

# Package ‘hgwr’

November 7, 2024

**Type** Package

**Title** Hierarchical and Geographically Weighted Regression

**Version** 0.6-0

**Date** 2024-11-06

**Maintainer** Yigong Hu <yigong.hu@bristol.ac.uk>

**Description** This model divides coefficients into three types, i.e., local fixed effects, global fixed effects, and random effects (Hu et al., 2022)<doi:10.1177/23998083211063885>. If data have spatial hierarchical structures (especially are overlapping on some locations), it is worth trying this model to reach better fitness.

**License** GPL (>= 2)

**URL** <https://github.com/HPDell/hgwr/>, <https://hpdell.github.io/hgwr/>

**Imports** Rcpp (>= 1.0.8)

**LinkingTo** Rcpp, RcppArmadillo

**Depends** R (>= 3.5.0), sf, stats, utils, MASS

**NeedsCompilation** yes

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0), furrr, progressr,

**SystemRequirements** GNU make

**RoxygenNote** 7.2.3

**VignetteBuilder** knitr

**Config/Needs/website** tidyverse, ggplot2, tmap, lme4, spdep, GWmodel

**Author** Yigong Hu [aut, cre],  
Richard Harris [aut],  
Richard Timmerman [aut]

**Repository** CRAN

**Date/Publication** 2024-11-07 09:50:02 UTC

## Contents

|                          |           |
|--------------------------|-----------|
| hgwrr-package            | 2         |
| coef.hgwr                | 3         |
| fitted.hgwr              | 4         |
| logLik.hgwr              | 4         |
| make_dummy               | 5         |
| multsam.test             | 6         |
| multisampling            | 7         |
| print.hgwr               | 8         |
| print.spahetbootres      | 9         |
| print.summary.hgwr       | 9         |
| print_table_md           | 10        |
| residuals.hgwr           | 11        |
| spatial_hetero_test      | 12        |
| spatial_hetero_test.hgwr | 13        |
| spatial_hetero_test_data | 17        |
| summary.hgwr             | 18        |
| wuhan.hp                 | 19        |
| <b>Index</b>             | <b>22</b> |

---

 hgwrr-package

*HGWR: Hierarchical and Geographically Weighted Regression*


---

## Description

An R and C++ implementation of Hierarchical and Geographically Weighted Regression (HGWR) model is provided in this package. This model divides coefficients into three types: local fixed effects, global fixed effects, and random effects. If data have spatial hierarchical structures (especially are overlapping on some locations), it is worth trying this model to reach better fitness.

## Details

|              |   |
|--------------|---|
| Package:     | hgwrr   |
| Type:        | Package   |
| Title:       | Hierarchical and Geographically Weighted Regression   |
| Version:     | 0.6-0   |
| Date:        | 2024-11-06  |
| Authors@R:   | c(person(given = "Yigong", family = "Hu", role = c("aut", "cre"), email = "yigong.hu@bristol.ac.uk")  |
| Maintainer:  | Yigong Hu <yigong.hu@bristol.ac.uk>   |
| Description: | This model divides coefficients into three types, i.e., local fixed effects, global fixed effects, and random effects.  |
| License:     | GPL (>= 2)  |
| URL:         | <a href="https://github.com/HPDell/hgwrr/">https://github.com/HPDell/hgwrr/</a> , <a href="https://hpdell.github.io/hgwrr/">https://hpdell.github.io/hgwrr/</a> |
| Imports:     | Rcpp (>= 1.0.8)   |
| LinkingTo:   | Rcpp, RcppArmadillo   |
| Depends:     | R (>= 3.5.0), sf, stats, utils, MASS  |

NeedsCompilation: yes  
 Suggests: knitr, rmarkdown, testthat (>= 3.0.0), furr, progressr,  
 SystemRequirements: GNU make  
 Roxygen: list(markdown = TRUE)  
 RoxygenNote: 7.2.3  
 VignetteBuilder: knitr  
 Config/Needs/website: tidyverse, ggplot2, tmap, lme4, spdep, GWmodel  
 Author: Yigong Hu [aut, cre], Richard Harris [aut], Richard Timmerman [aut]

## Note

Acknowledgement: We gratefully acknowledge support from China Scholarship Council.

