improve the precision of the estimation of δ.

We now illustrate the construction of $\mathbf{H}_1$ with an example. Suppose we use two spatial weighting matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ to generate the spatial lags for the dependent variable. So $\mathbf{W}^1 = (\mathbf{W}_1, \mathbf{W}_2)$. If we have $q = 2$, then $\mathbf{W}^2 = (\mathbf{W}_1\mathbf{W}_1, \mathbf{W}_1\mathbf{W}_2, \mathbf{W}_2\mathbf{W}_1, \mathbf{W}_2\mathbf{W}_2)$. Plug $\mathbf{W}^1$ and $\mathbf{W}^2$ into the definition of $\mathbf{H}_1$, and the instrument matrix $\mathbf{H}_1$ in this special case contains the linear independent columns in the following matrix:

$$\mathbf{H}_1 = [\mathbf{X}_\mathrm{f}, \mathbf{W}_1\mathbf{X}_\mathrm{f}, \mathbf{W}_2\mathbf{X}_\mathrm{f}, \mathbf{W}_1\mathbf{W}_1\mathbf{X}_\mathrm{f}, \mathbf{W}_1\mathbf{W}_2\mathbf{X}_\mathrm{f}, \mathbf{W}_2\mathbf{W}_1\mathbf{X}_\mathrm{f}, \mathbf{W}_2\mathbf{W}_2\mathbf{X}_\mathrm{f}]$$

GMM estimator of ρ based on 2SLS residuals

The initial GMM estimates of ρ solve the sample equivalent of the population moment conditions

$$\begin{aligned}(1/N)\,E(\boldsymbol{\epsilon}'\mathbf{A}_s\boldsymbol{\epsilon}) &= 0\\ (1/N)\,E(\boldsymbol{\epsilon}'\mathbf{B}_s\boldsymbol{\epsilon}) &= 0 \qquad \text{for } \; s \in \{1,\ldots,S\}\end{aligned}$$

where $\mathbf{A}_s = \mathbf{M}_s$ and $\mathbf{B}_s = \mathbf{M}_s'\mathbf{M}_s - \mathrm{diag}(\mathbf{M}_s'\mathbf{M}_s)$. See the estimator derived in Badinger and Egger (2011) and Prucha, Drukker, and Egger (2016) for details.

GS2SLS estimator of δ

The GS2SLS estimator of δ is based on the spatially Cochrane–Orcutt-transformed model.

$$\mathbf{y}_{nt} = \mathbf{Z}_*(\rho)\,\delta + \boldsymbol{\epsilon} \tag{3}$$

where $\mathbf{y}_{nt} = (\mathbf{I}_n - \sum_{s=1}^{S}\rho_s\mathbf{M}_s)\mathbf{y}$, $\mathbf{Z}_*(\rho) = (\mathbf{I}_n - \sum_{s=1}^{S}\rho_s\mathbf{M}_s)\mathbf{Z}$, and $\mathbf{I}_n$ is an $n \times n$ identity matrix.

Now, we apply the 2SLS estimator to (3) by using an instrument matrix $\mathbf{H}_2$ and replacing ρ with $\widetilde{\rho}$. The GS2SLS estimator of δ—say, $\widehat{\delta}$—is defined as

$$\widehat{\delta} = \left\{ \widehat{\mathbf{Z}_*(\widetilde{\rho})}' \mathbf{Z}_*(\widetilde{\rho}) \right\}^{-1} \widehat{\mathbf{Z}_*(\widetilde{\rho})}' \mathbf{y}_*(\widetilde{\rho})$$

where

$\mathbf{y}_*(\widetilde{\rho}) = (\mathbf{I}_n - \sum_{s=1}^{S} \widetilde{\rho}_s \mathbf{M}_s)\mathbf{y}$,

$\mathbf{Z}_*(\widetilde{\rho}) = (\mathbf{I}_n - \sum_{s=1}^{S} \widetilde{\rho}_s \mathbf{M}_s)\mathbf{Z}$,

$\widehat{\mathbf{Z}_*(\widetilde{\rho})} = \mathbf{P}_{\mathbf{H_2}} \mathbf{Z}_*(\widetilde{\rho})$, and

$\mathbf{P}_{\mathbf{H_2}} = \mathbf{H}_2 \left(\mathbf{H}_2'\mathbf{H}_2\right)^{-1} \mathbf{H}_2'$.

The instrument matrix $\mathbf{H}_2$ contains the linearly independent columns in

$$\mathbf{H}_2 = [\mathbf{H}_1, \mathbf{M_1}\mathbf{H}_1, \ldots, \mathbf{M_S}\mathbf{H}_1]$$

Efficient GMM estimator of ρ based on GS2SLS residuals

The form of the efficient GMM weighting matrix is given in Badinger and Egger (2011) and Prucha, Drukker, and Egger (2016). The matrix has one form in the default homoskedastic case and another in the heteroskedastic case. The form of the matrix causes the estimates of spatially autoregressive error correlations and the standard errors to differ when the `heteroskedastic` option is specified.

