

# Package ‘getspanel’

October 19, 2023

**Title** General-to-Specific Modelling of Panel Data

**Version** 0.1.5

**Date** 2023-10-18

**Description** Uses several types of indicator saturation and automated General-to-Specific (GETS) modelling from the 'gets' package and applies it to panel data. This allows the detection of structural breaks in panel data, operationalising a reverse causal approach of causal inference, see Pretis and Schwarz (2022) <[doi:10.2139/ssrn.4022745](https://doi.org/10.2139/ssrn.4022745)>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**URL** <https://github.com/moritzpschwarz/getspanel>,  
<http://moritzschwarz.org/getspanel/>

**BugReports** <https://github.com/moritzpschwarz/getspanel/issues>

**LazyData** true

**RoxygenNote** 7.2.3

**Suggests** testthat, knitr, rmarkdown, lfe, prettydoc, plm, fixest,  
lmtest, sandwich, cowplot

**Imports** gets, fastDummies, Matrix, ggplot2, stats, mvtnorm

**Depends** R (>= 3.5.0)

**VignetteBuilder** knitr, rmarkdown

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Felix Pretis [aut],  
Moritz Schwarz [aut, cre] (<<https://orcid.org/0000-0003-0340-3780>>)

**Maintainer** Moritz Schwarz <[moritz.schwarz@scmo.eu](mailto:moritz.schwarz@scmo.eu)>

**Repository** CRAN

**Date/Publication** 2023-10-19 11:30:03 UTC

## R topics documented:

break_uncertainty . . . . .	2
EUCO2residential . . . . .	3
EU_emissions_road . . . . .	4
felmFun . . . . .	5
fixestFun . . . . .	5
get_indicators . . . . .	6
identify_indicator_timings . . . . .	7
isatpanel . . . . .	7
logLik.plm . . . . .	11
pandata_simulated . . . . .	11
plmFun . . . . .	12
plot.isatpanel . . . . .	12
plot_counterfactual . . . . .	13
plot_grid . . . . .	14
plot_residuals . . . . .	15
print.isatpanel . . . . .	16
robust_isatpanel . . . . .	17
Within_plm . . . . .	18
<b>Index</b>	<b>19</b>

---

break_uncertainty	<i>Estimate Breakdate Uncertainty</i>
-------------------	---------------------------------------

---

### Description

Estimate Breakdate Uncertainty

### Usage

```
break_uncertainty(x, m = 15, interval = 0.99)
```

### Arguments

x	An object produced by the isatpanel function
m	Maximum range of interval (default is 15 time periods).
interval	Approximate level of interval. CI level will be at least $>$ interval. Default 0.99 is a 99% CI, so the time interval will always be the integer that results in at least $>$ 99% coverage.

### Value

A data.frame that indicates the uncertainty for each FESIS break. The time interval is given by the estimated date in the 'time' column with a confidence interval of +/- the interval in the tci column.

## Examples

```
data(EU_emissions_road)

# Group specification
EU15 <- c("Austria", "Germany", "Denmark", "Spain", "Finland", "Belgium",
         "France", "United Kingdom", "Ireland", "Italy", "Luxembourg",
         "Netherlands", "Greece", "Portugal", "Sweden")

# Prepare sample and data
EU_emissions_road_short <- EU_emissions_road[
EU_emissions_road$country %in% EU15 &
EU_emissions_road$year >= 2000,
]

# Run
result <- isatpanel(
  data = EU_emissions_road_short,
  formula = ltransport.emissions ~ lgdp + I(lgdp^2) + lpop,
  index = c("country", "year"),
  effect = "twoways",
  fesis = TRUE,
  plot = FALSE,
  t.pval = 0.01
)

break_uncertainty(result)
```

---

EUCO2residential

*CO2 Data for the EU Residential Sector*

---

## Description

CO2 Data for the EU Residential Sector

## Usage

EUCO2residential

## Format

A data frame with 1550 rows and 9 variables:

**country** Country

**year** Year

**lgdp** Log Gross Domestic Product

**lhdd** Log Heating Degree Days

**lcdd** Log Cooling Degree Days  
**urban** Urban Share  
**av.rate** EU Interest Rate  
**pop** Population  
**agg.directem** Aggregated Direct Emissions

#### Source

IEA

---

EU_emissions_road	<i>CO2 Data for EU Road Emissions</i>
-------------------	---------------------------------------

---

#### Description

CO2 Data for EU Road Emissions

#### Usage

EU\_emissions\_road

#### Format

A data frame with 1550 rows and 13 variables:

**X** Index  
**country** Country  
**year** Year  
**gdp** Gross Domestic Product  
**pop** Population  
**transport.emissions** Transport CO2 Emissions  
**lgdp** Log GDP  
**lpop** Log Population  
**ltransport.emissions** Log Transport CO2 Emissions  
**const** Constant  
**L1.ltransport.emissions** Lag 1 Log Transport CO2 Emissions  
**L1.lgdp** Lag 1 Log GDP  
**L1.lpop** Lag 1 Log Population

#### Source

EDGAR

---

felmFun	<i>Internal lfe/felm Estimation Method</i>
---------	--

---

**Description**

Internal lfe/felm Estimation Method

**Usage**

```
felmFun(y, x, effect, time, id, cluster = "individual", ...)
```

**Arguments**

y	dependent variable
x	matrix of regressors
effect	Fixed Effect specification
time	Character vector of name of the time variable
id	Character vector of the name of the group variable
cluster	Character vector of the variable(s) to cluster Standard Errors at
...	Further arguments to pass to gets::isat

**Value**

List to be used by gets::isat

---

fixestFun	<i>Internal fixest/feols Estimation Method</i>
-----------	--

---

**Description**

Internal fixest/feols Estimation Method

**Usage**

```
fixestFun(y, x, effect, time, id, cluster = "individual", ...)
```

**Arguments**

y	dependent variable
x	matrix of regressors
effect	Fixed Effect specification
time	Character vector of name of the time variable
id	Character vector of the name of the group variable
cluster	Character vector of the variable(s) to cluster Standard Errors at
...	Further arguments to pass to gets::isat

**Value**

List to be used by gets::isat

---

get_indicators	<i>Extract the retained indicators from an isatpanel object</i>
----------------	---

---

**Description**

Extract the retained indicators from an isatpanel object

**Usage**

```
get_indicators(object, uis_breaks = NULL)
```

**Arguments**

object            An object produced by the isatpanel function.  
 uis\_breaks        A string with the names of user-specified indicators.

**Value**

A list of indicators.

**Examples**

```
data(EU_emissions_road)

# Group specification
EU15 <- c("Austria", "Germany", "Denmark", "Spain", "Finland", "Belgium",
         "France", "United Kingdom", "Ireland", "Italy", "Luxembourg",
         "Netherlands", "Greece", "Portugal", "Sweden")

# Prepare sample and data
EU_emissions_road_short <- EU_emissions_road[
  EU_emissions_road$country %in% EU15 &
  EU_emissions_road$year >= 2000,
  ]

# Run
result <- isatpanel(
  data = EU_emissions_road_short,
  formula = ltransport.emissions ~ lgdp + I(lgdp^2) + lpop,
  index = c("country", "year"),
  effect = "twoways",
  fesis = TRUE,
  plot = FALSE,
  t.pval = 0.01
)
```

```
plot(result)
plot_grid(result)

# print the retained indicators
get_indicators(result)
```

---

identify\_indicator\_timings

*Internal function to identify the timing of selected indicators*

---

## Description

Internal function to identify the timing of selected indicators

## Usage

```
identify_indicator_timings(object, uis_breaks = NULL)
```

## Arguments

object	data.frame
uis_breaks	A character vector with the names of the UIS breaks if the uis argument was used in <a href="#">isatpanel</a> .

## Value

A list of data.frames

---

isatpanel

*Indicator Saturation for Panel Data*

---

## Description

This function is essentially a wrapper function around the `gets::isat()` function from the `gets` package. This function allows the running of various different indicator saturation techniques that can, for example, be used to answer reverse causal questions. Indicator Saturation techniques fully saturate a model with indicators (for example dummy-indicators or step-indicators) and then use an automated block-search algorithm to retain only relevant indicators that improve the model (based on a chosen information criterion).

