Package 'netpanel'

July 29, 2026

Type Package

Title Maximum Likelihood Estimation Routines for Panel Network Autocorrelation Models

Version 0.0.1

Description A set of maximum likelihood estimation routines for panel network autocorrelation models. The package contains functionality for fixed and random effects panel network autocorrelation model, in addition to dynamic panel network autocorrelation that contains time lags of the response and networks. Moreover, the package allows for multiple networks to be fitted. For information on the types of models estimated in this package, please see Anselin (1988) <doi:10.1007/978-94-015-7799-1>, Cook et al. (2023) <doi:10.1017/S0003055422000272>, Hays et al. (2010) <doi:10.1016/j.stamet.2009.11.005>, Lee an
2354.2012.00724.x>, Millo (2014) <doi:10.1016/j.csda.2013.07.024>, Millo and Piras (2012) <doi:10.18637/jss.v047.i01>, and Wang and Yu (2015) <doi:10.1016/j.econlet.2015.01.021>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports Rcpp (>= 1.1.0), RcppArmadillo, stats, Matrix

LinkingTo Rcpp, RcppArmadillo

URL https://github.com/kevinCarson/netpanel

BugReports https://github.com/kevinCarson/netpanel/issues

NeedsCompilation yes

Author Kevin Carson [aut, cre]

Maintainer Kevin Carson <kacarson@arizona.edu>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2026-07-29 16:20:22 UTC

Contents

`coef.pnam` *Extract the ML parameter estimates from* `pnam` *model fits*

Description

This function extracts the Maximum Likelihood (ML) parameter estimates for the exogenous covariates from a `pnam` model fit.

Usage

```
## S3 method for class 'pnam'
coef(object, ...)
```

Arguments

`object`	An object of class "pnam".
`...`	Additional arguments for other methods.

Value

The vector of estimated parameter values for the exogenous covariates.

Examples

```
data("net1", package = "netpanel") #the first panel network
data("simulated.data", package = "netpanel") #the simulated variables
#' a fixed effects panel network autocorrelation model
fe.pnam <- mlpnam(Y~x1+x2+x3, net.formula = ~ net1,
               data =  simulated.data, time = ~panel,
               actor = ~unit, model = "fixed",
               fixed.effect = "actor")
coef(fe.pnam) #the mle parameter values
```

`fitted.pnam` *Fitted values from* `pnam` *model fits*

Description

This function returns the model fitted values, $\hat{y}$, for a fitted `pnam` model. The function returns two types of fitted values to the user: linear regression fitted values and network regression fitted values. The network regression fitted values are formulated as: $\hat{y} = (I_{NT} - \sum_i \hat{\rho}_i W^i)^{-1}(X\hat{\beta})$, where W^i is the i^{th} included network. In comparison, the linear regression fitted values are formulated as: $\hat{y} = X\hat{\beta}$.

Usage

```
## S3 method for class 'pnam'
fitted(object, ...)
```

Arguments

`object`	An object of class `pnam`.
`...`	Additional arguments for other methods.

Value

A `data.frame` object that stores the fitted values from a `pnam` model fit, where the column "ols.fitted" stores the linear regression fitted values and "network.fitted" stores the network regression fitted values (see above).

Examples

```
data("net1", package = "netpanel") #the first panel network
data("simulated.data", package = "netpanel") #the simulated variables

# a fixed effects panel network autocorrelation model
fe.pnam <- mlpnam(Y~x1+x2+x3, net.formula = ~ net1,
               data =  simulated.data, time = ~panel,
               actor = ~unit, model = "fixed",
               fixed.effect = "actor")
fitted(fe.pnam) #the data.frame of fitted values
```

logLik.pnam — *Extract the estimated model log-likelihood from* `pnam` *model fits*

Description

This function extracts the estimated model loglikelhood from a `pnam` model fit.

Usage

```
## S3 method for class 'pnam'
logLik(object, ..., REML = FALSE)
```

Arguments

`object`	An object of class `pnam`.
`...`	Additional arguments for other methods.
`REML`	From the generic `logLik` function. Set to FALSE and does not need to changed by the user.

Value

The estimated model log-likelihood.