## Author(s)

Yigong Hu, Richard Harris, Richard Timmerman

## References

Hu, Y., Lu, B., Ge, Y., Dong, G., 2022. Uncovering spatial heterogeneity in real estate prices via combined hierarchical linear model and geographically weighted regression. *Environment and Planning B: Urban Analytics and City Science*. doi:10.1177/23998083211063885

---

|            |                                    |
|------------|------------------------------------|
| coef.hgwrn | <i>Get estimated coefficients.</i> |
|------------|------------------------------------|

---

## Description

Get estimated coefficients.

## Usage

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

## Arguments

object            An hgwrn object returned by `hgwr()`.  
 ...                Parameter received from other functions.

## Value

A `DataFrame` object consists of all estimated coefficients.

**See Also**

[hgwr\(\)](#), [summary.hgwrn\(\)](#), [fitted.hgwrn\(\)](#) and [residuals.hgwrn\(\)](#).

---

|              |                             |
|--------------|-----------------------------|
| fitted.hgwrn | <i>Get fitted response.</i> |
|--------------|-----------------------------|

---

**Description**

Get fitted response.

**Usage**

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

**Arguments**

|        |  |
|--------|--|
| object | An hgwrn object returned by <a href="#">hgwr()</a> . |
| ...    | Parameter received from other functions.             |

**Value**

A vector consists of fitted response values.

**See Also**

[hgwr\(\)](#), [summary.hgwrn\(\)](#), [coef.hgwrn\(\)](#) and [residuals.hgwrn\(\)](#).

---

|              |                                |
|--------------|--------------------------------|
| logLik.hgwrn | <i>Log likelihood function</i> |
|--------------|--------------------------------|

---

**Description**

Log likelihood function

**Usage**

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

**Arguments**

|        |                       |
|--------|-----------------------|
| object | An hgwrn object.      |
| ...    | Additional arguments. |

**Value**

An logLik instance used for S3 method [logLik\(\)](#).

**Description**

Function `make_dummy` converts categorical variables in a data frame to dummy variables.

Function `make_dummy_extract` converts a column to dummy variables if necessary and assign appropriate names. See the "detail" section for further information. Users can define their own functions to allow the model deal with some types of variables properly.

**Usage**

```
make_dummy(data)

make_dummy_extract(col, name)

## S3 method for class 'character'
make_dummy_extract(col, name)

## S3 method for class 'factor'
make_dummy_extract(col, name)

## S3 method for class 'logical'
make_dummy_extract(col, name)

## Default S3 method:
make_dummy_extract(col, name)
```

**Arguments**

|                   |   |
|-------------------|---|
| <code>data</code> | The data frame from which dummy variables need to be extracted. |
| <code>col</code>  | A vector to extract dummy variables.                            |
| <code>name</code> | The vector's name.  |

**Details**

If `col` is a character vector, the function will get unique values of its elements and leave out the last one. Then, all the unique values are combined with the `name` argument as names of new columns.

If `col` is a factor vector, the function will get its levels and leave out the last one. Then, all level labels are combined with the `name` argument as names of new columns.

If `col` is a logical vector, the function will convert it to a numeric vector with value `TRUE` mapped to 1 and `FALSE` to 0.

If `col` is of other types, the default behaviour for extracting dummy variables is just to copy the original value and try to convert it to numeric values.

**Value**

The data frame with extracted dummy variables.

**Examples**

```
make_dummy(iris["Species"])  
make_dummy_extract(iris$Species, "Species")  
make_dummy_extract(c("top", "mid", "low", "mid", "top"), "level")  
make_dummy_extract(factor(c("far", "near", "near")), "distance")  
make_dummy_extract(c(TRUE, TRUE, FALSE), "sold")
```

---

mulsam.test

*Simulated Spatial Multisampling Data For Test (DataFrame)*

---

**Description**

A simulation data set for testing use of spatial hierarchical structure and samples overlapping on certain locations.