**Usage**

```

isatpanel(
  data = NULL,
  formula = NULL,
  index = NULL,
  effect = c("individual"),
  na.remove = TRUE,
  engine = NULL,
  user.estimator = NULL,
  cluster = "none",
  ar = 0,
  iis = FALSE,
  jiis = FALSE,
  jsis = FALSE,
  fesis = FALSE,
  csis = FALSE,
  cfesis = FALSE,
  csis_var = colnames(mxreg),
  fesis_id = NULL,
  cfesis_var = colnames(mxreg),
  cfesis_id = NULL,
  uis = NULL,
  t.pval = 0.001,
  plot = TRUE,
  plm_model = "within",
  y = NULL,
  id = NULL,
  time = NULL,
  mxreg = NULL,
  ...
)

```

**Arguments**

<code>data</code>	The input data.frame object.
<code>formula</code>	Please specify a formula argument. The dependent variable will be the left-most element, separated by a ~ symbol from the remaining regressors. Note the intercept will always be removed, if effect is not "none" - this means that if any fixed effects are specified, the intercept will always be removed.
<code>index</code>	Specify the name of the group and time column in the format c("id", "time").
<code>effect</code>	Fixed Effect specification. Possible arguments: "twoways", "individual", "time", or "none".
<code>na.remove</code>	remove NAs
<code>engine</code>	Estimation function to use. Default is NULL, which uses the default estimation procedure of the gets package. Alternatives are "fixest", "plm", or "felm".
<code>user.estimator</code>	Use a user.estimator



cluster	cluster Standard Errors at this level. Default is "none". Possible values are: "individual", "time", or "twoways".
ar	Autoregressive Term to be included. default is 0.
iis	Logical. Use Impulse Indicator Saturation.
jiis	Logical. Use Joint Impulse Indicator Saturation (Outliers are common across all units). This is essentially just a time fixed effect, but this allows selection of FE.
jsis	Logical. Use Join Step Indicator Saturation (steps are common across all units). Will only be retained if time fixed effects are not included (i.e. effect = 'none' or 'individual'), as they are collinear otherwise.
fesis	Logical. Use Fixed Effect Step Indicator Saturation. Constructed by multiplying a constant (1) with group Fixed Effects. Default is FALSE.
csis	Logical. Use Coefficient Step Indicator Saturation. Constructed by Default is FALSE.
cfesis	Logical. Use Coefficient-Fixed Effect Indicator Saturation. Default is FALSE.
csis_var	The csis method can be conducted for all (default) variables or just a subset of them. If you want to use a subset, please specify the column names of the variable in a character vector.
fesis_id	The fesis method can be conducted for all (default) individuals/units (i.e. looking for breaks in individual countries) or just a subset of them (joint breaks in the coefficients for a few individuals). If you want to use a subset, please specify the individuals/units for which you want to test the stability in a character vector.
cfesis_var	The cfesis method can be conducted for all variables (default) or just a subset of them. If you want to use a subset, please specify the column names of the variable in a character vector.
cfesis_id	The cfesis method can be conducted for all individuals/units (default) or just a subset of them. If you want to use a subset, please specify the individuals/units to be tested in a character vector.
uis	Matrix or List. This can be used to include a set of UIS (User Specified Indicators). Must be equal to the sample size (so it is recommended to use this only with datasets without NA values. Default is NULL. See the reference by Genaro Sucarrat (2020) below for an explanation of the UIS system.
t.pval	numeric value between 0 and 1. The significance level used for the two-sided regressor significance t-tests
plot	Logical. Should the final object be plotted? Default is TRUE. The output is a combination of plot and <code>plot_grid()</code> using the cowplot package.
plm_model	Type of PLM model (only if engine = "PLM")
y	Deprecated. The dependent variable. Can be used when data, index, and formula are not specified.
id	Deprecated. Can be used when data, index, and formula are not specified. Must be a vector of the grouping variable as a character or factor
time	Deprecated. Can be used when data, index, and formula are not specified. Must be a vector of the time variable as an integer or numeric.
mxreg	The co-variates matrix
...	Further arguments to <code>gets::isat()</code>

**Value**

A list with class 'isatpanel'.

