

Package ‘spfrontier’

May 26, 2016

Type Package

Title Spatial Stochastic Frontier Models

Version 0.2.3

Date 2016-05-26

Author Dmitry Pavlyuk <Dmitry.V.Pavlyuk@gmail.com>

Maintainer Dmitry Pavlyuk <Dmitry.V.Pavlyuk@gmail.com>

Description A set of tools for estimation of various spatial specifications of stochastic frontier models.

License GPL (>= 2)

Depends R (>= 3.0),moments,tmvtnorm,mvtnorm,spdep,ezsims

Imports methods,parallel,optimx,numDeriv

ZipData no

RoxygenNote 5.0.1

Repository CRAN

Repository/R-Forge/Project spfrontier

Repository/R-Forge/Revision 51

Repository/R-Forge/DateTimeStamp 2016-05-26 17:58:16

Date/Publication 2016-05-26 22:05:03

NeedsCompilation no

R topics documented:

spfrontier-package	2
airports	2
airports.greece	3
genW	5
logLikelihood	6
ModelEstimates-class	7
spfrontier	9
spfrontier.true.value	10

Index	12
--------------	-----------

spfrontier-package *Spatial Stochastic Frontier*

Description

Spatial Stochastic Frontier

Details

A set of tools for estimation (MLE) of various spatial specifications of stochastic frontier models

Author(s)

Dmitry Pavlyuk <Dmitry.V.Pavlyuk@gmail.com>

airports *European airports statistical data*

Description

The spfrontier package includes the dataset airports, containing information about European airports infrastructure and traffic statistics in 2011.

Format

An unbalanced panel of 395 European airports in 2008-2012 (1763 observations) on the following 31 variables.

ICAO Airport ICAO code

AirportName Airport official name

Country Airport's country name

longitude Airport longitude

latitude Airport latitude

Year Observation year

PAX A number of carried passengers

ATM A number of of air transport movements served by an airport

Cargo A total volume of cargo served by an airport

Population100km A number of inhabitants, living in 100 km around an airport

Population200km A number of inhabitants, living in 200 km around an airport

Island 1 if an airport is located on an island; 0 otherwise

GDPpc Gross domestic product per capita in airport's NUTS3 region

RevenueTotal Airport total revenue

RevenueAviation Airport aviation revenue
RevenueNonAviation Airport non-aviation revenue
RevenueHandling Airport revenue from handling services
RevenueParking Airport revenue from parking services
EBITDA Airport earnings before interest, taxes, depreciation, and amortization
NetProfit Airport net profit
DA Airport depreciation, and amortization
StaffCount A number of staff employed by an airport
StaffCost Airport staff cost
RunwayCount A number of airport runways
CheckinCount A number of airport check-in facilities
GateCount A number of airport gates
TerminalCount A number of airport terminals
ParkingSpaces A number of airport parking spaces
RoutesDeparture A number of departure routes, served by an airport
RoutesArrival A number of arrival routes, served by an airport
Routes $(\text{RoutesDeparture} + \text{RoutesArrival})/2$

Source

Eurostat (2013). European Statistics Database, Statistical Office of the European Communities (Eurostat)
 Airports' statistical reports(2011)
 Open Flights: Airport, airline and route data <http://openflights.org/> (2013-05-31)
 TDC (2012). Informe de fiscalizacion de la imputacion por la entidad "Aeropuertos Espanoles y Navegacion Aerea" (AENA) a cada uno de los aeropuertos de los ingresos, gastos, e inversiones correspondientes a la actividad aeroportuaria, en los ejercicios 2009 y 2010., Tribunal de Cuentas, Spain, Doc 938.
 CIESIN, Columbia University. Gridded Population of the World: Future Estimates (GPWFE). (2005)

airports.greece

Greece airports statistical data

Description

The `sfrontier` package includes the dataset `airports`, containing information about Greece airports infrastructure and traffic statistics in 2011.

