# Package 'GWPR.light'

October 4, 2021

**Type** Package

**Title** Geographically Weighted Panel Regression (GWPR)

**Version** 0.1.0

**Author** Chao Li [aut, cre], Shunsuke Managi [aut]

**Maintainer** Chao Li <chaoli0394@gmail.com>

**Description** 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. Functions are as follows: bw.GWPR(), GWPR(), GWPR.moran.test(), GWPR.pFtest(), GWPR.phtest(), and GWPR.plmtest(). Data: California and TransAirPolCalif

**License** AGPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

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

**Depends** R (>= 2.10)

**Suggests** rmarkdown,
knitr

**VignetteBuilder** knitr

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

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

# R topics documented:



---

`GWPR.light-package` *A Package for Geographically Weighted Panel Regression (light version)*

---

## Description

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]

bw.GWPR *Bandwidth selection for basic GWPR*

## Description

A function for automatic bandwidth selection to calibrate a GWPR model

## Usage

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

## Arguments

| | |
|---|---|
| formula | The regression formula: : Y ~ X1 + ... + Xk |
| data | data.frame for the Panel data |
| index | A vector for the indexes : (c("ID", "Time")) |
| SDF | Spatial*DataFrame on which is based the data, with the "ID" in the index |
| 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 |
| bigdata | TRUE or FALSE, if the dataset exceeds 40,000, we strongly recommend set it true |
| upperratio | Set the ratio between upper boundary of potential bandwidth range and the forthest distance of SDF, if bigdata = T. (default value: 0.25) |
| effect | The effects introduced in the model, one of "individual" (default) , "time", "twoways", or "nested" |
| model | Panel model transformation: (c("within", "random", "pooling")) |
| random.method | Method of estimation for the variance components in the random effects model, one of "swar" (default), "amemiya", "walhus", or "nerlove" |
| approach | Score used to optimize the bandwidth, c("CV", "AIC") |
| 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 |
| 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 lower boundary of potential bandwidth range. |
| h.lower | The upper boundary of potential bandwidth range. |

## Value

The optimal bandwidth

## Author(s)

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

## References

Fotheringham, A. Stewart, Chris Brunsdon, and Martin Charlton. Geographically weighted regression: the analysis of spatially varying relationships. John Wiley & Sons, 2003.

## Examples

```
## Not run:
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.F <- bw.GWPR(formula = formula.GWPR, data = TransAirPolCalif, index = c("GEOID", "year"),
                SDF = California, adaptive = F, p = 2, bigdata = F, effect = "individual",
                   model = "within", approach = "CV", kernel = "bisquare", longlat = F)

bw.CV.F

bw.AIC.F <- bw.GWPR(formula = formula.GWPR, data = TransAirPolCalif, index = c("GEOID", "year"), SDF = Californ
                    adaptive = F, p = 2, bigdata = F, effect = "individual",
                   model = "within", approach = "AIC", kernel = "bisquare", longlat = F,
                    doParallel = T, cluster.number = 4)
bw.AIC.F

## End(Not run)
```

---

`California` *California (SpatialPolygonsDataFrame)*

---

## Description

The counties' boundary in California

## Usage

```
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

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

## End(Not run)
```

---

GWPR *Geographically Weighted Panel Regression Model*

---

## Description

This function implements GWPR

## Usage

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

## Arguments

| | |
|---|---|
| formula | The regression formula: : Y ~ X1 + ... + Xk |
| data | A data.frame for the Panel data |
| index | A vector for the indexes : (c("ID", "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 model, one of "individual" (default) , "time", "twoways", or "nested" |
| model | Panel model transformation: (c("within", "random", "pooling")) |
| random.method | Method of estimation for the variance components in the random effects model, one of "swar" (default), "amemiya", "walhus", or "nerlove" |
| 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 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 <managi.s@gmail.com>

## References

Fotheringham, A. Stewart, Chris Brunsdon, and Martin Charlton. Geographically weighted regression: the analysis of spatially varying relationships. John Wiley & Sons, 2003.

## Examples

```
## Not run:
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.AIC.F <- bw.GWPR(formula = formula.GWPR, data = TransAirPolCalif, index = c("GEOID", "year"), SDF = Californ
                    adaptive = F, p = 2, bigdata = F, effect = "individual",
                   model = "within", approach = "AIC", kernel = "bisquare", longlat = F,
                    doParallel = T, cluster.number = 4)

result.F.AIC <- GWPR(bw = bw.AIC.F, formula = formula.GWPR, data = TransAirPolCalif, index = c("GEOID", "year")
              SDF = California, adaptive = F, p = 2, effect = "individual", model = "within",
                     kernel = "bisquare", longlat = F)
summary(result.F.AIC$SDF$Local_R2)
library(tmap)
tm_shape(result.F.AIC$SDF) +
tm_polygons(col = "Local_R2", pal = "Reds",auto.palette.mapping = F,
            style = 'cont')

## End(Not run)
```