**Usage**

```
data(mulsam.test)
```

**Format**

A list of three items called "data", "coords" and "beta". Item "data" is a data frame with 873 observations at 25 locations and the following 6 variables.

$y$  a numeric vector, dependent variable  $y$   
 $g_1$  a numeric vector, group level independent variable  $g_1$   
 $g_2$  a numeric vector, group level independent variable  $g_2$   
 $z_1$  a numeric vector, sample level independent variable  $z_1$   
 $x_1$  a numeric vector, sample level independent variable  $x_1$   
group a numeric vector, group id of each sample

where  $g_1$  and  $g_2$  are used to estimate local fixed effects;  $x_1$  is used to estimate global fixed effects and  $z_1$  is used to estimate random effects.

**Author(s)**

Yigong Hu <yigong.hu@bristol.ac.uk>

**Examples**

```
data(mulsam.test)
hgwr(formula = y ~ L(g1 + g2) + x1 + (z1 | group),
      data = mulsam.test$data,
      coords = mulsam.test$coords,
      bw = 10, kernel = "bisquared")
```

---

multisampling

*Large Scale Simulated Spatial Multisampling Data (DataFrame)*

---

**Description**

A simulation data of spatial hierarchical structure and samples overlapping on certain locations.

**Usage**

```
data(multisampling)
```

**Format**

A list of three items called "data", "coords" and "beta". Item "data" is a data frame with 21434 observations at 625 locations and the following 6 variables.

*y* a numeric vector, dependent variable *y*  
*g1* a numeric vector, group level independent variable  $g_1$   
*g2* a numeric vector, group level independent variable  $g_2$   
*z1* a numeric vector, sample level independent variable  $z_1$   
*x1* a numeric vector, sample level independent variable  $x_1$   
*group* a numeric vector, group id of each sample

where *g1* and *g2* are used to estimate local fixed effects; *x1* is used to estimate global fixed effects and *z1* is used to estimate random effects.

**Author(s)**

Yigong Hu <yigong.hu@bristol.ac.uk>

**Examples**

```
## Not run:
data(multisampling)
hgwr(formula = y ~ L(g1 + g2) + x1 + (z1 | group),
      data = multisampling$data,
      coords = multisampling$coords,
      bw = 32)

## End(Not run)
```

---

```
print.hgwrn          Print description of a hgwrn object.
```

---

### Description

Print description of a hgwrn object.

### Usage

```
## S3 method for class 'hgwrn'
print(x, decimal.fmt = "%.6f", ...)
```

### Arguments

|                           |  |
|---------------------------|--|
| <code>x</code>            | An hgwrn object returned by <code>hgwr()</code> .  |
| <code>decimal.fmt</code>  | The format string passing to <code>base::sprintf()</code> .  |
| <code>...</code>          | Arguments passed on to <code>print_table_md</code>   |
| <code>col_sep</code>      | Column separator. Default to <code>" "</code> .  |
| <code>header_sep</code>   | Header separator. Default to <code>"-"</code> . If <code>header_sep</code> only contains one character, it will be repeated for each column. If it contains more than one character, it will be printed below the first row. |
| <code>row_begin</code>    | Character at the beginning of each row. Default to <code>col_sep</code> .  |
| <code>row_end</code>      | Character at the ending of each row. Default to <code>col_sep</code> .   |
| <code>table_before</code> | Characters to be printed before the table.   |
| <code>table_after</code>  | Characters to be printed after the table.  |
| <code>table_style</code>  | Name of pre-defined style. Possible values are <code>"plain"</code> , <code>"md"</code> , <code>"latex"</code> , or <code>"booktabs"</code> . Default to <code>"plain"</code> .  |

### Value

No return.

### See Also

`summary.hgwrn()`, `print_table_md()`.

