Package ‘TexMix’

March 1, 2020

Type Package

Title Supporting Functions and Data for Geo-Spatial Analytics Courses at UTDallas, Texas

Version 0.5.3

Maintainer Michael Tiefelsdorf <tiefelsdorf@utdallas.edu>

Description A collection of functions and data - mostly from Texas - is provided. These are used as teaching tools for geo-spatial data analytics courses in the GISciences program at The University of Texas at Dallas. In addition, several vignettes illustrate geo-spatial data analytics practices, such as relative risk kernel density estimations based on food store locations within Dallas County or the identification of homogenous and spatially contiguous market areas built on socio-economic, demographic and infrastructure census information. The spatial resolution of the data-sets ranges from 1623 food store locations, over geo-referenced areal data of 258 Texan counties, to 529 census tracts as well as 1669 block groups in Dallas County. Cartographic, specialized regression and data handling functions are provided.

Depends R (>= 3.5.0)

Imports sp, maptools, RColorBrewer, classInt, Formula

Suggests spatstat, smacpod, colorspace, car, effects, rgdal, spdep, DMwR, ClustGeo, knitr, rmarkdown, testthat

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

VignetteBuilder knitr

NeedsCompilation no

Author Michael Tiefelsdorf [aut, cre],
Yongwan Chun [ctb],
Laila Al Aswad [aut]

Repository CRAN

Date/Publication 2020-03-01 12:40:03 UTC

R topics documented:

`blockGroupShp` *Polygon layer of Dallas County, TX, census block groups*

Description

Block groups in Dallas County, Texas, in the longitude and latitude format (see `proj4string=CRS("+proj=longlat +ellps=WGS84")`).

Note: Several fields have missing observations and several variables are reported as absolute numbers.

Format

Spatial polygon data-frame with 1669 block groups. The variables are as follows:

ID Internal ID.

BLOCKGROUP Factor with the Census Bureau's block group number.

TRACT Factor with the Census Bureau's tract numbers.

AREA Calculated area of the block group in square miles.

LANDAREA Land area of the block group in square miles.

WATERAREA Water area of the block group in square miles.

DAYPOP Caliper's estimate of the absolute day time population 16 yrs and older.

NIGHTPOP Census's night time population counts in a block group.

MEDAGE Median population age in a block group.

WHITE Absolute number of white population in a block group.

PCTBLACK Absolute number of black population in a block group.

PCTASIAN Absolute number of asian population in a block group.

PCTHISPAN Absolute number of hispanic population in a block group.

PCTHIGH % of population 25+ with a high school degree.

PCTUNIVDEG % of population 25+ with a univeristy degrees.

TIME2WORK Average travel time to work. **Attention**: 1421 NA's.

PERCAPINC Per capita income in $.

HOUSEUNITS Absolute number of housing units

PCTOWNVAC % of vacant owner occupied housing units.

PCTRENTVAC % of vacant rental housing units.

Source

Based on 2018 ACS data, which were retrieved from Maptitude (https://www.caliper.com/).

Examples

```
data(blockGroupShp)
sp::summary(blockGroupShp)
```

---

bndShp *Boundary of Dallas County, TX*

Description

One polygon in the longitude and latitude format (see `proj4string=CRS("+proj=longlat +ellps=WGS84")`).

Examples

```
library(maptools)
validTractShp <- tractShp[!is.na(tractShp$BUYPOW), ]  # Remove 2 tracts with NA's
plot(tractShp, col="white", border="white", axes=TRUE,
     main="Dallas Census Tracts with Food Deserts")
plot(validTractShp, col="ivory2", border="white", add=TRUE)
plot(lakesShp, col="skyblue", border="skyblue",add=TRUE)
plot(hwyShp, col="cornsilk3", lwd=3, add=TRUE)
plot(foodDesertShp, border="magenta",lwd=2, add=TRUE)
plot(bndShp, border="black", add=TRUE)
box()
```

foodDesertShp *Boundaries of three Neighorhoods in the City of Dallas, TX*

Description

Three polygons of putative food deserts in the longitude and latitude format (see `proj4string=CRS("+proj=longlat +ellps=WGS84")`).