`GWPR.moran.test` *Moran's I Test for Panel Regression*

## Description

Moran's I Test for Panel Regression

## Usage

```
GWPR.moran.test(
  plm_model,
  SDF,
  bw,
  adaptive = F,
  p = 2,
  kernel = "bisquare",
  longlat = F,
  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 |
| `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.

**Excepted.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 chech the balanced panel data.

## Author(s)

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

## References

Beenstock, M., Felsenstein, D., 2019. The econometric analysis of non-stationary spatial panel data. Springer.

## Examples

```
## Not run:
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

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)

bw.AIC.F <- bw.GWPR(formula = formula.GWPR, data = TransAirPolCalif, index = c("GEOID", "year"), SDF = Californ
                    adaptive = F, p = 2, bigdata = F, effect = "individual",
                   model = "within", approach = "AIC", kernel = "bisquare", longlat = F,
                    doParallel = T, cluster.number = 4)

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

## End(Not run)
```

---

GWPR.pFtest *Locally F Test based on GWPR*

---

## Description

This function perform F test in each regression based on different subsamples

## Usage

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

## Arguments

| | |
|---|---|
| formula | The regression formula: : Y ~ X1 + ... + Xk |
| data | A data.frame for the Panel data. |
| index | A vector for the indexes : (c("ID", "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 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, df1, df2

## Author(s)

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

## Examples

```
## Not run:
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.AIC.F <- bw.GWPR(formula = formula.GWPR, data = TransAirPolCalif, index = c("GEOID", "year"), SDF = Californ
                    adaptive = F, p = 2, bigdata = F, effect = "individual",
                   model = "within", approach = "AIC", kernel = "bisquare", longlat = F,
                    doParallel = T, cluster.number = 4)
GWPR.pFtest.resu.F <- GWPR.pFtest(formula = formula.GWPR, data = TransAirPolCalif, index = c("GEOID", "year"),
                       SDF = California, bw = bw.AIC.F, adaptive = F, p = 2, effect = "individual",
                                  kernel = "bisquare", longlat = F)
library(tmap)
tm_shape(GWPR.pFtest.resu.F$SDF) +
```

```
    tm_polygons(col = "p.value", breaks = c(0, 0.05, 1))

## End(Not run)
```

---

`GWPR.phtest` *Locally Hausman Test based on GWPR*

---

## Description

Locally Hausman Test based on GWPR

## Usage

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

## Arguments

| | |
|---|---|
| `formula` | The regression formula: : Y ~ X1 + ... + Xk |
| `data` | A data.frame for the Panel data. |
| `index` | A vector for the indexes : (c("ID", "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" |
| `random.method` | Method of estimation for the variance components in the random effects model, one of "swar" (default), "amemiya", "walhus", or "nerlove" |
| `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 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 <managi.s@gmail.com>

## Examples

```
## Not run:
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.AIC.F <- bw.GWPR(formula = formula.GWPR, data = TransAirPolCalif, index = c("GEOID", "year"), SDF = Californ
                    adaptive = F, p = 2, bigdata = F, effect = "individual",
                   model = "random", approach = "AIC", kernel = "bisquare", longlat = F,
                    doParallel = T, cluster.number = 4)
GWPR.phtest.resu.F <- GWPR.phtest(formula = formula.GWPR, data = TransAirPolCalif, index = c("GEOID", "year"),
                       SDF = California, bw = bw.AIC.F, adaptive = F, p = 2, effect = "individual",
                                  kernel = "bisquare", longlat = F)
library(tmap)
tm_shape(GWPR.phtest.resu.F$SDF) +
     tm_polygons(col = "p.value", breaks = c(0, 0.05, 1))

## End(Not run)
```

---

# GWPR.plmtest *Locally Breusch-Pagan Lagrange Multiplier Test Based on GWPR*

---

## Description

This function perform Breusch-Pagan Lagrange Multiplier test in each regression based on different subsamples

## Usage

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

## Arguments

| | |
|---|---|
| formula | The regression formula: : Y ~ X1 + ... + Xk |
| data | A data.frame for the Panel data. |
| index | A vector for the indexes : (c("ID", "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 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 <managi.s@gmail.com>

## Examples

```
## Not run:
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.AIC.F <- bw.GWPR(formula = formula.GWPR, data = TransAirPolCalif, index = c("GEOID", "year"), SDF = Californ
                    adaptive = F, p = 2, bigdata = F, effect = "individual",
                    model = "within", approach = "AIC", kernel = "bisquare", longlat = F,
                    doParallel = T, cluster.number = 4)
GWPR.plmtest.resu.F <- GWPR.plmtest(formula = formula.GWPR, data = TransAirPolCalif, index = c("GEOID", "year"
                                    SDF = California, bw = bw.AIC.F, adaptive = F, p = 2,
                                    kernel = "bisquare", longlat = F)
library(tmap)
tm_shape(GWPR.plmtest.resu.F$SDF) +
     tm_polygons(col = "p.value", breaks = c(0, 0.05, 1))

## End(Not run)
```

`TransAirPolCalif` *Panel Dataset for Testing GWPR*

## Description

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

## Usage

```
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 develope 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

```
## Not run:
data(TransAirPolCalif)
head(TransAirPolCalif)

## End(Not run)
```

# Index