Package 'micromapST'

April 30, 2022

Type Package

Version 1.1.3

Date 2022-04-29

Title Linked Micromap Plots for U. S. and Other Geographic Areas

Author Jim Pearson [aut, cre, cph],
Dan Carr [aut, cph],
Linda Pickle [ctb, cph]

Maintainer Jim Pearson <jbpearson353@gmail.com>

Imports graphics, grDevices, labeling, RColorBrewer, stats, stringr,
utils

Depends R (>= 3.2.0)

Suggests R.rsp

VignetteBuilder R.rsp

Description Provides the users with the ability to quickly create
linked micromap plots for a collection of geographic areas.
Linked micromap plots are visualizations of geo-referenced data
that link statistical graphics to an organized series of small
maps or graphic images. The Help description contains examples
of how to use the 'micromapST' function. Contained in this
package are border group datasets to support creating linked
micromap plots for the 50 U.S. states and District of Columbia
(51 areas), the U. S. 20 Seer Registries, the 105 counties in
the state of Kansas, the 62 counties of New York, the 24
counties of Maryland, the 29 counties of Utah, the 32
administrative areas in China, the 218 administrative areas in
the UK and Ireland (for testing only), the 25 districts in the
city of Seoul South Korea, and the 52 counties on the Africa
continent.
A border group dataset contains the boundaries related to the
data level areas, a second layer boundaries, a top or third
layer boundary, a parameter list of run options, and a cross
indexing table between area names, abbreviations, numeric
identification and alias matching strings for the specific

1

geographic area. By specifying a border group, the package
create linked micromap plots for any geographic region. The
user can create and provide their own border group dataset for
any area beyond the areas contained within the package.
References: Carr and Pickle, Chapman and Hall/CRC, Visualizing
Data Patterns with Micromaps, CRC Press, 2010. Pickle, Pearson,
and Carr (2015), micromapST: Exploring and Communicating
Geospatial Patterns in US State Data., Journal of Statistical
Software, 63(3), 1-25., <https://www.jstatsoft.org/v63/i03/>.
Copyrighted 2013, 2014, 2015, 2016, and 2022 by Carr, Pearson
and Pickle.

License GPL (>= 2)

LazyData no

LazyLoad no

BuildResaveData best

ByteCompile yes

NeedsCompilation no

Repository CRAN

Date/Publication 2022-04-29 23:30:02 UTC

R **topics documented:**

---

micromapST-package   *A graphics package to easily and quickly create linked micromaps for a specified geographic collection of areas.*

Description

The micromapST package provides a means of creating multiple column graphics representing data for a collection of geographic areas. The originally micromapST was limited to creating linked micromaps for each US 50 state and the District of Columbia. With this release, the package has been updated to be able to create linked micromaps for any collection of areas that is defined in a border group dataset (see details). Each area's graphical element is linked to a small map by means of color.

Details

Package:	micromapST
Type:	Package
Version:	1.1.3
Date:	2022-04-29
License:	GPL-2
LazyLoad:	no

The package uses the following R released packages: stringr, RColorBrewer, graphics, stats, grDevices, and labeling. When the package is download, these support packages should also be loaded. If not, please install these packages before loading micromapST.

Linked micromap plots link statistical graphs to an organized set of small maps, thus adding geographical context to the graphs and data. The *micromapST* package has been expanded to be able to use boundary data from any collection of geographic areas through the use of border group datasets. Each border group dataset contains five R objects that define the boundaries, run parameters and

name, abbreviation, and ID relationships for a geographic region. When a border group is specified in the function call, the associated dataset is loaded and the five R objects become the key data structures used by micromapST to create the linked micromaps. The five R objects are:

- *areaParms* - run parameters for this border group
- *areaNamesAbbrsIDs* - the area name, abbreviation and numeric ID associates for each area.
- *areaVisBorders* - the boundary data for each area, indexed by the area abbreviation
- *L2VisBorders* - the boundary data for overlaying boundaries within the areas.
- *RegVisBorders* - the boundary data for a region of areas.
- *L3VisBorders* - the boundary data for the entire collection of areas.

The currently the supported sets of border groups contain in this package are:

- *USStatesBG* - The 50 U.S. states and the District of Columbia
- *USSeerBG* - The 20 U.S. Seer Areas within the U.S.
- *KansasBG* - The 105 counties in the state of Kansas
- *MarylandBG* - The 24 counties of Maryland
- *NewYorkBG* - The 62 counties in the state of New York
- *UtahBG* - The 29 counties of Utah
- *SeoulSKoreaBG* - The 25 districts in the city of Seoul.
- *UKIrelandBG* - The 218 counties in UK and Ireland, for testing only.
- *ChinaBG* - The 32 provinces, municipalities, autonomous regions, and Special Administrative divisions of China.
- *AfricaBG* - The 52 countries of the African continent.

The default border group is *USStatesBG* to allow users of previous releases of *micromapST* to run without changes to their code. To support users of the test release of *micromapSEER*, a function *micromapSEER* is included. The function *micromapSEER* sets up required parameters and then calls *micromapST* with the border group set to *USSeerBG* to create the same linked micromaps as the *micromapSEER* package created. All of the call parameters are the same. The two packages were merged to allow *micromapSEER* users to benefit from new features and fixes are released under *micromapST*.

Additional border groups will be added over time. The user may also create their own border group for use with *micromapST* (see paper on creating *micromapST* border groups.)
The entire micromap is created to fits on a single page. The page may be portrait or landscape and can range from an 8.5 x 11 up to a 11 to 17 page. Areas are grouped into panels from 3 to 5 areas each based on the sort variable, with the median-valued area set off in a separate panel in the middle of the page. If the median panel contains more than 1 area, a full link micromap panel is generated. Otherwise a single line representing the area is drawn and the median area is highlighted in the panels above and below the median. In the case of the U.S. Map and 51 states, there are 5 panels of 5 areas (states) above the median and 5 panels of 5 areas (states) below the median row.

The U.S. Seer Registry data may be groups of 9, 11, 13, 17 or 18 registries. Number of registries per panel and number of panels are dynamically setup based on the number of registries involved in the micromap. There are a variety of glyphs the caller can specify for each column of the micromap:

the US map with areas colored, a list of registry names or abbreviations (the default) and one or more statistical graphics. The order of these panels is specified by the caller.

The statistical glyphs implemented in this version are plots of dots, dots with significant, dots with confidence intervals, dots and intervals based on Standard Error, horizontal bars, arrows, time series with or without confidence bands, horizontal stacked (segmented (SEGBAR), normalized (NORM-BAR) or centered (CTRBAR)) bar charts, scatter plots and boxplots. The layout of the linked micromap plot is defined by the *panelDesc* data.frame that is passed to the *micromapST* function.

If the micromap cannot fit on one page, warnings are generated and the function is stopped. It is suggested the caller increase the size of the page (graphic space) being used to compensate.

The U.S. map of states and areas used by the *USStatesBG* and *USSeerBG* border groups are generalized boundary map, based on Mark Monmonier's visibility map. These maps are simplified to maximize the color areas shown for each state and to minimize the length of the boundary lines while still allowing identification of each area. At some future time, all border groups should have their boundaries characterized to enhance the linked micromap's readability.

One of the biggest enhancements in this version of *micromapST*, is support for other geographic areas. This has been added using border groups. Each border group data set contains the unique collection of information, run parameters, names, abbreviations, and boundary information for the geographic region. The package contains some pre-made border groups and the package user is encourage gather the required information and create their own border group. This is not a small task and required boundary file manipulation to reduce the complexity of the boundaries and researching to identify a suitable list of names, abbreviations and ID for each sub-area within the desired geographical area. The author is working on a guideline and a step by step procedure to help the user create their own border groups. This release includes border group to re-create the original link micromaps in earlier versions of *micromapST* for the 50 U. S. States and DC and micromapSEER for the 20 NCI Seer Registries. Several other test versions of border groups have been included in this release. The complete list of border groups included are:

- *USStatesBG* - the default border group to continue generating the original linked micromaps in earlier versions of micromapST.

- *USSeerBG* - a border group to support creating linked micromaps for NCI. A function micromapSEER is included to support earlier coding. This function automatically pulls in the USSeerBG border group.

- *KansasBG* - a border group to support the generation of linked micromaps for the 105 counties in the state of Kansas.

- *MarylandBG* - a border group to support the generation of linked micromaps for the 23 counties and the city of Baltimore in the state of Maryland.

- *NewYorkBG* - a border group to support the generation of linked micromaps for the 62 counties in the state of New York.

- *UtahBG* - a border group to support the generation of linked micromaps for the 29 counties in the state of Utah.

- *ChinaBG* - a border group to support the generation of linked micromaps for the 35 provinces, special administrative regions, and autonomous regions of China.

- *UKIrelandBG* - a border group for testing a large number of sub-areas (counties) in multiple regions. The package currently supports up to 110. This border group is designed to help in testing and developing support for up to 218 sub areas (counties, etc.)

- *SeoulSKoreaBG* - a border group to support the generation of linked micromaps for the 25 districts within the city of Seoul South Korea.
- *AfricaBG* - a border group to support creation of linked micromaps for 52 countries of Africa. This border group is for a continent with the sub-areas being countries.

The following datasets have also been included for use in many examples:

- *Educ8thData* - Education Data from an 8th grade survey.
- *TSdata* - Sample time series data.
- *statePop2010* - State Population data for 2010.
- *wflung00and95* - White Female Lung cancer data for 2000 and 1995.
- *wflung00and95US* - White Female Lung cancer data totals for 2000 and 1995.
- *wflung20cnty* - White Female Lung cancer sample data by county for 2000.
- *wmlung5070* - White Male Lung cancer sample data for 1950 and 1970.
- *wmlung5070US* - White Male Lung cancer sample data for 1950 and 1970 totals.

Refer to the chapter on each border group for definitions on the Names, Abbreviations, and IDs used in the border group to link the user data to the boundary information to draw the micromap maps.

The sort order of the rows (areas) is based one of the statistical data columns as specified by the user. Correlation between multiple statistical columns can be judged visually by comparing the pattern of one column's values from top to bottom of the page with that of the sorted column. Spatial clusters of states with similar values of the sorting variable can be identified on the small maps that are linked to the graphics by color.

A area linked micromap plot is generated by 4 steps:

```
# load the package

library(micromapST)

# Read, create or collect your data into a data.frame.

statsDFrame <- data.frame(a row per area,
                          column per variable to be plotted,
                          row.names set to the area names
                          or abbreviations)

# now set up a data frame that defines the labels,
#       panel and page layout

panelDesc<-data.frame(...)

# specify the data source, panelDesc, sorting variable and
#       order, and call the stateMicromap function

micromapST(statsDFrame, panelDesc, title=c("title1","title2"),
          details=list(options=values))
```

The package contains a set of examples of how to produce linked area micromaps. The datasets used in each example are provided to help you learn how to use micromapSEER.

Note

The packaged is tuned to work with an area 7.5" wide and 10.5" high. Testing has shown it works well with portrait or landscape orientation and areas up to 11" x 17".

The examples in this package the output is directed to a PDF file for best clarity and resolution. File types of SVG, PNG, JPEG or TIFF can also be used. If the output is directed to a window, it is suggested a `windows( 7.5, 10.5, xpinch=72, ypinch=72, pointsize=9 )` command is used to set up the window to best display the linked micromap. The results will vary based on the resolution of the monitor being used.

Author(s)

Daniel B. Carr `<dcarr@gmu.edu>` and James B Pearson, Jr `<jbpearson353@gmail.com>`, with contributions from Linda Pickle
Maintainer: "James B. Pearson Jr." `<jbpearson353@gmail.com>`
Package compiled by "James B, Pearson, Jr." `<jbpearson353@gmail.com>`

References

Daniel B. Carr and Linda Williams Pickle, Visualizing Data Patterns with Micromaps, CRC Press, 2010
Linda Williams Pickle, James B. Pearson Jr., Daniel B. Carr (2015), micromapST: Exploring and Communicating Geospatial Patterns in US State Data., Journal of Statistical Software, 63(3), 1-25., https://www.jstatsoft.org/v63/i03/

AfricaBG *AfricaBG border group datasets to support creating micromaps for the 52 countries in Africa*

Description

The *micromapST* function has the ability to generate linked micromaps for any geographical area. To specify the geographical area, the 'bordGrp' call argument is used to specify the border group dataset for the geographical area. The *AfricaBG* border group dataset supports creating linked micromaps for the 52 countries (sub-areas) on the African continent. When the 'bordGrp' call argument is set to *AfricaBG*, the appropriate name table (country names and abbreviations) and the 52 sub-areas (countries) boundary data is loaded in *micromapST*. The user's data is then linked to the boundary data via the country's name, abbreviation, alternate abbreviation, or ID based on the table below.

Details

The *AfricaBG* border group dataset contains the following data.frames:

areaParms - contains specific parameters for the border group

areaNamesAbbrsIDs - containing the names, abbreviations, and numerical identifier for the 59 countries in Africa.

areaVisBorders - the boundary point lists for each country in Africa.

L2VisBorders - the boundaries for an intermediate level and is not used in this border group and is set to L3VisBorders as a place holder.

RegVisBorders - the boundaries for regions in Africa. In this implementation of the border group, no regions are specified. This data frame is not used and is set to L3VisBorders as a place holder.

L3VisBorders - the boundary of the Africa continent.

The Africa continent border group contains 52 country sub-areas. Each country has a row in the areaNamesAbbrsIDs data.frame and a set of polygons in the areaVisBorders data.frame datasets. No regions are defined in the Africa border group, so the L2VisBorders dataset is not used and the regions option is disabled. The L3VisBorders dataset contains the outline of the Africa continent.

The details on each of these data.frame structures can be found in the "bordGrp" section of this document. The *areaNamesAbbrsIDs* data.frame provides the linkages to the boundary data for each sub-area (country) using the fullname, abbreviation, alternate abbreviation, and numerical identifier for each country to the *<statsDFrame>* data based on the setting of the 'rowNames' call argument.

A column or the data.frame row.names must match one of the types of names in the *areaNamesAbbrsIDs* data.frame name table. If the data row does not match a value in the name table, an warning is issued and the data is ignored. If no data is present for a sub-area (country) in the name table, the sub-area is mapped but not colored.

The following are a list of the names, abbreviations, alternate abbreviations and IDs for each country in the *AfricaBG* border group.

name	ab	alt_ab	id
Algeria	ALG	DZ	01
Angola	ANG	AO	02
Benin	BEN	BJ	03
Botswana	BOT	BW	04
Burkina Faso	BUF	BF	05
Burundi	BUR	BI	06
Cameroon	CAM	CM	07
Cape Verde	CAP	CV	08
Central African Republic	CAR	CF	09
Chad	CHA	TD	10
Comoros	COM	KM	11
Congo-Brazzaville	CNG	CG	12
Cote d'Ivoire	CDI	CI	13
Democratic Republic of Congo	ZAI	ZR	14
Djibouti	DJI	DJ	15
Egypt	EGY	EG	16

Equatorial Guinea	EQG	GQ	17
Eritrea	ERI	ER	18
Ethiopia	ETH	ET	19
Gabon	GAB	GA	20
Gambia	GAM	GM	21
Ghana	GHA	GH	22
Guinea	GIN	GN	23
Guinea-Bissau	GUB	GW	24
Kenya	KEN	KE	25
Lesotho	LES	LS	26
Liberia	LIB	LR	27
Libya	LAJ	LY	28
Madagascar	MAD	MG	29
Malawi	MAA	MW	30
Mali	MAL	ML	31
Mauritania	MAU	MR	32
Morocco	MOR	MA	33
Mozambique	MOZ	MZ	34
Namibia	NAM	NA	35
Niger	NIG	NE	36
Nigeria	NIR	NG	37
Rwanda	RWA	RW	38
Sao Tome and Principe	STP	ST	39
Senegal	SEN	SN	40
Sierra Leone	SIL	SL	41
Somalia	SOM	SO	42
South Africa	SOU	SA	43
Sudan	SUD	SD	44
Swaziland	SWA	SZ	45
Tanzania	TAN	TZ	46
Togo	TOG	TG	47
Tunisia	TUN	TN	48
Uganda	UGA	UG	49
Western Sahara	WES	WS	50
Zambia	ZAM	ZM	51
Zimbabwe	ZIM	ZW	52

When this border group was created, there appeared to be no consistant set of abbreviations for the African countries. Therefore, the two most commonly found sets of abbreviations are included as the *abbr* and *alt_abbr* abbreviation sets. Set 'rownames' to "ab" to reference the primary set and "alt_ab" to reference the second set of abbreviates in the name table.

The 'rowNames' = "alias" and the 'regionB' and 'dataRegionsOnly' features are not supported in the *AfricaBG* border group.

AfricaPopData   *Test data for the Africa border Group*

Description

This dataset contains the population and country data for the 52 countries in the African border group.

Usage

```
data(AfricaPopData)
```

Format

A data frame with 52 observations, 1 for each African country, on the following "x" variables.

Rank an integer rank of the country in Africa.

Name a character vector containing the Africa Country Name.

Abbr a character vector containing the African Country Abbreviation.

Projection a numeric vector of the number of the county's population

AvrRelGw a numeric vector of the average relative population growth.

AvrAbsGw a numeric vector of the average absolute population growth.

Est2Time a numeric vector of the estimated time to double the population - years.

OfficialPop a numeric vector of the official population.

MMDDYY the date the information was last updated.

PercOf a numeric vector representing the percentage the country's population is to the total population of Africa.

Details

This dataset was pulled from wikipedica on the population numbers for African countries.

Author(s)

Linda W. Pickle and Jim Pearson of StatNet Consulting, LLC, Gaithersburg, MD

---

bordGrp	*Defining different spatial areas for Linked Micromap using the micromapST package*

Description

The *micromapST* function can be used to create linked micromaps for may different spatial areas by using different border groups. Several border group (bordGrp) examples are contained in the package and include the 51 states and DC of the United States, the counties of Kansas, Maryland, New York, Utah, the countries and provinces of the U.K. and China, and the U. S. Seer Registries used by the National Cancer Institute. Each border group is a different dataset containing the unique boundaries and operational information to allow micromapST to work in a different spatial area. The structure of each border group dataset is identical with the same variable names and types of structures. A user can build their own border group dataset to meet their specific spatial area needs. Because the package contains several border group datasets, the use of lazydata or lazyloading has been disabled.

The name of the border group is specified in the 'bordGrp' call parameter. To permit a user to reference a border group dataset not contained in the package, and reside in a user's folder, the 'bordDir' must be used to direct the package to the border group. The border group must be saved under R using the *save* function with the file extension of ".rda". For example: bordGrp="private", bordDir="c:/SavedBorderGroups"

Each border group contain six (6) datasets by the same data.frame names. This allows the micromapST package the ability to quickly load a particular border group and create the requested micromaps. The five data.frames are: areaParms, areaNamesAbbrsIDs, areaVisBorders, L2VisBorders, RegVisBorders, and L3VisBorders. Since the same data.frame names are reused in each border group, the R lazyload feature is disabled in the package.

The following describes the purpose and structure of each data.frame in the border group dataset:

areaParms - contains specific parameters and operational information for the border group

areaNamesAbbrsIDs - containing the names (full text), name abbreviations, wildcard string for name matching, alternate name abbrevations, regional (Level 2) association of the sub-area, and a numerical identifier for the sub-area in the border group area. If the border group is for a state, the sub-areas would be the counties within the state. If the border group is for a continent, then the sub-areas are the countries on the continent. If the border group is for a country like the United States, then the sub-areas are the administrative sub-areas (like states, provinces, special administrative areas, or cities) within that country.

areaVisBorders - the boundary point lists for each sub-area in the border group area.

L2VisBorders - when a border group needs to have an intermediate set of boundaries drawn for clearity, the border group can provide the package a layer 2 set of boundaries via the *L2VisBorders* data.frame. The structure consists of the boundary information (point lists) of these areas and associated keys. Each sub-area is linked to it's L2 boundary via the *L2_ID* variable (column) in the name table (*areaNamesAbbrsIDs* data.frame. At this time only the U. S. Seer Registry border group uses the L2 boundary overlays. It uses L2VisBorders to draw the boundaries of U. S. States around Seer Registries within a state. The *areaParms$Map.L2Borders* variable in the border group must be set to *TRUE* for the package

to draw the layer 2 boundaries. If a border group does not use an intermediate level out-line, L2VisBorders should be set to L3VisBorders or NA and the *areaParms$Map.L2Borders* variable set to FALSE. In this case, no L2 boundaries are drawn.

RegVisBorders - when the border group wants to work with the sub-areas on a regional basis, the *regID* variable in the name table (*areaNamesAbbrsIDs*), the *RegVisBorders* boundary information data.frame, and the *areaParms$aP_Regions* are used to enable the feature and provide the information to map only regions of sub-areas and overlay areas with regional boundaries highlights. When the 'regions' call parameter is set to *TRUE* and the selected border group has the *areaParms$aP_Regions* set to *TRUE*, the package will scan the data provided by the caller and determine which regions are being referenced and which are not. The package uses the *regID* variable in the name table(*areaNamesAbbrsIDs*) to associate a sub-area with a region and as a key into the *RegVisBorders* data.frame to draw the region boundaries. Any region not containing data and any L2 area and sub-areas within those regions will not be drawn. If a border group does not use the regions feature the *RegVisBorders* data.frame in the border group should be set to the L3VisBorders data.frame or NA and the *areaParms$aP_Regions* variablel set to *FALSE*. This will disable the regions feature for the border group.

L3VisBorders - the boundary point list for an outline of the entire geographica area referenced by the border group. For the U.S. or a country border group, this is the outline of the country. For a state border group like Kansas, this is an outline of the state. For smaller areas like Seoul, it is an outline of the city. The L3VisBorders data frame must be present in the border group.

Details

The default border group is *USStatesBG* to be compatible with older R scripts using previous versions of the *micromapST* package.

This section provide the data frame structural details of each of data.frame and variables in the border group dataset.

areaParms data.frame The *areaParms* data.frame contains specific information about and in sup-port of its border group. It provides column headers for the map and id glyphics and controls several features that related to handling a border group by the *micromapST* package. There controls do not change from *micromapST* call to call and are therefore not part of the calling parameter set.

For example, there are several built in titles and labels for the map and id glyphics. These will change for different border groups. For the *USStatesBG* border group, the map title is always "U.S. States", while in the *USSeerBG* border group the map glyphic title is "U.S. Seer Areas".

This dataset contains the specific values related to a specific border group for the following variables.

The areaParms data.frame for each border group dataset contains the following variables:

areaUSData - logical variable. If *TRUE* then the border group represents the entire US area and labels are placed on the first map for Alaska, Hawaii, and DC. This is option is only used with the *USStatesDF* and *USSeerBG* border groups. For the state/county border groups and foreign country border groups, *areaUSData* must be set to *FALSE*.

enableAlias - If *TRUE*, enables the use of the 'rowNames'=*alias* call argument option. This permits a partial ("contains") match of the data in the 'rowNamesCol' link column in the *statsDFrame* or the row.names of the *statsDFrame* data.frame. This is only supported in

the *USSeerBG* bordGrp to allow direct use of data exported from the SEER*Stat website and match on the exported SEER*Stat registry names. This feature can be expanded when needed to other border groups.

aP_Regions - If *TRUE*, the package contains a valid *RegVisBorders* data.frame and the name table (*areaNamesAbbrsIDs*) contains a key for each sub-area to the region boundaries in the *regID* variable. If the caller set the 'regions' parameter to *TRUE*, the package will only draw regions and their boundaries if the region contains sub-areas with data. For examples: this allows you to provide data for the west coast U. S. states and not have the midwest, south, or northeastern states drawn. This feature is available to all border groups, but is currently only used by the USStatesBG, USSeerBG, UKIrelandBG and ChinaBG border groups. If set to *FALSE*, the regions feature is disabled and all regional information ignored.

The *RegVisBorders* should still exist, but should be set to equal the *L3VisBorders* data.frame. The 'regions' call parameter will be ignored. As an example: In the USStatesBG and USSeerBG border groups, the regions are set up using the 4 U. S. census regions of: West, South, Midwest, and NorthEast. If only data for states in the NorthEast are passed to micromapST, only the NorthEast region and its states will be mapped when 'regions' is set to *TRUE*. If 'regions' is set to *FALSE* then all of the boundaries for all of the U. S. States and DC are drawn eventhough data was only provided for the states in the northeast region. This feature also allows a border group with a large number of sub-area, like the UK and Ireland border group to be assembled and used on a regional basis with fewer sub-area were the full border group is not really usable with linked micromap presentations.

ID.Hdr.1 - 1st title for the id type glyphics column.

ID.Hdr.2 - 2nd title for the id type glyphics column.

Map.Hdr.1 - 1st title for the map type glyphics column.

Map.Hdr.2 - 2nd title for the map type glyphics column.

Map.Aspect - is the X/Y aspect ratio for the map borders in this border group. This is used to adjust the map glyphic to keep the area's aspect looking correct.