Format

Spatial polygon data-frame with 3 areas and the following variables:

ID Internal polygon ID

AREA Neighborhood area in square miles

DESERT Factor witht he name of the three neighborhoods

Source

Dallas Morning News Nov 16, 2016 (see https://www.dallasnews.com/opinion/editorials/2016/11/16/will-southern-dallas-food-deserts-get-relief)

Examples

```
library(maptools)
validTractShp <- tractShp[!is.na(tractShp$BUYPOW), ]  # Remove 2 tracts with NA's
plot(tractShp, col="white", border="white", axes=TRUE,
     main="Dallas Census Tracts with Food Deserts")
plot(validTractShp, col="ivory2", border="white", add=TRUE)
plot(lakesShp, col="skyblue", border="skyblue",add=TRUE)
plot(hwyShp, col="cornsilk3", lwd=3, add=TRUE)
plot(foodDesertShp, border="magenta",lwd=2, add=TRUE)
plot(bndShp, border="black", add=TRUE)
box()
```

foodStoresShp *Point layer of Stores selling Food in Dallas County, TX*

Description

Location of food stores in Dallas County, TX, in the longitude and latitude format (see `proj4string=CRS("+proj=longlat +ellps=WGS84")`)..

Format

Spatial polygon data-frame with 1623 verified store locations.

SALESVOL Reported total annual sales volume of goods in $

PROPFOOD Assumed proportion of food sales

FOODSALES Calculated annual sales volume of food in $

STORETYPE Factor distinguishing between stores selling nutritious food (grocery stores) and processed food (convenience stores)

Source

Reference USA, 2019 http://www.referenceusa.com.

Examples

```
library(spatstat)
library(rgdal)
library(sp)
proj4string(bndShp)                                      # Current system
projUTM <- CRS("+proj=utm +zone=14  +units=m")           # isotropic coordinate sytem
bndUTM <- spTransform(bndShp, projUTM)                   # Re-project boundary
storesUTM <- spTransform(foodStoresShp, projUTM)         # Re-project points
storesDf <- as.data.frame(storesUTM)                     # Extract data-frame
storesPts <- as.ppp(storesUTM)                           # Convert to .ppp
storesPts$marks <- NULL                                  # Clear marks
bndWin <- as.mask(as.owin(bndUTM), eps=200)            # pixel window with 200 m resolution
unitname(bndWin) <- list("meter","meters")               # set units
storesPts <- storesPts[bndWin]                           # assign window to pts
summary(storesPts)

## Evaluate weighted kernel density with bw=3000
allFoodIm <- density(storesPts, weights=storesDf$FOODSALES, sigma=3000)
plot(allFoodIm, main="All Stores Weighted Kernel Density\nbw = 3000 m")
plot(storesPts, cex=0.5, pch=16, col="green", add=TRUE)
box(); axis(1); axis(2)
```

---

hwyShp *Major Highways in Dallas County, TX*

Description

Basic line layer in the longitude and latitude format (see `proj4string=CRS("+proj=longlat +ellps=WGS84")`).

Examples

```
library(maptools)
validTractShp <- tractShp[!is.na(tractShp$BUYPOW), ]  # Remove 2 tracts with NA's
plot(tractShp, col="white", border="white", axes=TRUE,
     main="Dallas Census Tracts with Food Deserts")
plot(validTractShp, col="ivory2", border="white", add=TRUE)
plot(lakesShp, col="skyblue", border="skyblue",add=TRUE)
plot(hwyShp, col="cornsilk3", lwd=3, add=TRUE)
plot(foodDesertShp, border="magenta",lwd=2, add=TRUE)
plot(bndShp, border="black", add=TRUE)
box()
```

`lakesShp` *Lakes in Dallas County, TX*

Description

Polygon layer in the longitude and latitude format (see `proj4string=CRS("+proj=longlat +ellps=WGS84")`).