Examples

```
data("net1", package = "netpanel") #the first panel network
data("simulated.data", package = "netpanel") #the simulated variables

# a fixed effects panel network autocorrelation model
fe.pnam <- mlpnam(Y~x1+x2+x3, net.formula = ~ net1,
               data =  simulated.data, time = ~panel,
               actor = ~unit, model = "fixed",
               fixed.effect = "actor")
logLik(fe.pnam) #the estimated model log-likelihood.
```

mlpnam — *Fit a Maximum Likelihood Panel Network Autocorrelation Model (PNAM)*

Description

A function that contains Maximum Likelihood estimation routines to fit panel network autocorrelation models (MLPNAM). The function allows for (1) fixed and random actor-level effects, (2) the inclusion of time lagged outcome and network terms, (3) multiple networks to construct the network (spatial) filtering matrix, and (4) for the time-varying errors to exhibit network autocorrelation, that is, for the model to become a network mixed effects model (i.e., error + lag model). Please see the Details section for more information.

Usage

```
mlpnam(
  reg.formula,
  net.formula,
  data,
  time,
  actor,
  model = c("fixed", "random", "dynamic"),
  fixed.effect = c("actor", "time", "two-way"),
  dynamic.lag = c("network", "outcome", "network x outcome"),
  errors = c("idiosyncratic", "autocorrelated"),
  autocorrelated.network = NULL,
  optim.method = "L-BFGS-B",
  optim.control = list(),
  ...
)
```

Arguments

reg.formula	A formula object that is the symbolic description of the network autocorrelation model to be fitted. The dependent and exogenous regressors are extracted from the data argument. This is the same argument found in lm and glm.
net.formula	A formula object that is the symbolic description of the network structure for the network autocorrelation model, which allows multiple networks to be included. The dependent variable should not be defined. Each of the included covariates should be a list of length t (i.e., the number of time points/panels) and each element should be an N x N matrix, where N is the number of actors/units in the network. The included networks should be row-normalized. The networks are extracted from the environment.
data	An object of class data.frame that contains the variables from the reg.formula, time, and actor arguments. The data.frame should be stacked by time, and ordered within time with respect to their position in the network adjacency matrices. If you are unsure how the stacking of observations should be constructed, please see the structure of the simulated.data netpanel data object.
time	A formula object where the variable that represents the time index of the observation is on the right hand side of ~. The variable is extracted from the data argument.
actor	A formula object where the variable that represents the actor/unit index of the observation is on the right hand side of ~. The variable is extracted from the data argument.
model	The type of panel network autocorrelation model to be estimated (see the Details section). "fixed" indicates that a within-estimator fixed effects model should be estimated. The demeaning of the data structure is based upon the fixed.effect argument. "random" indicates that a random-intercept model should be estimated with random effects with respect to the actors/units. "dynamic" indicates that a lagged model will be estimated, and the specific lag structure is based upon the dynamic.lag argument.

`fixed.effect` **optional**. When `model` is set to "fixed", how should the data structure be demeaned? "actor" indicates the data will be demeaned by the unit-specific average. "time" indicates that the data will be time demeaned (i.e., the x_{i1t} value at time t will be subtracted by the mean of x_{1t}, which is the vector of x_1 values for all units at time t). "two-way" indicates that two-way fixed effects will be included (i.e., time and unit fixed effects).

`dynamic.lag` **optional**. When `model` is set to "dynamic", what lag structure should be included? The option "network" will add the time lag for each included network structure in the `net.formula` argument. The option "outcome" will add the time lag for the outcome variable (Y) based on the `reg.formula` argument. The "network x outcome" option will add the time lags for the outcome and network variables.

`errors` What is the assumed structure of time-varying errors? "idiosyncratic" indicates that the time-varying errors are assumed to be drawn from the following distribution: $N(0, \sigma^2)$, where σ^2 is the variance. "autocorrelated" assumes that the time-varying errors exhibit network autocorrelation and the model will become a network mixed effects model. See the Details section.

`autocorrelated.network`
optional. When `errors` is set to "autocorrelated", a `list` object with length t, where each element is an N x N matrix and represents the network to be used in the network error model. See the Details section.

`optim.method` **optional**. The numeric optimization procedure to be employed by `optim`. Defaults to "L-BFGS-B" and, in general, should not be changed.