**References**

Felix Pretis and Moritz Schwarz (2022). Discovering What Mattered: Answering Reverse Causal Questions by Detecting Unknown Treatment Assignment and Timing as Breaks in Panel Models. January 31, 2022. Available at SSRN: <https://ssrn.com/abstract=4022745> or <http://dx.doi.org/10.2139/ssrn.4022745>

Genaro Sucarrat. User-Specified General-to-Specific and Indicator Saturation Methods, The R Journal (2020) 12:2, pages 388-401. Available at: <https://journal.r-project.org/archive/2021/RJ-2021-024/index.html>

**See Also**

[gets::isat\(\)](#)

**Examples**

```
data(EU_emissions_road)

# Group specification
EU15 <- c("Austria", "Germany", "Denmark", "Spain", "Finland", "Belgium",
         "France", "United Kingdom", "Ireland", "Italy", "Luxembourg",
         "Netherlands", "Greece", "Portugal", "Sweden")

# Prepare sample and data
EU_emissions_road_short <- EU_emissions_road[
  EU_emissions_road$country %in% EU15 &
  EU_emissions_road$year >= 2000,
  ]

# Run
result <- isatpanel(
  data = EU_emissions_road_short,
  formula = ltransport.emissions ~ lgdp + I(lgdp^2) + lpop,
  index = c("country", "year"),
  effect = "twoways",
  fesis = TRUE,
  plot = FALSE,
  t.pval = 0.01
)
plot(result)
plot_grid(result)

# print the retained indicators
get_indicators(result)
```

---

logLik.plm	<i>Log-Likelihood Function for a plm object</i>
------------	---

---

**Description**

Log-Likelihood Function for a plm object

**Usage**

```
## S3 method for class 'plm'
logLik(object, ...)
```

**Arguments**

object	A plm object
...	Further Arguments

**Value**

The Log-Likelihood

---

pandata_simulated	<i>Simulated Panel Data</i>
-------------------	-----------------------------

---

**Description**

Simulated Panel Data

**Usage**

```
pandata_simulated
```

**Format**

A data frame with 400 rows and 9 variables:

**country** A random country  
**year** Year  
**gdp** A simulated Gross Domestic Product  
**temp** A simulated variable standing for temperature  
**const** The constant  
**country\_1** A dummy for country 1  
**country\_2** A dummy for country 2  
**country\_3** A dummy for country 3  
**country\_4** A dummy for country 4

...

**Source**

<https://github.com/moritzpschwarz/getspanel/>

---

plmFun	<i>plm Function to estimate isatpanel</i>
--------	---

---

**Description**

plm Function to estimate isatpanel

**Usage**

```
plmFun(y, x, time, id, cluster, effect, model = "pooling", ...)
```

**Arguments**

y	Dependent Variable
x	matrix or data.frame of regressors
time	Vector of time variable
id	Vector of group variable
cluster	cluster specification
effect	effect specification
model	model specification
...	Further arguments passed to plm

**Value**

A list to be used by gets::isat

---

plot.isatpanel	<i>Plotting an isatpanel object</i>
----------------	-------------------------------------

---

**Description**

Plotting an isatpanel object

**Usage**

```
## S3 method for class 'isatpanel'
plot(
  x,
  max.id.facet = 16,
  facet.scales = "free",
  title = NULL,
  zero_line = FALSE,
  ...
)
```

**Arguments**

x	An object produced by the isatpanel function
max.id.facet	The resulting plot will be faceted for each individual in the panel. Beyond a certain number, this might result in unreadable figures. Default set at 16.
facet.scales	To be passed to ggplot2::facet_wrap. Default is "free" (i.e. a separate y axis for each panel group/id). Alternatives are: "fixed", "fixed_y", and "fixed_x".
title	Plot title. Must be a character vector.
zero_line	Plot a horizontal line at y = 0. Default is FALSE.
...	Further arguments to be passed to ggplot2.