Format

A dataframe with 39 observations on the following 24 variables.

name Airport title

ICAO_code Airport ICAO code

lat Airport latitude

lon Airport longitude

APM_winter A number of passengers carried during winter period

APM_summer A number of passengers carried during summer period

APM A number of passengers carried (winter + summer)

cargo_winter A total volume of cargo served by an airport during winter period

cargo_summer A total volume of cargo served by an airport during summer period

cargo A number volume of cargo served by an airport (winter + summer)

ATM_winter A number of air transport movements served by an airport during winter period

ATM_summer A number of air transport movements served by an airport during summer period

ATM A number of air transport movements served by an airport (winter + summer)

opening_hours_winter A total number opening hours during winter period

opening_hours_summer A total number opening hours during summer period

opening_hours A total number opening hours (winter + summer)

runway_area A total area of airport runways

terminal_area A total area of airport terminal(s)

parking_area A total area of airport parking area

island 1 if an airpiort is located on an island; 0 otherwise

international 1 if an airpiort is international; 0 otherwise

mixed_use 1 if an airpiort is in mixed use; 0 otherwise

WLU A total volume of work load units (WLU) served by an airport

NearestCity A road network distance between an airport and its nearest city

Source

"Airport efficiency and public investment in Greece" (2010) In Proceeding of the 2010 International Kuhmo-Nectar Conference on Transport Economics, University of Valencia, Spain.

genW *Standard spatial contiguity matrixes*

Description

genW generates an spatial contiguity matrix (rook or queen)

rowStdrt standartizes spatial contiguity matrix by rows

constructW constructs a spatial contiguity matrix using object longitude and latitude coordinates

Usage

```
genW(n, type = "rook", seed = NULL)
```

```
rowStdrt(W)
```

```
constructW(coords, labels)
```

Arguments

n	a number of objects with spatial interaction to be arranged. See 'Details' for objects arranging principle
type	an optional type of spatial interaction. Currently 'rook' and 'queen' values are supported, to produce Rook and Queen Contiguity matrix. See references for more info. By default set to rook.
seed	an optional random number generator seed for random matrices
W	a spatial contiguity matrix to be standatised
coords	a matrix of two columns, where every row is a longitude-latitude pair of object coordinates
labels	a vector of object lables to mark rows and columns of the resulting contiguity matrix

Details

To generate spatial interaction between n objects the function arranges them on a chess board. A number of columns is calculated as a square root of n, rounded to the top. The last row contains empty cells, if n is not quadratic

The function divides every element in an argument matrix by the sum of elements in its row. Some spatial estimation requires this standartisation (generally - for faster calculations)

The function constructs a spatial contiguity matrix using object longitude and latitude coordinates. Euclidean distance is currently used.

References

Anselin, L. (1988). *Spatial Econometrics: Methods and Models*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Examples

```

# Completely filled 10x10 rook contiguity matrix
rookW <- genW(100)
rookW

# Partly filled 10x10 rook contiguity matrix
rookW <- genW(90)
rookW

# Completely filled 10x10 queen contiguity matrix
queenW <- genW(100, type="queen")
queenW

# Completely filled 10x10 queen contiguity matrix
queenW <- genW(100, type="queen")
queenW

# Standartisation
stQueenW <- rowStdrt(queenW)
stQueenW

data(airports)

W <- constructW(cbind(airports$lon, airports$lat),airports$Icao_code)

```

logLikelihood	<i>Calculation of the log likelihood function for the spatial stochastic frontier model</i>
---------------	---

Description

logLikelihood returns a value of the log likelihood function for the spatial stochastic frontier model

Usage

```

logLikelihood(formula, data, W_y = NULL, W_v = NULL, W_u = NULL,
  inefficiency = "half-normal", values, logging = c("quiet", "info",
  "debug"), costFrontier = F)

```

Arguments

formula	an object of class " formula "
data	data frame, containing the variables in the model
W_y	a spatial weight matrix for spatial lag of the dependent variable
W_v	a spatial weight matrix for spatial lag of the symmetric error term

W_u	a spatial weight matrix for spatial lag of the inefficiency error term
inefficiency	sets the distribution for inefficiency error component. Possible values are 'half-normal' (for half-normal distribution) and 'truncated' (for truncated normal distribution). By default set to 'half-normal'.
values	a vector of log likelihood function parameters
logging	an optional level of logging. Possible values are 'quiet','warn','info','debug'. By default set to quiet.
costFrontier	is designed for selection of cost or production frontier

Details

This function is exported from the package for testing and presentation purposes A list of arguments of the function exactly matches the corresponding list of the [spfrontier](#) function

ModelEstimates-class *Model Estimation Results*

Description

ModelEstimates stores information about MLE estimates of a spatial stochastic frontier model