Map.MinH - the minimum height of a group/row in inches

Map.MaxH - the maximum height of a group/row in inches.

bordGrp - a character vector of the name of the border group.

Map.L2VisBorders - logical variable. If *TRUE*, the L2VisBorder border overlay is drawn on the map glyphics. If *FALSE*, the *L2VisBorders* boundaries are not drawn. This option is currently only used by USSeerBG to draw state boundaries for states containing multiple Seer Registries.

aP_Regions - logical variable. If *TRUE*, the regions feature is enabled. When the feature is enabled, the *RegVisBorders* data.frame should be included in the border group, but it is not required. The key to the regional feature is the *regID* column in the name table (*areaNamesAbbrsIDs*) that identifies the region associated with the sub-area and the regional boundaries in the *RegVisBorders* data.frame. If *FALSE*, the regions feature is disabled.

aP_Proj - a character vector containing the projection used to create the areaVisBorders, L2VisBorders, RegVisBorders, and L3VisBorders boundary point lists.

aP_Units - a character vector containing the measurement units of the coordinates in the VisBorders boundary point lists.

Map.L3VisBorders - logical variable. If *TRUE*, the L2VisBorder boundary data is available to drawn on the map glyphics. If *FALSE*, the *L2VisBorders* boundaries are not available

and are not drawn. This option is currently only used by the following border groups: USSeerBG.

Map.RegVisBorders - logical variable. If *TRUE*, the *RegVisBorder* boundary data is available in the border group and can be used to drawn a regional boundaryy overlay on the map glyphics. If *FALSE*, the *RegVisBorders* boundary data is not available and regional boundaries are not drawn. This set of boundaries are only only available in the following border groups: USStatesBG, USSeerBG, ChinaBG, and UKIrelandBG. The drawing of the regional boundaries is controlled by the 'regionB' call parameter.

areaNamesAbbrsIDs The *areaNamesAbbrsIDs* data.frame provides the linkages between the full-name, abbreviation, and numerical identifier to link the *statsDFrame* data to the boundaries for the county areas. It is also used to validate the incoming data to ensure the linkage can be established. Within the boundary files, the area abbreviation is the key linkage. areaNamesAbbrsIDs data.frame.

areaNamesAbbrsIDs dataset provides a table to permit the translation of a numerical ID (e.g., FIPS codes), area abbreviatation (should be less than 4-6 characters), optional alternate abbreviation, and full area names in the *micromapST* function.

Name - character string of each sub-area name. Used to link the areas to boundaries when 'rowNames' = "full" is speccified. Used as the represented name of the sub-area in "ID" glyphics columns when 'plotNames' = "full" is spedified (default).

Abbr - the name abbreviation for each sub-area. Should be 2 to 3 character, but no more than 6 characters. Used to link the sub-areas to boundaries when 'rowNames' = "ab" is specified. Used as the represented name of the area in "ID" glyphics columns when 'plotNames' = "ab" is specified (default).

Alt_Abr - an alternate name abbreviation for each sub-area. Should be 2 to 3 characters, but no more than 6 characters. Most of the time this field is identical to the *Abbr* field. In some cases, multiple sets of authenticated abbreviations were found for the sub-areas in a continent or country. When this happened, the most common abbreviation should be placed in the *Abbr* field and the second abbreviation placed in the *Alt_Abr* field. This features allows the border group to be used by a wider audience. To access the *Alt_Abr* abbreviates, set the 'rowNames' call argument/parameter to "alt_ab". The labels in the *statsDFrame* statistics data.frame will be matched against the alternate abbreviations.

ID - numerical identifier for each sub-area. Used to link the data to boundary information when 'rowNames' = *id* is specified. The row.names in the user provided data.frame or speccified 'rowNamesCol' column must match the IDs in the name table. If no match a warning is generated and the data row ignored.

Alias - a character string for each area used to do a wildcard match ("contains") with the 'rowName' or 'rowNamesCol' specified in the *micromapST* call when the *USSeerBG* border group is used. When a match is completed, the abbreviation is used to link the user's data to the boundary data. The alias match is done when 'rowNames' is set to "alias". There should be only one match in the data for each sub-area alias. If more are found, an error is raised. This feature is only supported in the *USSeerBG* border group.

L2_ID - is the level 2 identifier. Used to link the sub-area to the L2VisBorders boundary data point data.frame. In the case of the USSeerBG border group, the L2 boundaries are state boundaries and the L2_ID value is the state 2 character abbreviation.

L2_ID_Name - is the full name L2 area.

regID - is the region identifier. Used to link the sub-area to the RegVisBorders boundary data point data.frame. The USStatesBG and USSeerBG border groups use this field to like

the sub-areas to the four (4) U. S. Census regions (Northeast, South, MidWest, and West) This association is used with the 'regions' call parameter to determine which regions and their sub- areas, etc. will be drawn when the caller provide data is mapped.

regName - is the full name of a region.

Key - is a character string used to link the name table to the boundary data in the VisBorders data.frames.

Link - when the initial border group is created, the *Key*, *Name* or the *Abbr* variables may not be able to provide a link to the original boundary data. When this happens, the border group creater must use an alternate "link" to tie the name table to the boundary data. The *link* field is used to accomplish this in the package provided border group building steps and functions. Once the border group is fully constructed, the *Link* field is not use.

MapLabel - when a sub-area is moved or enlarge on a map for better visibility, it may not be clear to the user what sub-area is being represented. In the U. S. border groups, the states of Alaska, Hawaii and the District of Columbia were modified and not in their normal position or size. This variable identifies these sub-areas and provide the information to drawn text labels near the sub-areas on the first map type glyph in a column. The *MapLabel* variable consists a character string consisting of three values: a text label and the numerical x and y coordinates on the map to draw the label. The text label should be as small as possible, at most 2 characters. This should only be done if absolutely necessary. If too many sub-area's are labeled, the map glyphs becomes unreadable. If value is missing (NA) or any of the components are missing (NA), then the label is not drawn. The x/y coordinate represent the lower left position of the location to print the string and must be in the same coordinates system used for the boundary data in the border group.

areaVisBorders, L2VisBorders, RegVisBorders, and L3VisBorders There are four boundary dataset - The boundaries of the areas being micromapped (counties) are in areaVisBorders. The boundaries of an intermediate level (2) for general orientation are in L2VisBorders. The boundaries of regional areas for highlight overlays and regional only mapping and are in RegVisBorders. The boundaries of the an outline of the entire mapping area is in L3VisBorders.

These four data.frames contain the boundary point lists for the area, regions and sub-areas. The L2VisBorders, RegVisBorders, and L3VisBorders data.frames are used to outline groups of sub-areas, regions of the area and the entire area. The areaVisBorders data.frame contains the point lists for each sub-areas and are keyed to the name table (*areaNamesAbbrsIDs*).

The data structure for each of the following four boundary data.frames is:

seq Key x y hole

seq - a numerical sequence number of the boundary points in the data.frame.

Key - the Key field had different uses in each of the four VisBorders structured data.frames. In the areaVisBorders data.frame, the *Key* is the unique key for the sub-area as defined in the name table (*areaNamesAbbrsIDs*). All of the points with the same *Key* form one or more polygons and represent boundaries of a sub-area. This allows the package to locate the boundary data for a specific sub-area when needed.

In the RegVisBorders boundary data.frame, the *Key* is the region ID associated with the boundary points (polygons). One or more sub-areas can be linked to an region boundary via the *regID* variable in the name table. The USStatesBG, USSeerBG, UKIreland BG,and ChinaBG border groups may use of the regions feature.

In the L3VisBorders, all of the *Key* field values area set to a name that represents the entire border group area. The *Key* field is not used in drawning the area's outline. The

level 3 boundary outline data.frame is always drawn when the entire geographic area is mapped. If not all of the regions are being mapped, then the L3 boundary is not drawn.

x - a numerical value for the *x* coordinates of a polygon. The end of the polygon is signaled by an *x* value of *NA*. The first point in the polygon does not have to be repeated. The polygon draw function used by micromapST will close the figure. An area may be composed of several polygons. Holes in areas are also represented by polygons and are associated with the sub-area.

y - a numerical value for the *y* coordinates of a polygon. The end of the polygon is signaled by an *y* value of *NA*.

hole - a logical value. If *TRUE*, the associated polygon is a hole within the sub-area identified by the *Key* field. A hole polygon is always drawn using the maps background color. For this reason, sub-areas containing holes (lakes, rivers, or other sub-areas), must proceed any sub-area in the data.frame that it may overlay with the hole. For example, if one sub-area "A" is contained within sub- area "B", sub-area "B" must have a hole where sub-area "A" is located and must preceed sub-area "A" in the VisBorders data.frame. In this way, sub-area "B" and its hole are drawn before sub-area "A" preventing sub- area "B" hole from overlaying sub-area "A".

Hole are processed by re-drawing the hole area with the current background color. Therefore, any area with holes must be in the data.frame prior to any areas that may occupy the hole's space in the map.

The order of the area's boundaries in these files is very important to allow correct processing of the holes and any areas that overlay holes. Holes are not unpainted polygons, but polygons repainted back to the background color. The order should be areas with holes must preceed areas that overlay hole space. This is required to ensure an area is not over-painted by an area's hole

Each data.frame should be validated before using to make sure they are clean and will not generate errors when used by *micromapST*.

Each border group contains the same six data.frames using the same six names. This allows the micromapST package the ability to quickly load a particular border group and create the requested micromaps.

See the write up on each included border group for details on the specific content of their border group dataset and the list of sub-area names, abbreviations, and id that should be used to link the data to the boundary information.

The 'regions' feature and RegVisBorder overlays are only supported in the following border groups:

USStatesBG USSeerBG UKIrelendBG ChinaBG

border groups.

ChinaBG   *ChinaBG border group datasets to support creating micromaps for the*
                                 32 providences and municipalities in the country of Republic of China

---

Description

The *micromapST* function has the ability to generate linked micromaps for any geographical area. To specify the geographical area, the 'bordGrp' call argument is used to specify the border group dataset for the geographical area. The *ChinaBG* border group dataset supports creating linked micromaps for the 34 provinces, special administrative regions, metropolitan areas in the China. When the 'bordGrp' call argument is set to *ChinaBG*, the appropriate name table (sub area names and abbreviations) and the 34 sub-areas (provinces, SAR, cities, etc.) boundary data is loaded in *micromapST*. The user's data is then linked to the boundary data via the county's name, abbreviated name or ID based on the table below.

Details

The *ChinaBG* border group dataset contains the following data.frames:

areaParms - contains specific parameters for the border group

areaNamesAbbrsIDs - containing the names, abbreviations, and numerical identifier for the providences and municipalities of China.

areaVisBorders - the boundary point lists for each area in China.

L2VisBorders - the boundaries for an intermediate level. This level is not used in this border group.

RegVisBorders - the boundaries for an regional level for China. This set of boundaries are used in conjunction with the regions call parameter.

L3VisBorders - the boundary of the country of China.

For the China border group, there are 34 county sub-areas listed in the areaNamesAbbrsIDs and areaVisBorders datasets. The *L2VisBorders* dataset is not used and is set to the *L3VisBorders* dataset as a placeholder. The *RegVisBorders* dataset represents the 6 regions of China in the *ChinaBG* border group. The L3VisBorders dataset contains the outline of the country of China.

The details on each of these data.frame structures can be found in the "bordGrp" section of this document. The *areaNamesAbbrsIDs* data.frame provides the linkages to the boundary data for each sub-area using the fullname, abbreviation, alternate abbreviation, and numerical identifier for each country to the *<statsDFrame>* data based on the setting of the 'rowNames' call argument. A column or the data.frame row.names must match one of the types of names in the *areaNamesAbbrsIDs* data.frame name table. If the data row does not match a value in the name table, an warning is issued and the data is ignored. If no data is present for a sub-area (county) in the name table, the sub-area is mapped but not colored.

The following are a list of the names, abbreviations, alternate abbreviations, ids, and the region for each county, province or metro area in the *ChinaBG* border group.

name	ab	alt_ab	id	region
Anhui	AH	CN.AH	34	Huadong
Beijing	BJ	CN.BJ	11	Huabei
Chongqing	CQ	CN.CQ	50	Xinan
Fujian	FJ	CN.FJ	35	Huadong
Gansu	GS	CN.GS	62	Xibei
Guangdong	GD	CN.GD	44	Zhongnan
Guangxi	GX	CN.GX	45	Zhongnan

Guizhou	GZ	CN.GZ	52	Xinan
Hainan	HI	CN.HA	46	Zhongnan
Hebei	HE	CN.HB	13	Huabei
Heilongjiang	HL	CN.HL	23	Dongbei
Henan	HA	CN.HE	41	Zhongnan
Hubei	HB	CN.HU	42	Zhongnan
Hunan	HN	CN.HN	43	Zhongnan
Jiangsu	JS	CN.JS	32	Huadong
Jiangxi	JX	CN.JX	36	Huadong
Jilin	JL	CN.JL	22	Dongbei
Liaoning	LN	CN.LN	21	Dongbei
Nei Menggu	NM	CN.NM	15	Huadong
Ningxia Hui	NX	CN.NX	64	Xibei
Qinghai	QH	CN.QH	63	Xibei
Shaanxi	SN	CN.SA	61	Xibei
Shandong	SD	CN.SD	37	Huadong
Shanghai	SH	CN.SH	31	Huadong
Shanxi	SX	CN.SX	14	Huabei
Sichuan	SC	CN.SC	51	Xinan
Tianjin	TJ	CN.TJ	12	Huabei
Xinjiang Uygur	XJ	CN.XJ	65	Xibei
Xizang	XZ	CN.XZ	54	Xinan
Yunnan	YN	CN.YN	53	Xinan
Zhejiang	ZJ	CN.ZJ	33	Buadong
Hongkong	HK	CN.HK	81	Zhongnan
Macau	MC	CN.MC	82	Zhongnan
Taiwan	TW	CN.TW	71	Huadong

The *ChinaBG* supports two sets of abbreviations for each county (province or metro area). No consistant source was found when the border group was originally created. Therefore, the two most common sets are included. The first abbreviation set can be referenced by setting the 'rowNames' call argument to "ab". The second (alternate) abbreviation set can be used by setting the 'rowNames' to "alt_ab".

The 'rowNames' = "alias" features are not supported in the *ChinaBG* border group.

---

cnPopData	*Test data for the China border Group*

Description

This dataset contains the 2014 population and average income per person in each of the China Areas

Usage

```
data(cnPopData)
```

Format

A data frame with 18 observations, 1 for each Seer area, on the following 12 variables.

area a character vector containing the China area full names.

pop2013 a numeric vector of the number of the county's population

Details

This dataset was pulled from the China ??? government website on Dec. 2014. It contains the population and average income per person for each of Kansas' 105 counties.

---

detailsVariables *Validation and Translation table for details variables*

Description

An internal table containing a list of the details variables and parameters to do validation of user provided variable overrides of the defaults. The table also contains the information needed to translate the existing (as of 9/17/2015) details structure into a new details structure segmented by each type of glyphic.

Usage

```
data(detailsVariables)
```

Format

A data frame with 6 variables:

varName a character string of the exact details variable name

method a character string describing the type of test required to validate the variable. The supported tests are: colors, integer, numeric, logical, and vector3

v1 first variable for the "test" for this variable.

v2 second variable (if needed) for the "test" for this variable.

usedBy a vector of character string listing the glyphics that use this variable. For Example: c('ts','tsconf')

newVarName the new variable name to be used within a glyphic. The glyphic name and variable name must be unique. This eliminates having to include the glyphic name in the variable name.

d_range defines the range of the dependency for the dependent relationship.

dependent indicates this variable is dependent on another variable and it's name.

default the default values or the variable.

comments operational comments on the variable.

Details

This dataset provide a table to *micromapST* for verification of the details variables provided by a user to override the packages default values. The *varName* is the exact name of the variable. If the variable name provide by the user does not match this list, it is flagged as an error and ignored. The test supported are: colors, integer, numeric, logical and vector3. The colors test calls the is.Color function. The integer and numeric tests use the range provided in *v1* and *v2* columns to check the range of the value for the variable. The logical test verifies the value is TRUE or FALSE. The vector3 test check to make sure the value is a vector or length 3 and each value is within the range in *v1* to *v2*.

More to be added later. *****

The micromapST package is being updated to use a new variable structure. The new structure will allow options/parameters to be specified on glyphic and column basis. This allows each column to be uniquely tuned to the user requirements. The variable names have also been simplifed to use the same name across glyphics when the purpose is the same. This should reduce the users learning curve.

Educ8thData   *Education Survey of 8th Grade Proficiency in Math*

Description

Math Proficiency Survey Results for 8th Graders in 2011 by State

Usage

```
data(Educ8thData)
```

Format

A data frame with 51 observations (one per state) on the following 7 variables.

StAbbrev a character string representing the 2 character state Id for this row.

State a character string factor of the state full name.

avgscore a numeric vector of average proficiency score for each state

PctBelowBasic a numeric vector of percentage of students with below basic scores

PctAtBasic a numeric vector of percentage of students at basic proficiency.

PctProficient a numeric vector of percentage of students at the proficient level

PctAdvanced a numeric vector of percentage of students scoring at the advanced level.

Details

The dataset contains 51 records, one for each state/area. The data represents the percentage of 8th grade students in 2011 in that state who tested at each proficiency level in math: less than basic, basic, proficient and advanced. The row name is the state abbreviation - 2 characters. This dataset is used by the *micromapSEER* examples using the "USStatesDF" border group.

Source

The National Center for Education Statistics, Department of Education
http://nces.ed.gov/nationsreportcard/states/

KansasBG	*KansasBG border group datasets to support creating micromaps for the 105 counties in the state of Kansas*

Description

The *micromapST* function has the ability to generate linked micromaps for any geographical area. To specify the geographical area, the 'bordGrp' call argument is used to specify the border group dataset for the geographical area. When *micromapST* function is used to micromap data for Kansas County area, the border group option ('bordGrp') is set to "KansasBG". This instructs micromapST to load the area Name, Abbreviation, ID and boundaries files for Kansas 105 counties. The datasets contained in the border group are areaNamesAbbrsIDs, areaVisBorders, L2VisBorders, and L3VisBorders for the counties of the state Kansas. The user's data is then linked to the boundary data via the county's name, abbreviated name or ID based on the table below.

Details

The border group contains the following data.frames::

areaParms - contains specific parameters for the border group

areaNamesAbbrsIDs - containing the names, abbreviations, and numerical identifier for the counties in the state of Kansas.

areaVisBorders - the boundary point lists for each county area in Kansas.

L2VisBorders - the boundaries for an intermediate level. For state areas, this boundary data is not used and set to L3VisBorders as a place holder.

RegVisBorders - the boundaries for an intermediate level. For state areas, this boundary data is not used and set to L3VisBorders as a place holder.

L3VisBorders - the boundary of the state of Kansas

The Kansas county border group contains 105 county sub-areas. Each county has a row in the areaNamesAbbrsIDs data.frame and a set of polygons in the areaVisBorders data.frame datasets. No regions are defined in the Kansas county border group, so the *L2VisBorders* and *RegVisBorders* datasets are not used and the regions feations is disabled. The L3VisBorders dataset contains the outline of the state of Kansas.

The details on each of these data.frame structures can be found in the "bordGrp" section of this document. The *areaNamesAbbrsIDs* data.frame provides the linkages to the boundary data for each sub-area (county) using the fullname, abbreviation, and numerical identifier for each country to the *<statsDFrame>* data based on the setting of the 'rowNames' call parameter.

A column or the data.frame row.names must match one of the types of names in the *areaNamesAbbrsIDs* data.frame name table. If the data row does not match a value in the name table, an warning

is issued and the data is ignored. If no data is present for a sub-area (county) in the name table, the sub-area (county) is mapped but not colored.

The following are a list of the names, abbreviations, and IDs for each country in the *KansasBG* border group.

name	ab	id
Allen	AL	20001
Anderson	AN	20003
Atchison	AT	20005
Barber	BA	20007
Barton	BT	20009
Bourbon	BB	20011
Brown	BR	20013
Butler	BU	20015
Chase	CS	20017
Chautauqua	CQ	20019
Cherokee	CK	20021
Cheyenne	CN	20023
Clark	CA	20025
Clay	CY	20027
Cloud	CD	20029
Coffey	CF	20031
Comanche	CM	20033
Cowley	CL	20035
Crawford	CR	20037
Decatur	DC	20039
Dickinson	DK	20041
Doniphan	DP	20043
Douglas	DG	20045
Edwards	ED	20047
Elk	EK	20049
Ellis	EL	20051
Ellsworth	EW	20053
Finney	FI	20055
Ford	FO	20057
Franklin	FR	20059
Geary	GE	20061
Gove	GO	20063
Graham	GH	20065
Grant	GT	20067
Gray	GY	20069
Greeley	GL	20071
Greenwood	GW	20073
Hamilton	HM	20075
Harper	HP	20077
Harvey	HV	20079
Haskell	HS	20081
Hodgeman	HG	20083
Jackson	JA	20085

Jefferson	JF	20087
Jewell	JW	20089
Johnson	JO	20091
Kearny	KE	20093
Kingman	KM	20095
Kiowa	KW	20097
Labette	LB	20099
Lane	LE	20101
Leavenworth	LV	20103
Lincoln	LC	20105
Linn	LN	20107
Logan	LG	20109
Lyon	LY	20111
Marion	MN	20115
Marshall	MS	20117
McPherson	MP	20113
Meade	ME	20119
Miami	MI	20121
Mitchell	MC	20123
Montgomery	MG	20125
Morris	MR	20127
Morton	MT	20129
Nemaha	NM	20131
Neosho	NO	20133
Ness	NS	20135
Norton	NT	20137
Osage	OS	20139
Osborne	OB	20141
Ottawa	OT	20143
Pawnee	PN	20145
Phillips	PL	20147
Pottawatomie	PT	20149
Pratt	PR	20151
Rawlins	RA	20153
Reno	RN	20155
Republic	RP	20157
Rice	RC	20159
Riley	RL	20161
Rooks	RO	20163
Rush	RH	20165
Russell	RS	20167
Saline	SA	20169
Scott	SC	20171
Sedgwick	SG	20173
Seward	SW	20175
Shawnee	SN	20177
Sheridan	SD	20179
Sherman	SH	20181

Smith	SM	20183
Stafford	SF	20185
Stanton	ST	20187
Stevens	SV	20189
Sumner	SU	20191
Thomas	TH	20193
Trego	TR	20195
Wabaunsee	WB	20197
Wallace	WA	20199
Washington	WS	20201
Wichita	WH	20203
Wilson	WL	20205
Woodson	WO	20207
Wyandotte	WY	20209

There are no alternate abbreviations or regions assocated with counties in Kansas.

The *id* field value is the U. S. state and county FIPS code.

The 'rowNames' = "alias" or "alt_ab" and the 'regionB' and 'dataRegionsOnly' features are not supported in the *KansasBG* border group.

KansPopInc   *Test data for the Kansas border Group*

Description

This dataset contains the 2014 population and average income per person in each of the 105 Kansas counties.

Usage

```
data(KansPopInc)
```

Format

A data frame with 18 observations, 1 for each Seer area, on the following 12 variables.

County a character vector containing the Kansas County Name.

Pop a numeric vector of the number of the county's population

AvgInc a numeric vector of the average person's income for the county.

Details

This dataset was pulled from the Kansas government website on Dec. 2014. It contains the population and average income per person for each of Kansas' 105 counties.

Author(s)

Linda W. Pickle and Jim Pearson of StatNet Consulting, LLC, Gaithersburg, MD

MarylandBG	*MarylandBG border group datasets to support creating micromaps for the 24 counties in the state of Maryland*

Description

The *micromapST* function has the ability to generate linked micromaps for any geographical area. To specify the geographical area, the 'bordGrp' call argument is used to specify the border group dataset for the geographical area. The *MarylandBG* border group dataset supports creating linked micromaps for the 24 counties in the state of Maryland. When the 'bordGrp' call argument is set to *MarylandBG*, the appropriate name table (county names and abbreviations) and the 24 sub-areas (countries) boundary data is loaded in *micromapST*. The user's data is then linked to the boundary data via the county's name, abbreviated or ID based on the table below.

Details

The *MarylandBG* border group dataset contains the following data.frames:

areaParms - contains specific parameters for the border group

areaNamesAbbrsIDs - containing the names, abbreviations, and numerical identifier for the counties in the state of Maryland.

areaVisBorders - the boundary point lists for each county area in Maryland.

L2VisBorders - the boundaries for an intermediate level. For this border group, this boundary data is not used and set to L3VisBorders as a place holder.

RegVisBorders - the boundaries for an intermediate level. For this border group, this boundary data is not used and set to L3VisBorders as a place holder.

L3VisBorders - the boundary of the state of Maryland

The Maryland county border group contains 24 county sub-areas. Each county has a row in the areaNamesAbbrsIDs data.frame and a set of polygons in the areaVisBorders data.frame datasets. No regions are defined in the Utah county border group, so the L2VisBorders and RegVisBorders data.frames is not used and the 'dataRegionsOnly' call parameter are is disabled. The L3VisBorders dataset contains the outline of the state of Maryland.