### Examples

```
data(mulsam.test)
model <- hgwr(
  formula = y ~ L(g1 + g2) + x1 + (z1 | group),
  data = mulsam.test$data,
  coords = mulsam.test$coords,
  bw = 10
)
print(model)
print(model, table.style = "md")
```



---

```
print.spahetbootres    Print the result of spatial heterogeneity test
```

---

**Description**

Print the result of spatial heterogeneity test

**Usage**

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

**Arguments**

|     |                         |
|-----|-------------------------|
| x   | A spahetbootres object. |
| ... | Other unused arguments. |

---

```
print.summary.hgwrn    Print summary of an hgwrn object.
```

---

**Description**

Print summary of an hgwrn object.

**Usage**

```
## S3 method for class 'summary.hgwrn'
print(x, decimal.fmt = "%.6f", ...)
```

**Arguments**

|              |   |
|--------------|---|
| x            | An object returned from <code>summary.hgwrn()</code> .  |
| decimal.fmt  | The format string passing to <code>base::sprintf()</code> .   |
| ...          | Arguments passed on to <code>print_table_md</code>  |
| col_sep      | Column separator. Default to "".  |
| header_sep   | Header separator. Default to "-". If header_sep only contains one character, it will be repeated for each column. If it contains more than one character, it will be printed below the first row. |
| row_begin    | Character at the beginning of each row. Default to col_sep.   |
| row_end      | Character at the ending of each row. Default to col_sep.  |
| table_before | Characters to be printed before the table.  |
| table_after  | Characters to be printed after the table.   |
| table_style  | Name of pre-defined style. Possible values are "plain", "md", "latex", or "booktabs". Default to "plain".   |

**Value**

No return.

**See Also**

[summary.hgwr\(\)](#), [print\\_table\\_md\(\)](#).

**Examples**

```
data(mulsam.test)
model <- hgwr(
  formula = y ~ L(g1 + g2) + x1 + (z1 | group),
  data = mulsam.test$data,
  coords = mulsam.test$coords,
  bw = 10
)
summary(model)
```

---

print\_table\_md

*Print a character matrix as a table.*

---

**Description**

Print a character matrix as a table.

**Usage**

```
print_table_md(
  x,
  col_sep = "",
  header_sep = "",
  row_begin = "",
  row_end = "",
  table_before = NA_character_,
  table_after = NA_character_,
  table_style = c("plain", "md", "latex", "booktabs"),
  ...
)
```

**Arguments**

|            |   |
|------------|---|
| x          | A character matrix.   |
| col_sep    | Column separator. Default to "".  |
| header_sep | Header separator. Default to "-". If header_sep only contains one character, it will be repeated for each column. If it contains more than one character, it will be printed below the first row. |

|              |   |
|--------------|---|
| row_begin    | Character at the beginning of each row. Default to col_sep.   |
| row_end      | Character at the ending of each row. Default to col_sep.  |
| table_before | Characters to be printed before the table.  |
| table_after  | Characters to be printed after the table.   |
| table_style  | Name of pre-defined style. Possible values are "plain", "md", "latex", or "booktabs". Default to "plain". |
| ...          | Additional style control arguments.   |

### Details

When table\_style is specified, col\_sep, header\_sep, row\_begin and row\_end would not take effects. Because this function will automatically set their values. For each possible value of table\_style, its corresponding style settings are shown in the following table.

|            | plain | md  | latex |
|------------|-------|-----|-------|
| col_sep    | ""    | " " | "&"   |
| header_sep | ""    | "_" | ""    |
| row_begin  | ""    | " " | ""    |
| row_end    | ""    | " " | "\\"  |

In this function, characters are right padded by spaces.

### Value

No return.

### See Also

[print.hgwrn\(\)](#), [summary.hgwrn\(\)](#).

---

|                 |                       |
|-----------------|-----------------------|
| residuals.hgwrn | <i>Get residuals.</i> |
|-----------------|-----------------------|

---

### Description

Get residuals.

### Usage

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

### Arguments

|        |  |
|--------|--|
| object | An hgwrn object returned by <a href="#">hgwr()</a> . |
| ...    | Parameter received from other functions.             |

**Value**

A vector consists of residuals.