ML estimator

We implement a quasi–maximum likelihood (QML) estimator for the first-order SARAR model. We can write SARAR(1, 1) [see (1)] as

$$\begin{aligned} \mathbf{y} &= \mathbf{X}\beta + \overline{\mathbf{X}}\gamma + \lambda\mathbf{W}\mathbf{y} + \mathbf{u} = \mathbf{X}_{\mathrm{f}}\zeta + \lambda\mathbf{W}\mathbf{y} + \mathbf{u} \\ \mathbf{u} &= \rho\mathbf{M}\mathbf{u} + \boldsymbol{\epsilon} \end{aligned} \tag{4}$$

where

$\mathbf{X}_{\mathrm{f}} = [\mathbf{X}, \overline{\mathbf{X}}]$ is an $n \times L$ matrix containing exogenous covariates and spatial lags for the exogenous variables, with $L = K + P$;

$\zeta = (\beta', \gamma')'$ is an $L \times 1$ vector of coefficients;

$\mathbf{W}$ and $\mathbf{M}$ are $n \times n$ spatial weighting matrices with 0 diagonal elements; and

λ and ρ are scalar parameters.

Log-likelihood function

We give the log-likelihood function assuming that $\boldsymbol{\epsilon} \sim N(0, \sigma^2\mathbf{I}_n)$. Lee (2004) gives formal results on the consistency and asymptotic normality of the QML estimator when the innovations are i.i.d. but not necessarily normally distributed. Violations of the assumption that the innovations are i.i.d. can cause the QML estimator to produce inconsistent results.

The reduced form of (4) is

$$\mathbf{y} = (\mathbf{I}_n - \lambda\mathbf{W})^{-1}\mathbf{X}_\mathrm{f}\,\zeta + (\mathbf{I}_n - \lambda\mathbf{W})^{-1}(\mathbf{I}_n - \rho\mathbf{M})^{-1}\boldsymbol{\epsilon}$$

The unconcentrated log-likelihood function is

$$\begin{aligned}\ln L(\mathbf{y}\,|\,\zeta,\lambda,\rho,\sigma^2) = &-\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) + \ln||\mathbf{I}_n - \lambda\mathbf{W}|| + \ln||\mathbf{I}_n - \rho\mathbf{M}|| \\ &+ \frac{1}{2\sigma^2}\left\{(\mathbf{I}_n - \lambda\mathbf{W})\mathbf{y} - \mathbf{X}_\mathrm{f}\,\zeta\right\}'(\mathbf{I}_n - \rho\mathbf{M})'(\mathbf{I}_n - \rho\mathbf{M})\left\{(\mathbf{I}_n - \lambda\mathbf{W})\mathbf{y} - \mathbf{X}_\mathrm{f}\,\zeta\right\}\end{aligned} \tag{5}$$

We can concentrate the log-likelihood function by taking first-order derivatives with respect to ζ and σ^2 in (5) and setting them to 0, yielding the maximizers

$$\begin{aligned}\widehat{\zeta}(\lambda,\rho) &= \left\{\mathbf{X}_\mathrm{f}'(\mathbf{I}_n - \rho\mathbf{M})'(\mathbf{I}_n - \rho\mathbf{M})\mathbf{X}_\mathrm{f}\right\}^{-1}\mathbf{X}_\mathrm{f}'(\mathbf{I}_n - \rho\mathbf{M})'(\mathbf{I}_n - \rho\mathbf{M})(\mathbf{I}_n - \lambda\mathbf{W})\mathbf{y} \\ \widehat{\sigma^2}(\lambda,\rho) &= \frac{1}{n}\left\{(\mathbf{I}_n - \lambda\mathbf{W})\mathbf{y} - \mathbf{X}_\mathrm{f}\,\widehat{\zeta}(\lambda,\rho)\right\}'(\mathbf{I}_n - \rho\mathbf{M})'(\mathbf{I}_n - \rho\mathbf{M}) \\ &\quad\times\left\{(\mathbf{I}_n - \lambda\mathbf{W})\mathbf{y} - \mathbf{X}_\mathrm{f}\,\widehat{\zeta}(\lambda,\rho)\right\}\end{aligned}$$

Substituting $\widehat{\zeta}(\lambda,\rho)$ and $\widehat{\sigma^2}(\lambda,\rho)$ into the log-likelihood function in (5), we have the concentrated log-likelihood function

$$\ln L_c(\mathbf{y}\,|\,\lambda,\rho) = -\frac{n}{2}\left\{\ln(2\pi) + 1\right\} - \frac{n}{2}\ln\{\sigma^2(\lambda,\rho)\} + \ln||\mathbf{I}_n - \lambda\mathbf{W}|| + \ln||\mathbf{I}_n - \rho\mathbf{M}||$$

The QML estimates for $\widehat{\lambda}$ and $\widehat{\rho}$ can be computed by maximizing the concentrated log likelihood. Then, we can calculate the QML estimates for ζ and σ^2 as $\widehat{\zeta}(\widehat{\lambda},\widehat{\rho})$ and $\widehat{\sigma^2}(\widehat{\lambda},\widehat{\rho})$.