The details on each of these data.frame structures can be found in the "bordGrp" section of this document. The *areaNamesAbbrsIDs* data.frame provides the linkages to the boundary data for each sub-area (county) using the fullname, abbreviation, and numerical identifier for each country to the <*statsDFrame*> data based on the setting of the 'rowNames' call argument.

A column or the data.frame row.names must match one of the types of names in the *areaNamesAbbrsIDs* data.frame name table. If the data row does not match a value in the name table, an warning is issued and the data is ignored. If no data is present for a sub-area (county) in the name table, the sub-area is mapped but not colored.

The following are a list of the names, abbreviations, and ids for each country in the *MarylandBG* border group.

name	ab	id
Allegany	AL	24001
Anne Arundel	AA	24003
Baltimore	BL	24005
Baltimore City	BC	24510
Calvert	CV	24009
Caroline	CL	24011
Carroll	CR	24013
Cecil	CC	24015
Charles	CH	24017
Dorchester	DR	24019
Frederick	FR	24021
Garrett	GR	24023
Harford	HR	24025
Howard	HW	24027
Kent	KN	24029
Montgomery	MG	24031
Prince George's	PG	24033
Queen Anne's	QA	24035
St. Mary's	SM	24039
Somerset	SS	24037
Talbot	TB	24041
Washington	WH	24043
Wicomico	WC	24045
Worcester	WR	24047

There are no alternate abbreviations or regions assocated with counties in Maryland border group.

The *id* field value is the U. S. state and county FIPS code.

The 'rowNames' = "alias" or "alt_ab" and the 'regionB' and 'dataRegionsOnly' call parameters are not supported in the *MarylandBG* border group.

---

mdPopData	*Test data for the Maryland border Group*

Description

This dataset contains the county populations for 1970, 1980, 1990, 2000, and 2010 and the projected estimated populations for 2015, 2020, 2025, 2030, 2035 and 2040 for each Maryland county and Baltimore City.

Usage

```
data(mdPopData)
```

Format

A data frame with 24 rows, 1 for each county, with 12 variables for each county:

County a character vector containing the Maryland county name and Baltimore City.

X1970 a numeric vector of the county's population in 1970.

X1980 a numeric vector of the county's population in 1980.

X1990 a numeric vector of the county's population in 1990.

X2000 a numeric vector of the county's population in 2000.

X2010 a numeric vector of the county's population in 2010.

X2015 a numeric vector of the county's estimated population in 2015.

X2020 a numeric vector of the county's estimated population in 2020.

X2025 a numeric vector of the county's estimated population in 2025.

X2030 a numeric vector of the county's estimated population in 2030.

X2035 a numeric vector of the county's estimated population in 2035.

X2040 a numeric vector of the county's estimated population in 2000. .

Details

This dataset was pulled from the Maryland government website in January, 2015. It contains the population and estimated population for each of Maryland counties and Baltimore City for 1970, 1980, 1990, 2000, and 2010 and estimated populations for years 2015, 2020, 2025, 2030, 2035, and 2040.

---

messages *micromapST Generated Warning and Error Messages*

Description

micromapST verifies as much of its operation and the data provided by the user to try and identify and document to the user problems before they cause an R warning or error exception and throw a R like cryptic message. Each message, warning and error message is documented with the general form of the message and a friendly explanation and advice on what may be wrong and how to fix the issue identified.

The micromapST messages all start with "***" to help quickly find them in the warnings() logs and general output.

The general format of the messages is:

```
***XXXX NNNNNN text of message

  or

***XXXX NNNNNN CC text of message
```

where the XXXX is the message alphanumeric identifier, NNNNNN is the name of the function or glyph issuing the message and CC is the glyph graphic column number in the *panelDesc* structure.

The four alphanumeric message identifier following the "***" is a unique message identifer to help find the explanation in this document and to help discuss problems over the email more accurately. It's always best to include the log of the preparation and execution of micromapST as well as the data used when requesting help.

The first two digits/characters of the message identifier indicates which logical segment of the package generated the message:

01xx Main package initialization, startup, call argument validation, user data validation and *panelDesc* structure and contents validation.

02xx panelDesc and glyph parameters and data validation, processing and graphic generation

03xx Border Group data and structure validation and setup,

04xx Internal Operation

05xx Informational Messages

The second field of the message contains the name of the section of code generating the message. If a glyph generated the message, the name of the glyph (type=xxx) and the panel column number are displayed to help you focus on the parameters and data that may need to be inspected. Each message contains specific information (like *panelDesc* named list, checked value, etc.) to help identify the cause of the message and help lead the user to a fix.

The remainder of the message contain a text explanation of the issue and a list of parameters: glyph column number/name, data row or column, variable name, and value related to the issue.

Details

Conventions: The user provides the data.frames for the statsDFrame and panelDesc structures. In this document, there variables are represented by the generic terms of *statsDFrame* and *panelDesc*. In the messages produced by the package, these values are replaced with the names of the user variables provided in the micromapST call.

The following values are also substituted with the real run values at time of execution:

<sortVarCol> List of column names/numbers used in the 'sortVar' call parameter. Used to identify invalid *statsDFrame* column names and numbers used in the 'sortVar' call parameter.

<PDvarName> panelDesc variable name. The panelDesc variables include *lab1*, *lab2*, *lab3*, *lab4*, *refVal*, *refTxt*, *col1*, *col2*, *col3*, *type*, and *panelData*.

<PDcol> panelDesc glyph column number. This helps identify which graphic column (glyph) is being processing when the error/warning is signaled. The far left column number is 1.

<PDvarList? a list of variables from the panelDesc data.frame. Used in messages to identify one or more *panelDesc* variables that are invalid.

<detVar> detailed list variable name

<lastVarName> last variable name used. Used to locate a value or absents of a value in a list by using the last good variable name.

<bordGrp> a character string of the name of the border group being referenced. The border group names does not include the .RData or .rda extension used on the data file.

<detVarName> detailed list variable name

<method> The labeling method requested for generating Axis labels.

<list of unused sub-areas> A list of the sub-areas (counties, etc.) that did not have data.

<listRowNames> list of row.names (sub-areas)

<litrowNamesCol> The literal rowNamesCol value submitted by the caller.

<class(xxx)> The class of the variable names "xxx"

<number items> Number of items in list or vector.

<invalid name list> List of invalid names that were unexpected.

<list> a general list of options, variables, or names.

<statsDFrame> The name of the data.frame used as the first call parameter. This data.frame is used for the statsDFrame and provides the numeric values for the plots.

<panelDesc> The name of the data.frame serving as the panelDesc panel column description.

<gName> The name of the glyph being drawn.

<glyphName> The name of the glyhp begin drawn.

<sDFName> The name of statistics data.frame for the linked micromap.

<pdColNum> The panelDesc data.frame column number (glyph column).

<dupList> Is the list of duplicate entries in a list.

<rowNumbers> References the row number in the statsDFrame data.frame.

<list of rows> Is a list of rows related to the message.

<rowNumbers> One or more row numbers related to the message.

<NotUsedNames> Identifies a not used name from name table.

<future> Reserved for future usage.

The following is a listing of all micromapST generated messages and a description of possible causes and solutions.

01yx Initialization, Setup and Call argument validation, *panelDesc* and User data validation. where y indicates the specific argument or area.

    01Zx General Operation ("Z"):

        01Z0 CARG Key call arguments are missing, NULL, wrong type, or NA, Execution stopped.
        The micromapST function requires the *statsDFrame* and *panelDesc* data frames be provided. This is the minimum set of call parameters required. If either or both parameters are missing (the first two in the function call), the execution of the function is stopped.

        01Z1 CARG Errors found in call arguments. Execution stopped.
        Errors are found in the micromapST calling parameters/arugments. See the log to determine the problem and fix.

    010x *statsDFrame* **call argument (1st argument):**

        0101 CARG-DF First argument (*statsDFrame***) is missing or not a data.frame.**
        The first calling argument/parameter must be the *statsDFrame* data frame containing the data to be graphed. If the first argument/parameter is missing, NULL or NA or not a data frame, the function will stop. Provide the correctly built *statsDFrame* and re- run. ***

0102 CARG-DF The following sub-area(s) in the name table were not referenced in the user data> <not Used Sub-Areas> <
Each row in the *statsDFrame* data frame represents a sub-area (county, district, etc.) in a border group. The names of the sub-areas in the *statsDFrame* (either in a column or as the row.names) must match the names, abbreviations or IDs in the border group's name table. This error message lists the names in the *statsDFrame* that do not match the name table. Correct the sub-area name in the data frame and re-run micromapST. micromapST will attempt to use the data for matching rows, but the overall results are un- predicatable and may not provide the desired results. ***

0103 CARG-DF The *<sDFName>* **statsDFrame data.frame has no columns or rows.**
The <sDFName> data frame provided as the *<statsDFrame>* does not have any columns or rows. It's empty. Provide a valid *<statsDFrame>* and re-run the function. ***

0104 CARG-DF There are duplicate entries in the *statsDFrame* **data.frame. Duplicate entires are ignored.**
The *statsDFrame* data frame contains rows with duplicate names. Only the first row is used, duplicates are ignored. Check the sub-area name values in the data frame and correct as needed to make sure each row represents a unique sub-area. The following message provides a list of the duplicate sub-areas. ***

0105 CARG-DF The duplicate rows are: <listRowNames>
This message provides a list of the duplicate sub-area names, abbreviations or IDs in the *statsDFrame* data frame. Correct the duplicate entries and re-run micromapST. ***

0106 CARG-DF The following row in the *<sDFName>* **statsDFrame data.frame are not found in the name table:**
See continued message (0107) for details

0107 CARG-DF <listRowNames>
This is a two line error message. The sub-area names provided in *statsDFrame* data frame must match the names in the border group name table. This warning provides a list of the sub-araa names not found in the name table. Please select the correct border group or correct the sub-area names in the data frame and re-run micromapST. Unless requested, the function will stop.

0108 CARG-DF The rows not matched to boundaries will be removed and not mapped. ???? This warning is confirming the function call parmmeter 'ignoreNoMatch' is set to TRUE. The unmatch sub-areas reported in the previous warning (0106 and 0107) will be ignored and the data ignored.

0109 CARG-DF Data rows must all match entries in the name table to proceed. Call stopped.
If 'ignoreNoMatch' is FALSE and a data row does not have an entry in the boundary name table then the package can not continue to run. All data must have valid names or abbreviations associated with each row that has sub-area boundaries in the border group. Execution is stopped. Verify the sub-area identifiers in the supplied data, correct and retry.

011x panelDesc call argument (2nd argument):

0111 CARG-PD The second call argument, the panelDesc structure, is missing or not a data.frame or list.
The second call argument/parameter is the *panelDesc* data frame. Without this data frame, micromapST does not know which glyphs to draw or where the supporting

data is located in the *statsDFrame* data frame. The data frame is missing, NULL, NA or is not a data frame structure. Correct and re-run micromapST.

0113 CARG-PD The following named lists in the *<pDName>* **panelDesc data.frame are not valid:** *<PDVarList>*

The *panelDesc* variables *col1*, *col2*, and *col3* identify the columns in the *<sDF-Name>* data frame that contain data for each sub-area to be used by the glyph for the column. The column number or name provided in the *panelDesc* structure is not a valid *statsDFrame* column number (out of range) or name. This error message provides a list of the invalid column numbers or names.

0114 CARG-PD The required *type* **named list is missing in the** *<pDName>* **panelDesc data.frame.**

The *panelDesc* list named *type* does not exist in the *panelDesc* data frame. This named list is required to identify the type of glyph to be generated for the specific column. Inspect the *panelDesc* data frame, check for spelling and correct the source of the problem and re-run.

Example: *panelDesc* = list(type = c("map","dot","bar"), ...)

0115 CARG-PD The *<panelDesc>* *type* **named list contains one or more invalid glyph name(s):** *<pdTypeList>***.**

The type named list in the *<panelDesc>* panelDesc data frame contains invalid glyph names. The invalid names are listed in this error message. The valid glyph names are: map, mapcum, mapmedian, maptail, id, arrow, bar, boxplot, dot, dotse, dotconf, dotsignif, segbar, normbar, ctrbar, ts, tsconf, and scatdot.

012x call argument column name/number verification:

When a call argument/parameter contains a column name or number referencing the *<statsDFrame>* data.frame, the arguments values are checked during the initialization of the micromapST processing. The following errors may be detected and the function stopped.

The call arguments/parameters that contain column names/numbers are:

```
        rowNamesCol     RNC
        sortVar         SORT
```

The abbreviated name (*<callArgAbbr>*) listed above for the call argument/parameter ('<callArg>') is included in the message to assist in identifying and correcting the problem.

0120 CARG-'<callArgAbbr>' A column index number in the '<callArg>' call argument is negative or zero: *<value>*

The '<callArg>' value(s) are numeric and must be a positive number greater than 0 to reference a *<statsDFrame>* column. The bad value is displayed in the *<value>* part of the message. Check the column index number provided and re-run.

0121 CARG-'<callArgAbbr>' A column index number in the '<callArg>' call argument is greater than the number of columns in *<statsDFrame>* **data.frame:** *<value>*

The '<callArg>' value(s) are numeric and must be in a range from 1 to *n*, where *n* is the number of columns in the *<statsDFrame>* data.frame. The bad value is displayed in n the *<value>* part of the message. Check the column index number(s) and re-run.

0122 CARG-'<callArgAbbr>' The call argument/parameter, '<callArg>', is not a valid variable type. It must be a numeric or character type value.

The '<callArg>' value(s) must be a numeric vector (column numbers) or a character vector (column names and/or numbers). Logical, function, list, data.frame, etc. are not acceptable. The arugment should be coded as:

                    sortVar=c("rates","pop")   or   rowNamesCom=c(2)

Column names and numbers can be intermixed in the vector. Correct the call argument and re-run.

0123 CARG-'<callArgAbbr>' A column name in the '<callArg>' call argument does not exist in the *<sDFName>* data.frame: *<value>*

One or more of the column names in the '<callArg>' call argument/parameter does not exist in the *<statsDFrame>* data.frame. The bad value is displayed in the *<value>* part of the message. Verify that the column name exists in the *<statsDFrame>* data.frame and re-run.

0124 CARG-'<callArgAbbr>' The '<callArg>' call argument is empty. Argument ignored.

The '<callArg>' value was found to have a length of zero - empty. The argument can not be checked and is ignored.

0125 CARG-'<callArgAbbr>' A column index number in the '<callArg>' call argument did not convert from character to integer: *<value>*

The '<callArg>' value was validated as a numeric, but R was not able to convert it from a character string into an integer. Check the value and re-run.

015x 'bordGrp' and 'bordDir' parameter:

0150 BGBD When the 'bordDir' parameter is set to NULL, the 'bordGrp' must be one contain in the package. *<backageName>*

When the 'bordDir' call parameter is not provied, or set to NULL or NA, the value of the 'bordGrp' parameter must be one of the internal border groups provided with the package. The border groups provided with the package are:

USStatesBG U.S. States and the District of Columbia

USSeerBG The 18 U.S. Seer Registry areas

KansasBG The counties of the state of Kansas

MarylandBG The counties of the state of Maryland

NewYorkBG The counties of the state of New York

UtahBG The counties of the state of Utah

UKIrelandBG The provinces, counties and cities of the United Kingdom and Ireland

ChinaBG The provinces, special administration areas, metropolitan cities of China

SeuolSKBG The districts of the South Korean city of Seoul

If you are using a custom or private border group, then the 'bordDir' parameter must be provide to specify the file directory containing the border group .rda file.

See the appropriate documentation section for details on each border group contained in this package.

0151 BGBD After loading <bordGrp> border group data set, the following objects are missing: <MissBGObjects>

The border group has been located and loaded into micromapST. However, the <Miss-BGObjexts> are missing and the border group can not be used. The border group must be rebuilt to include the missing objects.

0153 BGBD The directory specified in the 'bordDir' argument/parameter does not exist. Value=<*bordDir*>

The path specified does not exist, Processing is stopped. Correct the path name and re-run. Check to make sure the slashes in the path name are correct and that the path exists.

0154 BGBD The 'bordGrp' filename must have an ".rda" or ".RData" file extension.

When private border group files are created, they must be saved as ".rda" or ".RData" type "R" files. Please re-create your private border group file and re-run. Execution is stopped.

0155 BGBD The 'bordGrp' file in the 'bordDir' directory does not exist.

The package attempted to open the 'bordGrp' file in the 'bordDir' directory, but the file does not exist. If the 'bordGrp' filename was specified without a file extension, the package appended an ".rda" file extension. Verify the 'bordGrp' filename exists in the 'bordDir' directory and re-run. Execution is stopped..

0156 BGBD The 'bordGrp' parameter has not been specified and is required when the *bordDir* is provided.

When the 'bordDir' call argument/parameter is specified, the package will load a users or private border group into the package. Therefore, the 'bordGrp' call argument/parameter MUST be specified to identify the file in the directory to load. Verify the 'bordGrp' filename is present and exists in the 'bordDir' directory and re-run. Execution was stopped.

0157 BGBD System error encountered when loading the border group. See error message:

The package attempted to load the user/private border group. The R system load() function failed and returned an error message. The error message is logged following this message using the same message indicator. Inspect the file and resolve the R system load() error mesage and re-run. Execution was stopped.

??? Second line to message. ???

0158 BGBD In the areaParms data.frame the Map.L2Borders has been set to FALSE, because no L2 level boundaries are available.

The border groups areaParms Map.L2Borders variable is set to TRUE to request the overprinting of level 2 boundaries on the maps. However, there is no L2VisBorders data.frame in the border group. The variable has been set to FALSE and no level 2 boundaries will be drawn.

0159 BGBD The areaParms variable aP_Regions is TRUE, but no regional boundaries exist in border group. Regions feature disabled.

This warning does not impact the creation of linked micromaps. If the caller wants to use the regions feature, then the border group must be corrected to provide the appropriate regional boundary information in the RegVisBorders data.frame in the border group.

017x 'rowNamesCol' call argument:

To be able to link up the *statsDFrame* rows to sub-areas in the border group and its boundary information, the *statsDFrame* must provide a column containing the sub-area names, abbreviations or IDs that match the name table entries in the border group. The values can be supplied via the row.names on the data frame or in a column in the data frame. The 'rowNamesCol' call parameter allows the user to specify which *statsDFrame* column

to use as the link to the border group name table. The rowNames argument/parameter is used to specify if the link is the name, abbreviation or ID of the sub-area.

0171 CARG-RNC The row names in column <rowNamesCol> of the *<statsDFrame>* **data frame contain duplicates. Only one row per sub-area is permitted. Duplicate rows are: <listRowNames>.**

The *statsDFrame* data frame contains rows with the same sub-area names. Only one row is allowed per sub-area. This error message contains a list of sub-area names that are duplicated. The execution is stopped. Correct the sub-area names associated with the rows involved and re-run.

Note: a possible later work-a-round is to have the package delete duplicate rows from the data.frame and continue.

0172 CARG-RNC The 'rowNamesCol' argument value must be a character or numeric value. It was found to be: <class('rowNamesCol')>.

Value provided in the 'rowNamesCol' must ba a character string (column name) or a numeric/integer (column number). Any other type of value is not acceptable and reported in this error message. Execution is stopped since the sub-area names in the *<statsDFrame>* data.frame cannot be located.

0173 CARG-RNC The 'rowNamesCol' argument value must have a length = 1. Only first value used.

The 'rowNamesCol' value must have a length of 1 and should be a simple vector. Only one column can be specified to contain the sub-area names (row names.) If more then one value is provided, only the first value in the data will be used.

018x 'sortVar' and 'ascend' call arguments:

The 'sortVar' calling argument/parameter specifies the *statsDFrame* column numbers and/or names to be used to sort the data and the rows presented in the linked micromaps. The sorting is executed in the order of the columns listed in the numeric or character vector provided via the 'sortVar' argument/parameter. The column numbers and names are verified before the sorting is attempted. All references must be valid. The default sort uses the sub-area names as spedified in the 'rowNames' argument/parameter.

The 'ascend' flag indicates which direction the sorting will be done. 'ascend' = TRUE requests ascending order. 'ascend' = FALSE requests descending order.

0186 CARG-AS The 'ascend' parameter is not a logical variable. Must be TRUE or FALSE.

The 'ascend' call argument/parameter must be a logical vector with a length of 1. Vectors with multiple values are not allowed. Numeric, Integer or Character vectors are not supported.

019x 'rowNames' call argument and 'alias' name matching feature:

The 'rowNames' call argument/parameter specifies what type of sub-area name/string should be used to match the sub-area name in the *statsDFrame* data frame, boxplot and time series data with the border group's name table. The valid rowNames values are: 'full', 'ab', 'alt_ab', 'id', and 'alias'. The default is 'ab'.

When 'alias' is specified, the package attempt to link the *<statsDFrame>* data rows to the boundary data using a partial match between the sub-area names in the *<statsDFrame>* data.frame and the *alias* column in the name table. This feature was implemented to handle special cases were the data source does not use the regularly used names or abbreviations for the sub-areas. See the discussion on the USSeerBG border group for more details and an example. At this time, only the USSeerBG border group makes use of this feature.

0190 CARG-RN Invalid '`rowNames`' argument value. The value must be '`ab`', '`alt_ab`', '`id`', '`alias`', or '`full`'. The default of '`ab`' will be used.

The '`rowNames`' call argument/parameter contains an invalid value. The default of '`ab`' is used.

0191 CARG-RN rowNames='`alias`' is not supported for this bordGrp. The default of '`ab`' will be used.

The rowNames value of '`alias`' is only supported in border groups containing the an alias column in the name table and have the border group areaParm variable *enableAlias* set to TRUE. The only border group supplied with the package that supports the alias name matching is the the USSeerBG border group.

0192 CARG-RN rowNames='`seer`' is only supported for the USSeerBG bordGrp. The default of '`ab`' will be used.

(Delete message replace with 0191 CARG-RN.)

0195 ALIAS Alias Names(s) in the data no not match the name table for the area. The unmatched data rows are: <listRowNames>

The rowNames value of '`alias`' was specified. During partial wildcard matching of the "alias" column in the border group name table with the data column in the *statsDFrame* data frame specified by the '`rowNamesCol`' call argument/parameter, one or more data rows did not successfully match the name table. The <listRowNames> part of the message lists the data row values that did not match. Review the *statsDFrame* values used for sub-area names and the documentation of the alias strings provided in the selected border group and make adjustments as required.

0196 ALIAS Sub-area names in the data have duplicate name in rows: <listRowNames> Only one row per sub-area is permitted.

There are multiple rows in the *statsDFrame* that match a single sub-area in the border group name table. Only the first matching data row will be used. The <listRowNames> section of the message contains a list of the duplicate rows. Correct row names to eliminate any duplicates and re-run.

01Ax '`title`' call argument:

The '`title`' calling argument/parameter is used to specify a one or two line title for the linked micromap. If not '`title`' is provided, the titles are left empty - blank.

01A0 CARG-TL The '`title`' argument contains more than 2 items. Only the first two will be used.

The '`title`' call argument/parameter contains more that 2 character strings. Only two title lines are supported. Only the first two values (lines) are used.

01A1 CARG-TL The typeof/class of the '`title`' parameter is not character. Only character vectors are supported. The '`title`' argument is ignored.

The type and class of the '`title`' call argument/parameter must be a character vector. Numeric, integer and logical vectors are not supported.

01A2 CARG-TL The '`title`' argument/parameter is empty. Recommend providing a title for the linked micromap.

The '`title`' call argument/parameter is empty (length of 0). It's recommend a set of titles be specified for the linked micromap. Please provide a '`title`' character vector to help identify the linked micromap and re-run.

01Bx '`plotNames`' call argument:

The '`plotNames`' calling argument/parameter specifies the type of sub-area name to be used in the ID glyph. The allowed values are '`ab`' and '`full`'.

01B0 CARG-PL Invalid 'plotNames' argument value. The value must be 'ab' or 'full'.

The value of the 'plotNames' call argument/parameter is not valid, The value must be 'ab' or 'full'. The default value of 'ab' is used. Check the spelling and re-run. ***check***

01Cx 'grpPattern' call argument:

The package creates a pattern of how many sub-area rows will be placed in each glyph group row based on the total number of sub-area with data. The user can override this generated pattern by providing their own pattern in the form of an integer vector. Each entry represents one glyph group row. The maximum value for an entry is 5. The values must be in decending order toward the median sub-area and must sum to the number of sub-areas in the data. For example: grpPattern = c(5,5,4,3,4,5,5) for 31 sub-areas. If these rules are not followed, the following messages are generated and the package created pattern is used.

01C0 CARG-GP The 'grpPattern' call parameter is invalid. It must be an integer vector.

The vector provided must be an numeric or integer vector. The package created pattern is used. Correct and re-run.