`optim.control` **optional**. A list control options for the `optim` function. Please see the `optim` function for the set of options.

`...` Additional arguments.

Details

This function fits a panel network autocorrelation model with time-varying social adjacency matrices via Maximum Likelihood Estimation. The estimated model takes the form:

$$Y_t = \rho_1 W_1^t Y_t + \cdots + \rho_k W_k^t Y_t + X_t \beta + v_t, \;\; t = 1, 2, \cdots, T$$

$$v_t = \mu_N + e_t, \;\; t = 1, 2, \cdots, T$$

where Y_t is the outcome vector for N actors at time t, $\rho_k W_k^t Y_t$ is the network autocorrelation term for the k^{th} social network at time t (taken from the `net.formula` argument), X_t are the set of exogenous time-varying (and invariant) regressors (taken from the `reg.formula` argument). Additionally, v_t is the time-varying error term that is the sum of the time-invariant unit-specific unobserved factors, μ_N, thought to explain Y_t and e_t are the time- and unit-varying unobserved factors. Dependent upon the `model` argument, the function will (1) estimate a fixed-effects network regression, where the within estimator demeans the data by time, actor, or both, (2) estimate a random effects network regression, or (3) estimate a dynamic panel network autocorrelation where the time lags of the outcome and network variables are added to the set of exogenous covariates. Please see the set of references below for more information of network (spatial) cross-sectional and panel models and their respective estimation strategies (e.g., fixed effects).

Moreover, the model can also estimate the network mixed effects model, where the estimated formula is:

$$Y_t = \rho_1 W_1^t Y_t + \cdots + \rho_k W_k^t Y_t + X_t \beta + v_t, \;\; t = 1, 2, \cdots, T$$
$$v_t = \mu_N + e_t, \;\; t = 1, 2, \cdots, T$$
$$e_t = \lambda W_t^e e_t + \epsilon_t, \;\; t = 1, 2, \cdots, T$$

where λ is the error autocorrelation parameter, W_t^e is the error social adjacency matrix (if applicable, the network included in the `autocorrelated.network` argument), and ϵ_t are a set of i.i.d normal errors at time t.

Internally, the function follows the two-step computation strategy discussed in Millo and Piras (2012). In the first step, the set of parameter estimates for ρ, λ, and θ are found via numerical optimization routines. Importantly, within each search, the values for β and σ^2 are updated via their GLS estimators. In the second step, once the ML parameter estimates for ρ, λ, and θ are found, the final GLS estimates for β and σ^2 are computed. The asymptotic variance-covariance matrix for ρ, λ, and θ are the inverse of the negative Hessian matrix and the variance-covariance matrix for β is found via the standard GLS variance estimator. For sake of brevity, please see the below listed references for more information on these estimators.

Value

An object of class `pnam` as a `list` containing the following elements:

- `N` - The number of unique units in the panel network dataset.
- `t` - The number of panels (time periods) in the panel network dataset.
- `type` - The type of model estimated per the `model` argument.
- `error.structure` - The assumed structure for the time-varying errors per the `errors` argument.
- `fe.type` - If a fixed effects model is estimated, the type of fixed effects demeaning per the `fixed.effect` argument.
- `lag.type` - If a dynamic model is estimated, the type of lag structure included in the model per the `dynamic.lag` argument
- `max.eigen` - The maximum absolute eigenvalue of the rho*W matrix.
- `lambda` - If the time-varying errors are assumed to be autocorrelated, the estimated value for lambda.
- `rho` - The vector of estimated maximum likelihood rho parameters.
- `vcov.rho` - The asymptotic variance-covariance of the estimated rho parameters.
- `coefficients` - The vector of estimated exogenous ML parameters.
- `vcov` - The asymptotic variance-covariance of the exogenous ML parameters.
- `sigma2` - The estimated variance of the time-varying residual.
- `residuals` - The residuals of the fitted model.
- `fitted.values` - The fitted values from the estimated model.
- `logLik` - The estimated model log-likelihood.
- `convergence` - The converge value returned from the optim function. 0 indicates successful convergence.