Examples

```
library(maptools)
validTractShp <- tractShp[!is.na(tractShp$BUYPOW), ]  # Remove 2 tracts with NA's
plot(tractShp, col="white", border="white", axes=TRUE,
     main="Dallas Census Tracts with Food Deserts")
plot(validTractShp, col="ivory2", border="white", add=TRUE)
plot(lakesShp, col="skyblue", border="skyblue",add=TRUE)
plot(hwyShp, col="cornsilk3", lwd=3, add=TRUE)
plot(foodDesertShp, border="magenta",lwd=2, add=TRUE)
plot(bndShp, border="black", add=TRUE)
box()
```

`lmHetero` *Function: Multiplicately weighted regression model*

Description

`lmHetero` accounts for heteroscedasticity in regression models

Usage

```
lmHetero(
  formula,
  hetero = ~1,
  data,
  subset,
  na.action,
  contrasts = NULL,
  iter = FALSE,
  ...
)
```

Arguments

`formula`	Formula object (perhaps, multiple parts formula `y~x1+x2+...| log(z1)+log(z2)+...`) linking the dependent variable to its set of associated independent variables x and a second expression separated by `|` modelling the local variances with the variables z. The formula `y~x1+x2+...` assumes homoscedasticity. **Important:** the weights variables z must be entered in their log-transformed form! This is the preferred specification of the function call.
`hetero`	Optional formula specification without "`| log(z1)+log(z2)+...`": A second parameter modeling the heteroscedasticity with a right-handed formula by a set of variables z. That is, "`hetero=~log(z1)+log(z2)+...`". Omitting the second parameter assumes "`hetero=~1`". This is only included for backward compatibility).
`data`	An optional data frame containing the variables in the model. By default the variables are taken from the environment of the formula.
`subset`	An optional vector specifying a subset of observations to be used in fitting the model.
`na.action`	A function that indicates what should happen when the data contain `NA`s. The default is set by the "`na.action`" option
`contrasts`	An optional list. See the "`contrasts.arg`" of `model.matrix.default`.
`iter`	Logical indicating whether the interation history should be displayed. The default setting if "`FALSE`".
`...`	Currently not in use.

Details

This function estimates the parameters of a regression model whose normally distributed disturbances have a variance that multiplicatively depends on a set of strictly positive weights variables. That is,

$$\sigma_i^2 = \exp(\gamma_0 + \gamma_1 \cdot \log(z_{i1}) + ...)$$

The weights variables z must be entered in their logarithmic forms. The paramater $\exp(\gamma_0)$ expresses the global variance.

Value

a list with 10 elements:

`CALL`	function call
`sigma2`	global variance estimate $exp(gamma_0)$
`gamma`	vector of estimated gamma coefficients
`namesGamma`	vector of variable names expressed by Z
`beta`	vector of estimated weight adjusted regression parameters
`weights`	vector of weights $1/\sigma_i^2$ estimates for each observation. It can be used in the call `lm(...,weights=weights)` to adjust for heteroscedasticity
`covBeta`	covariance matrix of the estimated regression coefficients
`covGamma`	covariance matrix of the estimated gamma coefficients
`logLikeH1`	log-likelihood of the heteroscedastic adjusted regression model
`logLikeH0`	log-likelihood of the unadjusted regression model

Author(s)

Michael Tiefelsdorf (<tiefelsdorf@utdallas.edu>) & Yongwan Chun

Source

The maximum likelihood estimation procedure for multiplicately weighted regression is given in Greene W. H. (2000). Econometric Analysis. 4th edition. Upper Saddle River: Prentice Hall. pp 516-521 (Note: page numbers will differ for other editions)