01C1 CARG-GP The 'grpPattern' call parameter is not valid. The total of the rows per group must equal the number of rows in the <*statsDFrame*> data.frame

The sum of the sub-area rows in the grpPattern vector is not equal to the number of sub-areas present in the *statsDFrame* data.frame. For example, if the <*statsDFrame*> data.frame has 51 rows after duplicates are removed, then the sum of the grpPattern vector must be 51. A grpPattern of c(5,5,5,5,5,1,5,5,5,5,5) would be valid, but c(4,4,4,4,4,1,4,4,4,4,4) would be invalid. The package will use a calculated pattern. Correct and re-run.

01C2 CARG-GP The 'grpPattern' call parameter contains a value > 5 rows per group.

Each element in the grpPattern vector represents the number of sub-area rows per glyph group/row. The maximum number of sub-area per group is 5. The package will use a calculated pattern. Correct and re-run.

01C3 CARG-GP The 'grpPattern' call parameter is not properly ordered. The number of rows per group must be in desending order to the median sub-area.

The number of rows per glyph group must be arranged in descending order from the ends to the median sub-area row in the middle of the list. For example: grgPattern = c(5,5,4,3,4,5,5). A grpPattern of c(3,4,5,5,4,3) is acceptable. The package will created and use a calculated pattern. Correct and re-run.

01Dx 'axisScale' call argument:

The 'axisScale' calling argument/parameter is a single value character vector that specifies the type of axis scaling the micromapST should use on the glyphs. The acceptable values are:

o original axis labeling provided by the pretty function. Scaling is done in the same way as previous releases of micromapST.

e labeling is done using the extended algorithm from the labeling package.

s labeling is done using the extended algorithm and label scaling by applying a multiplier of hundreds, thousands, etc and a subtitle used to identify the multiplier units.

```
Example:
```

```
        before:     1000  2000  3000  4000

        after:            in hundreds
                      1     2     3     4
```

sn labeling is done using the extended algorithm and each label is scaled by applying a multiplier of hundreds, thousands, etc. and a character identifying the scaling units postpended to the label.

```
        Example:
          before:    0    5000  10000  15000  20000

          after:     0     5K    10K    15K    20K
```

'axisScale' must be a character vector with a single item.

01D0 CARG-SC The 'axisScale' argument is invalid, 'axisScale' can only be set to 'o', 'e', 's', or 'sn'. The default value of 'e' will be used.
The 'axisScale' call argument/parameter does not contain a valid value. It must be 'o', 'e', 's', or 'sn'. The default of 'e' will be used. ***check***

01D1 CARG-SC The 'axisScale' argument is not a character vector of length 1.
The 'axisScale' call argument/parameter must be a character vector containing only one value. Numerical, integer and logical vectors are not supported. Only simple vector structures are support. Correct the value assigned with 'axisScale' and re-sun. ***check***

01Ex 'staggerLab' call argument:

01E0 CARG-SL The 'staggerLab' argument is not a logical value. Setting 'staggerLab' to FALSE.
The 'staggerLab' parameter must be a TRUE or FALSE value. If it is not a logical variable, the call parameter is ignored and the default is used. ***check***

01Fx 'colSize' call argument:
This feature is not implemented in the version. The argument name is reserved for future use and development.

01F0??? CARG-CS The length of the 'colSize' argument must be the same as the number of columns in the *panelDesc* data.frame.
The number of elements in the 'colSize' vector must be equal to the number of glyph graphic columns specified in the *panelDesc* data.frame. The 'colSize' argument/parameter is ignored. ***check***

01F1 CARG-CS In the 'colSize' parameter in the *panelDesc* data.frame has values for fixed width glyph columns. Values will be set to the mean value for *colSize*.
The calculated and r******equested ('colSize') columns width require *<reqWidth>* inches to build the linked micromap, only *<availWidth>* inches are available in the plotting area. Increase the size of the plotting area to the required width plus 5 ***check***

01F2 CARG-CS The edited 'colSize' parameter in the *panelDesc* data.frame has at least one invalid or missing value. Values changed to mean of valid 'colSize' values: <mean>
The calculated a***************nd requested ('colSize') columns width require *<reqWidth>* inches to build the linked micromap, only *<availWidth>* inches are

available in the plotting area. Increase the size of the plotting area to the required width plus 5 ***check***

01F4 CARG-CS The '`colSize`' parameter in the *panelDesc* data.frame contains non-numeric entries: <list of entries> Entry(s) set to mean.

The calculat**************ed and requested ('`colSize`') columns width require *<reqWidth>* inches to build the linked micromap, only *<availWidth>*. Entry(s) set to mean of colSize values. ***check***

01F5 CARG-CS The '`colSize`' entries in *panelDesc* data.frame are out of range (<= 0 or > 200). Values: <list of bad values>. Values set to mean.

The calculated and requested ('`colSize`') columns width require *<reqWidth>* inches to build the linked micromap, only *<availWidth>* inches are available in the plotting area. Increase the size of the plotting area to the required width plus 5 ***check***

01F6 CARG-CS The 'colSize' parameter in <pDName> contains no useful information and will be ignored.

The calculated and requested ('`colSize`') columns width require *<reqWidth>* inches to build the linked micromap, only *<availWidth>* inches are available in the plotting area. Increase the size of the plotting area to the required width plus 5 ***check***

01Kx '`colors`' call argument:

01K0 COLORS An invalid single value is provided for the '`colors`' argument. It must be 'BW', 'greys', or 'grays'. The argument is ignored.

In the micromapST function call, a single value was provided for the '`colors`' parameter. The value must be *BW*, *greys*, or *grays*. If not, the parameter is ignored.

01K1 COLORS The colors vector has the incorrect number of items. It must have 1 or 24 items. <number items> provided.

In the micromapST function call, a single value or a vector 24 values can be supplied. If a different number of values are provided, the parameter is ignored and the default colors are used.

01K2 COLORS The colors vector type is invalid. It must be a character vector.

In the micromapST function call, the '`colors`' parameter contains non-character values. The '`colors`' contained in the vector must be character strings identifying the colors to be used for the 24 color parameters used by micromapST. Numerical values related to the current color palette are not acceptable.

01K3??? COLORS The colors vector contains invalid color names. <invalid name list>

The color names supplied in the vector must be character string and must match the color names returned by the colors() function or a valid hex color value in R. FUTURE ***check***

01Nx '`details`' call argument

The '`details`' call argument/parameter contains a named list of micromapST parameters values the user wants to override or change. Most of the parameters relate to minor functional variations in how the glyphs are drawn. Several other control the details on how the graphic page is layed out and should not be changed under normal situations. Changes of these variable can cause unpredictable results.

Future: The *panelDesc* data frame has been extended to support glyph parameter modifications on a per column per glyph basis and is the preferred way of working with the glyphs variables. Any changes made via the '`details`' parameter are global changes and will effect all fo the glyphs.

01N0 DETS The <tag> does not have a valid value: <value> Check type <method> used.

The ??????? 'details' call argument/parameter must be a list structure. If it is not, 'details' value is ignored.

01N1 DETS The 'details' parameter is not a list.

The 'details' call argument/parameter must be a list structure. If it is not, 'details' value is ignored.

01N2 DETS Invalid variable name: <detVar> in the 'details' list. The variable is IGNORED.

The named list provided by the 'details' argument/parameter must have named elements that match the defined list of variable for the package. (See micromap-STSetDefaults for details.) If the name in the list does not match, the variable is ignored. This message contains a list of the all ignored variables in the 'details' call argument/parameter.

01N3 DETS Zero length variable name found in the 'details' list after the *<last-VarName>* **variable.**

This error frequently occurs when the 'details' parameter list contain two commas and no name/value. The position of the two commas is preceded by the <lastVar-Name> variable in the list. The empty variable/list will be ignored, but please remove the extra comma for future runs.

01NA DETS The *Id.Dot.pch* **variable is can only be set to a value in the range from 1 to 25. Using the default of 22.**

The *IdDotpch* variable is the symbol code for the character preceeding the labels in the id glyph. Only pch values of 1 to 25 are supported.. The variable is ignored and the default character value of 22 is used.

Over time each more 'details' variable will be checked and additional error message will be provided.

*** What other 'details' variable should be validated?

The *panelDesc* is being enhanced to allow appropriate variables to be defined on a glyph graphic column basis. As this feature becomes available the documentation will be updated.

02xx panelDesc and glyphs data messages:

As micromapST setups to create the linked micromaps, each glyphs preforms additional validate on the *panelDesc* variables and the associated *statsDFrame* data.

The message identifier (4 alphanumeric) in the message is followed by the glyph name (type) and the *panelDesc* panel column number (1 through "N"). This helps the user quickly identify which glyphs column in the *panelDesc* definition is related to and therefore the data in the *statsDFrame*. The text of the message with specific information to pin point the problem.

For the map, mapcum, mapmedian, maptail and id glyphs there are no additional data from the user and no additional validation is required.

For the arrow, bar, dot, dotsignif, dotse, dotconf, ctrbar, norbar, segbar, and scatdot glyphs uses data in the *statsDFrame* data.frame and use the *col1*, *col2*, and *col3* named lists for the panel column to identify the *statsDFrame* columns with the required data. The *col1*, *col2*, and *col3* pointers are validated and the data in the *statsDFrame* data.frame is inspected based on the type of glyph requested. The messages use the 3rd digit to identify which *panelDesc* variable was validated. 1 = *col1*, 2 = *col2*, 3 = *col3*. The name of the *panelDesc* named list is also included in the message.

For the segmented stacked bar glyphs (CTRBAR, SEGBAR, NORBAR), *col1* and *col2* specify the first and last *statsDFrame* column of the contiguous set of columns to use for the data for the segment values/sizes. The message related to the inspection of these columns use the 3rd digit of the message number to indicate the relative column number, 1 being the first column of data up to a maximum of 9 for the last column of data. The column name is also included in the messages to help.

The following is a summary of the usage of the *col1*, *col2*, and *col3* variables by each glyph:

Variable Usage by glyph:

glyph		col1	col2	col3
dot	=	dotvalue	-	-
bar	=	bar length/value	-	-
arrow	=	start value	end value	-
dotsignif	=	dot value	p_value	-
dotse	=	dot value	standard error	-
scatdot	=	x coordinates	y coordinates	-
ctrbar	=	first data column in set	last data column in set	-
normbar	=	first data column in set	last data column in set	-
segbar	=	first data column in set	last data column in set	-
dotconf	=	dot value	lower confidence	upper confidence

The *col1*, *col2*, and *col3* variables are not used by the map, mapcum, maptail, mapmedian, id, boxplot, ts, and tsconf glyphs.

The 3rd character in the message identifier *panelDesc* data column (varcol1, var(col2), or var(col3)) pointer related to the warning message. The glyph graphic column number associated with the warning is provided in the <cc> text of the message.

02yx *panelDesc* **parameter checks for** *panelDesc* **variables:**

This top level checks for the values provided in the *col1*, *col2*, *col3* variables are done by the internal function CheckColx and include:

a) If does not exist, create dummy vector to for variable

b) Verify type of vector is numeric, integer, or character. All other types not permitted.

c) Fill empty string items with "NA" where appropriate.

The rest of the checking is done in the glyphs based on which variables needed.

*** Note, the use of <PDvar> appears to be a duplication of the *xx* variable in the message, is this true? The following warnings relate to the *panelDesc* variables in general:

0200????? PDCOL The the <*pdVarName*> **named list in the** <*panelDesc*> **structure has a type of** <*pdVarName-Type*>**. It must have a type of numeric, integer or character.**

The <*panelDesc*> variable indicated by <*pdVarName*> contains one or more negative numbers. The indexes to the <*statsDFrame*> data frame, but be positive from 1 up to the number of columns in the data frame. Correct and re-run. **checked** ***check***

0201 PDCOL In the <*pdVarName*> **named list in the** <*panelDesc*> **panelDesc structure there are one or more negative values.**

The <glyph> determined the <*statsDFrame*> column number of <*PDColOrig*> in <*PDVar*> for glyph column *cc* has a zero or negative value. It must be a positive

integer no greater than the number of columns in the user provided *<statsDFrame>* data.frame. ***check***

0202 PDCOL One or more of the values in the *<pdVarName>* **named list in the** *<panelDesc>* **structure is greater than the columns in the** *<statsDFrame>* **data.frame:** *<valList>*

Some of the column numbers in the *<pdVarName>* named list in the *<panelDesc>* panelDesc structure have values greater than the number of columns in the *<statsDFrame>* data frame. The <glyph> determined the *<statsDFrame>* column number of *<PDColOrig>* in *<PDVar>* for glyph column "cc" The *col1*, *col2*, and *col3* panelDesc variables (named lists) are used to point to the data columns in the *<statsDFrame>* data.frame. When provided as numbers, they must be in the range from 1 to the maximum number of columns in the *<statsDFrame>* data.frame.

0205 <gNameList> The length of the glyph type list is different the length of the variables list.

<gNameList> ... Some of the column numbers in the *<pdVarName>* named list in the *<panelDesc>* panelDesc structure have ?????????tsDFrame> data.frame. When provided as numbers, they must be in the range from 1 to the maximum number of columns in the *<statsDFrame>* data.frame.

020A   <gName> <pdColNum> The first column name/number (<stColName1>) must proceed the last column name/number (<stColName2>) in the <sDFName> data frame.

<gName> .....Some of the column numbers in the *<pdVarName>* named list in the *<panelDesc>* panelDesc structure have ?????????tsDFrame> data.frame. When provided as numbers, they must be in the range from 1 to the maximum number of columns in the *<statsDFrame>* data.frame.

020B   <gName> <pdColNum> The first column name/number (<stColName1>) must proceed the last column name/number (<stColName2>) in the <sDFName> data frame.

<gName> ....Some of the column numbers in the *<pdVarName>* named list in the *<panelDesc>* panelDesc structure have ?????????tsDFrame> data.frame. When provided as numbers, they must be in the range from 1 to the maximum number of columns in the *<statsDFrame>* data.frame.

02y1xx   <gName> <pdColNum> The number of segments is <value>. It must be between 2 and 9. If over 9, only the first 9 will be used.

(DUPLICATE-KILL) The <<glyph> in *panelDesc* column "y" determined the data type of the var<PDvar> to be invalid. The *<glyph>* routine expected the value to be an integer. Correct and re-run. DUPLICATE OF 02y0... ***check***

02y2 PDCOL <glyph> cc The column number of <PDColOrig> in <PDVar> is negative or zero. Must be a positive integer or the name of a column in *statsDFrame***.**

Duplicate of 02y1? But better (keep). The <glyph> determined the *<statsDFrame>* column number of *<PDColOrig>* in *<PDVar>* for glyph column "cc" has a zero or negative value. It must be a positive integer no greater than the number of columns in the user provided *<statsDFrame>* data.frame. ***check***

02y3 PDCOL <glyph> cc The column number of *<PDColOrig>* **in** *<PDVar>* **is greater than the number of columns in the** *<statsDFrame>* **data.frame.**

Duplicate of 02y1? But better (keep). The <glyph> determined the *<statsDFrame>* column number of *<PDColOrig>* in *<PDVar>* *panelDesc* variable for glyph column

cc is out of range. Its value is greater than the number of columns in the user provided *statsDFrame* data.frame. ***check***

It appears the "y" is the col"y" indicator and "cc" is the glyph column number. If so, the "y" is redundent.

The following warning deal with the column name or number specified in the *panelDesc* data.frame variables *col1*, *col2*, and *col3*:

02y4 \<glyph\> cc No *\<statsDFrame\>* **column was specified in** *\<PDVarName\>* **in the** *\<panelDesc\>* **panelDesc data.frame. A data column name/number is required.** *\<usage\>*

The \<glyph\> function for column *\<cc\>* requires a valid *\<statsDFrame\>* data.frame column be specified in the *panelDesc \<PDVarName\>* variable. The value is missing. Correct the *\<PDVarName\>* for column *cc* and re-run.

02y5 \<glyph\> cc The required panelDeac variable *\<pdVarName\>* **is missing from the** *\<pdDFname\>* **data frame.** *\<usage\>*

When using the "\<glyph\>" in glyph column "cc", requires the *\<pdVarName\>* variable to locate the data in the *\<statsDFrame\>* data frame. This is general related to the *col1*, *col2* and *col3* variable lists in the *panelDesc* data frame. Check the glyph descriptions and make sure all of the panelDesc variables required for the glyph are provided.

02y6 \<glyph\> cc The specified column name or number in *\<pdVarName\>* **panelDesc variable (\<stColName\>) does not exist in** *\<statsDFrame\>* **data frame or is out of range.** *\<usage\>*

The \<glyph\> determined the *statsDFrame* column name or number provided in the *\<pdVarName\>* does not exist or is out of range. If a column name was provided, it did not match any column names in the *\<statsDFrame\>* data frame. If a column number was provided, it has a value < 0 or greater than the number of columns in the *\<statsDFrame\>* data frame. Verify the column name or number in the panelDesc column variable and re-run.

02y7 \<glyph\> cc The data provided in the *\<stColName\>* **column of the** *\<statsDFrame\>* **data frame contains one or more non-numeric entries.** *\<usage\>*

For the glyph \<glyph\> in panel column *cc*, the data provided in the *\<stColName\>* are character vectors and one or more values contain non-numeric characters. This may result in an improper translation to numerics. The entries are treated as NA values. Inspect the data in column *\<stColName\>* and correct.

02y8 \<glyph\> cc The data provided in the *\<stColName\>* **column in the** *\<statsDFrame\>* **data frame contains one or more entries that could not be converted to numeric values.** *\<usage\>*

For the glyph *\<glyph\>* in column *cc*, an attempt was made to convert the character data in the *\<stColName\>* column of the *\<statsDFrame\>* to numeric values. One or more entries did not convert correctly resulting in missing values. Inspect the data in the *\<stColName\>* column of the *\<statsDFrame\>* data frame and correct.

02y9 \<glyph\> cc The *\<stColName\>* **data column in** *\<statsDFrame\>* **data frame is not a character or numeric vector.** *\<usage\>*

The *\<stColName\>* data column in the *\<statsDFrame\>* is not a character or numeric column. The glyph requires numerical values to generate the glyph. Inspect the data and correct. Logical. List and Matrix type data is not supported. If the column is character vectors, the character vectors must only contains numerical images and

must be able to be converted to numerics to draw the glyph. Inspect the data column and correct.

02yA <glyph> cc The *<stColName>* **data column in** *<statsDFrame>* **data frame does not contain any numerical data. No rows will be drawn.** *<usage>*

The *<statsDFrame>* data frame column *<stColName>* does not contain ANY numerical data to permit drawing of the requested glyph. Inspect the data column and re-run.

02yB <glyph> cc The *<pdVarName>* **data column in** *<statsDFrame>* **data frame contains all missing values (NA) and cannot be graphed.** *<usage>*

In the <glyph> for column *cc*, the data column in the user provided *statsDFrame* pointed to by the *<PDVarName>* panelDesc variable contains a *NA* value. The glyph for that data row will not be drawn. If this is an error, correct the data in the *statsDFrame* data.frame and re-run. ***check***

02yD <glyph> cc The *<stColName>* **data column in** *<statsDFrame>* **data frame does not contain any data. Data vector has length of zero.** *<usage>*

The *<statsDFrame>* data frame column *<stColName>* has a length of zero. If the *<statsDFrame>* is properly constructed with at least one sub-area, then this should not happen. If this error occurs, check the *<statsDFrame>* to make sure it has not been corrupted and has rows for each sub-area being graphed and mapped.

The following warning related to the data pointed to by the *panelDesc col1*, *col2*, and *col3* variable in the *statsDFrame* data.frame. The "y" indicates which panelDesc col (1, 2, or 3) is being referenced.

02yC <glyph> cc The *<stColName>* **column in** *<statsDFrame>* **data frame contains one or more missing values.** *<usage>*

Description needed. **checked** arrow, bar, dot, dotsignif, dotconf, ***check***

The following warnings are specific to the dot significance glyph:

022P <glyph> cc One or more P_Value data entries in *<stColName>* **are missing.**

The data in the *<statsDFrame>* pointed to for use by the dotsignif glyph (varcol2) for the significance P Value is not in the range of 0 to 1 probability. Review the data and correct the values to a range of 0 to 1. This is only used by the DOTSIGNIF glyph.

022Q <glyph> cc One or more P_Value data entries in *<stColName>* **are out of the range.**

The data in the *<statsDFrame>* pointed to for use by the dotsignif glyph (varcol2) for the significance P Value is not in the range of 0 to 1 probability. Review the data and correct the values to a range of 0 to 1. This is only used by the DOTSIGNIF glyph.

The following special case that do not reference a particular panelDesc column variable. These are generated by the stacked bar glyphs with regard to the *<statsDFrame>* data and the panelDesc col1 and col2 specifications.:

020A <glyph> cc The first column name/number (<stColName>**) must proceed the last column name/number (**stColName>**) in the** *<statsDFrame>* **data frame.**

In segbar, normbar and ctrbar stacked bar glyphs, the user provides the location of the first and last data column in the *<statsDFrame>* data frame. Each data column between the first and last columns are used to create the stacked bar glyph. The processing is done from lower column number to higher column number. Therefore, the first column must preceed the last column in the data frame. That is must have a lower column number or a column name that preceeds the last column name in the data frame.

020B \<glyph> cc The number of segments is \<numSegs>. It must be between 2 and 9. If over 9, only the first 9 will be used.
Stacked bar glyphs use 2 to 9 data columns from the *statsDFrame* data.frame to create segmented stacked bar graphs (CTRBAR, SEGBAR, NORBAR). Each data column represents the length of one segment and is validated for the glyph prior to use. If only 1 segment is specified, no stacked bar glyph is drawn. If more than 9 segments exists, only the first 9 will be used in the segbar, normbar and ctrbar glyph.

020C \<glyph> cc The number of segments is \<numSegs> At least 2 data columns must be defined.
****Duplicate of 020B Used by segbar, normbar, ctrbar glyphs. ***check***

020D \<glyph> cc The data provided cannot be centered around the value of \<center value>.
The ctrbar glyph is designed to center the stacked bars around the value of zero. The data provided is all higher than or lower than zero. A future feature is planned to allow the center value for the ctrbar glyphs to be specified. Until then the center value is set to 0 (zero). Used by ctrbar glyphs. ***check***

BoxPlot specific processing messages (boxplot data list):

02B1 BOXPLOT cc The *panelData* value of (\<pDataValue> in the \<panelData> data frame does not exist or is not accessible.
The *panelData* variable in the \<panelDesc> data frame is used to pass completed data structures to glyphs that cannot be contained in the \<statsDFrame> data frame. In this case, the variable name provided \<pDataValue> does not exist or is not accessible by the package. Make sure the name is spelled correctly and exists in the .GlobalEnv (calling) environment. This message is used by the BOXPLOT, TS and TSCONF glyphs. **checked** boxplot,

02B3 BOXPLOT cc The \<pdDataName> data for the boxplot is not is list.
Must be a list of boxplot sets of data - one per sub-area.
Needs description

02B4 BOXPLOT cc The \<pdDataName> structure does not have any name attributes for the boxplot data.
No names, no way to link to the boundary data.
Needs description

02B5 BOXPLOT cc The \<pdDataName> boxplot data is not a valid structure. Must contain 6 boxplot sub lists.
The \<pdDataName> structure must contain the 6 data lists for each sub-area. Less than 6 or greater than 6 lists were found..
Need description

02B6 BOXPLOT cc The \<pdDataName> boxplot data does not contain all of the lists of boxplot function output. Invalid structure.
The \<pdDataName> structure must contain the following 6 names list for each sub-area: stats, names, out, group, n, and conf. If these named lists are not found, the package considers the data structure invalid and will not attempt to draw the boxplot glyph. Verify the \<pdDataName> is the correct variable and was properly created by the boxplot function.

02B7 BOXPLOT cc In the \<PDDataName> boxplot data, the '$name' named list contains one or more missing values (NA).
The '$name' list is used to link the boxplot data to the boundary data. If the data cannot be match with a sub-area, the sub-areas glyph row will not be drawn. Check the

boxplot data and ensure each individual boxplot has a valid sub-area name associated with it.

02B8 BOXPLOT cc There are duplicate sets of boxplot data in \<boxnam\> for the same sub-area. Only the first one will be used.
The '$name' list in the boxplot data contain duplicates sub-area names. Only one boxplot can be drawn per sub-area. The first set of data will be used to draw the boxplot. Any other data with the same name will be ignored. If a sub-area does not have any boxplot data, that sub-areas glyph will be omitted.