- `netfilter` - The filtering matrix of the estimated model ($I_{NT} - \{\sum_j rho_j * W^j\}$), where W^j is the jth network from the `net.formula` argument.
- `k` - The total number of estimated parameters.
- `df` - The model degrees of freedom.
- `AIC` - The AIC of the estimated PNAM.
- `BIC` - The BIC of the estimated PNAM.
- `response` - The outcome vector (Y).
- `covariates` - The covariate matrix (X).
- `optim.information` - The returned list from the optim function.
- `error.dis` - If a random effects model is estimated, a list that contains the residual variances for the time-varying residuals (idiosyncratic), the unit-/actor-specific residual variances (unit), and the estimated ratio of the variances (theta).
- `theta` - If a random effects model is estimated, the estimated value for theta (the ratio of the unit and idiosyncratic variances.)
- `vcov.theta` - If a random effects model is estimated, the asymptotic variance of the estimated theta value.

References

Anselin, Luc. 1988. *Spatial Econometrics: Methods and Models*. Dordrecht: Springer Netherlands.

Cook, Scott J., Jude C. Hays and Robert J. Franzese. 2023. “STADL Up! The Spatiotemporal Autoregressive Distributed Lag Model for TSCS Data Analysis.” *American Political Science Review* 117(1):59–79.

Doreian, Patrick. 1982. “Maximum Likelihood Methods for Linear Models: Spatial Effect and Spatial Disturbance Terms.” *Sociological Methods & Research* 10(3):243–269.

Doriean, Patrick, Klaus Teuter and Chi-Hsein Wang. 1984. “Network Autocorrelation Models: Some Monte Carlo Results.” *Sociological Methods & Research* 13(2):155–200.

Duxbury, Scott. 2023. *Longitudinal Network Models*. Vol. 192 of *Quantitative Applications in the Social Sciences* SAGE Publications.

Hays, Jude C., Aya Kachi, and Robert J. Franzese. 2010. "A Spatial Model Incorporating Dynamic, Endogenous Network Interdependence: A Political Science Application." *Statistical Methodology* 7(3): 406-428.

Lee, Lung-fei and Jihai Yu. 2010. “Estimation of Spatial Autoregressive Panel Data Models with Fixed Effects.” *Journal of Econometrics* 154(2):165–185.

Lee, Lung-fei and Jihai Yu. 2012a. “QML Estimation of Spatial Dynamic Panel Data Models with Time Varying Spatial Weights Matrices.” *Spatial Economic Analysis* 7(1):31–74.

Lee, Lung-fei and Jihai Yu. 2012b. “Spatial Panels: Random Components Versus Fixed Effects.” *International Economic Review* 53(4):1369–1412.

Millo, Giovanni. 2014. "Maximum Likelihood Estimation of Spatially and Serially Correlated Panels with Random Effects." *Computational Statistics & Data Analysis* 71: 914-933.

Millo, Giovanni and Gianfranco Piras. 2012. "splm: Spatial Panel Data Models in R." *Journal of Statistical Software* 47: 1-38.

Neuman, Eric J. and Mark S. Mizruchi. 2010. “Structure and Bias in the Network Autocorrelation Model.” *Social Networks* 32(4):290–300.

Wang, Wei and Jihai Yu. 2015. "Estimation of Spatial Panel Data Models with Time Varying Spatial Weights Matrices.” *Economic Letters* 128:95-99.

See Also

coef.pnam, vcov.pnam, logLik.pnam, AIC, residuals.pnam, fitted.pnam, netimpacts.

Examples

```
data("net1", package = "netpanel") #the first panel network
data("net2", package = "netpanel") #the second panel network
data("net3", package = "netpanel") #the third panel network
data("simulated.data", package = "netpanel") #the simulated variables

# a random effects panel network autocorrelation model
re.pnam <- mlpnam(Y~x1+x2+x3, net.formula = ~ net1,
               data =  simulated.data, time = ~panel,
               actor = ~unit, model = "random")
summary(re.pnam)

# a fixed effects panel network autocorrelation model
fe.pnam <- mlpnam(Y~x1+x2+x3, net.formula = ~  net1 + net2,
               data =  simulated.data, time = ~panel,
               actor = ~unit, model = "fixed",
               fixed.effect = "actor")
summary(fe.pnam)

# a dynamic panel network autocorrelation model with autocorrelated errors
dyn.pnam <- mlpnam(Y~x1+x2+x3, net.formula = ~  net1 + net2,
               data =  simulated.data, time = ~panel,
               actor = ~unit, model = "dynamic",
               dynamic.lag = "outcome")
summary(dyn.pnam)



# a fixed effects panel network autocorrelation model with autocorrelated errors
fe.pnam.error <- mlpnam(Y~x1+x2+x3, net.formula = ~ net1,
               data =  simulated.data, time = ~panel,
               actor = ~unit, model = "fixed",
               errors = "autocorrelated",
               autocorrelated.network = net3)
summary(fe.pnam.error)
```