**See Also**

[hgwr\(\)](#), [summary.hgwrm\(\)](#), [coef.hgwrm\(\)](#) and [fitted.hgwrm\(\)](#).

---

spatial\_hetero\_test    *Generic method to test spatial heterogeneity*

---

**Description**

Generic method to test spatial heterogeneity

**Usage**

```
spatial_hetero_test(x, ...)

## Default S3 method:
spatial_hetero_test(x, ...)

## S3 method for class 'matrix'
spatial_hetero_test(x, coords, ...)

## S3 method for class 'numeric'
spatial_hetero_test(x, coords, ...)

## S3 method for class 'vector'
spatial_hetero_test(x, coords, ...)

## S3 method for class 'data.frame'
spatial_hetero_test(x, coords, ...)

## S3 method for class 'sf'
spatial_hetero_test(x, ...)
```

**Arguments**

|          |  |
|----------|--|
| x        | The data to be tested.   |
| ...      | Arguments passed on to <a href="#">spatial_hetero_test_data</a> , <a href="#">spatial_hetero_test_data</a> |
| resample | The total times of resampling with replacement. Default to 5000.   |
| poly     | The number of polynomial terms used by the polynomial estimator. Default to 2.                             |
| bw       | The adaptive bandwidth used by the polynomial estimator. Default to 10.                                    |
| kernel   | The kernel function used by the polynomial estimator.  |
| verbose  | The verbosity level. Default to 0.   |

coords            The coordinates used for testing. Accepts a matrix or vector. For matrix, it needs to have the same number of rows as x. For vector, it indicates the columns in x and the actual coordinates will be taken from x.

### Methods (by class)

- `spatial_hetero_test(default)`: Default behavior.
- `spatial_hetero_test(matrix)`: For the matrix, `coords` is necessary.
- `spatial_hetero_test(numeric)`: Takes `x` as values of a series variables stored by column, and `coords` as coordinates for each row in `x`.
- `spatial_hetero_test(vector)`: Takes `x` as values of the variable, and `coords` as coordinates for each element in `x`.
- `spatial_hetero_test(data.frame)`: Takes `x` as variable values (each column is a variable), and `coords` as coordinates for each row in `x`.
- `spatial_hetero_test(sf)`: For the `sf` object, coordinates of centroids are used. Only the numerical columns are tested.

---

spatial\_hetero\_test.hgwrn

*Hierarchical and Geographically Weighted Regression*

---

### Description

A Hierarchical Linear Model (HLM) with group-level geographically weighted effects.

### Usage

```
## S3 method for class 'hgwrn'
spatial_hetero_test(
  x,
  round = 99,
  statistic = stat_gls,
  parallel = FALSE,
  verbose = 0,
  ...
)

hgwr(
  formula,
  data,
  ...,
  bw = "CV",
  kernel = c("gaussian", "bisquared"),
  alpha = 0.01,
  eps_iter = 1e-06,
```

```
    eps_gradient = 1e-06,
    max_iters = 1e+06,
    max_retries = 1e+06,
    ml_type = c("D_Only", "D_Beta"),
    f_test = FALSE,
    verbose = 0
)

## S3 method for class 'sf'
hgwr(
  formula,
  data,
  ...,
  bw = "CV",
  kernel = c("gaussian", "bisquared"),
  alpha = 0.01,
  eps_iter = 1e-06,
  eps_gradient = 1e-06,
  max_iters = 1e+06,
  max_retries = 1e+06,
  ml_type = c("D_Only", "D_Beta"),
  f_test = FALSE,
  verbose = 0
)

## S3 method for class 'data.frame'
hgwr(
  formula,
  data,
  ...,
  coords,
  bw = "CV",
  kernel = c("gaussian", "bisquared"),
  alpha = 0.01,
  eps_iter = 1e-06,
  eps_gradient = 1e-06,
  max_iters = 1e+06,
  max_retries = 1e+06,
  ml_type = c("D_Only", "D_Beta"),
  f_test = FALSE,
  verbose = 0
)

hgwr_fit(
  formula,
  data,
  coords,
  bw = c("CV", "AIC"),
```

```

kernel = c("gaussian", "bisquared"),
alpha = 0.01,
eps_iter = 1e-06,
eps_gradient = 1e-06,
max_iters = 1e+06,
max_retries = 1e+06,
ml_type = c("D_Only", "D_Beta"),
f_test = FALSE,
verbose = 0
)