02BA BOXPLOT cc The $stats matrix in the \<boxnam\> boxplot data does not have 5 values per area.list is not the same length as the $name list.
Need description

02BB BOXPLOT cc The $stats matrix in the *\<pcDataName\>* boxplot data must *\<bp-NumNames\>* elements.
Need description

02BC BOXPLOT cc The $stats matrix in the *\<pcDataName\>* boxplot data has missing values. Sub-areas with missing values will not be drawn.
Need description

02BD BOXPLOT cc The sub-area names/abbrevations in the *\<pdDataName\>* boxplot data $names values do not match the border group names: *\<nameList\>*.
Need description

02BE BOXPLOT cc 02BE BOXPLOT cc There are one or more of rows in the *\<statsDFrame\>* that does not have matching boxplot data, (*\<pdDataName\>*) entries.
Need description

02BF BOXPLOT cc 02BE BOXPLOT cc The missing sub-areas are: *\<subAreaList\>*
Need description

02Bx BOXPLOT cc The $n list is not the same length as the $name list.
Need description ***check***

02B5 BOXP-0x cc The $conf list is not the same length as the $name list.
Need description ***check***

02B6 BOXP-0x cc The $group list is not the same length as the $name list.
Need description ***check***

02B7 BOXP-06 cc The $stats matrix in the boxplot data does not have 5 columns of data per sub-area.
Need description ***check***

02B9 BOXP-09 cc The boxplot $stats data contains missing values NA.
Need Description ***check***

For the time series (TS) and time series with confidence band (TSCONF), data for the graphics is provided through a separate data structure via the panelData column in the *panelDesc* data.frame. The *statsDFrame* data.frame is not used and the *colx* indexes are not used.

02D1 \<glyphs\> cc The *panelData* value of (*\<pDataValue\>* in the *\<panelData\>* data frame does not exist or is not accessible.
The *panelData* variable in the *\<panelDesc\>* data frame is used to pass completed data structures to glyphs that cannot be contained in the *\<statsDFrame\>* data frame. In this case, the variable name provided *\<pDataValue\>* does not exist or is not accessible by

the package. Make sure the name is spelled correctly and exists in the .GlobalEnv (calling) environment. This message is used by the BOXPLOT, TS and TSCONF glyphs. ***check***

The following warnings are issued during the processing of the sub-area identifiers in the *<statsDFrame>* data frame and the border group name table.

02P3 <glyph> cc There are no links in the *<PDDataName>* **data to the following** *statsDFrame* **sub-areas: <list of** *statsDFrame* **sub-areas.**
Sub-areas in the *statsDFrame* do not have data in *<PDDataName>*. ***check***

02P4 <glyph> cc The following sub-area links in *<PDDataName>* **data do not link to** *<statsDFrame>* **sub-areas and will not be used: <listOfNames>**
Sub-areas in the *<PDDataName>* do not have links to the *<statsDFrame>* data.frame. ***check***

02P5 <glyph> cc There are no sub-area links provided in *<PDDataName>*. **Cannot link data to the** *statsDFrame* **data.frame.**
<PDDataName> does not have values to link to *<statsDFrame>* or the name table. ***check***

02P6 <glyph> cc There are duplicate sub-areas in the *<PDDataName>* **structure. Only the first match will be used.**
<PDDataName> has duplicate entries. ***check***

Only validate panelData entries for boxplot, ts and tsconf glyphs. For boxplot, datatype must be list. For ts and tsconf, datatype must be 3 dimensional array.

Implementation: *statsDFrame* and the sorted link table drives the glyph generation. Must do this to match up with the other columns. If no data is in *<PDDataName>* for the *statsDFrame* link, no glyphs is drawn for the row.

If duplicates exist in *<PDDataName>* only the first will be used.

If extra entries exist in *<PDDataName>* that don't match the *statsDFrame*, even if they match the name table, they are not used.

TS and TSConf specific processing:

02T0 TSxxxx-11 cc The time series array does not have rownames assigned to the 1st dimension. Data cannot be paired up with area.
Need description ***check***

02T2 TSxxxx-02 cc The data structured passed in the panelData field is not an array. Structure name = <dataNam>.
Need description ***check***

02T3 TSxxxx-03 cc The time serial array's 1st dimension is not <numRows> sub-areas to match statsDFrrame. It is <dimDArr[1]>.
Need description ***check***

02T4 TSxxxx-04 cc The time serial array's 2nd dimension must have at least 2 values (time periods 2 to "n"). It is <dimDArr[2]>.
Need description ***check***

02T5 TSxxxx-05 cc The time series array's 3rd dimension is not 4 values for TSConf (X, Y, LowY and HighY). It is <dimDArr[3]>.
Need description (Duplicate) ***check***

02T6 TSxxxx-06 cc The time series array's 3rd dimension must be 2 or 4. It is <dimDArr[3]>. (TS: X & Y, TSCONF: X, Y, LowY, HighY.
Need description ***check***

02T7 TSxxxx-10 cc The time series array's 3nd dimension must be at least 2. (TS: X & Y) It is <dimDArr[3]>.
Need description. Duplicate - relook at. ***check***

03xx bordGrp and bordDir Validation:

03yy - asdf
description of error group ***check***

04xx panel layout calculations:

042x column sizing calculations:

0420 PANEL Calculated column widths is less than minimum <colSizeMin> inches - too many columns specified.
Need description ***check***

0421 PANEL Column width is to small to be useful, Package stopped.
Need description ***check***

043x row sizing calculations:

0430 PANEL panelLayout - The calculated width of <calculated-width> is too large for the available space of <space-width>.
Need description ***check***

0431 PANEL panelLayout - The calculated GrpRow Height is too small to be used.
Need description ***check***

0432 PANEL panelLayout - The calculated GrpRow Height is <GrpRowHeight> inches. The minimum size limit is: <minimum row height.
Need description ***check***

xsh title
Need description ***check***

045x panel functions:

0450 PANEL panelLengthen - invalid vector length. < 2
Need description ***check***

0451 PANEL panelSelect - Dimension error. Program error - index 'i' or 'j' is out of bounds.
Need description ***check***

0452 PANEL panelSelect - Bad label region name. Must be left, right, top or bottom.
Need description ***check***

049x Internal Package Messages:

0490 DMP Error in axisMethod selection in Dot and DotSignif code.
Need description ***check***

0491 INB is.between.r The r range value is not a vector with length of 2. FALSE returned.
Function is.between.r — Need description **checked** ***check***

05xx Informational:

050x Panel Messages:

0501 PANEL Number of parameters overlaid = <numOverlaid.
Need description ***check***

xxx asdf
     Need description ***check***
xxx asdf
     Need description ***check***

Back to the details section - Rest of the details dection

End of details section.

---

micromapGDefaults   *micromapGDefaults data.frame structure*

Description

The micromapGDefaults data.frame provides all of the detailed structure, colors, sizing, font sizes, separation distances, line weights and types, spacing, etc. required to physically construct the requested micromapST graphic in portrait or landscape modes from a letter size (8.5 x 11) up to a tabloid (11 x 17) page. The data.frame is mainly used internal to micromapST, but a copy can be obtained by a user when a large number of changes are required. This is not recommended. The primary purpose of this section is to provide a list and description of many of the variables in the details list that can be used to enable or disable functions of micromapSEER and it's glyphics. This variable are identified by a "*" after the variable name. These are the only variables that generally safe to be modified by the user.

Format

The data.frame contains two lists: colors and details.

colors The colors vector is the name of a color palette or a vector of 7 or 14 color names or values ("#xxxxxx"). The vector defines the colors used for:

- The 5 colors in each group for the states and symbols in the glyphcs. One color per row (state).
- 1 color for the median state and glyphics,
- 1 foreground color for highlighted states in the map. This is used to highlight states already referenced previously or have meaning depend on the type of map requested. The usage is as follows:

```
        "map"       - not used.
        "mapcum"    - highlight states previously referenced
                       above.
        "maptail"   - highlight states previously referenced
                       above the median row and highlight
                       remaining states below the median row.
        "mapmedian" - highlight all states above the median
                       in maps above the median row and highlight
                       all states below the median in maps
                       below the median row.
```

- When 7 additional colors are specified, they are used as the translucent colors in the time series glyphics bands. If only 7 colors are provided, the additional 7 translucent colors are generated from the original colors as a 20% transparent version of the original color e.g., adjustcolors(colors,0.2).

If colors = "greys" or "grays" or "bw", the entire plot will be done using the packages standard black/white/gray shades. Additional color palettes may be supported in future releases.
If a colors vector is not provided, the package default colors will be used:

- 5 state colors: "red", "orange", "green", "greenish blue", "lavender"
- 1 median state color: "black"
- 1 highlighed states: "light yellow" and
- 7 translucent colors using the above colors at 20%.

It is strongly recommended to use the default.

DetailsPanel is a list structure containing the variables for the particular list of areas and glyphics columns for a particular micromap creation. They are created using the charactistics of the statsDFrame passed to the package and the variables in the *details* structure. The list of calculated variables is:

numRows = number of rows (areas) in statsFrame data.frame.

maxRows = maximum number of rows (areas) supported by micromaps.

numGrps = number of logic panel groups. Based on the table lookup by number of rows (areas).

GrpPattern = vector containing the pattern of number of rows per group. Number of rows per group must be in decending order from the outside groups to the median (center) group.

GrpEntry = vector of the number of rows per each panel group, Length of vector equals the number of groups.

GrpMedEntry = logical vector of number of panel groups with median panel = TRUE.

rowSep = c(0,0,0,0,0,0.1,0.1,0.0.0.0.0), location of row separations in each column of panels (0.1 around the median row, 0 for others)(inches). Calculated based on the table of rows per panel. Length equals the number of panel groups + 1.

rowSize = c(7,7,7,7,7,1.6,7,7,7,7,7), size of rows in each panel (1.6 row units for median row, 7 row units for the 5 item groups). Based on the lookup table of rows per panel group. Length equals the number of panel groups + 1.

medGrp = the index number of the median panel group, if the number of groups is odd. Otherwise, the value is zero (0).

medRow = the index number of the median row (area), if the number of areas is odd. Otherwise, the value is zero (0).

medGrpSize = number of rows in the median group when there is an odd number of groups.

medRowBlw = the index number of the row/group below the median group when there is an odd number of groups.

medRowAbv = the index number of the row/group above the median group when there is an odd number of groups.

groupedRowSize = c(35, 1.65, 35), size of panels (1.65 row units for median panel of 1 area, 7 row units for all other panels.) For the full US States, =c(35,1.65,35). For the US Seer with 18 areas = c(14,14). The vector varies based on the number of panel groups and if the median group represents only one area.

groupedRowSep = c(0,0.1,0.1,0), separation between panels. The normal value is zero. If there is an odd number of panel groups, then there is a separation added on each side of the median group. If the number of panel groups is even, then a single separation is added to the middle of the panel groups. (e.g., c(0,0,0,0,0.1,0,0.0.0). The separation value is set to the value of the *rowSepGap* variable.

ib = for the US States micromaps, the value is c(1,6,11,16,21,26,27,32,37,42,47), index for first items in each of the 11 groups. For any other micromap, it's value is calculated based on the number of panel groups used and the number of rows (areas) in each panel.

ie = for the US States micromaps, the value is c(5,10,15,20,25,26,31,36,41,46,51), index for last items in each of the 11 groups. For any other micromap, it's value is calculated based on the numberof panel groups used and the number of rows (areas) in each panel. The *ib* and *ie* variables are used to index into the sorted list of states to pull out the states for a particular group/row.

As with the other details data, all of these variables are copied into the packages memory and are accessible by all internal functions. Most of the vectors are calculated from the number of rows and groups in the linked micromap and the specified gap sizes between key groups.

details is a list structure that contains the variables and values for the detailed layout and operation of micromapST. The details parameters have been generalize to support other border and area sets beyond the 51 US States. The number of rows is set by the rows in the data, and the acceptable list of areas in the supporting data frame (see USStatesDF and USSeerDF descriptions) The list of variables that define the page and panel layout is:

topMar = 1.2, top margin in inches

botMar = 0.5, bottom margin in inches
botMarLegend = 0.5, bottom margin for legend
botMardif = 0.2, bottom margin difference

leftMar = 0.0, left margin in inches
leftMarAxis = 0.2, left margin when Y axis labels and title are required
rightMar = 0.0, right margin in inches

borderSize = 0.4 inches. The border space between the page edges and the margins. This value is used for the top, bottom, left and right border spacing.

mgpTop = c(3.2,0.1,0), Top margin line for "X" axis labels and axis line.
mgpBottom = c(3.2,0,0), Bottom margin line for "X" axis labels and axis line.
padjBottom = -0.35, Axis tick label placement adjustment
mgpLeft = c(0.75, 0.1, 0), Left "Y" axis margin line for axis labels and axis line.

sc = 1.08, x axis scale expansion factor. Applied to the data range to calculate the graph's range.

pad = 0.67, y axis padding for integer plotting locations

padex = 0.34 inches, total panel padding (i.e., 0.17 at top and bottom of panel)

padMinus = 0.63 inches, spacing to keep reference line off panel edge

The following variables may be changed by the user, if careful. They impact the gaps between groups, position of column titles, grid and panel colors, general text size, and the minimum and maximum size of the panels.

rowSepGap = 0.1 units. Space between panel groups at or around the median panel. There are 7 units per panel. The average unit is between 1/10 and 1/8 inches.

rowSizeMn = 0.5 inches. Minimum panel group height.

rowSizeMx = 1.25 inches. Maximum panel group height.

rowSizeMin = 1.65 units. Minor panel group height in units for single area panel.

rowsizeMaj = 7 units. Major panel group height for all panels, except single area panel.

colSepGap = 0.1 inches. Glyphics column separator space.

colSizeMin = 0.75 inches. Minimum Glyphics column size.

colSizeMax = 1.8 inches. Maximum Glyphics column size.

rcRatioMin = 0.25, minimum row size to col size ratio permited. (not implemented)

rcRatioMax = 2, maximum row size to col size ratio permited. (not implemented)

XAxis.L.mcex = 0.88888 Size used for large XAxis labels

XAxis.M.mcex = 0.77778 Size used for medium XAxis labels

XAxis.S.mcex = 0.66667 Size used for small XAxis labels

XAxis.Sp.mcex = 0.2 Size used for XAxis space between lines of labels

XAxis.offset = 0.0 inches. XAxis offset

XAxis.indent = 0.10 1000th inches. First and Last label indents from edge.

XAxis.nGridpIn = 3.4 labels per inch in XAxis - initial objective.

XAxis.gapPC = 0.75 (*100 for precentage). Percentage of the width of a space used to determine label overlaps.

XAxis.staggered = TRUE, enable staggered label feature - NOT USED

YAxis.cex = 0.33332 cex size of YAxis labels

YAxis.offset = 0.0 lines. Offset of YAxis labels from panel edge.

YAxis.nGridpIn = 5 labels. Number of labels per inche for YAxis - initial goal.

YAxis.staggered = TRUE. Enable staggered labels on YAxis - NOT USED.

YAxis.width = 0.2 inches. Extra column gap size required to support drawing an Y-Axis for TS and ScatDot glyphics

staggered = FALSE. Position of last staggered label. FALSE = low, TRUE = high.

Title.Line.1.pos = 1.27 inches, top panel 1st line placement above the first panel, used with lab1

Title.Line.2.pos = 0.64 inches, top panel 2nd line placement above the first panel, used with lab2

Title.Line.2x.pos = 0.01 inches, top panel X-Axis line placement above the first panel

Title.Line.3x.pos = 0.01 inches, bottom panel X-Axis line placement below the last panel

Title.Line.3.pos = 0.64 inches, bottom panel line placement below the last panel, used with lab3

Title.Line.4.pos = 1.27 inches, reference line legend below last panel, used with reftext

Title.Line.5.pos = 0.35 inches, Y axis label placement (to the left of panel), used with lab4

Title.cex = 1.0, text size of title, used with title

Grid.Line.col = "white" * Grid line color

Grid.Line.lwd = 1, Grid line width

Panel.Fill.col = "#EBEBEB", * defaults to light gray

Panel.Outline.col = "black", * color of panel outlines

Text.cex = 0.75 cex, general text size

The following variable relate to the reference text and line features:

Ref.Val.lty = "dashed", * set reference value line to dashed

Ref.Val.lwd = 1.5, * line width of reference line

Ref.Val.col = "midgreen", * color of reference line when color is used.

Ref.Val.BW.col = "black", * color of reference line when grays are used.

Ref.Text.cex = 0.75, * size of reference line text

Ref.Text.col = "black", * color of reference line text when color is used.

Ref.Text.BW.col = "black", * color of reference line text when grays are used.

The following variables are used by the 'arrow' glyph:

Arrow.lwd = 2.5, line width of arrow.

Arrow.cex = 0.08, * size of arrow (*** not implemented ***)

Arrow.Head.length = 0.08, length of arrow.

Arrow.Shadow.lwd = 4.0, * line width of arrow shadow to create outline (*** not implemented ***)

Arrow.Shadow.col = "black", * arrow shadow color (*** not implemented ***)

Arrow.Dot.pch = 21, arrow-dot symbol 19-25.

Arrow.Dot.pch.size = 0.9 cex, arrow-dot size.

Arrow.Dot.pch.lwd = 0.5, line weight used on filled arrow-dot symbols.

Arrow.Dot.Outline = FALSE, include dot outline when filled. FALSE=NO, TRUE=YES

Arrow.Dot.Outline.col = "black", color used for arrow-dot outline.

Arrow.Dot.Outline.lwd = 0.5, line weight for arrow-dot outline.

The following variables are used by the 'bar' glyph:

Bar.barht = 2/3, * fraction of line height for bar. Should never be > .90. Usable range is 0.333 to 0.90

Bar.Outline.col = "black", * color of bar outline.

Bar.Outline.lwd = 0.5, * line width for bar outline

Bar.Outline.lty = "solid", * line type for bar outline

The following variables are used by the 'boxplot' glyph:

BoxP.thin = 0.2, line width of box.

BoxP.thick = 0.6, thick line width.

BoxP.Use.Black = FALSE, * whether to outline the outlier points.

BoxP.Median.col = "black", * color of median box

BoxP.Median.Line = 0.80, * line width of median line.

BoxP.Median.Dot.pch = 19, * solid circle symbol.

BoxP.Median.Dot.col = "white", * color of median dot.

BoxP.Median.Dot.cex = 0.95, * size of circles.

BoxP.Median.Dot.lwd = 2, * line width for median.

BoxP.Outline.col = "#262626FF", * boxplot outline color

BoxP.Outlier.pch = 20, symbol for outlier - 19-25.

BoxP.Outlier.lwd = 0.4, * line width of outlier outlines.

BoxP.Outlier.cex = 0.7, * size of outlier dots.

BoxP.Outlier.BW.col = "#4c4c4cFF" * color of outliner lines when greys used.

The following variables are used by the 'ctrbar' glyph:

CBar.varht = FALSE, * enables variable height bars.

CBar.two.ended = FALSE, * request two ended variable height bars be used

CBar.Center.Line.enable = FALSE, * request a line is drawn at center point.

CBar.Center.value = 0, * value of the center stacked bar glyph. def=0

CBar.Zero.Line.col = "white", * centered bar zero vertical line color

CBar.Zero.Line.lwd = 1, * line width for centered bar zero vertical line

CBar.Zero.Line.lty = "dotted", * type of centered bar zero line. (small to large to small). (*** Not implemented ***)

The following variables are used by all of the horizontal stacked bar glyphs ('ctrbar', 'segbar' and 'normbar'):

CSNBar.barht = 0.66667, * fixed height of bar when variable height bars are is not used. Should never be greater than 0.90. Usable range is 0.333 to 0.90.

CSNBar.First.barht = 0.3333, * height of first bar when variable height bars are used. Must be less than SBar.Last.barht and in the range of 0.333 to 0.6667, SNBar.varht or CBar.varht must be TRUE for this option of function.

CSNBar.Last.barht = 0.80, * height of last bar when variable height bars are used. Must
   be greater than CSNBar.First.barht and in the range of 0.6667 to 0.90. CSNBar.varht or
   CBar.varht must be TRUE for this option of function.

CSNBar.Outline.col = "black", * color of stacked bar outlines.

CSNBar.Outline.lwd = 0.75, * line weight for bar segment outline in segmented bar plots.

CSNBar.Outline.lty = "solid", * line type for bar segment outline in segmented bar plots.

The following variables are used by the 'dot', 'dotconf', and 'dotse' glyphs:

Dot.pch = 21, * solid circle (S compatible).

Dot.pch.size = 0.9 cex, * size of dot.

Dot.pch.lwd = 0.5, * linen weight for dot outline when 0:18 dot used.

Dot.Outline =FALSE, * whether to outline the dots.

Dot.Outline.col = "black", * color of dot outline.

Dot.Outline.lwd = 0.5, * line width of dot outline.

The following variables are used by the 'dotconf' glyph:

Dot.Conf,pch = 21, * solid circle (S compatible).

Dot.Conf.pch.size = 0.9 cex, * size of dot.

Dot.Conf.pch.lwd = 0.5, * linen weight for dot outline when 0:18 dot used.

Dot.Conf.Outline =FALSE, * whether to outline the dots.

Dot.Conf.Outline.col = "black", * color of dot outline.

Dot.Conf.Outline.lwd = 0.5, * line width of dot outline.

Dot.Conf.lwd = 2, * line width of confidence interval lines.

Dot.Conf.size = 0.55, * size of confidence interval

The following variables are used by the 'dotse' glyph:

Dot.SE,pch = 21, * solid circle (S compatible).

Dot.SE.pch.size = 0.9 cex, * size of dot.

Dot.SE.pch.lwd = 0.5, * linen weight for dot outline when 0:18 dot used.

Dot.SE.Outline =FALSE, * whether to outline the dots.

Dot.SE.Outline.col = "black", * color of dot outline.

Dot.SE.Outline.lwd = 0.5, * line width of dot outline.

Dot.SE = 95, percent confidence interval

Dot.SE.lwd = 2, * line width of confidence interval lines.

Dot.SE.size = 0.55, * size of confidence interval

The following variables are used by the 'dotsignif' glyph:

Dot.Signif.pch = 4, overprint character "x" when not significance.

Dot.Signif.pch.size = 0.9*1.2 * cex size of the overprint character.

Dot.Signif.pch.lwd = 0.5, * linen weight for dot outline when 0:18 dot used.

Dot.Signif.pch.col = "black", color of overlaid symbol on DOT.

Dot.Signif.Outline =FALSE, * whether to outline the dots.

Dot.Signif.Outline.col = "black", * color of dot outline.

Dot.Signif.Outline.lwd = 0.5, * line width of dot outline.

Dot.Signif.pvalue = 0.05, * p-value for testing significance.

Dot.Signif.range = c(0,1), * valid range for significant test data for p-value

The following variables are used by the 'id' glyph:

Id.Cex.mod = 1, fudge adjustment for ID Text.

Id.Title.Line.1.pos = 0.9 inches, top panel 1st line id title placement above the first panel, used with lab1

Id.Title.Line.2.pos = 0.1 inches, top panel 2nd line id title placement above the first panel, used with lab2

Id.Text.cex = 0.65, text side of ID column

Id.Dot.pch = 22, * pch symbol value to plot next to state name/abbrev.

Id.Dot.cexm = 1.5, * size of dot symbol for state ID

Id.Dot.lwd = 0.8, * size of solid dot symbol for state ID

Id.Dot.Outline.col = "dark gray", * color of outlines of ID symbols

Id.Dot.Outline.lwd = 0.8, * line weight for outlines of ID symbols

Id.Dot.width = 0.1 inches. With of the ID symbol (box)

Id.Space = 0.03125, width of a space in inches.

Id.Start = 0.055, offset from left for start of ID column.

The following variables are used by all of the "map" type glyphs:

Map.Area.Spec.cex = 0.32, * font size for state labels

Map.Bg.col = "#262626FF", (grey(0.88)) * color of background (not active) sub-areas fill in maps

Map.Bg.Line.col = "white", * background of maps

Map.Bg.Line.lwd = 1, * line weight for map background boundaries

Map.Bg.Line.lty = "solid", * line type for map background boundaries

Map.Fg.Line.col = "black", * foreground color of maps

Map.Fg.Line.lwd = 1, * line weight for map foreground boundaries

Map.Fg.Line.lty = "solid", * line type for map foreground boundaries

Map.L2.Line.col = "black", * color of Layer 2 outline in maps

Map.L2.Line.lwd = 1. * line weight for Layer 2 boundaries

Map.L2.Line.lty = "solid", * line type for Layer 2 boundaries

Map.L3.Line.col = "black", * color of Layer 3 (national) outline in maps

Map.L3.Line.lwd = 1. * line weight for Layer 3 (national) boundaries

Map.L3.Line.lty = "solid", * line type for Layer 3 (national) boundaries

Map.Lab.Box.Width = 0.09, * width in inches of box symbols using in titles for maps

Map.Max.width = 2.5 inches, maximum width of each map

Map.Min.width = 1.5 inches, minimum width of each map

Map.Panel.col = "white", * background color of map panels

Map.Median.text = "Median for Sorted Panel", * text used in single row median panel instead of a map.

Map.Unu.col = "lightest grey", * color of sub-areas not referenced in maps

The following variable is used by the 'rank' glyph.

Rank.width = 0.25 inches - column fixed width

Rank.method = 1, rank method - to be defined.