Method status returns estimation status

Method resultParams returns raw estimated coefficients

Method hessian returns Hessian matrix for estimated coefficients

Method stdErrors returns standard errors of estimated coefficients

Method efficiencies returns efficiency estimates

Method show prints estimated coefficients

Method coefficients returns estimated coefficients

Method fitted returns model fitted values

Method residuals returns residuals

Method summary prints summary of the estimated model

Usage

status(object)

resultParams(object)

hessian(object)

stdErrors(object)

efficiencies(object)

```

## S4 method for signature 'ModelEstimates'
show(object)

## S4 method for signature 'ModelEstimates'
coefficients(object)

## S4 method for signature 'ModelEstimates'
resultParams(object)

## S4 method for signature 'ModelEstimates'
fitted(object)

## S4 method for signature 'ModelEstimates'
efficiencies(object)

## S4 method for signature 'ModelEstimates'
residuals(object)

## S4 method for signature 'ModelEstimates'
stdErrors(object)

## S4 method for signature 'ModelEstimates'
hessian(object)

## S4 method for signature 'ModelEstimates'
status(object)

## S4 method for signature 'ModelEstimates'
summary(object)

```

Arguments

object an object of ModelEstimates class

Details

ModelEstimates stores all parameter estimates and additional statistics, available after estimation of a spatial stochastic frontier model.

Slots

coefficients estimated values of model parameters
resultParams raw estimated values
status model estimation status:
 0 - Success
 1 - Failed; convergence is not achieved
 1000 - Failed; unexpected exception
 1001 - Failed; Initial values for MLE cannot be estimated

1002 - Failed; Maximum likelihood function is infinite

logL value of the log-likelihood function

logLcalls information about a number of log-likelihood function and its gradient function calls

hessian Hessian matrix for estimated coefficients

stdErrors standard errors of estimated coefficients

residuals model residuals

fitted model fitted values

efficiencies estimates of efficiency values for sample observations

spfrontier

Spatial stochastic frontier model

Description

spfrontier estimates spatial specifications of the stochastic frontier model.

Usage

```
spfrontier(formula, data, W_y = NULL, W_v = NULL, W_u = NULL,
  inefficiency = "half-normal", initialValues = "errorsarlm",
  logging = c("quiet", "info", "debug"), control = NULL, onlyCoef = F,
  costFrontier = F)
```

Arguments

formula	an object of class " formula ": a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	data frame, containing the variables in the model
W_y	a spatial weight matrix for spatial lag of the dependent variable
W_v	a spatial weight matrix for spatial lag of the symmetric error term
W_u	a spatial weight matrix for spatial lag of the inefficiency error term
inefficiency	sets the distribution for inefficiency error component. Possible values are 'half-normal' (for half-normal distribution) and 'truncated' (for truncated normal distribution). By default set to 'half-normal'. See references for explanations
initialValues	an optional vector of initial values, used by maximum likelihood estimator. If not defined, estimator-specific method of initial values estimation is used.
logging	an optional level of logging. Possible values are 'quiet', 'warn', 'info', 'debug'. By default set to quiet.
control	an optional list of control parameters, passed to optim estimator from the ' stats ' package
onlyCoef	allows calculating only estimates for coefficients (with inefficiencies and other additional statistics). Developed generally for testing, to speed up the process.
costFrontier	is designed for selection of cost or production frontier

Details

Models for estimation are specified symbolically, but without any spatial components. Spatial components are included implicitly on the base of the model argument.

References

Kumbhakar, S.C. and Lovell, C.A.K (2000), Stochastic Frontier Analysis, Cambridge University Press, U.K.

Examples

```
data( airports )
airports2011 <- subset(airports, Year==2011)
W <- constructW(cbind(airports2011$longitude, airports2011$latitude),airports2011$ICAO)
formula <- log(PAX) ~ log(Population100km) + log(Routes) + log(GDPpc)
ols <- lm(formula , data=airports2011)
summary(ols )
plot(density(stats::residuals(ols)))
skewness(stats::residuals(ols))

# Takes >5 sec, see demo for more examples
# model <- spfrontier(formula , data=airports2011)
# summary(model )

# model <- spfrontier(formula , data=airports2011, W_y=W)
# summary(model )
```

spfrontier.true.value *True value for simulation*

Description

spfrontier.true.value returns true parameter values for a simulation process

ezsimspfrontier tests estimators of a spatial stochastic frontier model with different parameters