`spregress, ml` uses a grid search to find reasonable initial values for λ and ρ.

The formula for the robust VCE is given in Lee (2004).

Pseudo-R^2

The pseudo-R^2 is calculated as $\{\mathrm{corr}(y,\widehat{y})\}^2$, where $\widehat{y}$ is the reduced-form prediction of y.

References

Arraiz, I., D. M. Drukker, H. H. Kelejian, and I. R. Prucha. 2010. A spatial Cliff–Ord-type model with heteroskedastic innovations: Small and large sample results. *Journal of Regional Science* 50: 592–614. https://doi.org/10.1111/j.1467-9787.2009.00618.x.

Badinger, H., and P. H. Egger. 2011. Estimation of higher-order spatial autoregressive cross-section models with heteroscedastic disturbances. *Papers in Regional Science* 90: 213–235. https://doi.org/10.1111/j.1435-5957.2010.00323.x.

Baum, C. F., and S. Hurn. 2021. *Environmental Econometrics Using Stata*. College Station, TX: Stata Press.

Britt, C. L. 1994. Crime and unemployment among youths in the United States, 1958–1990: A time series analysis. *American Journal of Economics and Sociology* 53: 99–109. https://doi.org/10.1111/j.1536-7150.1994.tb02680.x.

Cliff, A. D., and J. K. Ord. 1973. *Spatial Autocorrelation*. London: Pion.

——. 1981. *Spatial Processes: Models and Applications*. London: Pion.

Cressie, N. 1993. *Statistics for Spatial Data*. Rev. ed. New York: Wiley.

Darmofal, D. 2015. *Spatial Analysis for the Social Sciences*. New York: Cambridge University Press.

Drukker, D. M., P. H. Egger, and I. R. Prucha. 2013. On two-step estimation of a spatial autoregressive model with autoregressive disturbances and endogenous regressors. *Econometric Reviews* 32: 686–733. https://doi.org/10.1080/07474938.2013.741020.

Drukker, D. M., I. R. Prucha, and R. Raciborski. 2013. Maximum likelihood and generalized spatial two-stage least-squares estimators for a spatial-autoregressive model with spatial-autoregressive disturbances. *Stata Journal* 13: 221–241.

Gini, C. 1909. Concentration and dependency ratios (in Italian). English translation in *Rivista di Politica Economica 1997* 87: 769–789.

Kelejian, H. H., and I. R. Prucha. 1998. A generalized spatial two-stage least squares procedure for estimating a spatial autoregressive model with autoregressive disturbances. *Journal of Real Estate Finance and Economics* 17: 99–121. https://doi.org/10.1023/A:1007707430416.

——. 1999. A generalized moments estimator for the autoregressive parameter in a spatial model. *International Economic Review* 40: 509–533. https://doi.org/10.1111/1468-2354.00027.

——. 2010. Specification and estimation of spatial autoregressive models with autoregressive and heteroskedastic disturbances. *Journal of Econometrics* 157: 53–67. https://doi.org/10.1016/j.jeconom.2009.10.025.

Lee, L.-F. 2004. Asymptotic distributions of quasi-maximum likelihood estimators for spatial autoregressive models. *Econometrica* 72: 1899–1925. https://doi.org/10.1111/j.1468-0262.2004.00558.x.

Lee, L.-F., X. Liu, and X. Lin. 2010. Specification and estimation of social interaction models with network structures. *Econometrics Journal* 13: 145–176. https://doi.org/10.1111/j.1368-423X.2010.00310.x.

LeSage, J., and R. K. Pace. 2009. *Introduction to Spatial Econometrics*. Boca Raton, FL: Chapman & Hall/CRC.

Messner, S. F., L. Anselin, D. F. Hawkins, G. Deane, S. E. Tolnay, and R. D. Baller. 2000. An Atlas of the Spatial Patterning of County-Level Homicide, 1960–1990. Pittsburgh: National Consortium on Violence Research.

Prucha, I. R., D. M. Drukker, and P. H. Egger. 2016. Simultaneous equations models with higher-order spatial or social network interactions. Working paper, Department of Economics, University of Maryland. http://econweb.umd.edu/~prucha/papers/WP_IRP_PHE_DMD_2016.pdf.

Waller, L. A., and C. A. Gotway. 2004. *Applied Spatial Statistics for Public Health Data*. Hoboken, NJ: Wiley.

Whittle, P. 1954. On stationary processes in the plane. *Biometrika* 434–449. https://doi.org/10.2307/2332724.

Also see

[SP] **spregress postestimation** — Postestimation tools for spregress

[SP] **estat moran** — Moran's test of residual correlation with nearby residuals

[SP] **Intro** — Introduction to spatial data and SAR models

[SP] **spivregress** — Spatial autoregressive models with endogenous covariates

[SP] **spmatrix** — Categorical guide to the spmatrix command

[SP] **spxtregress** — Spatial autoregressive models for panel data

[R] **regress** — Linear regression

[U] **20 Estimation and postestimation commands**