The following variables are used by the scatter dot glyph ('scatdot'):

SCD.Axis,cex = 0.49, font size for Y axis labels for scatter dots

SCD.Bg.pch = 21, * type of point/symbol to be used for background data points (not active) - state's dots.

SCD.Bg.pch.fill = "transparent", * fill color for not selected state's dots.

SCD.Bg.pch.col = "black", * border color used for non-active data points.

SCD.Bg.pch.lwd = 0.6, * line width of outline of point/symbol used as non-active data points.

SCD.Bg.pch.size = 0.75, * size of point/symbol used as non-active data points.

SCD.Fg.pch = 21, * type of point/symbol for active data points.

SCD.Fg.pch.col = "black", * border color for foreground dots.

SCD.Fg.pch.lwd = 0.6, * Scatter dot symbol outline line weight for active data points.

SCD.Fg.pch.size = 1, * size of point/symbol for active data points in scatter dots.

SCD.Median.pch = 21, * shape of filled symbol for median value - scatter dots.

SCD.Median.pch.col = "black", * border color for median symbol - scatter dots.

SCD.Median.pch.lwd = 0.6, * line width used on the median symbol - scatter dots.

SCD.Median.pch.size = 1, * symbol size median value - scatter dots.

SCD.Median.pch.fill = "black", * color of filled symbol for median value - scatter dots.

SCD.hGrid = FALSE, * whether or not to include horizontal grid lines in panel.

SCD.xsc = 1.08, x range multiplier to keep dots from being clipped

SCD.ysc = 1.12, y range multiplier to keep dots from being clipped

SCD.DiagLine = TRUE, * whether or not to include x=y sloped line

SCD.DiagLine.col = colGrid, * color of sloped line, default, grid line color. SCD.DiagLine must be TRUE.

SCD.DiagLine.lwd = 1.25, * line weight of sloped line, default, grid line color. SCD.DiagLine must be TRUE.

SCD.DiagLine.lty = "solid", * line type of sloped line, default, grid line color. SCD.DiagLine must be TRUE.

The following variables are used by the 'segbar' and 'normbar' glyphs:

SNBar.varht = FALSE, * enables variable height bars from SBar.First.barht to SBar.Last.barht.

SNBar.two.ended = FALSE, * request two ended variable height bars be used (small to large to small).(*** Not implemented ***)

SNBar.Middle.Dot =FALSE, * request a dot be draw in at the mid point in the segmented bars.

SNBar.MDot.pch =21, * type of point/symbol used as the mid point dot. SNBar.Middle.Dot must be TRUE for this parameter to function.

SNBar.MDot.pch.fill ="white", * color of the point/symbol used as the mid point dot. SNBar.Middle.Dot must be TRUE for this parameter to function.

SNBar.MDot.pch.size =0.3, * size of point/symbol used as the mid point dot. SNBar.Middle.Dot must be TRUE for this parameter to function.

SNBar.MDot.pch.border.lwd =NA, * line width of outline of point/symbol used as the mid point dot. SNBar.Middle.Dot must be TRUE for this parameter to function.

SNBar.MDot.pch.border.col =NA, * color of outline of point/symbol used as the mid point dot. SNBar.Middle.Dot must be TRUE for this parameter to function.

The following variables are used by the time series glyphs ('ts' and 'tsconf'):

TS.lwd = 1.1, * time series line weight

TS.Axis.cex = 0.49, * font size for Y axis labels

TS.hGrid = FALSE, * whether or not to include horizontal grid lines in panel

The following variable is reserved for package testing only and should not be used.

MST.Debug = 0, disabled. Do not use.

When changing the colors list, then the entire list must be specified. To change the values of items in the details list, only the variable(s) requiring change need to be specified as a list for the details= parmater in the call. In general, the beginning of the variable names indicates the glyph the variable is associated, in most cases.

```
Arrow.     -> arrow glyph
Bar.       -> bar glyph
BoxP       -> boxplot glyph
CBar       -> ctrbar glyph
CSNBar     -> ctrbar, segbar and normbar glyphs.
Dot.       -> dot, dotconf, and dotse glyphs
Dot.conf.  -> dotconf glyph
Dot.Signf  -> dot with significance overlay.
Grid       -> grid elements of all glyphs
Id.        -> id glyph
Map.       -> map glyphs
Panel      -> general glyph panel
Rank       -> area ranking glyphs.
Ref        -> Reference text and line
SCD.       -> scatter dot glyph
SNBar      -> segbar and normbar glyphs
Title      -> page and column labels and titles
TS         -> ts and tsconf glyph
```

```
            TSconf    -> tsconf glyph
```

The remaining variables deal with page and glyph layout and spacing and should not be adjusted.
For example: to turn off the midpoint dot in the segmented bar glyphics, all that is required is:

```
            details = list(SNBar.Middle.Dot=FALSE)
```

Details

The micromapGDefaults data.frame is built by the micromapGSetDefaults function when the micromapSEER package is called. Once built it cannot be changed. To change one or two (a few) variables, construct a list of these variables and pass it to micromapSEER via the details parameter in the call. To do large scale customization, call the micromapGSetDefaults function to get a copy of the entire data.frame and modify this copy. This is not recommended.

Author(s)

Daniel B. Carr, George Mason University, Fairfax VA, with contributions from Jim Pearson and Linda Pickle of StatNet Consulting, LLC, Gaithersburg, MD

See Also

micromapGSetDefaults, micromapGSetPanelDef,

---

micromapGSetDefaults   *function to build the micromapGDefault data.frame*

Description

The micromapGSetDefaults function generates a data.frame containing two lists: the colors and details list. Each list contains the operational parameters to instruct micromapST on how to physically construct a micromapST graphic. The colors list contains 14 rgb colors:

The 7 basic colors:

5 colors for the state maps in each group of 5 states that make up one panel row.

1 color for the median state (in the middle)

1 color for the cumulative hightlight color for the states already presented in other panel rows.

and a repeat of the 7 rgb colors with an alpha transparency value of 10%.

The details list contains the spacing, margins, text size, etc. information to guide the construction of the micromapST graphic. These lists may be modified by the user, but this is not recommended. See the micromapSTDefault description for more information on these lists.

Usage

```
micromapGSetDefaults()
```

Details

The default colors in the colors list are: red, orange, green, greenish-blue and lavender for the state colors, black for the median state color, and light yellow for the cummulative hightlighting on the map. The default values in the details list can be found in the micromapGDefaults documentation.

This function is primarily used internally by micromapST. However, if the user wants to see values of all of the internal variables or wants to make wholesale changes to the layout and operational changes to the colors and details list, this function can be used to create a copy of the full colors and details lists.

Value

[micromapGDefaults](#)

Author(s)

Daniel B. Carr, George Mason University, Fairfax VA, and Jim Pearson, StatNet Consulting, LLC, Gaithersburg, MD

---

micromapGSetPanelDef   *function to build the micromapGPanelDef data.frame*

Description

The micromapGSetPanelDef function generates a data.frame to supplement the micromapGDefaults structure. The resulting structure along with the values from micromapGSetDefaults are copies into the package's memory and used by the "panel functions" to manage the panels during the setup and glyphics creation.

Usage

```
micromapGSetDefaults()
```

Details

The variable in the structure are::

Var 1 for the state maps in each group of 5 states that make up one panel row.

Var 2 for the median state (in the middle)

Var 3 for the cumulative hightlight color for the states already presented in other panel rows.

is not recommended. See the micromapGDefault description for more information on these lists. More to come.

Value

[micromapGSetPanelDef](#)

Author(s)

Daniel B. Carr, George Mason University, Fairfax VA, and Jim Pearson, StatNet Consulting, LLC, Gaithersburg, MD

---

micromapSEER   *A function to create linked micromaps for the 20 U. S. Seer Registries*

Description

The *micromapSEER* function or the *micromapST* function with the 'bordGrp' set to *USSeerBG* can be used to create linked micromaps for the 20 U. S. Seer Registries.

Usage

```
micromapSEER(statsDFrame,panelDesc,...)
```

Arguments

statsDFrame	- data frame of the data required for creating the graphical glyphics with one row per Seer Area being mapped. Each row is linked to the boundary data using a Seer Area abbreviation or name.
panelDesc	- data frame containing information and pointers for each glyphic column to be generated. The data frame specifies the type of glyphic and the columns in the *statsDFrame* data frame that contain the data for the glyphic or the external data structure via the panelData list.
...	- the remaining parameters required and used by the *micromapST* function call.

Details

More details to follow.

Value

None

Author(s)

Daniel B. Carr, George Mason University, Fairfax VA, and Jim Pearson, StatNet Consulting, LLC, Gaithersburg, MD

See Also

[micromapST](#)

micromapST	*Linked Micromap Graphics Package*

Description

Provides a easy and quick means of creating Linked Micromaps for any collection of geographically associated areas. The micromapST package uses the standard graphics and RColorBrewer packages to rapidly create highly readable linked micromap plots. This gives the user the ability to explore different views of their data quickly.

micromapST uses the border and name information contained in border group datasets to define the geographical areas used in creating the linked micromaps.

The *micromapSEER* function is included to help users of the specialized *micromapST* for NCI Seer areas called *micromapSEER* migrate to this package. The *micromapSEER* function, calls *micromapST* with the border group set to *USSeerBG* to generate linked micromaps for the 20 U. S. Seer Areas.

The *micromapST* contains border group datasets with the boundary and name information for the following::

- *USStatesBG* - data from the original *micromapST* package for . the U. S. 50 states and the District of Columbia.
- *USSeerBG* - data for the 21 U. S. Seer Areas.
- *KansasBG* - data for the 105 counties in the state of Kansas.
- *NewYorkBG* - data for the 62 counties in the state of New York.
- *MarylandBG* - data for the 24 counties in the state of Maryland.
- *UtahBG* = data for the 29 counties in the state of Utah
- *ChinaBG* - 34 administrative areas in the country of China.
- *UKIrelandBG* - 218 administrative areas in UK, Ireland and Isle of Man.
- *SeoulKoreaBG* - 25 districts in Seoul S. Korea.
- *AfricaBG* - 52 countries of the Africa continent.

Each plot row represents a single sub-area (state, county or province) within the border group area. Each column can be defined to present a different graphical representations of the user's data.
For linked micromaps, the primary columns are a MAP type, ID (sub-area name) and one or more glyphics. The statistical data is presented in the glyphics columns as one of the following graphics (glyphics):

- arrows,
- bars,
- boxplots,
- dots,
- dots with confidence intervals,
- dots with standard error,

- dot with a significance marker,

- time series line plots with or without confidence bands,

- scatter plots, and

- horizontal stacked (segmented, centered, and normalized) bars.

All border groups are distrbuted with the package as .rda datasets.

Usage

```
micromapST ( statsDFrame,
                    panelDesc,
                    rowNamesCol = NULL,
                    rowNames   = NULL,
                    sortVar    = NULL,
                    ascend     = TRUE,
                    title      = c("", ""),
                    plotNames  = NULL,
                    axisScale  = NULL,
                    staggerLab = NULL,
                    bordGrp    = NULL,
                    bordDir    = NULL,
                    dataRegionsOnly = NULL,
                    regionsB   = NULL,
                    grpPattern = NULL,
                    maxAreasPerGrp = NULL,
                    ignoreNoMatches = FALSE,
                    colors     = NULL,
                    details    = NULL)
```

Arguments

statsDFrame   a data.frame containing data used with the following plots/glyphs:. *arrow*, *bar*, *segbar*, *normbar*, *ctrbar*, *dot*, *dotse*, *dotconf*, *dotsignif*, and *scatdot* plots. The data for the *boxplot* and time series plots (*TS* and *TSConf*) is more complex and multi-dimensional and is passed to the glyph generation routines via the *panelData* parameter (see below for more details.)

The *row.names* of *statsDFrame* data.frame are used as the link identifier between the data row to the map boundary data. For U.S. state data, the link identifier must be the the state's 2 character abbreviations, full names, or 2-digit US FIPS codes as the *id*. For Seer Areas, the area identifier is the Seer Area abbreviates as defined in the *USSeerBG* document. Refer to the documentation on each border group for the exact names, abbrevviations and ids defined for each sub-area.

The data columns in the statsDFrame are associated with each graphic using the *col1*, *col2*, and *col3* vectors in the panelDesc data.frame by column name or number.

panelDesc	a data.frame that defines the description of each column: types, associated data columns in the *stateFrame* data.frame, column titles (top and bottom), reference values and text, and names of additional data.frames for complex glyphics (time series and boxplots). See section on panelDesc data.frame. for more details.
rowNames	defines the type of value used as the row.names in the *stateFrame*. The options are:

- ‘ab’ - an abbreviation for the sub-area: 2 character state ID, postal code abbreviation, ISO abbreviation, or a generally accepted abbreviation for the sub-area;
- ‘id’ - sub-area ID, numerical integer identifiers. In the U.S. border groups, the state and county FIPs codes are used. An alias for ‘id’ is ‘FIPS’.;
- ‘full’ - the full sub-area names;
- ‘alias’ - partial match aliases. Used only with the Seer Registry sub-area registry names as outputted by the SeerStat programs; or
- ‘alt_ab’ - an alternate sub-area abbreviation. In some areas, there area two accepted set of abbreviations. If the border group has an alternate abbreviation defined, this option allows the alternative abbreviation to be used as the *rowNames* in the data.

See the documentation on each border group for details on the full names, abbreviatations, optional alternate abbreviations and numeric identifiers defined for each area in the particular border group in the *areaNamesAbbrsIDs* R object.
The default is ‘ab’.

If a border group does not have one of these sets of name data, then the corresponding option will not be available.

The linkage between data and boundaries is accomplished using the strings in the statsDFrame column identified by *rowNamesCol* or if no *rowNamesCol* is specified in the row.names information of the *statsDFrame*,

The linkage values are validated to against the *areaNamesAbbrsIDs* data.frame for the specified border group for the *micromapST* call. If a sub-area in the border group is not referenced in the data, it is outlined on the map, but not colored. If the sub-area link cannot be matched, a warning message is generated and the execution of the package is stopped. Unless, the user has set the ‘noMatchIgnore’ call argument to TRUE and the data will be ignored.

The ‘alias’ option is implemented to support with the *USSeerBG* border group and data generated by the Seer Stat program. The registry column identifying the Seer Area contains the Seer Area name and additional information. This option along with the *alias* field in the border group allows the package to use the registry column as the linkage to the boundary data using a wildcard or partial string allows the registry column generated by Seer Stat to be used as the linkage. This option is controlled the *enableAlias* variable in the border group's *areaParms* data.frame.

rowNamesCol	allows the user to specify the data.frame column that contains the sub-area string to be used to link the data row to the boundary data for the sub-area. The *rowNames* option above specifies which name information (full name, abbreviation, id, alternate abbreviations) the sub-area string is matched to in the border group

name information. (see the border group documentation for more details.) The *rowNamesCol* value must be a column number or column name within the *stats-DFrame* data.frame. The default value for *rowNamesCol* is to use row.names of the passed data.frame.

sortVar   defines the column name(s) or number(s) in the *statsDFrame* data.frame to be used to sort the *statsDFrame* data.frames before creating the state micromap. A vector of column names or numbers can be used sort on multiple columns and to break ties.

For Example: `sortVar=c(4,5)` where columns 4 and 5 in the *statsDFrame* are used in the sort. If the user needs to sort the data based on information in the boxplot or time-series data, the best practice is to copy the data into the *statsD-Frame*.

ascend   a logical value. If *TRUE*, *sortVar* will be sorted in ascending order. If *FALSE*, *sortVar* will be sorted in descending order. The default value is *TRUE*.

title   A character vector with one or two character strings to be used as the title of the overall micromap plot page.

```
For example:
    \code{title = "micromapST Title"} or
    \code{title = c("title line 1","title line 2")}
```

plotNames   defines the type of state names to be displayed when an *id* glyph column requested. The options are: 'ab' or 'full'. 'ab' will display the sub-area abbreviations as defined in the border groups name information. 'full' will display the full sub-area name as defined in the border group name information. (For the U.S. Stata name information, the full name for District of Columbia is shown as Dist. of Col.. because of space limitations).
The default is 'ab'.

axisScale   defines the type of axis labels to be used and if scaling is applied. The acceptable values are 'o', 'w', 's', and 'sn'. 'o' is the original method using the pretty function limiting the values to the range of the data. 'w' is the default and uses the Wilkinson algorithm to generate the axis labels for a set of data, 's' uses the wilkinson algorithm, adjusts the range to cover the labels, and determines a scaling to apply to the labels. If 1000 is the scaling, "in the thousands" is added to the column titles. 'sn' uses the wilkinson algorithm, adjusts the range to cover the labels, and checks each value to determine if it should be scaled. If it is scales, the scale identifier is added as a suffix. (e.g., 1240334 become 1.24M) "w" is the wilkinson algorithm without any scaling.

staggerLab   is a logical variable the controls if the axis labels are staggered and drawn on on two lines instead of one. If set to FALSE (the default), the labels are not staggered. If TRUE, two lines are used to draw the axis labels, alternating labels on each line.

bordDir   specifies the path to a border group dataset (.rda) that is external to the *micromapST* package. The name of the border group is specified in the *bordGrp* parameter.

bordGrp   specifies which preloaded border group to use for the area names, abbreviations,
                   numeric identifier, and area boundary data. The supported border groups are
                   USStatesBG and *USSeerBG*. The default *bordGrp* value is *USSeerBG*.

dataRegionsOnly

                   specifies to only map regions containing data when *aP_Regions* is set to *TRUE*.
                   This indicates the name table contains the information to draw partial area maps
                   of regions. When set to *TRUE*, only regions within the area are drawn that con-
                   tain at least one sub-area with data provided by the caller. If *FALSE*, the entire
                   area is drawn. The default is *FALSE*. Retional boundaries are not required, but
                   suggested. See border group documentation for more details.

regionsB   when regional boundaries are provided, controls whether to overlay the bound-
                   aries on the map. If dataRegionsOnly is set to *TRUE* and only a subset of regions
                   will be mapped and regional boundaries are provided, regionsB is et to *TRUE*.
                   The default is *FALSE*. Independent of 'dataRegionsOnly', if set to *TRUE*, if
                   present, regional boundaries are overlayed on the map, If set to *FALSE*, no re-
                   gional boundaries are drawn. See border group documentation for more details.

grpPattern   The micromapST package generates the pattern of rows to panel groups auto-
                   matically. The pattern is based on a maximum of 5 rows per group and number
                   of rows can only decend from the edges toward the median row. The pattern gen-
                   erated by the package can be overriden by using the *grpPattern* to specify the
                   pattern to use as a numeric vector. The provided vector must pass the following
                   checks: a) must be a numeric vector, b) the sum of the values in the vector must
                   equal the number of rows in the user supplied statsDFrame data frame c) the
                   maximum number of rows per panel group is 5. d) the number of rows per panel
                   must be integers and decending from the outside to the middle of the vector.

maxAreasPerGrp   This parameter defines the maximum number of areas to be represented by a
                   group-row in the resulting linked micromap. The value can be 5 or 6.

ignoreNoMatches

                   The *micromapST* package will automatically handle situations where there is
                   no data for a sub-area in an area. However, it will stop processing if there is
                   data for a non-existing sub-area. To instruct the package to ignore rows of data
                   for sub-areas that do not have boundary information in the border group, set the
                   call argument 'ignoreNoMatches' to *TRUE* to get the package to detail the data
                   rows from the processing.

colors   is a vector containing a vector of 12 or 24 color names or values ("#xxxxxx") or
                   the name of a color palette. The vector of 12 or 24 color names or "#xxxxxx"
                   values are used to define the colors used for:

                   • The 6 colors in each group for the states and symbols in the glyphs. one
                     color per row (state).
                   • 1 color for the median state and glyphics,
                   • 1 foreground color for highlighted states in the map. This is used to high-
                     light states already referenced previously or have meaning depend on the
                     type of map requested. The usage is as follows:

                              "map"   - not used.
                              "mapcum"   - highlight areas previously referenced
                                             above.

```
                        "maptail"  - highlight areas previously referenced
                                     above the median row and highlight
                                     remaining states below the median row.
                        "mapmedian"- highlight all areas above the median
                                     in maps above the median row and
                                     highlight all areas below the median
                                     in maps below the median row.
```

- 2 accent colors for "mapmedian" sub-area colors for above median and be-
  low median.
- When an 12 additional colors are specified, they are used as the translucent
  colors in the 'tsconf' confidence intervals bands. If only 12 colors are
  provided, the additional 12 translucent colors are generated using a 20%
  transparent version of the original color.

```
                        e.g., \code{adjustcolors(colors,0.2)}
```

The only color palette support is a gray palette to permit publication of the linked
micromaps using a gray scale instead of color. By setting *colors* = "greys",
"grays", or "bw", the entire plot will be generated using gray scale that has
been balanced to maintain readability and reproduction without the use of color
printing. Additional color palettes may be supported in future releases.

If a *colors* vector is not provided, the package default colors will be used:

- 6 state colors: "red", "orange", "green", "greenish blue", "lavender"
- 1 median state color: "black"
- 1 highlighed states: "light yellow" for "map", "mapcum", "maptail"
- 2 highlighed states: "light red" and "light green" for "mapmedian" and
- 12 translucent colors using the above colors at 20%.

It is strongly recommended to use the default.

details   defines the spacing, line widths and many other details of the plot layout, struc-
                 ture and content; see *micromapGDefaults$details* for more details. Generally
                 details does not need to be specified, the default values will always be used and
                 are strongly recommended. However, in a few cases, it may be desireable to
                 turn off or disable a feature. In these cases, the user can specify just the specific
                 variable and value in a list and pass it to *micromapST* via the *details* parameter.
                 For example:

```
                     details=list(SNBar.Middle.Dot=FALSE,SBar.varht=FALSE)
```

                 The entire details variable list does not have to be passed. See the section on the
                 micromapGDefaults$details for more details.

Details

The *micromapST* function creates a linked micromap plot for data referencing a collection of ge-
ographic related areas, like the 50 US States and DC geographical areas or U.S. Seer Areas. The
function provides links from a US state map to several forms of graphical charts: dot ('dot'), dot

with confidence intervals ('dotconf'), dot standard error ('dotse'), dot with significance mark ('dotsignif'), arrow ('arrow'), bar chart ('bar'), time series ('ts'), time series with a confidence band ('tsconf'), horizontal stacked (segmented) bar ('segbar'), normalized bar ('normbar'), centered bar charts ('ctrbar'), scattered dot ('scatdot'), and box plots ('boxplot'). The data values for each column of graphs and each area are provided in the *statsDFrame* data.frame. The panelDesc data.frame specifies the type of chart, the column numbers in the *statsDFrame* with the statistics for the chart, column titles, reference values, etc. Additional data for boxplots and time series plots are provided through the panelData data.frame column.