Examples

```
library(sp)
data(tractShp)
validTractShp <- tractShp[!is.na(tractShp$BUYPOW), ]         # Remove 2 tracts with NA's
## Population at risk
totPop <- validTractShp$MALE+validTractShp$FEMALE

## H0 model (homoscedasticity)
mod.lm <- mod.lmH <- lmHetero(PERCAPINC~PCTNOHINS+PCTMINOR+PCTUNIVDEG+PCTWHITE,
                              data=validTractShp)
summary(mod.lm)

## Preferred heteroscedasticiy function call
mod.lmH <- lmHetero(PERCAPINC~PCTNOHINS+PCTMINOR+PCTUNIVDEG+PCTWHITE|log(totPop),
                    data=validTractShp)
summary(mod.lmH)

## Alternative equivalent heteroscedasticiy function call
mod.lmH <- lmHetero(PERCAPINC~PCTNOHINS+PCTMINOR+PCTUNIVDEG+PCTWHITE, hetero=~log(totPop),
                    data=validTractShp)
summary(mod.lmH)
```

```
## Use estimated weights as input for weighted lm-model.
## This also to perform further model diagnostics.
mod.lmW <- lm(PERCAPINC~PCTNOHINS+PCTMINOR+PCTUNIVDEG+PCTWHITE, weights=mod.lmH$weights,
              data=validTractShp)
summary(mod.lmW)
hist(weighted.residuals(mod.lmW))
```

---

mapBiPolar *Function: Map a bipolar theme broken around a neutral value*

Description

`mapBiPolar` generates a map of a bipolar theme variable

Usage

```
mapBiPolar(var.name,shape, break.value=0, neg.breaks=4, pos.breaks=neg.breaks,
                  map.title="", legend.title=deparse(substitute(var.name)),
                  legend.pos="bottomleft", legend.cex=1, add.to.map=FALSE)
```

Arguments

`var.name`	A variable to be mapped in a bipolar theme. If it is in a data-frame, then the data-frame must be refered to, e.g., `df$var`
`shape`	An existing spatial polygon or spatial polygon data-frame
`break.value`	Neutral value separating the negative branch from the positive branch of the variable
`neg.breaks`	Number of classes in the negative branch (default=4)
`pos.breaks`	Number of classes in the positive branch (default=`neg.breaks`)
`map.title`	Character string with map title
`legend.title`	Character string with legend title (default=`var.name`)
`legend.pos`	Location of legend in the map frame (default=`"bottomleft"`)
`legend.cex`	Relative font size of the legend
`add.to.map`	Logical to start a new map frame if `FALSE` or overlay onto an existing map frame if `TRUE`

Details

The function `mapBiPolar` maps a *bipolar variable* with a *divergent color ramp* around a specific break values. A legend is generated. Below values are coded blue and above values red. Each branch is broken into 'quantiles'. Therefore, the number of class in each branch should be proportional to the number of observations in each class. NA's are permitted.

Value

NULL

Author(s)

Michael Tiefelsdorf <tiefelsdorf@utdallas.edu>

Examples

```
library(maptools)
validTractShp <- tractShp[!is.na(tractShp$BUYPOW), ]         # Remove 2 tracts with NA's
mapColorQual(validTractShp$CITYPERI, validTractShp,
             map.title="Cities and Peripherie in Dallas County",
             legend.title="Regions")

mapColorRamp(validTractShp$bad1500D, validTractShp, breaks=9,
             map.title="Density of Convenience Stores in Dallas County\nbw=1500 meters",
             legend.title="Junk Food")

hist(tractShp$LRRmedD)
mapBiPolar(validTractShp$LRRmedD, validTractShp, break.value=0,
           neg.breaks=5, pos.breaks=5,
           map.title="LRR: log(f(junk food),f(healthy food))\nbw=medium",
           legend.title="log relative risk")
```

`mapColorQual` *Function: Maps a qualitative theme for a maximum of 12 categories*

Description

`mapColorQual` generates a map of a qualitative variable

Usage

```
mapColorQual(var.name, shape, map.title="", legend.title=deparse(substitute(var.name)),
                       legend.pos="bottomleft", legend.cex=1, add.to.map=FALSE)
```

Arguments

`var.name`	A factor, perhaps with NA's, to be mapped with a maximum of 12 categories. If the factor is in a data-frame, then the data-frame must be explicitly referred to, e.g., `df$var`
`shape`	An existing spatial polygon or spatial polygon data-frame
`map.title`	Character string with map title
`legend.title`	Character string with legend title (default=`var.name`)
`legend.pos`	Location of legend in the map frame (default=`"bottomleft"`)

`legend.cex` Relative font size of the legend

`add.to.map` Logical to start a new map frame if `FALSE` or overlay onto an existing map frame if `TRUE`

Details

The function `mapColorQual` maps a *categorical variable* with a *set of distinct colors*. A legend is generated. The maximum number of valid factor levels should not exceed 12. NA's are permitted.