net1 — *A `list` of simulated panel social networks (1)*

Description

A `list` of length 10 where each element is an simulated row-normalized 50 x 50 social adjacency matrix, where a non-zero W_{ijt} entry denotes a directed relationship from i to j at time t.

Usage

```
data(net1)
```

Format

`net1`: A `list` of length 10 where each element is an simulated row-normalized 50 x 50 social adjacency matrix, where a non-zero W_{ijt} entry denotes a directed relationship from i to j at time t.

net2 — *A `list` of simulated panel social networks (2)*

Description

A `list` of length 10 where each element is an simulated row-normalized 50 x 50 social adjacency matrix, where a non-zero W_{ijt} entry denotes a directed relationship from i to j at time t.

Usage

```
data(net2)
```

Format

`net2`: A `list` of length 10 where each element is an simulated row-normalized 50 x 50 social adjacency matrix, where a non-zero W_{ijt} entry denotes a directed relationship from i to j at time t.

`net3` *A `list` of simulated panel social networks (3)*

Description

A `list` of length 10 where each element is an simulated row-normalized 50 x 50 social adjacency matrix, where a non-zero W_{ijt} entry denotes a directed relationship from i to j at time t.

Usage

```
data(net3)
```

Format

`net3`:
A `list` of length 10 where each element is an simulated row-normalized 50 x 50 social adjacency matrix, where a non-zero W_{ijt} entry denotes a directed relationship from i to j at time t.

`netimpacts` *Compute the direct, indirect, and total impacts from a fitted `pnam` object*

Description

This function returns the estimated average direct, indirect, and total impacts from a fitted panel network autocorrelation model.

Usage

```
netimpacts(object, return.impact.matrix = FALSE, digits = 5)
```

Arguments

`object` An object of class `pnam`.

`return.impact.matrix` TRUE/FALSE. If TRUE, the function will also return the estimated impact matrix for each estimated covariate. Set to FALSE by default.

`digits` The number of digits to round the estimates after the decimal point.

Details

In classical linear regression, the marginal effect of a regressor, x_1, on the outcome, Y, is $\hat{\beta}_1$. In panel network (and spatial) effects models, the marginal effect is:

$$\frac{\partial Y}{\partial x_1} = (I_{NT} - \sum_k \hat{\rho}_k W^k)^{-1} \hat{\beta}_1$$

The direct impact is the average of the diagonal value of the above matrix, the indirect impact is the average of the row sum of the off-diagonal values, and the total impact is sum of both. The function will also return the impact matrix for each estimated effect if `return.impact.matrix` is set to TRUE.

Value

A `data.frame` object that stores the direct, indirect, and total impact for each estimated effect in the fitted `pnam` model.

Examples

```
data("net1", package = "netpanel") #the first panel network
data("simulated.data", package = "netpanel") #the simulated variables

# a fixed effects panel network autocorrelation model
fe.pnam <- mlpnam(Y~x1+x2+x3, net.formula = ~ net1,
               data =  simulated.data, time = ~panel,
               actor = ~unit, model = "fixed",
               fixed.effect = "actor")
netimpacts(fe.pnam) #compute the impact values
```

`netpanel` *Maximum Likelihood Estimation Routines for Panel Network Autocorrelation Models*

Description

The `netpanel` package contains maximum likelihood estimation routines to fit panel network autocorrelation models (PNAMs) with time-varying social adjacency matrices. The package contains functionality for (1) fixed and random effects PNAMs, (2) dynamic PNAMs that include time lags of the outcome and networks, and (3) the estimation of network mixed effects models where, in addition to the lag of the outcome, the time-varying residuals are assumed to exhibit network autocorrelation. Importantly, the package allows for multiple network lags to be fitted simultaneously. Please see the list of below references for more information on panel network (spatial) autocorrelation models.