Value

None

Author(s)

Daniel B. Carr, George Mason University, Fairfax VA, with contributions from Jim Pearson and Linda Pickle of StatNet Consulting, LLC, Gaithersburg, MD

References

Daniel B. Carr and Linda Williams Pickle, Visualizing Data Patterns with Micromaps, CRC Press, 2010
Linda Williams Pickle, James B. Pearson Jr., Daniel B. Carr (2015), micromapST: Exploring and Communicating Geospatial Patterns in US State Data., Journal of Statistical Software, 63(3), 1-25., https://www.jstatsoft.org/v63/i03/

See Also

micromapST, micromapSEER

Examples

```
######
#   micromapST - Example # 01 - map with no cumulative shading,
#     2 columns of statistics: dot with 95% confidence interval,
#     boxplot sorted in descending order by state rates, using
#     the default border group of "USStatesBG", with default symbols.
####
# load sample data, compute boxplot
TDir <- paste0(tempdir(),"/")   # get a temp directory for the output
                                # PDF files for the example.
cat("TempDir:",TDir,"\n")

   # replace this directory name with the location if you want to same
   # the output from the examples.

data(wflung00and95,wflung00and95US,wflung00cnty,envir=environment())

wfboxlist = boxplot(split(wflung00cnty$rate,wflung00cnty$stabr),
```

```
                          plot=FALSE)

# set up 4 column page layout

panelDesc01 <- data.frame(
  type=c("map","id","dotconf","boxplot"),
  lab1=c("","","State Rate","County Rates"),
  lab2=c("","","and 95% CI","(suppressed if 1-9 deaths)"),
  lab3=c("","","Deaths per 100,000","Deaths per 100,000"),
  col1=c(NA,NA,1,NA),col2=c(NA,NA,3,NA),col3=c(NA,NA,4,NA),
  refVals=c(NA,NA,wflung00and95US[1,1],NA),
  refTexts=c(NA,NA,"US Rate 2000-4",NA),
  panelData= c("","","","wfboxlist")
  )
panelDesc <- panelDesc01
# set up PDF output file, call package

pdf(file=paste0(TDir,"Ex01-WFLung-2000-2004-State-Dot-County-Box.pdf"),
    width=7.5,height=10)

micromapST(wflung00and95, panelDesc, sortVar=1, ascend=FALSE,
       title=c("Ex01-White Female Lung Cancer Mortality, 2000-2004",
               "State Rates & County Boxplots")
         )

dev.off()

##End Example 01

######
#   micromapST - Example # 02 - map with cumulative shading
#                  from top down (mapcum), arrow and bar charts,
#                  sorted in descending order by starting
#                  value of arrows (1950-69 rates) using default
#                  border group of "USStatesDF".  This
#                  example also provides custom colors for the
#                  linked micromaps, highlights, etc.
#
####
# Load example data from package.

data(wmlung5070,wmlung5070US,envir=environment())

panelDesc02 <- data.frame(
   type=c("mapcum","id","arrow","bar"),
   lab1=c("","","Rates in","Percent Change"),
   lab2=c("","","1950-69 and 1970-94","1950-69 To 1970-94"),
   lab3=c("MAPCUM","","Deaths per 100,000","Percent"),
   col1=c(NA,NA,"RATEWM_50","PERCENT"),
   col2=c(NA,NA,"RATEWM_70",NA)
 )

colorsRgb = matrix(c(                    # the basic 7 colors.
```

```
213,  62,  79,   #region 1: red      #D53E4F - Rust Red
252, 141,  89,   #region 2: orange      #FC8D59 - Brn/Org
253, 225, 139,   #region 3: green       #FEE08B - Pale Brn
153, 213, 148,   #region 4: greenish blue  #99D594 - med Green
 50, 136, 189,   #region 5: lavendar      #3288BD - Blue
255,   0, 255,   #region 6              #FF00FF - Magenta
.00, .00, .00,   #region 7: black for median #000000 - Black
230, 245, 152,   #non-highlighted foreground #E6F598 - YellowGreen
255, 174, 185,   # alternate shape upper   #FFAEB9 - Mauve
191, 239, 255,   # alternate shape lower   #BFEFFF - Cyan
242, 242, 242,   # lightest grey for non-referenced sub-areas  #F2F2F2
234, 234, 234),  # lighter grey for bkg - non-active sub-areas. #EAEAEA

  ncol=3,byrow=TRUE)

xcolors = c( grDevices::rgb(colorsRgb[,1],colorsRgb[,2],colorsRgb[,3],
                        maxColorValue=255),
             # set solid colors
           grDevices::rgb(colorsRgb[,1],colorsRgb[,2],colorsRgb[,3],64,
                        maxColorValue=255))
             # set translucent colors for time series.


# set up reference names for color set
names(xcolors) =c("rustred","orange","lightbrown","mediumgreen",
                "blue","magenta", "black","yellowgreen",
                "mauve","cyan","lightest grey","lighter grey",
                "l_rustred","l_orange","vlightbrown","lightgreen",
                "lightblue","l_black","l_yelgreen","l_mauve",
                "l_cyan","l_lightest grey","l_lighter grey")

pdf(file=paste0(TDir,"Ex02-WmLung50-70-Arrow-Bar.pdf"),width=7.5,height=10)

micromapST(wmlung5070,panelDesc02,sortVar=1,ascend=FALSE,
        title=c("Ex02-Change in White Male Lung Cancer Mortality Rates",
                "from 1950-69 to 1970-94-Diff colors"), colors=xcolors
            )

dev.off()
##End Example 02


######
#   micromapST - Example # 03 - Time Series Line Plots with
#     Confidence Bands maptail option highlights states from extremes
#     to middle state read in time series data set example using the
#     default border group of "USStatesDF".
####

data(TSdata,envir=environment())
temprates      <-  data.frame(TSdata[,,2])

# TSdata structure is array of size c(51,15,4),
# dimensions = 51 states, 15 years, (year label, point value, low limit,
# high limit)
```

```
panelDesc03   <- data.frame(
    type=c("maptail","id","tsconf","dot"),
    lab1=c("","","Time Series","Female"),
    lab2=c("","","Annual Rate per 100,000","Most Recent Rate (2010)"),
    lab3=c("","","Years","Deaths per 100,000"),
    lab4=c("","","Rate",""),
    col1=c(NA,NA,NA,15),
    panelData =c(NA,NA,"TSdata",NA)
    )
ExTitle   <- c("Ex03-Time Series with Confidence bands",
               "Annual Female Lung Cancer Mortality Rates, 1996-2010")

pdf(file=paste0(TDir,"Ex03-Time-Series-with-Conf.pdf"),
    width=7.5,height=10)

micromapST(temprates,panelDesc03,sortVar="P15",ascend=FALSE,
           title=ExTitle)

dev.off()

# end of example 03


######
#   micromapST - Example 04 - dot followed by a scatter dot columns
#     use same data as Example 3 to compare 1996 & 2010 rates
#     mapmedian option shades states above or below the median
#     (light yellow) using the default border group of "USStatesBG"
#
#   USES data loaded for Example 03 (temprates).
#
####

panelDesc04 <- data.frame(
    type=c("mapmedian","id","dot","scatdot"),
    lab1=c("","","Female Lung Cancer Mortality","Comparison of Rates"),
    lab2=c("","","Rate in 1996 (Sort Variable)",
                      "in 1996 (x axis) and 2010 (y axis)"),
    lab3=c("","","Deaths per 100,000","Deaths per 100,000 in 1996"),
    lab4=c("","","","Rate in 2010"),
    col1=c(NA,NA,1,1),
    col2=c(NA,NA,NA,15)
    )
ExTitle <- c("Ex04-Dot Plot for 1996, Scatter Plot Comparing 1996 to 2010",
             "Female Lung Cancer Mortality Rates")

pdf(file=paste0(TDir,"Ex04-Scatter-Dots.pdf"),width=7.5,height=10)

micromapST(temprates,panelDesc04,sortVar=1,ascend=FALSE,title=ExTitle)

dev.off()
```

```
##End Example 04

######
#   micromapST - Example 05 - horizontal stacked (segmented) bars
#     segbar plots the input data, normbar plots percent of total
#     package computes the percents from input data
#     input for the categories for each state must be in consecutive
#     columns of the input data.frame using the default border group
#     of "USStatesBG"
####

data(statePop2010,envir=environment())

panelDesc05 <- data.frame(
    type=c("map","id","segbar","normbar"),
    lab1=c("","","Stacked Bar","Normalized Stacked Bar"),
    lab2=c("","","Counts","Percent"),
    col1=c(NA,NA,"Hisp","Hisp"),
    col2=c(NA,NA,"OtherWBH","OtherWBH")
    )

pdf(file=paste0(TDir,"Ex05-Stkd-Bar-var-height.pdf"),
     width=7.5,height=10)

micromapST(statePop2010, panelDesc05, sortVar="OtherWBH", ascend=FALSE,
   title=c("Ex05-Stkd Bars: 2010 Census Pop by Race, Sorted by Cnt Other Race",
           "Cat-L to R: Hispanic, non-Hisp White, Black, Other-sn-varbar"),
     details=list(SNBar.varht=TRUE), axisScale="sn" )

dev.off()
## End Example 05

######
#
#   micromapST - Example 06 - horizontal stacked (segmented) bars
#     segbar plots the input data, normbar plots percent of total
#     package computes the percents from input data
#     input for the categories for each state must be in consecutive
#     columns of the input data.frame using the default border group
#     of "USStatesBG".
#
#     Turning off the variable bar height and the midpoint dot features
#     in the horizontal stacked bars (segmented)
#
#  USES data loaded for Example 05 above - statePop2010.
#
####

panelDesc06= data.frame(
      type=c("map","id","segbar","normbar"),
      lab1=c("","","Stacked Bar","Normalized Stacked Bar"),
      lab2=c("","","Counts","Percent"),
      col1=c(NA,NA,"Hisp","Hisp"),
```

```
        col2=c(NA,NA,"OtherWBH","OtherWBH")
    )

pdf(file=paste0(TDir,"Ex06-Stkd-Bar-fixedheight-nodot.pdf"),
    width=7.5,height=10)

micromapST(statePop2010,panelDesc06,sortVar=4,ascend=FALSE,
  title=c("Ex7-Stacked Bars: 2010 Census Pop by Race, Sorted by Other Race",
          "Cat-L to R: Hisp, non-Hisp White, Black, Other,ID-diamond"),
        details=list(SNBar.Middle.Dot=FALSE,SNBar.varht=FALSE,Id.Dot.pch=23)
     )
dev.off()

## End Example 06


######
#   micromapST - Example 07 - centered (diverging) stacked bars
#
#     National 8th grade Math Proficiency NAEP Test Scores Data for 2011
#     source: National Center for Education Statistics,
#     http://nces.ed.gov/nationsreportcard/naepdata/
#     bar segment values - % in each of 4 categories:
#          % < Basic, % at Basic, % Proficient, % Advanced
#     using the default border group of "USStatesBG".
####

data(Educ8thData,envir=environment())

# columns = State abbrev, State name, Avg Score, %s \<basic,
#           basic, proficient, advanced

panelDesc07 <- data.frame(
    type=c("map","id","dot","ctrbar"),
    lab1=c("","","Avg. Scores","Math Proficiency"),
    lab2=c("","","","<Basic, Basic, Proficient, Advanced"),
    lab3=c("","","","% to Left of 0            |    % to Right"),
    col1=c(NA,NA,"avgscore","PctBelowBasic"),col2=c(NA,NA,NA,"PctAdvanced")
  )
ExTitle <- c("Ex07-Stkd Bars:Educational Progress (NAEP) in Math-2011, 8th Grade",
             "Centered at Not-Prof vs. Prof")

pdf(file=paste0(TDir,"Ex07-Educ-Centered-Bar.pdf"),width=7.5,height=10)

micromapST(Educ8thData,panelDesc07,
             sortVar=3,
             title=ExTitle)

dev.off()

### End of example 07

#
```

```
#  Example # 08 - Maps Seer Registries using the micromapST function
#    with the bordGrp = "USSeerBG".
#

data(Seer18Area,envir=environment())

# set up 4 column page layout

panelDesc08 = data.frame(
  type=c("mapcum","id","dotsignif","arrow")
  ,lab1=c("","","Rate Trend APC", "Rate Change")
  ,lab2=c("","","Dot-Signif","2002-06 to 2007-11")
  ,lab3=c("","","","")
  ,col1=c(NA,NA,"RateTrendAPC","Rate20022006")
  ,col2=c(NA,NA,"pValue", "Rate20072011")
  )

ExTitle       <- c("Ex08 - SeerStat Data - 2002-6 and 2007-11",
                     "Dot with Signif., Arrow and Bar")

pdf(file=paste0(TDir,"Ex08-DotSignificants.pdf"),width=7.5,height=10)

micromapST(Seer18Area,panelDesc08,
        sortVar="Rate20022006",ascend=FALSE,
        title=ExTitle,
        rowNames="alias",rowNamesCol='Registry',
        bordGrp="USSeerBG",
        plotNames="ab")

dev.off()
#
#  Both calls are effectively identical.
#
#### End of examples 08

#####
#
#  Example # 10 - Counties in Kansas on an 11 x 17 page
#

# set up 4 column page layout

data(KansPopInc,envir=environment())

panelDesc10 = data.frame(
  type=c("map","id","dot",        "dot")
  ,lab1=c("",       "",  "Population", "Average Inc.")
  ,lab2=c("",       "",  "in 2000",    "per year")
  ,lab3=c("",       "",  "People",     "")
  ,col1=c(NA,       NA,  "Pop",        "AvgInc")
  )

ExTitle        <- c("Ex10 - Kansas Pop data 11x17",
```

```
                              "Current Pop and Average Inc - scaling=e")

    pdf(file=paste0(TDir,"Ex10-Kansas Population and Inc-11x17.pdf"),
        width=10, height=16)
                # tabloid size page (11x17) to handle 105 counties.

    #  Use default scaling = "e" and no staggered labels,
    #  Use full county names for data to boundary matching,
    #  but presented abbreviated county names
    #  in "id" glyphic column with large page.

    micromapST(KansPopInc, panelDesc10,
            sortVar=c("AvgInc","Pop"), ascend=FALSE,
            title=ExTitle,
            rowNames="full", rowNamesCol='County',
            bordGrp="KansasBG",
            plotNames="ab")

    dev.off()
    #
    #
    #### end of # Ex 10


    #####
    #
    #   micromapST - Example 11 - A linked micromap of the counties
    #       of the state of New York state using the border group
    #       "NewYorkDF".  The pop/inc data is shown using two dot glyphs,
    #       an arrow and bar glyph (2010 Population, an arrow showing the
    #       change in population from 2000 to 2010, Population in 2000,
    #       and a bar showing the amount of the change.)
    #
    #####

    data(nyPopData,envir=environment())

    nyPopData$Dif00_10 <- nyPopData$Pop_2010 - nyPopData$Pop_2000

    # set up 6 column page layout with colSize

    panelDesc11 <- data.frame(
      type=c("map","id","dot",          "arrow",        "dot",     "bar")
      ,lab1=c("",  "",  "Population in", "Increase from","Pop 2005","Incre")
      ,lab2=c("",  "",  "2010",          "2000",         "",        "2000to2010")
      ,lab3=c("",  "",  "",              "",             "",        "")
      ,col1=c(NA,  NA,  "Pop_2010",      "Pop_2000",     "Pop_2000","Dif00_10")
      ,col2=c(NA,  NA,  "",              "Pop_2010",     "",        "")
      ,colSize=c(NA,NA, 15,              20,             5,         20)
      )

    ExTitle      <- c("Ex11 - New York Population data",
                        "2010 Pop and since 2000-colSize,sn,stag")
```

```
pdf(file=paste0(TDir,"Ex11-New York Pop 2010 and Change-sn colSize.pdf"),
     width=7.5, height=10.5)

micromapST(nyPopData, panelDesc11,
        sortVar="Pop_2000", ascend=FALSE,
        title=ExTitle,
        rowNames="full",rowNamesCol="Area",
        axisScale="sn", staggerLab=TRUE,
        bordGrp="NewYorkBG"
      )

dev.off()
#
#
#### End of Examples 11


#####
#   micromapST - Example 12 - A linked micromap of the counties of
#       the state of Maryland using the border group "MarylandBG".
#       The MarylandPopInc data is shown using two dot glypics - current
#        population and average increase per county.
#       A "maptail" state map is used to show the counties in relationship
#       to the median county as sorted by the 1970 population.
#####

data(mdPopData,envir=environment())

# set up 5 column page layout

panelDesc12 = data.frame(
  type=c("maptail","id","dot","dot","arrow")
  ,lab1=c("","","Population", "Population","Change")
  ,lab2=c("","","in 1970","in 2000", "from 1970 to 2000")
  ,lab3=c("","","","","")
  ,col1=c(NA,NA,"X1970","X2010","X1970")
  ,col2=c(NA,NA,"","","X2010")
 )

ExTitle       <- c("Ex12 - Maryland Population-map",
                      "1970 and 2010 Pop and Change,stag,sn")

pdf(file=paste0(TDir,"Ex12-MD Pop 1970 and 2010 plus change-map.pdf"),
     width=7.5, height=10.5)

micromapST(mdPopData, panelDesc12,
        sortVar=2, ascend=FALSE,
        title=ExTitle,
        rowNames="full", rowNamesCol='County',
        bordGrp="MarylandBG",
        axisScale="sn", staggerLab=TRUE,
        plotNames="ab")
```

```
dev.off()
#
#
#### End of Example 12


#####
#    micromapST - Example 13 - A linked micromap of the provinces,
#        municipalities, autonomous regions and special administrative
#        regions of China using the border group of "ChinaDF".
#        The ChinaPopInc data is shown using two dot glypics - current
#        population and average increase per area.
#####

data(cnPopData,envir=environment())

# set up 4 column page layout

panelDesc13 = data.frame(
  type=c("map","id","dot","bar")
  ,lab1=c("","","Population", "Population")
  ,lab2=c("","","in 2013","in 2013")
  ,lab3=c("","","","")
  ,col1=c(NA,NA,"pop2013","pop2013")
  )

ExTitle       <- c("Ex13 - China Population",
                      "in 2013 by area")

pdf(file=paste0(TDir,"Ex13-China 2013 Population.pdf"),
    width=7.5, height=10.5)

micromapST(cnPopData, panelDesc13,
        sortVar="pop2013", ascend=FALSE,
        title=ExTitle,
        rowNames="full", rowNamesCol='area',
        bordGrp="ChinaBG",
        plotNames="full")

dev.off()
#
#
#### End of Example 13



#####
#    micromapST - Example 14 - A linked micromap of the districts
#        of the city Seoul South Korea, using the border group of
#        "SeoulSKoreaBG". The included SeoulPopData dataset provides
#        population and district area statistics for 2012.
#        The micromapST generates two glyphics, a sorted dot
#        glyphic based on the population and a bar graph based on
#        the area.
```

```
#####

data(SeoulPopData,envir=environment())

# set up 4 column page layout

panelDesc14 = data.frame(
  type=c("map","id","dot","bar")
  ,lab1=c("","","Population", "Area")
  ,lab2=c("","","in 2012","in 2012")
  ,lab3=c("","","","sqkm")
  ,col1=c(NA,NA,"Pop.2012","Area")
  )

ExTitle       <- c("Ex14 - Seoul Population",
                    "in 2012 by district")

pdf(file=paste0(TDir,"Ex14-Seoul 2012 Population.pdf"),
     width=7.5, height=10.5)

micromapST(SeoulPopData,panelDesc14,
        sortVar=3, ascend=FALSE,    # sort based on the population
        title=ExTitle,
        rowNames="full", rowNamesCol='District',
        bordGrp="SeoulSKoreaBG",
        plotNames="full"
      )

dev.off()
#
#
#### End of Example 14

#####
#   micromapST - Example 15 - A linked micromap of the counties in the
#       state of Utah.  The UtahPopData data is shown using two dot glypics
#       - current population and average increase per area.
#####

data(UtahPopData,envir=environment())

#
#  Get population differences from 2011 to 2001 and 1991.
#   Data contains ",".  The comma's must be removed and values are
#      converted to numbers.
#   If data is factors, need to add "as.character()" function
#      to the formula below.

UtahPopData2 <- as.data.frame(sapply(UtahPopData,
                       function(x) gsub(",","",x)),stringsAsFactors=FALSE)

UtahPopData2$Del1101 <- as.numeric(UtahPopData2$X2011)
                         - as.numeric(UtahPopData2$X2001)
```

```
UtahPopData2$Del0191 <- as.numeric(UtahPopData2$X2001)
                         - as.numeric(UtahPopData2$X1991)



# set up 5 column page layout

panelDesc15 = data.frame(
  type=c("map","id","dot","arrow","arrow")
  ,lab1=c("","","Population", "2001-2011","Chg 1991-2001")
  ,lab2=c("","","in 2011","pop change","pop change")
  ,lab3=c("","","","","")
  ,col1=c(NA,NA,"X2011","X2011","X2001")
  ,col2=c(NA,NA,NA,"X2001","X1991")
  )

ExTitle       <- c("Ex15 - Utah county population 2011",
                     " and changes last two decades,sn")

pdf(file=paste0(TDir,"Ex15-Utah Population.pdf"),
     width=7.5,height=10.5)

micromapST(UtahPopData, panelDesc15,
        sortVar="X2011", ascend=FALSE,
        title=ExTitle,
        rowNames="full",rowNamesCol='County',
        axisScale="sn",
        bordGrp="UtahBG",
        plotNames="ab"
      )

dev.off()
#
#
#### End of Example 15

####
#
#  Example 16 - use of state.x77 data table as data source
#     Data does not contain a row for DC, a missing sub-area.
#     Example also uses a smaller then 7.5 x 10 graphic space.
#

data(state,envir=environment())

stateData <- as.data.frame(state.x77)

rownames(stateData) <- state.abb

panelDesc16 <- data.frame(type = c("maptail", "id", "dot"),
                    lab1 = c("", "", "Murder"),
                    lab3 = c("", "", "Murders per 100K Population"),
                    col1 = c(NA, NA, 5))
```

```
ExTitle <- c("Ex16 - LM Plot of Murders in the United States",
                        "No DC row entry.")

pdf(file = paste0(TDir,"Ex16_state.x77_no_DC.pdf"),
    width = 5, height = 9)

micromapST(stateData, panelDesc16,
            sortVar = 5, ascend = FALSE,
            title = ExTitle)

dev.off()

#
#  Example 17 - US state map based on data from state.x77 table with
#      DC row added to complete data.frame, but with missing values (NAs).
#      The DC row will be sorted to the bottom of the list size
#      it does not contain any data.
#
#   Used data and the panelDesc data.frames (stateData and panelDesc16)
#   used in example 16.
#

# add DC as 51st state with missing data "NA" to stateData.

stateData <- rbind(stateData, DC = rep(NA, 8))
                    # missing values for DC row.

ExTitle <- c("Ex17 - LM Plot of Murders in the United States",
                    "DC row added with NA, decending.")

pdf(file =paste0(TDir,"Ex17_state.x77_DCasNA_D.pdf"),
     width = 5, height = 9)

micromapST(stateData, panelDesc16,
            sortVar = 5, ascend = FALSE,
            title = ExTitle)

dev.off()

#
#
##### end of example # 17

####
#
#  Example 18 - use of Africa population data as data source
#      Demonstrates support for vertical oriented geographical
#        areas.
#

data(AfricaPopData,envir=environment())

panelDesc18 <- data.frame(
```

```
                type = c("map", "id", "dot",      "dot",          "dot"),
                lab1 = c("",       "","Population","Percentage Of","Est x2 Time"),
                lab3 = c("",       "","People",    "Total",        "Years"),
                col1 = c(NA,       NA,"Projection","PercOf",       "Est2Time")
             )


pdf(file = paste0(TDir,"Ex18_Africa Micromap-11x17.pdf"),
     width = 11, height = 17)


ExTitle <- c("Ex18-Africa Population Data",
               "Sorted by Population on 11x17")


micromapST(AfricaPopData, panelDesc18,
          sortVar = "Projection", ascend = TRUE,
          title = ExTitle,
          rowNames = "ab", rowNamesCol = "Abbr",
          bordGrp = "AfricaBG" )

dev.off()
#
##### End of Example 18

##### Other data available
#
#  Other dataset available in the package to experiment with other
#   border groups are:

#  Data for segmented stacked bar charts
data(Educ8thData)
str(Educ8thData)

#  Data for time series gryph
data(TSdata)
str(TSdata)

#  US states population data
data(statePop2010)
str(statePop2010)

#  White Female Lung data
data(wflung00and95)
str(wflung00and95)

data(wflung00and95US)
str(wflung00and95US)

data(wflung00cnty)
str(wflung00cnty)

data(wmlung5070)
str(wmlung5070)
```

```
data(wmlung5070US)
str(wmlung5070US)

#
#
#####

unlink(TDir)
```

NewYorkBG	*NewYorkBG border group datasets to supports building micromaps for the 62 counties in the state of New York.*

Description

The *micromapST* function has the ability to generate linked micromaps for any geographical area. To specify the geographical area, the 'bordGrp' call argument is used to specify the border group dataset for the geographical area. The *NewYorkBG* border group dataset supports creating linked micromaps for the 62 counties in the state of New York. When the 'bordGrp' call argument is set to *NewYorkBG*, the appropriate name table (county names and abbreviations) and the 62 sub-areas (countries) boundary data is loaded in *micromapST*. The user's data is then linked to the boundary data via the county's name, abbreviated or ID based on the table below.

Details

The *NewYorkBG* border group dataset contains the following data.frames:

areaParms - contains specific parameters for the border group

areaNamesAbbrsIDs - containing the names, abbreviations, numerical identifier and alias matching string for each of the 62 counties in New York

areaVisBorders - the boundary point lists for each area.

L2VisBorders - the boundaries for an intermediate level. For this border group, this boundary data.frame is not used and set to L3VisBorders as a place holder.

RegVisBorders - the boundaries for an intermediate level. For this border group, this boundary data.frame is not used and set to L3VisBorders as a place holder.

L3VisBorders - the boundary of the state of New York.

The New York county border group contains 62 county sub-areas. Each county has a row in the areaNamesAbbrsIDs data.frame and a set of polygons in the areaVisBorders data.frame datasets. No regions are defined in the New York county border group, so the L2VisBorders dataset is not used and the regions option is disabled. The L3VisBorders dataset contains the outline of the state of New York.

The details on each of these data.frame structures can be found in the "bordGrp" section of this document. The *areaNamesAbbrsIDs* data.frame provides the linkages to the boundary data for each sub-area (county) using the fullname, abbreviation, and numerical identifier for each country to the *<statsDFrame>* data based on the setting of the 'rowNames' call argument.

A column or the data.frame row.names must match one of the types of names in the *areaNamesAbbrsIDs* data.frame name table. If the data row does not match a value in the name table, an warning is issued and the data is ignored. If no data is present for a sub-area (county) in the name table, the sub-area (county) is mapped but not colored.