```

### Arguments

|              |   |
|--------------|---|
| x            | An hgwrn object   |
| round        | The number of times to sampling from model.   |
| statistic    | A function used to calculate the statistics on the original data and bootstrapped data. Default to the variance of standardlised GLSW estimates.  |
| parallel     | If TRUE, use furrr package to parallel.   |
| verbose      | An integer value. Determine the log level. Possible values are:<br><b>0</b> no log is printed.<br><b>1</b> only logs in back-fitting are printed.<br><b>2</b> all logs are printed.   |
| ...          | Further arguments for the specified type of data.   |
| formula      | A formula. Its structure is similar to <code>lmer</code> function in <b>lme4</b> package. Models can be specified with the following form:<br><br>$\text{response} \sim \text{L}(\text{glsw}) + \text{fixed} + (\text{random} \mid \text{group})$ For more information, please see the formula subsection in details. |
| data         | The data.   |
| bw           | A numeric value. It is the value of bandwidth or "CV". In this stage this function only support adaptive bandwidth. And its unit must be the number of nearest neighbours. If "CV" is specified, the algorithm will automatically select an optimized bandwidth value.  |
| kernel       | A character value. It specify which kernel function is used in GWR part. Possible values are<br>gaussian Gaussian kernel function $k(d) = \exp\left(-\frac{d^2}{b^2}\right)$<br>bisquared Bi-squared kernel function. If $d < b$ then $k(d) = \left(1 - \frac{d^2}{b^2}\right)^2$ else $k(d) = 0$                     |
| alpha        | A numeric value. It is the size of the first trial step in maximum likelihood algorithm.  |
| eps_iter     | A numeric value. Terminate threshold of back-fitting.   |
| eps_gradient | A numeric value. Terminate threshold of maximum likelihood algorithm.   |

|             |  |
|-------------|--|
| max_iters   | An integer value. The maximum of iteration.  |
| max_retries | An integer value. If the algorithm tends to be diverge, it stops automatically after trying <i>max_retries</i> times.  |
| ml_type     | An integer value. Represent which maximum likelihood algorithm is used. Possible values are:<br>D_Only Only <i>D</i> is specified by maximum likelihood.<br>D_Beta Both <i>D</i> and <i>beta</i> is specified by maximum likelihood. |
| f_test      | A logical value. Determine whether to do F test on GLSW effects. If <i>f_test</i> =TRUE, there will be a <i>f_test</i> item in the returned object showing the F test for each GLSW effect.  |
| coords      | A 2-column matrix. It consists of coordinates for each group.  |

## Details

### Effect Specification in Formula:

In the HGWR model, there are three types of effects specified by the formula argument:

**Group-level spatially weighted (GLSW, aka. local fixed) effects** Effects wrapped by functional symbol *L*.

**Sample-level random (SLR) effects** Effects specified outside the functional symbol *L* but to the left of symbol *|*.

**Fixed effects** Other effects

For example, the following formula in the example of this function below is written as

$$y \sim L(g1 + g2) + x1 + (z1 | group)$$

where *g1* and *g2* are GLSW effects, *x1* is the fixed effects, and *z1* is the SLR effects grouped by the group indicator *group*. Note that SLR effects can only be specified once!

## Value

A list describing the model with following fields.

*gamma* Coefficients of group-level spatially weighted effects.

*beta* Coefficients of fixed effects.

*mu* Coefficients of sample-level random effects.

*D* Variance-covariance matrix of sample-level random effects.

