Package ‘GWPR.light’

October 18, 2021

Type Package

Title Geographically Weighted Panel Regression

Version 0.1.1

Author Chao Li [aut, cre] (<https://orcid.org/0000-0002-6854-4456>),
Shunsuke Managi [aut] (<https://orcid.org/0000-0001-7883-1427>)

Maintainer Chao Li <chaoli0394@gmail.com>

Description Geographically weighted panel regression is grounded in a branch of spatial statistics. Using geographically weights, the geographically weighted panel regression is try to solve the residuals from panel regression clustering spatially. To investigate whether the residuals cluster spatially, the Moran's I test is also improved. Furthermore, three local statistic tests are contained to help the users select model. The major references are Fotheringham et al. (2003, ISBN:978-0-470-85525-6) and Beenstock and Felsenstein (2019, ISBN:978-3-030-03614-0).

License AGPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Imports data.table, doParallel, dplyr, foreach, GWmodel, iterators,
    lmtest, methods, parallel, plm, sp, stats

Depends R (>= 2.10)

Suggests rmarkdown, knitr, rgeos, tmap

VignetteBuilder knitr

URL https://github.com/MichaelChaoLi-cpu/GWPR.light

BugReports https://github.com/MichaelChaoLi-cpu/GWPR.light/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2021-10-18 09:10:06 UTC
Description

This package are grounded in a branch of spatial statistics. Using geographically weights, the geographically weighted panel regression is try to solve the residuals from panel regression clustering spatially. To investigate whether the residuals cluster spatially, the Moran’s I test is also improved. Furthermore, three local statistic tests are contained to help the users select model. This package includes the function for the optimal bandwidth selection in GWPR, the function for GWPR, the function for the local Hausman test, the function for the local F test for individual effects, the function for the local Lagrange Multiplier Breusch-Pagan test, and the function for panel Moran’s I test. The functions have been optimized, which require the less memory in the calculation.

Details

Package: GWPR.light
Type: Package
Version: 0.1.0
Date: 2021-10-02
License: AGPL (>= 3)
LazyLoad: yes

Author(s)

Chao Li <chaoli0394@gmail.com> [aut, cre] Shunsuke Managi <managi@doc.kyushu-u.ac.jp> [aut]
Description

A function for automatic bandwidth selection to calibrate a GWPR model

Usage

bw.GWPR(formula, data, index, SDF, adaptive = FALSE, p = 2, bigdata = FALSE,
         upprerratio = 0.25, effect = "individual",
         model = c("pooling", "within", "random"),
         random.method = "swar", approach = c("CV", "AIC"), kernel = "bisquare",
         longlat = FALSE, doParallel = FALSE, cluster.number = 2,
         human.set.range = FALSE, h.upper = NULL, h.lower = NULL)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>formula</td>
<td>The regression formula: ( Y \sim X_1 + \ldots + X_k )</td>
</tr>
<tr>
<td>data</td>
<td>data.frame for the Panel data</td>
</tr>
<tr>
<td>index</td>
<td>A vector of the two indexes: (c(&quot;ID&quot;, &quot;Time&quot;))</td>
</tr>
<tr>
<td>SDF</td>
<td>Spatial*DataFrame on which is based the data, with the &quot;ID&quot; in the index</td>
</tr>
<tr>
<td>adaptive</td>
<td>If TRUE, adaptive distance bandwidth is used, otherwise, fixed distance band- width.</td>
</tr>
<tr>
<td>p</td>
<td>The power of the Minkowski distance, default is 2, i.e. the Euclidean distance</td>
</tr>
<tr>
<td>bigdata</td>
<td>TRUE or FALSE, if the dataset exceeds 40,000, we strongly recommend set it TRUE</td>
</tr>
<tr>
<td>upprerratio</td>
<td>Set the ratio between upper boundary of potential bandwidth range and the forthest distance of SDF, if bigdata = T. (default value: 0.25)</td>
</tr>
<tr>
<td>effect</td>
<td>The effects introduced in the model, one of &quot;individual&quot; (default) , &quot;time&quot;, &quot;twoways&quot;, or &quot;nested&quot;</td>
</tr>
<tr>
<td>model</td>
<td>Panel model transformation: (c(&quot;within&quot;, &quot;random&quot;, &quot;pooling&quot;))</td>
</tr>
<tr>
<td>random.method</td>
<td>Method of estimation for the variance components in the random effects model, one of &quot;swar&quot; (default), &quot;amemiya&quot;, &quot;walhus&quot;, or &quot;nerlove&quot;</td>
</tr>
<tr>
<td>approach</td>
<td>Score used to optimize the bandwidth, c(&quot;CV&quot;, &quot;AIC&quot;)</td>
</tr>
</tbody>
</table>
| kernel    | bisquare: \( wgt = \left(1-(vdist/bw)^2\right)^2 \) if vdist < bw, \( wgt=0 \) otherwise (default);