**Value**

A ggplot2 plot that plots an 'isatpanel' object and shows observed data, the fitted values, and all identified breaks and impulses.

---

plot\_counterfactual    *Plot the Counterfactual Path*

---

**Description**

Plot the Counterfactual Path

**Usage**

```
plot_counterfactual(
  x,
  plus_t = 5,
  facet.scales = "free",
  title = NULL,
  zero_line = FALSE
)
```

**Arguments**

x	An object produced by the isatpanel function
plus_t	Number of time periods for the counterfactual to be displayed (default = 5).
facet.scales	To be passed to ggplot2::facet_wrap. Default is "free" (i.e. a separate y axis for each panel group/id). Alternatives are: "fixed", "fixed_y", and "fixed_x".
title	Plot title. Must be a character vector.
zero_line	Plot a horizontal line at y = 0. Default is FALSE.

**Value**

A ggplot2 plot that plots an 'isatpanel' object and shows the counterfactuals for each break.

**Examples**

```

data(EU_emissions_road)

# Group specification
EU15 <- c("Austria", "Germany", "Denmark", "Spain", "Finland", "Belgium",
         "France", "United Kingdom", "Ireland", "Italy", "Luxembourg",
         "Netherlands", "Greece", "Portugal", "Sweden")

# Prepare sample and data
EU_emissions_road_short <- EU_emissions_road[
EU_emissions_road$country %in% EU15 &
EU_emissions_road$year >= 2000,
]

# Run
result <- isatpanel(
  data = EU_emissions_road_short,
  formula = ltransport.emissions ~ lgdp + I(lgdp^2) + lpop,
  index = c("country", "year"),
  effect = "twoways",
  fesis = TRUE,
  plot = FALSE,
  t.pval = 0.01
)
plot(result)
plot_grid(result)
plot_counterfactual(result)

```

---

plot\_grid

*Plotting an isatpanel object*


---

**Description**

Plotting an isatpanel object

**Usage**

```
plot_grid(x, title = NULL, regex_exclude_indicators = NULL, ...)
```

**Arguments**

x	An object produced by the isatpanel function
title	Plot title. Must be a character vector.
regex_exclude_indicators	A regex character vector to exclude the inclusion of certain indicators in the plot. Default = NULL. Use with care, experimental.
...	Further arguments to be passed to ggplot2.

**Value**

A ggplot2 plot that plots an 'isatpanel' object and shows all indicators as a grid to give a good and quick overview.

**Examples**

```
data(EU_emissions_road)

# Group specification
EU15 <- c("Austria", "Germany", "Denmark", "Spain", "Finland", "Belgium",
         "France", "United Kingdom", "Ireland", "Italy", "Luxembourg",
         "Netherlands", "Greece", "Portugal", "Sweden")

# Prepare sample and data
EU_emissions_road_short <- EU_emissions_road[
  EU_emissions_road$country %in% EU15 &
  EU_emissions_road$year >= 2000,
  ]

# Run
result <- isatpanel(
  data = EU_emissions_road_short,
  formula = ltransport.emissions ~ lgdp + I(lgdp^2) + lpop,
  index = c("country", "year"),
  effect = "twoways",
  fesis = TRUE,
  plot = FALSE,
  t.pval = 0.01
)
plot(result)
plot_grid(result)
```

---

plot\_residuals

*Plot Residuals from 'isatpanel' against OLS*

---

**Description**

Plot Residuals from 'isatpanel' against OLS

**Usage**

```
plot_residuals(isatpanelobject)
```

**Arguments**

isatpanelobject

An output from the 'isatpanel' function

**Value**

A ggplot2 plot that plots an 'isatpanel' object and shows the residuals over time in comparison to an OLS model.