*sigma* Variance of errors.

*effects* A list including names of all effects.

*call* Calling of this function.

*frame* The DataFrame object sent to this call.

*frame.parsed* Variables extracted from the data.

*groups* Unique group labels extracted from the data.

*f\_test* A list of F test for GLSW effects. Only exists when *f\_test*=TRUE. Each item contains the F value, degrees of freedom in the numerator, degrees of freedom in the denominator, and *p* value of  $F > F_{\alpha}$ .



**Functions**

- `spatial_hetero_test(hgwr)`: Test the spatial heterogeneity with bootstrapping.
- `hgwr_fit()`: Fit a HGWR model

**Examples**

```
data(mulsam.test)
hgwr(
  formula = y ~ L(g1 + g2) + x1 + (z1 | group),
  data = mulsam.test$data,
  coords = mulsam.test$coords,
  bw = 10
)

mod_Ftest <- hgwr(
  formula = y ~ L(g1 + g2) + x1 + (z1 | group),
  data = mulsam.test$data,
  coords = mulsam.test$coords,
  bw = 10
)
summary(mod_Ftest)
```

---

`spatial_hetero_test_data`

*Test the spatial heterogeneity in data based on permutation.*

---

**Description**

Test the spatial heterogeneity in data based on permutation.

**Usage**

```
spatial_hetero_test_data(
  x,
  coords,
  ...,
  resample = 5000,
  poly = 2,
  bw = 10,
  kernel = c("bisquared", "gaussian"),
  verbose = 0
)
```

**Arguments**

|          |  |
|----------|--|
| x        | A matrix of data to be tested. Each column is a variable.                      |
| coords   | A matrix of coordinates.   |
| ...      | Additional arguments.  |
| resample | The total times of resampling with replacement. Default to 5000.               |
| poly     | The number of polynomial terms used by the polynomial estimator. Default to 2. |
| bw       | The adaptive bandwidth used by the polynomial estimator. Default to 10.        |
| kernel   | The kernel function used by the polynomial estimator.                          |
| verbose  | The verbosity level. Default to 0.   |

**Value**

A spahetbootres object of permutation-test results with the following items:

vars The names of variables.  
 t0 The value of the statistics on original values.  
 t The value of the same statistics on permuted values.  
 p The p-value for each variable.

Currently, variance is used as the statistics.

---

|               |                                 |
|---------------|---------------------------------|
| summary.hgwrn | <i>Summary an hgwrn object.</i> |
|---------------|---------------------------------|

---

**Description**

Summary an hgwrn object.

**Usage**

```
## S3 method for class 'hgwrn'
summary(object, ..., test_hetero = FALSE, verbose = 0)
```

**Arguments**

|             |   |
|-------------|---|
| object      | An hgwrn object returned from <code>hgwr()</code> .   |
| ...         | Other arguments passed from other functions.  |
| test_hetero | Logical/list value. Whether to test the spatial heterogeneity of GLSW effects. If it is set to FALSE, the test will not be executed. If it is set to TRUE, the test will be executed with default parameters (see details below). It accepts a list to enable the test with specified parameters. |
| verbose     | An Integer value to control whether additional messages during testing spatial heterogeneity should be reported.  |

**Details**

The parameters used to perform test of spatial heterogeneity are

`bw` Bandwidth (unit: number of nearest neighbours) used to make spatial kernel density estimation.

Default: 10.

`poly` The number of polynomial terms used in the local polynomial estimation. Default: 2.

`resample` Total resampling times. Default: 5000.

`kernel` The kernel function used in the local polynomial estimation. Options are "gaussian" and "bisquared". Default: "bisquared".

**Value**

A list containing summary informations of this `hgwr` object with the following fields.

`diagnostic` A list of diagnostic information.

`random.stddev` The standard deviation of random effects.

`random.corr` The correlation matrix of random effects.

`residuals` The residual vector.

**See Also**

[hgwr\(\)](#).