\[
gaussian: \quad wgt = \exp\left(-0.5\left(vdist/bw\right)^2\right); 
\text{exponential:} \quad wgt = \exp\left(-vdist/bw\right); 
\text{tricube:} \quad wgt = \left(1-(vdist/bw)^3\right)^3 \] if vdist < bw, \( wgt=0 \) otherwise; 
\text{boxcar:} \quad wgt=1 \text{ if dist < bw, } wgt=0 \text{ otherwise}  |
| longlat   | If TRUE, great circle distances will be calculated                         |
| doParallel| If TRUE, "cluster": multi-process technique with the parallel package would be used. |
cluster.number The number of the clusters that user wants to use

human.set.range

    If TRUE, the range of bandwidth selection could be set by the user

h.upper The upper boundary of the potential bandwidth range.
h.lower The lower boundary of the potential bandwidth range.

Value

The optimal bandwidth

Author(s)

Chao Li <chaoli0394@gmail.com> Shunsuke Managi

References


Examples

```
data(TransAirPolCalif)
data(California)
formula.GWPR <- pm25 ~ co2_mean + Developed_Open_Space_perc + Developed_Low_Intensity_perc +
                 Developed_Medium_Intensity_perc + Developed_High_Intensity_perc +
                 Open_Water_perc + Woody_Wetlands_perc + Emergent_Herbaceous_Wetlands_perc +
                 Deciduous_Forest_perc + Evergreen_Forest_perc + Mixed_Forest_perc +
                 Shrub_perc + Grassland_perc + Pasture_perc + Cultivated_Crops_perc +
                 pop_density + summer_tmmx + winter_tmmx + summer_rmax + winter_rmax

bw.CV.Fix <- bw.GWPR(formula = formula.GWPR, data = TransAirPolCalif,
                      index = c("GEOID", "year"),
                      SDF = California, adaptive = FALSE, p = 2, bigdata = FALSE,
                      effect = "individual", model = "within", approach = "CV",
                      kernel = "bisquare", longlat = FALSE)

bw.CV.Fix

bw.AIC.Fix <- bw.GWPR(formula = formula.GWPR, data = TransAirPolCalif,
                       index = c("GEOID", "year"),
                       SDF = California, adaptive = FALSE, p = 2, bigdata = FALSE,
                       effect = "individual", model = "within", approach = "AIC",
                       kernel = "bisquare", longlat = FALSE, doParallel = FALSE)

bw.AIC.Fix
```
**California**

**California (SpatialPolygonsDataFrame)**

**Description**

The counties’ boundary in California

**Usage**

```r
data(California)
```

**Format**

A `sp::SpatialPolygonsDataFrame` with 'GEOID':