Value

`NULL`

Author(s)

Michael Tiefelsdorf <tiefelsdorf@utdallas.edu>

Examples

```
library(maptools)
validTractShp <- tractShp[!is.na(tractShp$BUYPOW), ]         # Remove 2 tracts with NA's
mapColorQual(validTractShp$CITYPERI, validTractShp,
             map.title="Cities and Peripherie in Dallas County",
             legend.title="Regions")

mapColorRamp(validTractShp$bad1500D, validTractShp, breaks=9,
             map.title="Density of Convenience Stores in Dallas County\nbw=1500 meters",
             legend.title="Junk Food")

hist(tractShp$LRRmedD)
mapBiPolar(validTractShp$LRRmedD, validTractShp, break.value=0,
           neg.breaks=5, pos.breaks=5,
           map.title="LRR: log(f(junk food),f(healthy food))\nbw=medium",
           legend.title="log relative risk")
```

`mapColorRamp` *Function: Maps a sequential color theme*

Description

`mapColorRamp` generates a map with a sequential theme of an interval scaled variable

Usage

```
mapColorRamp(var.name, shape, breaks=8, map.title="",
                      legend.title=deparse(substitute(var.name)),
                      legend.pos="bottomleft", legend.cex=1, add.to.map=FALSE)
```

Arguments

- `var.name` A variable to be mapped in a bipolar theme. If it is in a data-frame, then the data-frame must be refered to, e.g., `df$var`
- `shape` An existing spatial polygon or spatial polygon data-frame
- `breaks` Number of qunatiles. It needs to range the range of 3 to 9
- `map.title` Character string with map title
- `legend.title` Character string with legend title (default=`var.name`)
- `legend.pos` Location of legend in the map frame (default=`"bottomleft"`)
- `legend.cex` Relative font size of the legend
- `add.to.map` Logical to start a new map frame if `FALSE` or overlay onto an existing map frame if `TRUE`

Details

The function `mapColorRamp` maps an *interval scaled variable* by a *sequetial color ramp*. Quantiles values are coded in gradually increasing intensities of oranges. A legend is generated. NA's are permitted.

Value

`NULL`

Author(s)

Michael Tiefelsdorf <tiefelsdorf@utdallas.edu>

Examples

```
library(maptools)
validTractShp <- tractShp[!is.na(tractShp$BUYPOW), ]         # Remove 2 tracts with NA's
mapColorQual(validTractShp$CITYPERI, validTractShp,
             map.title="Cities and Peripherie in Dallas County",
             legend.title="Regions")

mapColorRamp(validTractShp$bad1500D, validTractShp, breaks=9,
             map.title="Density of Convenience Stores in Dallas County\nbw=1500 meters",
             legend.title="Junk Food")

hist(tractShp$LRRmedD)
mapBiPolar(validTractShp$LRRmedD, validTractShp, break.value=0,
           neg.breaks=5, pos.breaks=5,
           map.title="LRR: log(f(junk food),f(healthy food))\nbw=medium",
           legend.title="log relative risk")
```

plotBoxesByFactor — *Layout of box-plots: variables by factor*

Description

`plotBoxesByFactor` generates box-plots of several variable by a factor.

Usage

```
plotBoxesByFactor(xVars, groups, ncol=3, zTrans=TRUE, varwidth=FALSE)
```

Arguments

`xVars`	data-frame of metric variables.
`groups`	factor variable.
`ncol`	number of layout columnes.
`zTrans`	logical. `xTrans=TRUE` performs a z-transformation of all variables.
`varwidth`	logical. `varwidth=TRUE` makes the box width proportional to the number of observation used to generate the box for a specific factor level.

Details

This function organizes several box-plots of metric variables broken by a factor variable in to a layout. By default 3 box-plots are shown per row and the variables are standardized by the z-transformation. As the number of factor levels increases the number of plots per row should be decreased to maintain a reasonable visual resolution.