Details

The `netpanel` package primarily works through the wrapper function `mlpnam`. The `mlpnam` function returns an S3 object of class `pnam` that stores the relevant results of the fitted PNAM. The full set of functions (including user-helper S3 methods) are:

- `mlpnam`: fits a panel network autocorrelation model to an empirical panel network dataset via Maximum Likelihood Estimation.
- `netimpacts`: computes the direct, indirect, and total network impacts for each included regressor from a fitted `pnam` S3 object.
- `coef.pnam`: extracts the vector of ML parameter estimates from a fitted `pnam` S3 object.
- `vcov.pnam`: extracts the asymptotic variance-covariance matrix from a fitted `pnam` S3 object.
- `logLik.pnam`: extracts the estimated log-likelihood from a fitted `pnam` S3 object.
- `residuals.pnam`: extracts the model residuals from a fitted `pnam` S3 object.
- `fitted.pnam`: extracts the linear and network regression fitted values from a fitted `pnam` S3 object.

Additionally, the `netpanel` package includes a set of simulated pseudo data objects for users to start fitting panel network autocorrelation models. Please see the `simulated.data` `netpanel` help page for more information on the simulated dataset and respective networks.

Author(s)

Kevin A. Carson kacarson@arizona.edu

References

Anselin, Luc. 1988. *Spatial Econometrics: Methods and Models*. Dordrecht: Springer Netherlands.

Cook, Scott J., Jude C. Hays and Robert J. Franzese. 2023. “STADL Up! The Spatiotemporal Autoregressive Distributed Lag Model for TSCS Data Analysis.” *American Political Science Review* 117(1):59–79.

Doreian, Patrick. 1982. “Maximum Likelihood Methods for Linear Models: Spatial Effect and Spatial Disturbance Terms.” *Sociological Methods & Research* 10(3):243–269.

Doriean, Patrick, Klaus Teuter and Chi-Hsein Wang. 1984. “Network Autocorrelation Models: Some Monte Carlo Results.” *Sociological Methods & Research* 13(2):155–200.

Duxbury, Scott. 2023. *Longitudinal Network Models*. Vol. 192 of *Quantitative Applications in the Social Sciences* SAGE Publications.

Hays, Jude C., Aya Kachi, and Robert J. Franzese. 2010. "A Spatial Model Incorporating Dynamic, Endogenous Network Interdependence: A Political Science Application." *Statistical Methodology* 7(3): 406-428.

Lee, Lung-fei and Jihai Yu. 2010. “Estimation of Spatial Autoregressive Panel Data Models with Fixed Effects.” *Journal of Econometrics* 154(2):165–185.

Lee, Lung-fei and Jihai Yu. 2012a. “QML Estimation of Spatial Dynamic Panel Data Models with Time Varying Spatial Weights Matrices.” *Spatial Economic Analysis* 7(1):31–74.

Lee, Lung-fei and Jihai Yu. 2012b. “Spatial Panels: Random Components Versus Fixed Effects.” *International Economic Review* 53(4):1369–1412.

Millo, Giovanni. 2014. "Maximum Likelihood Estimation of Spatially and Serially Correlated Panels with Random Effects." *Computational Statistics & Data Analysis* 71: 914-933.

Millo, Giovanni and Gianfranco Piras. 2012. "splm: Spatial Panel Data Models in R." *Journal of Statistical Software* 47: 1-38.

Neuman, Eric J. and Mark S. Mizruchi. 2010. "Structure and Bias in the Network Autocorrelation Model." *Social Networks* 32(4):290–300.

Wang, Wei and Jihai Yu. 2015. "Estimation of Spatial Panel Data Models with Time Varying Spatial Weights Matrices." *Economic Letters* 128:95-99.

See Also

Useful links:

- `https://github.com/kevinCarson/netpanel`
- Report bugs at `https://github.com/kevinCarson/netpanel/issues`

`print.pnam` *Print Method for* `pnam` *model objects.*

Description

Print Method for `pnam` model objects.

Usage

```
## S3 method for class 'pnam'
print(x, digits = 5, ...)
```

Arguments

`x`	An object of class `pnam`.
`digits`	The number of digits to print after the decimal point.
`...`	Additional arguments (currently unused).