- **GEOID**: a numeric vector, fips IDs of the counties

**Author(s)**

Chao Li <chaoli0394@gmail.com> Shunsuke Managi <managi.s@gmail.com>

**Examples**

```r
## Not run:
data(California)
plot(California)
## End(Not run)
```

---

**GWPR**

**Geographically Weighted Panel Regression Model**

**Description**

This function implements GWPR

**Usage**

```r
GWPR(formula, data, index, SDF, bw = NULL, adaptive = FALSE, p = 2,
     effect = "individual", model = c("pooling", "within", "random"),
     random.method = "swar", kernel = "bisquare", longlat = FALSE)
```
Arguments

formula  The regression formula: \( Y \sim X_1 + \ldots + X_k \)
data  A data.frame for the Panel data
index  A vector of the two indexes: \( \text{c(}"\text{ID}"\text{, }"\text{Time}"\text{)} \)
SDF  Spatial*DataFrame on which is based the data, with the "ID" in the index
bw  The optimal bandwidth, either adaptive or fixed distance
adaptive  If TRUE, adaptive distance bandwidth is used, otherwise, fixed distance bandwidth.
p  The power of the Minkowski distance, default is 2, i.e. the Euclidean distance
effect  The effects introduced in the model, one of "individual" (default), "time", "twoways", or "nested"
model  Panel model transformation: \( \text{c(}"\text{within}"\text{, }"\text{random}"\text{, }"\text{pooling}"\text{)} \)
random.method  Method of estimation for the variance components in the random effects model, one of "swar" (default), "amemiya", "walhus", or "nerlove"
kernel  bisquare: \( \text{wgt} = (1-(\text{vdist}/\text{bw})^2)^2 \) if vdist < bw, \( \text{wgt}=0 \) otherwise (default); gaussian: \( \text{wgt} = \exp(-.5*(\text{vdist}/\text{bw})^2) \); exponential: \( \text{wgt} = \exp(-\text{vdist}/\text{bw}) \); tricube: \( \text{wgt} = (1-(\text{vdist}/\text{bw})^3)^3 \) if vdist < bw, \( \text{wgt}=0 \) otherwise; boxcar: \( \text{wgt}=1 \) if dist < bw, \( \text{wgt}=0 \) otherwise
longlat  If TRUE, great circle distances will be calculated

Value

A list of result:

GW.arguments  a list class object including the model fitting parameters for generating the report
file
R2  global r2
index  the index used in the result, Note: in order to avoid mistakes, we forced a rename of the individuals’ ID as id.
plm.result  an object of class inheriting from plm, see plm
raw.data  the data.frame used in the regression
GWPR.residuals  the data.frame includes Y, Y hat, and residuals from GWPR
SDF  a Spatial*DataFrame (either Points or Polygons, see sp) integrated with fit.points, GWPR coefficient estimates, coefficient standard errors and t-values in its data slot.

Author(s)

Chao Li <chaoli0394@gmail.com> Shunsuke Managi

References

Examples

data(TransAirPolCalif)
data(California)
formula.GWPR <- pm25 ~ co2_mean + Developed_Open_Space_perc + Developed_Low_Intensity_perc +
       Developed_Medium_Intensity_perc + Developed_High_Intensity_perc +
       Open_Water_perc + Woody_Wetlands_perc + Emergent_Herbaceous_Wetlands_perc +
       Deciduous_Forest_perc + Evergreen_Forest_perc + Mixed_Forest_perc +
       Shrub_perc + Grassland_perc + Pasture_perc + Cultivated_Crops_perc +
       pop_density + summer_tmmx + winter_tmmx + summer_rmax + winter_rmax

# precomputed bandwidth
bw.AIC.Fix <- 2.010529

result.F.AIC <- GWPR(bw = bw.AIC.Fix, formula = formula.GWPR, data = TransAirPolCalif,
                     index = c("GEOID", "year"), SDF = California, adaptive = FALSE,
                     p = 2, effect = "individual", model = "within",
                     kernel = "bisquare", longlat = FALSE)

summary(result.F.AIC$SDF$Local_R2)
library(tmap)
  tm_shape(result.F.AIC$SDF) +
  tm_polygons(col = "Local_R2", pal = "Reds",auto.palatte.mapping = FALSE,
               style = 'cont')

GWPR.moran.test

Moran’s I Test for Panel Regression

Description

Moran’s I test for spatial autocorrelation in residuals from an estimated panel linear model (plm).