Usage

```
spfrontier.true.value()
```

```
ezsimspfrontier(runs, params, inefficiency = "half-normal",
  logging = "info", control = list())
```

Arguments

runs	a number of simulated samples
params	a set with parameters to be used in simulation.
inefficiency	sets the distribution for inefficiency error component. Possible values are 'half-normal' (for half-normal distribution) and 'truncated' (for truncated normal distribution). By default set to 'half-normal'. See references for explanations
logging	an optional level of logging. Possible values are 'quiet','warn','info','debug'. By default set to quiet.
control	an optional list of control parameters for simulation process. Currently the procedure supports: ignoreWy (TRUE/FALSE) - the spatial contiguity matrix for a dependent variable is not provided to <code>spfrontier</code> estimator (but used in DGP) ignoreWv (TRUE/FALSE) - the spatial contiguity matrix for a symmetric error term is not provided to <code>spfrontier</code> estimator (but used in DGP) ignoreWu (TRUE/FALSE) - the spatial contiguity matrix for a inefficiency error term is not provided to <code>spfrontier</code> estimator (but used in DGP) parallel (TRUE/FALSE) - whether to use parallel computer seed - a state for random number generation in R. If NULL (default), the initial state is random. See <code>set.seed</code> for details. auto_save - saves intermediate results to files. See <code>ezsim</code> for details.

Details

The `spfrontier.true.value` function should not be used directly, it is exported for supporting `ezsim`

The `ezsimspfrontier` function executes multiple calls of the `spfrontier` estimator on a simulated data set, generated on the base of provided parameters. The resulting estimates can be analysed for biasedness, efficiency, etc.

See Also

`ezsim`

Examples

```
params000 <- list(n=c(50, 100),beta0=5,
                beta1=10,
                beta2=1,
                sigmaV=0.5,
                sigmaU=2.5)
ctrl <- list(seed=999, cores=1)
res000 <- ezsimspfrontier(2, params = params000,
                        inefficiency = "half-normal",
                        logging = "info",
                        control=ctrl)
summary(res000)
```

Index

- *Topic **frontier**,
 - spfrontier.true.value, 10
- *Topic **frontier**
 - genW, 5
 - spfrontier, 9
 - spfrontier-package, 2
- *Topic **simulation**
 - spfrontier.true.value, 10
- *Topic **spatial**
 - genW, 5
 - spfrontier, 9
 - spfrontier-package, 2
 - spfrontier.true.value, 10
- *Topic **stochastic**
 - genW, 5
 - spfrontier, 9
 - spfrontier-package, 2
 - spfrontier.true.value, 10
- airports, 2
- airports.greece, 3
- coefficients (ModelEstimates-class), 7
- coefficients, ModelEstimates-method (ModelEstimates-class), 7
- constructW (genW), 5
- efficiencias (ModelEstimates-class), 7
- efficiencias, ModelEstimates-method (ModelEstimates-class), 7
- eZsim, 11
- eZsimspfrontier (spfrontier.true.value), 10
- fitted (ModelEstimates-class), 7
- fitted, ModelEstimates-method (ModelEstimates-class), 7
- formula, 6, 9
- genW, 5
- hessian (ModelEstimates-class), 7
- hessian, ModelEstimates-method (ModelEstimates-class), 7
- logLikelihood, 6
- ModelEstimates-class, 7
- optim, 9
- residuals (ModelEstimates-class), 7
- residuals, ModelEstimates-method (ModelEstimates-class), 7
- resultParams (ModelEstimates-class), 7
- resultParams, ModelEstimates-method (ModelEstimates-class), 7
- rowStdrt (genW), 5
- set.seed, 11
- show, ModelEstimates-method (ModelEstimates-class), 7
- spfrontier, 7, 9, 11
- spfrontier-package, 2
- spfrontier.true.value, 10
- stats, 9
- status (ModelEstimates-class), 7
- status, ModelEstimates-method (ModelEstimates-class), 7
- stdErrors (ModelEstimates-class), 7
- stdErrors, ModelEstimates-method (ModelEstimates-class), 7
- summary, ModelEstimates-method (ModelEstimates-class), 7