Author(s)

Michael Tiefelsdorf <tiefelsdorf@utdallas.edu>

Examples

```
varsKeep <- c("PCTWHITE","PCTBLACK","PCTASIAN","PCTHISPAN","MEDAGE","MEDVALHOME")
myData <- tractShp@data
plotBoxesByFactor(myData[,varsKeep], tractShp$CITYPERI, ncol=2, zTrans=TRUE, varwidth=FALSE)
```

`summary.lmHetero` *Function: Summerize lmHetero model estimates*

Description

`summary.lmHetero` provides relevant summary information of a heteroscedastic regression model estimated by `lmHetero`.

Usage

```
## S3 method for class 'lmHetero'
summary(object, ...)
```

Arguments

`object`	An object of S3-class `lmHetero`.
`...`	Currently not in use.

Details

Regression tables for the estimate regression coefficients and the coefficients of the weights model are provided as well as a maximum likelihood ratio test is performed against a model, which assumes homoscedasticity.

Value

the input 'object' is returned silently.

Author(s)

Michael Tiefelsdorf `<tiefelsdorf@utdallas.edu>`

Examples

```
library(sp)
data(tractShp)
validTractShp <- tractShp[!is.na(tractShp$BUYPOW), ]        # Remove 2 tracts with NA's
## Population at risk
totPop <- validTractShp$MALE+validTractShp$FEMALE

## H0 model
mod.lm <- mod.lmH <- lmHetero(PERCAPINC~PCTNOHINS+PCTMINOR+PCTUNIVDEG+PCTWHITE,
                              data=validTractShp)
summary(mod.lm)

## Preferred function call
mod.lmH <- lmHetero(PERCAPINC~PCTNOHINS+PCTMINOR+PCTUNIVDEG+PCTWHITE|log(totPop),
                    data=validTractShp)
summary(mod.lmH)
```

```
## Alternative function call
mod.lmH <- lmHetero(PERCAPINC~PCTNOHINS+PCTMINOR+PCTUNIVDEG+PCTWHITE, hetero=~log(totPop),
                    data=validTractShp)
summary(mod.lmH)

## Use estimated weights as input for weighted lm
mod.lmW <- lm(PERCAPINC~PCTNOHINS+PCTMINOR+PCTUNIVDEG+PCTWHITE, weights=mod.lmH$weights,
              data=validTractShp)
summary(mod.lmW)
hist(weighted.residuals(mod.lmW))
```

`tractShp` *Polygon layer of Dallas County, TX, census tracts*

Description

Census tracts in Dallas County, Texas, in the longitude and latitude format (see `proj4string=CRS("+proj=longlat +ellps=WGS84")`).

Attention: Two census tracts do not have a night time population and therefore NA's in most of their variables. These tracts are the airports of Love Field and DFW (see `tractShp$TRACT=="48113980000" | tractShp$TRACT=="48113980100"`).

Format

Spatial polygon data-frame with 529 census tracts. The variables are as follows:

ID Internal ID.

SeqId Sequence ID of tracts from 1 to 529.

TRACT Factor with the Census Bureau's tract numbers.

AREA Calculated area of the census tract in square miles. **Proportional to the denominator of the average store weighted kernel density.**

LANDAREA Land area of the census tract in square miles.

WATERAREA Water area of the census tract in square miles.

nCells Number of 100x100 meters raster cells within each census tract. **Denominator of the average store weighted kernel density in each census tract.**

all1500D Average weighted kernel density of all stores with a bandwith of 1500 meters. Weights are based on store's food sales volume.

all2250D Average weighted kernel density of all stores with a bandwith of 2250 meters. Weights are based on store's food sales volume.

all3000D Average weighted kernel density of all stores with a bandwith of 3000 meters. Weights are based on store's food sales volume.

good2000D Average weighted kernel density of grocery stores with a bandwith of 2000 meters. Weights are based on store's food sales volume.

good3000D Average weighted kernel density of grocery stores with a bandwith of 3000 meters. Weights are based on store's food sales volume.

good4000D Average weighted kernel density of grocery stores with a bandwith of 4000 meters. Weights are based on store's food sales volume.

bad1000D Average weighted kernel density of convenience stores with a bandwith of 1000 meters. Weights are based on store's food sales volume.

bad1500D Average weighted kernel density of convenience stores with a bandwith of 1500 meters. Weights are based on store's food sales volume.

bad2000D Average weighted kernel density of convenience stores with a bandwith of 2000 meters. Weights are based on store's food sales volume.