Usage

GWPR.moran.test(plm_model, SDF, bw, adaptive = FALSE, p = 2,
                 kernel = "bisquare", longlat = FALSE, alternative = "greater")

Arguments

plm_model An object of class inheriting from "plm", see plm
SDF Spatial*DataFrame on which is based the data, with the "ID" in the index
bw The optimal bandwidth, either adaptive or fixed distance
adaptive If TRUE, adaptive distance bandwidth is used, otherwise, fixed distance bandwidth.
p The power of the Minkowski distance, default is 2, i.e. the Euclidean distance
kernel bisquare: wgt = (1-(vdist/bw)^2)^2 if vdist < bw, wgt=0 otherwise (default);
gaussian: wgt = exp(-.5*(vdist/bw)^2); exponential: wgt = exp(-vdist/bw); tricube: wgt = (1-(vdist/bw)^3)^3 if vdist < bw, wgt=0 otherwise; boxcar: wgt=1 if dist < bw, wgt=0 otherwise
GWPR.moran.test

longlat
If TRUE, great circle distances will be calculated

alternative
A character string specifying the alternative hypothesis, must be one of greater (default), less or two.sided.

Value
A list of result:

statistic the value of the standard deviate of Moran’s I.
p.value the p-value of the test.
Estimated.I the value of the observed Moran’s I.
Expected.I the value of the expectation of Moran’s I.
V2 the value of the variance of Moran’s I.
alternative a character string describing the alternative hypothesis.

Note
Current version of panel Moran’s I test can only check the balanced panel data.

Author(s)
Chao Li <chaoli0394@gmail.com> Shunsuke Managi

References

Examples

data(TransAirPolCalif)

formula.GWPR <- pm25 ~ co2_mean + Developed_Open_Space_perc + Developed_Low_Intensity_perc +
Developed_Medium_Intensity_perc + Developed_High_Intensity_perc +
Open_Water_perc + Woody_Wetlands_perc + Emergent_Herbaceous_Wetlands_perc +
Deciduous_Forest_perc + Evergreen_Forest_perc + Mixed_Forest_perc +
Shrub_perc + Grassland_perc + Pasture_perc + Cultivated_Crops_perc +
pop_density + summer_tmmx + winter_tmmx + summer_rmax + winter_rmax

pdata <- plm::pdata.frame(TransAirPolCalif, index = c("GEOID", "year"))
moran.plm.model <- plm::plm(formula = formula.GWPR, data = pdata, model = "within")

summary(moran.plm.model)

# precomputed bandwidth
bw.AIC.Fix <- 2.010529

# Moran's I test
GWPR.moran.test(moran.plm.model, SDF = California, bw = bw.AIC.Fix, kernel = "bisquare",
adaptive = FALSE, p = 2, longlat = FALSE, alternative = "greater")
Locally F Test Based on GWPR

Description

This function performs an F test in each regression based on different subsamples.

Usage

GWPR.pFtest(formula, data, index, SDF, bw = NULL, adaptive = FALSE, p = 2, 
effect = "individual", kernel = "bisquare", longlat = FALSE)