**Examples**

```
data(mulsam.test)
m <- hgwr(
  formula = y ~ L(g1 + g2) + x1 + (z1 | group),
  data = mulsam.test$data,
  coords = mulsam.test$coords,
  bw = 10
)
summary(m)
summary(m, test_hetero = TRUE)
summary(m, test_hetero = list(kernel = "gaussian"))
```

---

wuhan.hp

*Wuhan Second-hand House Price and POI Data (DataFrame)*


---

**Description**

A data set of second-hand house price in Wuhan, China collected in 2018.

**Usage**

```
data(multisampling)
```

**Format**

A list of two items called "data" and "coords". Item "data" is a data frame with 13862 second-hand properties at 779 neighbourhoods and the following 22 variables.

Price House price per square metre.

Floor.High 1 if a property is on a high floor, otherwise 0.

Floor.Low 1 if a property is on a low floor, otherwise 0.

Decoration.Fine 1 if a property is well decorated, otherwise 0.

PlateTower 1 if a property is of the plate-tower type, otherwise 0.

Steel 1 if a property is of 'steel' structure, otherwise 0.

BuildingArea Building area in square metres.

Fee Management fee per square meter per month.

d.Commercial Distance to the nearest commercial area.

d.Greenland Distance to the nearest green land.

d.Water Distance to the nearest river or lake.

d.University Distance to the nearest university.

d.HighSchool Distance to the nearest high school.

d.MiddleSchool Distance to the nearest middle school.

d.PrimarySchool Distance to the nearest primary school.

d.Kindergarten Distance to the nearest kindergarten.

d.SubwayStation Distance to the nearest subway station.

d.Supermarket Distance to the nearest supermarket.

d.ShoppingMall Distance to the nearest shopping mall.

lon Longitude coordinates (Projected CRS: EPSG 3857).

lat Latitude coordinates (Projected CRS: EPSE 3857).

group Group id of each sample.

The following variables are group level:

- Fee - d.Commercial - d.Greenland - d.Water - d.University - d.HighSchool - d.MiddleSchool  
- d.PrimarySchool - d.Kindergarten - d.SubwayStation - d.Supermarket - d.ShoppingMall

The following variables are sample level:

- Price - Floor.High - Floor.Low - Decoration.Fine - PlateTower - Steel - BuildingArea

Item "coords" is a 779-by-2 matrix of coordinates of all neighbourhoods.

**Author(s)**

Yigong Hu <yigong.hu@bristol.ac.uk>

### Examples

```
## Not run:
data(wuhan.hp)
hgw(
  formula = Price ~ L(d.Water + d.Commercial + d.PrimarySchool +
    d.Kindergarten + Fee) + BuildingArea + (Floor.High | group),
  data = wuhan.hp$data,
  coords = wuhan.hp$coords, bw = 50, kernel = "bisquared")

## End(Not run)
```

# Index

`base::sprintf()`, 8, 9

`coef.hgwr`, 3  
`coef.hgwr()`, 4, 12

`fitted.hgwr`, 4  
`fitted.hgwr()`, 4, 12

`hgwr (spatial_hetero_test.hgwr)`, 13  
`hgwr()`, 3, 4, 8, 11, 12, 18, 19  
`hgwr_fit (spatial_hetero_test.hgwr)`, 13  
`hgwr`-package, 2

`lmer`, 15  
`logLik.hgwr`, 4

`make_dummy`, 5  
`make_dummy_extract (make_dummy)`, 5  
`mulsam.test`, 6  
`multisampling`, 7

`print.hgwr`, 8  
`print.hgwr()`, 11  
`print.spahetbootres`, 9  
`print.summary.hgwr`, 9  
`print_table_md`, 8, 9, 10  
`print_table_md()`, 8, 10

`residuals.hgwr`, 11  
`residuals.hgwr()`, 4

`spatial_hetero_test`, 12  
`spatial_hetero_test.hgwr`, 13  
`spatial_hetero_test_data`, 12, 17  
`summary.hgwr`, 18  
`summary.hgwr()`, 4, 8–12

`wuhan.hp`, 19