LRRlowD Average low Log-relative risk $log(bad1000D/good2000D)$.

LRRmedD Average medium Log-relative risk $log(bad1500D/good3000D)$.

LRRhiD Average high Log-relative risk $log(bad2000D/good4000D)$.

DES3NEIG **Factor** distinguishing the three putative food desert neighborhood against the remaining census tracts.

FOODDES **Factor** distinguishing the 14 putative food desert census tracts against the remaining 515 census tracts.

CITYPERI **Factor** distinguishing the census tracts in the city of Dallas, the Park cities, North, East, South and West census tracts.

DAYPOP Caliper's estimate of the absolute day time population.

NIGHTPOP Census's night time population counts in a census tract.

PCTDAYPOP % day time population: $DAYPOP/(DAYPOP + NIGHTPOP)$.

POPDEN Population density: $(0.4*DAYPOP+0.6*NIGHTPOP)/LANDAREA$. **Relative measure for potential food demand.**

BUYPOW Absolute buying power in a census tract in \$. Source: IRS 2016 records. **Absolute measure for potential food demand.**

MALE Census's male population count in a census tract.

FEMALE Census's female population count in a census tract.

MEDAGE Median population age in a census tract.

PCTWHITE % white population in a census tract.

PCTBLACK % black population in a census tract.

PCTASIAN % asian population in a census tract.

PCTHISPAN % hispanic population in a census tract.

PCTMINOR % of population belonging to a minority.

PCTBADENG % of population, which does not speak English well.

PCTNOHIGH % of population 25+ without a high school degree.

PCTUNIVDEG % of population 25+ with a univeristy degrees.

PCTNOVEH % of households not owning a car.

PCTPUB2WRK % of employed population taking public transportation to work.

TIME2WORK Average travel time to work. **Attention**: 110 NA's.

PCTNOHINS % of civilian population without health insurance.

PCTUNEMP % of population in the labor force, which is unemployed.

PCTFAMPOV % of family below the poverty threshold.

PCTPOPPOV % of population below the poverty threshold.

HHMEDINC Median household income in $.

MEDFAMINC Median family income in $.

PERCAPINC Per capita income in $.

PCTHUVAC % of vacant housing units.

MEDVALHOME Median home value. **Attention**: 27 NA's.

PCTB2010 % of homes built from 2010 to 2018.

PCTB2000 % of homes built from 2000 to 2009.

PCTB1990 % of homes built from 1990 to 1999.

PCTB1980 % of homes built from 1980 to 1989.

PCTB1970 % of homes built from 1970 to 1979.

PCTB1960 % of homes built from 1960 to 1969.

PCTB1950 % of homes built from 1950 to 1959.

PCTB1940 % of homes built from 1940 to 1949.

PCTBPRE % of homes built before 1940.

Source

Based on 2018 ACS data, which were retrieved from Maptitude (https://www.caliper.com/). The store density statistics by census tract were calculated by the authors.

Examples

```
library(maptools)
validTractShp <- tractShp[!is.na(tractShp$BUYPOW), ]         # Remove 2 tracts with NA's
mapColorQual(validTractShp$CITYPERI, validTractShp,
             map.title="Cities and Peripherie in Dallas County",
             legend.title="Regions")

mapColorRamp(validTractShp$bad1500D, validTractShp, breaks=9,
             map.title="Density of Convenience Stores in Dallas County\nbw=1500 meters",
             legend.title="Junk Food")

hist(tractShp$LRRmedD)
mapBiPolar(validTractShp$LRRmedD, validTractShp, break.value=0,
           neg.breaks=5, pos.breaks=5,
           map.title="LRR: log(f(junk food),f(healthy food))\nbw=medium",
           legend.title="log relative risk")
```

Index