Arguments

- `formula`: The regression formula: \( Y \sim X_1 + \ldots + X_k \)
- `data`: A data.frame for the Panel data.
- `index`: A vector of the two indexes: \((\text{"ID"}, \text{"Time"})\).
- `SDF`: Spatial*DataFrame on which is based the data, with the "ID" in the index.
- `bw`: The optimal bandwidth, either adaptive or fixed distance.
- `adaptive`: If TRUE, adaptive distance bandwidth is used, otherwise, fixed distance bandwidth.
- `p`: The power of the Minkowski distance, default is 2, i.e. the Euclidean distance.
- `effect`: The effects introduced in the fixed effects model, one of "individual" (default), "time", "twoways".
- `kernel`: `bisquare`: \( wgt = (1-(vdist/bw)^2)^2 \) if \( vdist < bw \), \( wgt=0 \) otherwise (default); `gaussian`: \( wgt = \exp(-.5*(vdist/bw)^2) \); `exponential`: \( wgt = \exp(-vdist/bw) \); `tricube`: \( wgt = (1-(vdist/bw)^3)^3 \) if \( vdist < bw \), \( wgt=0 \) otherwise; `boxcar`: \( wgt=1 \) if \( dist < bw \), \( wgt=0 \) otherwise.
- `longlat`: If TRUE, great circle distances will be calculated.

Value

A list of results:

- `GW.arguments`: a list class object including the model fitting parameters for generating the report file.
- `SDF`: a Spatial*DataFrame (either Points or Polygons, see sp) integrated with fit.points, test value, p value, df1, df2.

Author(s)

Chao Li <chaoli0394@gmail.com> Shunsuke Managi
Examples

```r
data(TransAirPolCalif)
data(California)
formula.GWPR <- pm25 ~ co2_mean + Developed_Open_Space_perc + Developed_Low_Intensity_perc + Developed_Medium_Intensity_perc + Developed_High_Intensity_perc + Open_Water_perc + Woody_Wetlands_perc + Emergent_Herbaceous_Wetlands_perc + Deciduous_Forest_perc + Evergreen_Forest_perc + Mixed_Forest_perc + Shrub_perc + Grassland_perc + Pasture_perc + Cultivated_Crops_perc + pop_density + summer_tmmx + winter_tmmx + summer_rmax + winter_rmax

# precomputed bandwidth
bw.AIC.Fix <- 2.010529
GWPR.pFtest.resu.F <- GWPR.pFtest(formula = formula.GWPR, data = TransAirPolCalif, index = c("GEOID", "year"), SDF = California, bw = bw.AIC.Fix, adaptive = FALSE, p = 2, effect = "individual", kernel = "bisquare", longlat = FALSE)
library(tmap)
 tm_shape(GWPR.pFtest.resu.F$SDF) +
   tm_polygons(col = "p.value", breaks = c(0, 0.05, 1))
```

**GWPR.pFtest**

*Locally Hausman Test Based on GWPR*

**Description**

Locally Hausman Test Based on GWPR

**Usage**

```r
GWPR.pFtest(formula, data, index, SDF, bw = NULL, adaptive = FALSE, p = 2, effect = "individual", random.method = "swar", kernel = "bisquare", longlat = FALSE)
```

**Arguments**

- **formula**
  The regression formula: \( Y \sim X_1 + \ldots + X_k \)
- **data**
  A data frame for the Panel data.
- **index**
  A vector of the two indexes: (c("ID", "Time").
- **SDF**
  SpatialDataFrame on which is based the data, with the "ID" in the index.
- **bw**
  The optimal bandwidth, either adaptive or fixed distance.
- **adaptive**
  If TRUE, adaptive distance bandwidth is used, otherwise, fixed distance bandwidth.
- **p**
  The power of the Minkowski distance, default is 2, i.e. the Euclidean distance.
- **effect**
  The effects introduced in the fixed effects model, one of "individual" (default), "time", "twoways"
random.method  Method of estimation for the variance components in the random effects model, one of "swar" (default), "amemiya", "walhus", or "nerlove"

kernel  bisquare: \( \text{wgt} = (1-(v\text{dist}/bw)^2)^2 \) if \( v\text{dist} < bw \), \( \text{wgt}=0 \) otherwise (default); gaussian: \( \text{wgt} = \exp(-.5*(v\text{dist}/bw)^2) \); exponential: \( \text{wgt} = \exp(-v\text{dist}/bw) \); tricube: \( \text{wgt} = (1-(v\text{dist}/bw)^3)^3 \) if \( v\text{dist} < bw \), \( \text{wgt}=0 \) otherwise; boxcar: \( \text{wgt}=1 \) if dist < \( bw \), \( \text{wgt}=0 \) otherwise

longlat  If TRUE, great circle distances will be calculated

Value  
A list of result:

GW.arguments  a list class object including the model fitting parameters for generating the report file

SDF  a Spatial*DataFrame (either Points or Polygons, see sp) integrated with fit.points, test value, p value, df

Note  
If the random method is "swar", to perform this test, bandwidth selection must guarantee that enough individuals in the subsamples. Using bw.GWPR function can avoid mistake.

Author(s)  
Chao Li <chaoli0394@gmail.com> Shunsuke Managi

Examples  

data(TransAirPolCalif)
data(California)

formula.GWPR <- pm25 ~ co2_mean + Developed_Open_Space_perc + Developed_Low_Intensity_perc + Developed_Medium_Intensity_perc + Developed_High_Intensity_perc + Open_Water_perc + Woody_Wetlands_perc + Emergent_Herbaceous_Wetlands_perc + Deciduous_Forest_perc + Evergreen_Forest_perc + Mixed_Forest_perc + Shrub_perc + Grassland_perc + Pasture_perc + Cultivated_Crops_perc + pop_density + summer_tmmx + winter_tmmx + summer_rmax + winter_rmax

# precomputed bandwidth
bw.AIC.Fix <- 7.508404

GWPR.phtest.resu.F <- GWPR.phtest(formula = formula.GWPR, data = TransAirPolCalif, index = c("GEOID", "year"), SDF = California, bw = bw.AIC.Fix, adaptive = FALSE, p = 2, effect = "individual", kernel = "bisquare", longlat = FALSE)

library(tmap)
tm_shape(GWPR.phtest.resu.F$SDF) +
  tm_polygons(col = "p.value", breaks = c(0, 0.05, 1))
Locally Breusch-Pagan Lagrange Multiplier Test Based on GWPR

**Description**
This function performs a Breusch-Pagan Lagrange Multiplier test in each regression based on different subsamples.

**Usage**
```r
GWPR.plmtest(formula, data, index, SDF, bw = NULL, adaptive = FALSE, p = 2,
              kernel = "bisquare", longlat = FALSE)
```

**Arguments**
- `formula`: The regression formula: \( Y \sim X_1 + \ldots + X_k \)
- `data`: A data.frame for the Panel data.
- `index`: A vector for the indexes: \( \{\text{"ID"}, \text{"Time"}\} \).
- `SDF`: Spatial*DataFrame on which is based the data, with the "ID" in the index.
- `bw`: The optimal bandwidth, either adaptive or fixed distance.
- `adaptive`: If TRUE, adaptive distance bandwidth is used, otherwise, fixed distance bandwidth.
- `p`: The power of the Minkowski distance, default is 2, i.e. the Euclidean distance.
- `kernel`: bisquare: \( wgt = (1-(vdist/bw)^2)^2 \) if \( vdist < bw \), \( wgt=0 \) otherwise (default); gaussian: \( wgt = \exp(-.5*(vdist/bw)^2) \); exponential: \( wgt = \exp(-vdist/bw) \); tricube: \( wgt = (1-(vdist/bw)^3)^3 \) if \( vdist < bw \), \( wgt=0 \) otherwise; boxcar: \( wgt=1 \) if \( dist < bw \), \( wgt=0 \) otherwise
- `longlat`: If TRUE, great circle distances will be calculated

**Value**
A list of result:
- `GW.arguments`: a list class object including the model fitting parameters for generating the report
- `SDF`: a Spatial*DataFrame (either Points or Polygons, see sp) integrated with fit.points, test value, p value, df1, df2