Value

No return value. Prints out the main results of a `pnam` object.

`print.summary.pnam` *Print Method for* `summary.pnam` *Panel Network Autocorrelation Model Summary*

Description

Print Method for `summary.pnam` Panel Network Autocorrelation Model Summary

Usage

```
## S3 method for class 'summary.pnam'
print(x, digits = 5, ...)
```

Arguments

`x`	An object of class `summary.pnam`.
`digits`	The number of digits to print after the decimal point.
`...`	Additional arguments (currently unused).

Value

No return value. Prints out the summary of a `pnam` object.

`residuals.pnam` *Residuals for* `pnam` *model fits*

Description

This function returns the model residuals for a `pnam` model fit.

Usage

```
## S3 method for class 'pnam'
residuals(object, ...)
```

Arguments

`object`	An object of class `pnam`.
`...`	Additional arguments for other methods.

Value

The vector of `pnam` model fit residuals.

Examples

```
data("net1", package = "netpanel") #the first panel network
data("simulated.data", package = "netpanel") #the simulated variables

# a fixed effects panel network autocorrelation model
fe.pnam <- mlpnam(Y~x1+x2+x3, net.formula = ~ net1,
               data =  simulated.data, time = ~panel,
               actor = ~unit, model = "fixed",
               fixed.effect = "actor")
residuals(fe.pnam) #the vector of model residuals
```

`simulated.data` *Simulated panel network dataset*

Description

A simulated panel network dataset that contains a response variable (Y) and three exogenous regressors (x_1, x_2, and x_3). The exogenous regresors x_1 and x_3 vectors are drawn from standard normal distribution, and x_2 is a vector of random bernoulli variables. The `panel` variable represents the time of data collection and the `unit` variable indexes the unique actors across the panels. The outcome is generated from the following deterministic equation:

$$Y_t = \rho_1 W^1 Y_t + \rho_2 W^w Y_t + \beta_0 + \beta_1 x_t 1 + \beta_2 x_t 2 + \beta_3 x_t 3 + v_t, \;\; t = 1, 2, \cdots, 10$$

$$v_t = \mu_t + e_t, \;\; t = 1, 2, \cdots, 10$$

$$e_t = \lambda W^3 e_t + \epsilon_t, \;\; t = 1, 2, \cdots, 10$$

where W^1 is the `net1` `netpanel` data object, W^2 is the `net2` `netpanel` data object, and W^3 is the `net3` `netpanel` data object.

Usage

```
data(simulated.data)
```

Format

`simulated.data`:

A simulated panel network dataset with size 500×5 the following variables: Y, x_1, x_2, x_3, `panel`, and `unit`.

summary.pnam — *Summary Method for* pnam *Objects*

Description

Summarizes the results of a fitted pnam model object.

Usage

```
## S3 method for class 'pnam'
summary(object, digits = 5, ...)
```

Arguments

object	An object of class pnam.
digits	The number of digits to print after the decimal point.
...	Additional arguments (currently unused).

Value

A list of summary statistics for the panel network autocorrelation model, such as the estimated log-likelihood and the estimated parameters.

vcov.pnam — *Extract the asymptotic variance-covariance matrix from* pnam *model fits*

Description

This function extracts the asymptotic variance-covariance matrix from pnam model fits for the exogenous covariates. For the endogenous variance matrices (such as the variance-covariance matrix for ρ), please extract those from the model fit. For example, the variance matrix for ρ can be extracted as `model$vcov.rho`, where `model` is the pnam object name.

Usage

```
## S3 method for class 'pnam'
vcov(object, ...)
```

Arguments

object	An object of class pnam.
...	Additional arguments for other methods.

Value

The asymptotic variance-covariance matrix for the exogenous covariates.

Examples

```
data("net1", package = "netpanel") #the first panel network
data("simulated.data", package = "netpanel") #the simulated variables

# a fixed effects panel network autocorrelation model
fe.pnam <- mlpnam(Y~x1+x2+x3, net.formula = ~ net1,
               data =  simulated.data, time = ~panel,
               actor = ~unit, model = "fixed",
               fixed.effect = "actor")
vcov(fe.pnam) #the asymptotic variance-covariance matrix
```

Index