**Examples**

```
data(EU_emissions_road)

# Group specification
EU15 <- c("Austria", "Germany", "Denmark", "Spain", "Finland", "Belgium",
          "France", "United Kingdom", "Ireland", "Italy", "Luxembourg",
          "Netherlands", "Greece", "Portugal", "Sweden")

# Prepare sample and data
EU_emissions_road_short <- EU_emissions_road[
  EU_emissions_road$country %in% EU15 &
  EU_emissions_road$year >= 2000,
]

# Run
result <- isatpanel(
  data = EU_emissions_road_short,
  formula = ltransport.emissions ~ lgdp + I(lgdp^2) + lpop,
  index = c("country", "year"),
  effect = "twoways",
  fesis = TRUE,
  plot = FALSE,
  t.pval = 0.01
)
plot(result)
plot_residuals(result)
```

---

print.isatpanel

*Printing isatpanel results*


---

**Description**

Printing isatpanel results

**Usage**

```
## S3 method for class 'isatpanel'
print(x, ...)
```

**Arguments**

x                    An isatpanel object.  
 ...                  Further arguments passed to print



**Value**

Print output of the 'isatpanel.result' list element of the 'isatpanel' object.

---

robust_isatpanel	<i>Get robust Standard Errors for the isatpanel result</i>
------------------	--

---

**Description**

Get robust Standard Errors for the isatpanel result

**Usage**

```
robust_isatpanel(
  object,
  robust = TRUE,
  HAC = FALSE,
  lag = NULL,
  type = "HC0",
  cluster = "group"
)
```

**Arguments**

object	An isatpanel object
robust	Logical (TRUE or FALSE). Should the Standard Errors be robustified for Heterogeneity? This uses <a href="#">plm::vcovHC</a> with the specified type (default is "HC0").
HAC	Should Heteroscedasticity and Autocorrelation Robust Standard Errors be used? This uses <a href="#">plm::vcovNW</a> , which uses the Newey-West estimator.
lag	Maximum Number of Lags to be used with <a href="#">plm::vcovNW</a> using the Newey-West estimator. Cannot be specified when HAC = FALSE. Default is NULL.
type	Character string. Type of Robust procedure e.g. 'HC0' for White SE or 'HC3' for Lang.
cluster	Should an object with clustered S.E. be included? Choose between 'group' or 'time' or FALSE. Uses <a href="#">plm::vcovHC</a> with the cluster argument.

**Value**

A list with robust estimates

**Examples**

```

data(EU_emissions_road)

# Group specification
EU15 <- c("Austria", "Germany", "Denmark", "Spain", "Finland", "Belgium",
         "France", "United Kingdom", "Ireland", "Italy", "Luxembourg",
         "Netherlands", "Greece", "Portugal", "Sweden")

# Prepare sample and data
EU_emissions_road_short <- EU_emissions_road[
EU_emissions_road$country %in% EU15 &
EU_emissions_road$year >= 2000,
]

# Run
result <- isatpanel(
  data = EU_emissions_road_short,
  formula = ltransport.emissions ~ lgdp + I(lgdp^2) + lpop,
  index = c("country", "year"),
  effect = "twoways",
  fesis = TRUE,
  plot = FALSE,
  t.pval = 0.01
)
robust_isatpanel(result)

```

---

 Within\_plm

*Use the within transformation from the plm package*


---

**Description**

Use the within transformation from the plm package

**Usage**

```
Within_plm(df, effect = "twoways")
```

**Arguments**

df	A data.frame object
effect	The fixed effect specification. Values possible: "twoways" (default), "individual", "time", "nested"

**Value**

A data.frame object with the transformation complete

# Index

## \* datasets

- EU\_emissions\_road, 4
- EUCO2residential, 3
- pandata\_simulated, 11

break\_uncertainty, 2

EU\_emissions\_road, 4  
EUCO2residential, 3

felmFun, 5  
fixestFun, 5

get\_indicators, 6  
gets::isat(), 7, 9, 10

identify\_indicator\_timings, 7  
isatpanel, 7, 7

logLik.plm, 11

pandata\_simulated, 11  
plm::vcovHC, 17  
plm::vcovNW, 17  
plmFun, 12  
plot.isatpanel, 12  
plot\_counterfactual, 13  
plot\_grid, 14  
plot\_grid(), 9  
plot\_residuals, 15  
print.isatpanel, 16

robust\_isatpanel, 17

Within\_plm, 18