**Author(s)**
Chao Li <chaoli0394@gmail.com> Shunsuke Managi
**Examples**

```r
data(TransAirPolCalif)
data(California)
formula.GWPR <- pm25 ~ co2_mean + Developed_Open_Space_perc + Developed_Low_Intensity_perc + Developed_Medium_Intensity_perc + Developed_High_Intensity_perc + Open_Water_perc + Woody_Wetlands_perc + Emergent_Herbaceous_Wetlands_perc + Deciduous_Forest_perc + Evergreen_Forest_perc + Mixed_Forest_perc + Shrub_perc + Grassland_perc + Pasture_perc + Cultivated_Crops_perc + pop_density + summer_tmmx + winter_tmmx + summer_rmax + winter_rmax

# precomputed bandwidth
bw.AIC.Fix <- 2.010529

GWPR.plmtest.resu.F <- GWPR.plmtest(formula = formula.GWPR, data = TransAirPolCalif, index = c("GEOID", "year"), SDF = California, bw = bw.AIC.Fix, adaptive = FALSE, p = 2, kernel = "bisquare", longlat = FALSE)

library(tmap)
tm_shape(GWPR.plmtest.resu.F$SDF) +
  tm_polygons(col = "p.value", breaks = c(0, 0.05, 1))
```

---

**TransAirPolCalif**  
*Panel Dataset for Testing GWPR*

**Description**

Panel dataset to estimate the relationship between county-level PM2.5 concentration and on-road transportation in California.

**Usage**

```r
data(TransAirPolCalif)
```

**Format**

A data.frame with 23 variables, and 928 observations, which are:

- **GEOID**  a numeric vector, fips IDs of the counties
- **year**  a numeric vector, year
- **pm25**  a numeric vector, annually average PM2.5 concentration in the counties
- **co2_mean**  a numeric vector, geographically average CO2 emission from on-road transportation in each year, million tons/km2
- **Developed_Open_Space_perc**  a numeric vector, percentage of developed open space of total area in each county
- **Developed_Low_Intensity_perc**  a numeric vector, percentage of low-intensity developed area of total area in each county
Developed_Medium_Intensity_perc  a numeric vector, percentage of medium-intensity developed area of total area in each county

Developed_High_Intensity_perc  a numeric vector, percentage of high-intensity developed area of total area in each county

Open_Water_perc  a numeric vector, percentage of open water of total area in each county

Woody_Wetlands_perc  a numeric vector, percentage of woody wetland of total area in each county

Emergent_Herbaceous_Wetlands_perc  a numeric vector, percentage of emergent herbaceous wetland of total area in each county

Deciduous_Forest_perc  a numeric vector, percentage of deciduous forest of total area in each county

Evergreen_Forest_perc  a numeric vector, percentage of evergreen forest of total area in each county

Mixed_Forest_perc  a numeric vector, percentage of mixed forest of total area in each county

Shrub_perc  a numeric vector, percentage of shrub of total area in each county

Grassland_perc  a numeric vector, percentage of grassland of total area in each county

Pasture_perc  a numeric vector, percentage of pasture of total area in each county

Cultivated_Crops_perc  a numeric vector, percentage of cultivated crops of total area in each county

pop_density  a numeric vector, average population density in each county

summer_tmmx  a numeric vector, average temperature in summer

winter_tmmx  a numeric vector, average temperature in winter

summer_rmax  a numeric vector, average humidity in summer

winter_rmax  a numeric vector, average humidity in winter

Author(s)

Chao Li <chaoli0394@gmail.com> Shunsuke Managi <managi.s@gmail.com>

Examples

```r
## Not run:
data(TransAirPolCalif)
head(TransAirPolCalif)
## End(Not run)
```
Index

* datasets
  California, 5
  TransAirPolCalif, 13

* package
  GWPR.light-package, 2

bw.GWPR, 3

California, 5

GWPR, 5
GWPR.light (GWPR.light-package), 2
GWPR.light-package, 2
GWPR.moran.test, 7
GWPR.pFtest, 9
GWPR.phtest, 10
GWPR.plmtest, 12

TransAirPolCalif, 13