The following are a list of the names, abbreviations, and IDs for each country in the *NewYorkBG* border group.

name	ab	id
Albany	ALBA	36001
Allegany	ALLE	36003
Bronx	BRON	36005
Broome	BROO	36007
Cattaraugus	CATT	36009
Cayuga	CAYU	36011
Chautauqua	CHAU	36013
Chemung	CHEM	36015
Chenango	CHEN	36017
Clinton	CLIN	36019
Columbia	COLU	36021
Cortland	CORT	36023
Delaware	DELA	36025
Dutchess	DUTC	36027
Erie	ERIE	36029
Essex	ESSE	36031
Franklin	FRAN	36033
Fulton	FULT	36035
Genesee	GENE	36037
Greene	GREE	36039
Hamilton	HAMI	36041
Herkimer	HERK	36043
Jefferson	JEFF	36045
Kings	KING	36047
Lewis	LEWI	36049
Livingston	LIVI	36051
Madison	MADI	36053
Monroe	MONR	36055
Montgomery	MONT	36057
Nassau	NASS	36059
New York	NEWY	36061
Niagara	NIAG	36063
Oneida	ONEI	36065
Onondaga	ONON	36067
Ontario	ONTA	36069
Orange	ORAN	36071

Orleans	ORLE	36073
Oswego	OSWE	36075
Otsego	OTSE	36077
Putnam	PUTN	36079
Queens	QUEE	36081
Rensselaer	RENS	36083
Richmond	RICH	36085
Rockland	ROCK	36087
St. Lawrence	STLA	36089
Saratoga	SARA	36091
Schenectady	SCHE	36093
Schoharie	SCHO	36095
Schuyler	SCHU	36097
Seneca	SENE	36099
Steuben	STEU	36101
Suffolk	SUFF	36103
Sullivan	SULL	36105
Tioga	TIOG	36107
Tompkins	TOMP	36109
Ulster	ULST	36111
Warren	WARR	36113
Washington	WASH	36115
Wayne	WAYN	36117
Westchester	WEST	36119
Wyoming	WYOM	36121
Yates	YATE	36123

There are no alternate abbreviations or regions assocated with counties in New York border group.

The *id* field value is the U. S. state and county FIPS code.

The 'rowNames' = "alias" or "alt_ab", the 'regionB' and 'dataRegionsOnly' features are not supported in the *NewYorkBG* border group.

---

nyPopData	*Test data for the New York border Group*

Description

This dataset contains the 2014 population and average income per person in each of the New York counties.

Usage

```
data(nyPopData)
```

Format

A data frame with 63 rows, 1 for each county and 14 variables per county:

Area a character vector containing the New York county and buorgh name.

Est_2000 a numeric vector of the county's estimated population for 2000.

Est_2001 a numeric vector of the county's estimated population for 2001.

Est_2002 a numeric vector of the county's estimated population for 2002.

Est_2003 a numeric vector of the county's estimated population for 2003.

Est_2004 a numeric vector of the county's estimated population for 2004.

Est_2005 a numeric vector of the county's estimated population for 2005.

Est_2006 a numeric vector of the county's estimated population for 2006.

Est_2007 a numeric vector of the county's estimated population for 2007.

Est_2008 a numeric vector of the county's estimated population for 2008.

Est_2009 a numeric vector of the county's estimated population for 2009.

Est_2010 a numeric vector of the county's estimated population for 2010.

Pop_2000 a numeric vector of the county's actual population for 2000.

Pop_2010 a numeric vector of the county's actual population for 2010.

Details

This dataset was pulled from the New York government website in January, 2015. It contains the actual and estimate populations of each county from 2000 to 2010.

---

panelDesc   *micromapST panel description data.frame structure*

Description

The panelDesc data.frame provides the *micromapST* function with the information required to process the *statsFrame* data and panelData data.frames and to generate the required linked micromap plot.

It specifies which columns in the *statsFrame* data.frame contain the data for each glyph column, the column types, labels, reference values and text, and when more complex data is needed by a glyph (boxplot and time series) what the name of the data structure..

```
Example
  panelDesc = data.frame(
      type=c("mapcum","id","dotconf","dotconf"),
      lab1=c("","","White Males","White Females"),
      lab2=c("","","Rate and 95% CI","Rate and 95% CI"),
      lab3=c("","","Deaths per 100,000","Deaths per 100,000"),
      col1=c(NA,NA,"Rate",9),
      col2=c(NA,NA,4,11),
```

```
                    col3=c(NA,NA,5,12),
                    colSize=c(NA,NA,5,5),
                    refVals=c(NA,NA,NA,wflungUS[,1]),
                    refTexts=c(NA,NA,NA,"US Rate"),
                    panelData=c("","","","")
```

The panelDesc data.frame (which does not have to be named "panelDesc", any name will do) provides the means of defining how many columns to create, the type of glyph per column, where the data required by the glyph is located in the *statsFrame* (column number or name) or the name of a supplimental data structure when the glyph is boxplots or time series (via the *panelData* list entry), the column titles, and the column's reference value and label for the link micromap generation.

In the following description the term "AREA" represents the geographic unit being mapped and associated with data in the *statsFrame*. The naming used must match the border group specified. If the border group of "USStatesDF" is used, the areas are U.S. States and DC and 51 data rows must be present. If the border group of "USSeerDF" is used, the areas are U.S. Seer areas as defined by NCI and the number of data rows can be 9, 11, 13, 17 or 18. In all cases, the abbreviations and names defined in the border group dataset must be used in preparing the statsFrame and panelData structures.

Glyph Types

The *type* vector defines the type of glyph to be used for each column. The available glyphs are:

Map types: "map", "mapcum","maptail","mapmedian"

State or Area ID and/or Name: "id"

Ranking: "rank"

Graphical Type: "dot", "dotse","dotconf", "dotsignf", "bar", "arrow", "ts", "tsconf","scatdot", "segbar", "normbar", "ctrbar", "boxplot"

The following provides a description of each panel type:

map - US map with active areas colored

mapcum - US map with active areas colored and previously active area highlighted generating an accumulation from top to bottom

maptail - US map with active areas colored and previously active area highlighted until the median area, then the reverse to the end (areas that have not been active are highlighted.)

mapmedian - US map with active areas colored. Maps above the median area have areas with values above the median highlighted. Maps below the median area have areas with values below the median highlighted. This helps define the above and below median area groups.

id - generates a column with a colored identifier (a square) and the area or area name or abbreviation.

rank - number the area in rank order, sequentially.

arrow - an arrow between two values with a head.

bar - a single bar chart.

boxplot - a boxplot per area with box, upper and lower whiskers and outliers.

dot - a dot for a single value.

dotse - a dot for a single value and its standard error.

dotconf - a dot for a single value and its confidence interval.

dotsignf - a dot for a single value with an indicator of its significants.

ts - a time series line for up to 30 sets "x" and "y" values for each area.

tsconf - a time series line for a up to 30 sets of "x", "y" and upper "y" and lower "y" values as a confidence interval band for each area.

segbar - a horizontal stacked (segmented) bar plot starting at 0 for 2 to 9 bars.

normbar - a stacked bar plot where the data is normalized for each area by dividing the bar segment values by the sum of the values for all of the bars. Up to 9 bars are supported.

ctrbar - a stacked bar plot where the bar segments are centered around the 0.Up to 9 bars are supported.

scatdot - a set of points for each area with an "x" and "y" value.

Labels (Column Headers and Footers)

micromapST supports up to 3 column labels or titles: *lab1*, *lab2* and *lab3*, where *lab1* and *lab2* are header titles for the column. *lab3* is the footer title for the column. All titles are optional. *lab3* is used to indicate the unit of measure at the bottom of the columns, but is not limited to this use. For example:

```
    lab1=c("Col1-Title", "Col2=Title", "Col3-Title" ) # 1st title for columns
    lab2=c("Col1-Sub",   "Col2-Sub",   "Col3-Sub"   ) # 2nd title for columns
   lab3=c("Col1-Footer","Col2-Footer","Col3-Footer") # Footer title for columns
```

lab4 is used only when time series or scatter dot glyphs are used to provide a Y axis title for the column. All label-title vectors are optional and only required when an title or label is needed.

Data References

Depending on the type of glyph selected for the column, 1 to 3 data values for each area may be required: The *col1*, *col2* and *col3* vectors serve as indexes to columns in the *statsFrame* data.frame passed in the arguments of the *micromapST* function call. The values can be either the numeric number of the row in *statsFrame* data.frame or the column name. If no index is required, the entry should be set to *NA*.

If the glyph requires one value, then only the *col1* index is used and the *col2* and *col3* indexes are set to *NA* if present . If 2 values are required, then *col1* and *col2* indexes are used and the *col3* index is set to *NA*, if present. If 3 values are required, then *col1*, *col2*, and *col3* indexes are used.

The *statsFrame* column indexes can be provided as an integer or the column name. If the integer value is less than 1 or greater than the number of columns in *statsFrame* or a column name is used that does not exist in statsFrame, the *micromapST* function will stop and generate an error message.

Glyph Name	Meaning	col1	col2	col3	panelData
arrow	Arrow	Beginning Values	Ending Values (arrow head)	NA	NA

bar	Horizontal bar	Bar end values (length)	NA	NA	NA
segbar	Horizontal stacked bar	Values for first (left-most) segment (length)	Values for the last (right-most) bar segment (length)	NA	NA
normbar	Horizontal stacked bar, normalized to total 100%	Values for first (left-most) bar segment (length)	Values for last (right-most,bar segment (length)	NA	NA
ctrbar	Horizontal stacked bar, centered on the middle bar	Values for first (left-most) bar segment (length)	Values for last (right-most,bar segment (length)	NA	NA
boxplot	Horizontal box plot	NA	NA	NA	Name of output list from call to boxplot(...plot=F)
dot	Dot	Values for dots	NA	NA	NA
dotconf	Dot with confidence interval line	Values for dots	Values of lower limits	Values for upper limits tab	NA
dotse	Dot with line length +/- standard	Values for dots	Standard errors	NA	NA

error

dotsignf	Dot overprinted if not significant	Values for dots	P value associated with dot	NA	NA
scatdot	Scater plot of dots	Values on horizontal (x) axis	Values on vertical (y) axis	NA	NA
ts	Time Series (line) plot	NA	NA	NA	Name of array with dimensions of c(51,t,2), where t = # of time points (max 15), x values in [,,1], y values in [,,2]
tsconf	Time Series (line) plot with confidence limits	NA	NA	NA	Name of array with dimensions of c(51,t,4), as ts lower limit is [,,3] amd the upper limit is [,,4]

The panelData data.frame is only used when a glyph requires more data per area than can be provided by the *statsFrame* columns. Only glyphs using this vector are boxplots and time series.

In the case of the 'boxplot' glyph, the boxplot function with plot=F is used to generate the boxplot statistical details for each area. The name of the resulting list of 51 sets of boxplot statistics (one for each area) is placed in the panelData data.frame element for the boxplot column.

For the time series and time series with confidence interval, the glyphs require a 3 dimensional array of data. The first dimension ([*area*,,]) represents the areas. The second dimension ([,t,]) ranges from 2 to *n*. There is no upper limit, but 200-250 samples is a practical limit. One for each data point. The third dimension ([,,*v*]) provides the values at data point var*t* for area *st*. [,,var1] is the *x* axis value. For time series, is usually just the value 1 to *n* to order the *y* values. [,,*2*] is the median *y* value. For time series with confidence intervals: [,,*3*] is the lower value *y* and [,,*4*] is the upper value *y*.

Reference Lines

Reference lines can be created in arror, bar, dot, dotconf, dotse, and segbar glyphs by specifying

the reference values in the *RefVal=* vector. A label appearing at the bottom of the column can be specified using the *RefTxt=* vector in the *panelDesc* data.frame.

Format

The parameters in the panelDesc data.frame structure are:

type= The types of graphics for each column of panels can be specified by the following keywords in the "type variable":

The following are the type of glyphs that can be specified in the type vector:

Map types: "map", "mapcum","maptail","mapmedian"

State ID and/or Name: "id"

Glyph Type: "dot", "dotse","dotconf", "dotsignf", "bar", "arrow", "ts", "tsconf","scatdot", "segbar", "normbar", "ctrbar", "boxplot"

The following provides a description of each panel type:

map - a non-highlighted map

mapcum - maps show the accumulated areas top to bottom

maptail - maps show the accumulated areas from the top and bottom toward median area.

mapmedian - the maps above the median highlight the areas above the median area and maps below the median highlight areas below the median area based on the sorting variable.

id - generates a column with a color identifier (a filled in square) and the area abbreviation or name. The plotNames parameter in the *micromapSEER* call controls whether the area's full name or 2 character abbreviation is displayed.

rank - sequentially number areas from 1 (highest rank) to "n" (lowest rank)

arrow - an arrow from value 1 to value 2 with value 2 the head of the arrow.

bar - a bar for a single set of values, The values can be positive or negative.

boxplot - a boxplot for each area using a data.frame generated by the boxplot function with plot=F. The name of the boxplot data.frame is passed to *micromapSEER* using the *panelData* vector.

dot - a dot for a single value using one set of values.

dotse - a dot for a single value and its standard error using two values.

dotconf - a dot for a single value and its confidence interval using three values.

dotsignf - a dot for a single value overlaid if value is not significant using two values: value for dot and P value.

ts - a time series line plot for each area. The glyph use the *panelData* vector to get the name of a three (3) dimensional array the data for the plot. The array contains one entry per area, 1 to 30 data points and the *x* and *y* values. See section on *panelData* below for more details. A reasonable upper limit to the number of points is between 200-300.

tsconf - a time series line and confidence interval band for each area. The glyph use the *panelData* vector to get the name of a three (3) dimensional array the data for the plot. The array contains one entry per area, 1 to *n* data points and the *x*, *y*, *lower y* and *upper y* values. See section on *panelData* below for more details. A reasonable upper limit to the number of points is between 200-300.

segbar - a horizontal stacked (segmented) bar plot starting at 0 using data in the statsFrame data.frame. The *col1* and *col2* columns are used to indicate the first and last columns in

the statsFrame data.frame that contain the contiguous bar segment values (lengths). For example: The data for a 5 segment bar glyph is in columns 4 through 8 in the *statsFrame* (5 columns). *col1* is set to 4 to identify the first column and *col2* is set to 8 to identify the last column in the sequence. Column names may be used, but the column identified in *col1* must preceed the column identified in *col2*.

normbar - a stacked bar plot where the data is normalized for each area by dividing the bar segment values by the sum of the values for all of the bars. The stacked bar plot for each area then ranges from 0 to 100% (edge to edge). The *col1* and *col2* columns are used to identify the first and last columns for bar data in the *statsFrame* in the same way as for the "segbar" glyph (see above.)

ctrbar - a stacked bar plot where the bar segments are centered around the middle of the data. If there is an even number of segments, the 0 point is between the lower half and the upper half of the segments. If there is an odd number of segments, the center is the midpoint of the middle segment. The other segments are plotted to the left and right of the center point. The *col1* and *col2* columns are used to indicate the first and last columns in the *statsFrame* data.frame that contain the contiguous bar segment values. (See "segbar" type above for more information.)

scatdot - a set of 51 points with an *x* and *y* value per area. All points are plotted in each panel with the key areas in the panel highlighted. *col1* indicates *statsFrame* column containing the *x* values and *col2* indicates the column containing the *y* values.

Example: type=c("id","map","rank", "boxplot") To specify a *micromapSEER* with three columns, left to right, containing the area label, a map and a boxplot.

col1=, *col2=*, *col3=* Vectors of index numbers or names of columns in *statsFrame* data.frame to be used as data for graphics. The uses of these three vectors are defined below:

any "map" type, id, boxplots, ts, and tsconf glyphs do not use the *col1*, *col2*, or *col3* vectors to locate data in the *statsFrame* data.frame. If these vectors are present, the corresponding entires should be NA for the respective columns.

dot uses *col1* to specify a single data column in *statsFrame* data.frame to be ploted.

bar uses *col1* to specify the data column in *statsFrame* data.frame for the length of the bar. The data value can be positive or negative.

dotse uses *col1* and *col2* to specify the data columns in *statsFrame* data.frame to be used as the estimate and standard error values, respectively.

dotsignf uses *col1* and *col2* to specify the data columns in *statsFrame* data.frame to be used as the value for the dot and its associated P value.

arrow uses *col1* and *col2* to specify the data columns in statsFrame data.frame for the beginning and end values of the arrow.

segbar, normbar, ctrbar uses *col1* and *col2* to specify the first and last columns in the *statsFrame* data.frame. The *statsFrame* data.frame columns from *col1* to *col2* are used for the length values of each bar in the glyph. *col1* must preceed *col2* in the *statsFrame* data.frame. The minimum number of data columns is 2 columns with a maximum of 9 columns.

scatdot uses *col1* and *col2* to specify the *x* and *y* values respectivefully for a dot for each of the 51 areas and DC in a scatter dot plot.

dotconf uses *col1*, *col2*, and *col3* to specify the data columns in *statsFrame* data.frame for the estimate value, lower confidence interval, and upper confidence interval values.

See the table above.

colSize= A numeric vector used to specify the proportional width size of a glyph column in relation to all other glyph columns. If used, values must be included for all glyph columns except for the map and id glyphs, which are fixed width columns. The width of a glyph column is determined by summing all of the colSize values and dividing the sum into the value for each glyph column to yield a percentage of the available width to be allocated to each column. For example: colSize=c(NA,NA,10,10,5,15), does not affect columns 1 and 2. The percentages for columns 3 through 6 are 25%, 25%, 12.5% and 37.5%. If 4 inches of space is available, the columns will be allocated: 1, 1, 0.5, and 1.5 inches. The column widths are still regulated by the minimum and maximum column widths set in the package. If a value is missing for non-map or id glyph, the package will a value equal to the average of the provided values.

lab1=, *lab2=* Character vectors provide the two column labels (titles) lines at the top of each column. If no label is required, use *""* for a blank line.

lab3= Character vector used as a label at the bottom of each column. This is typically used to show units of measure. If no label is required, use *""* for a blank line.

lab4= Character vector used as the vertical (y) axis label for *ts*, *tsconf*, and *scatdot* glyphs. If no label is required, use *""* for a blank line.

refVals= Is a list of object names providing the reference values for each graphic column. The reference value is displayed as a dashed vertical line for each panel in the specified column.

refTexts= Is a list of 1 or 2 labels to be displayed at the bottom of each column to identify the reference value.

panelData= List of object names containing the boxplot data list and/or an array of time series data for each area. If boxplot and time series data are not used in a column, then associated object names should be NA.

For 'boxplot' data, each row name in the boxplot list must be the area abbreviation (2 character) for the area associated with the data. There must be 51 rows in the data.frame. Each row must be data produced by the 'boxplot' function. The area abbreviation must be used as the boxplot$names to be able to associate the individual boxplots to each area. The *names* attribute must contain the area name.

For the time series glyph (ts), the data must be a three (3) dimensional array. The first dimension $[st,,]$ represent one entry for each area (1 to 51). The second dimension $[,t,]$ indexes up to 30 data points for the area. The third dimension $[,,v]$ are the data point values. $[,,var\{1\}]$ is the *x* value and $[,,2]$ is the median *y* value for the data point. The rownames associated with the first dimension must be the area abbreviationss to link the elements of this structure the presentation order of the areas.

For the time series with confidence intervals glyph (tsconf), the array is extended to include: $[,,3]$ and $[,,4]$ for the *lower y* and *upper y* values.

For time series data, the order of the first dimension of the array must match the area order in the *statsFrame*. For example, the data in dataArray$[1,,]$ is the the area identified in statsFrame$[1,]$

Details

The panelDesc data.frame is used to describe the content of the micromapST plot to the function. It contains the index of the data in the statsFrame data.frame, the types of graphics to be used in each column, titles, column headers, reference values and labels, etc.

Note

A descriptor may be omitted if none of the panel plots need it.

Author(s)

Daniel B. Carr, George Mason University, Fairfax VA, with contributions from Jim Pearson and Linda Pickle of StatNet Consulting, LLC, Gaithersburg, MD

See Also

micromapST

Seer18Area	*Test data for 18 U.S. Seer Registries (of 20) for general dot, arrow, and bar glyphics.*

Description

Randomly generated statistics to create rate, count and population data for the 18 U.S. Seer registries for testing of micromapSEER or micromapST using the "USSeerBG" border group. The data was generated by the SeerStat program and exported as a CSV file. The non-data lines at the bottom of the file must be removed.

Usage

```
data(Seer18Area)
```

Format

A data frame with 18 observations, 1 for each Seer registries (sub-areas), on the following 11 variables.

Registry a character vector of the name of the Seer Registry

Rate20022006 a numeric vector of the incident rates from 2002 to 2006

Count20022006 a numeric vector of the incident counts

Pop20022006 a numeric vector of the population

Rate20072011 a numeric vector of the incident rates from 2007 to 2011

Count20072011 a numeric vector of the incident counts

Pop20072011 a numeric vector of the population

RateTrendAPC a numeric vector of the rate trends per Seer registry

pValue a numeric vector of the P Value related to the Rate Trend

LowerCI a numeric vector of the lower confidence interval

UpperCI a numeric vector of the upper confidence interval

Details

This dataset is a randomly generated collection of data from 2002 to 2011 Seer Registry data to support testing of micromapST functions for the 18 U.S. Seer registries. The data has no relationship to the Seer registries and is only for test purposes. The row names are the Seer registry abbreviations. This dataset is used in the micromapST examples.

Author(s)

Linda W. Pickle and Jim Pearson of StatNet Consulting, LLC, Gaithersburg, MD

SeoulPopData   *Test data for the Seoul South Korea city district border Group*

Description

This dataset contains the 2012 population and area data for each of the districts in city of Seoul.

Usage

```
data(SeoulPopData)
```

Format

A data frame with 25 observations, 1 for each district, on the following 5 variables.

District a character vector containing the full name of the Seoul district.

City a character vector containing the city name => Seoul.

Pop.2012 a numeric vector of the number of the district's 2012 population.

Area a numeric vector of the area square kilometers for the district.

Founded a character vector containing the dates each district was founded.

Details

This dataset was pulled from the China ??? government website on Dec. 2014. It contains the population and average income per person for each of Kansas' 105 counties.

| SeoulSKoreaBG | *Seoul South Korea border group datasets to support creating micromaps for the 25 districts in the Korean city of Seoul.* |

Description

The *micromapST* function has the ability to generate linked micromaps for any geographical area. To specify the geographical area, the 'bordGrp' call argument is used to specify the border group dataset for the geographical area. The *SeoulSKoreaBG* border group dataset supports creating linked micromaps for the 25 districts in the city of Seoul South Korea. When the 'bordGrp' call argument is set to *SeoulSKoreaBG*, the appropriate name table (county names and abbreviations) and the boundary data for the 25 districts are loaded in *micromapST*. The user's data is then linked to the boundary data via the district's name, abbreviated or ID based on the table below.

Details

The *SeoulSKoreaBG* border group contains the following data.frames::

areaParms - contains specific parameters for the border group

areaNamesAbbrsIDs - containing the names, abbreviations, and numerical identifier for the districts in the city of Seoul.

areaVisBorders - the boundary point lists for each district in the city of Seoul Korea.

L2VisBorders - the boundaries for an intermediate level. For this border group, this boundary data is not used and set to L3VisBorders as a place holder.

RegVisBorders - the boundaries for an regional level. For this border group, this boundary data is not used and set to L3VisBorders as a place holder.

L3VisBorders - the boundary of the city of Seoul South Korea

The Seoul district border group contains 25 district sub-areas. Each district has a row in the areaNamesAbbrsIDs data.frame and a set of polygons in the areaVisBorders data.frame datasets. No regions or L2 boundaries are defined for the Seoul district border group. The RegVisBorders and and L2VisBorders data.frames are set the L3VisBorders data.frame. The regional feature is disabled. The L3VisBorders dataset contains the outline of the city of Seoul.

The details on each of these data.frame structures can be found in the "bordGrp" section of this document. The *areaNamesAbbrsIDs* data.frame provides the linkages to the boundary data for each sub-area (district) using the fullname, abbreviation, and numerical identifier for each country to the *<statsDFrame>* data based on the setting of the 'rowNames' call parameter.

A column or the data.frame row.names must match one of the types of names in the *areaNamesAbbrsIDs* data.frame name table. If the data row does not match a value in the name table, an warning is issued and the data is ignored. If no data is present for a sub-area (district) in the name table, the sub-area is mapped but not colored.

The following are a list of the names, abbreviations, and ids for each country in the *SeoulSKoreaBG* border group.

| name | ab | id |

Dobong	DO	11100
Dongdaemun	DN	11060
Dongjak	DG	11200
Eunpyeong	EU	11120
Gangdong	GA	11090
Gangbuk	GN	11250
Gangnam	GG	11230
Gangseo	GS	11160
Geumcheon	GE	11180
Guro	GU	11170
Gwanak	GW	11210
Gwangjin	GJ	11050
Jongro	JO	11010
Jung	JU	11070
Jungnang	JN	11020
Mapo	MA	11140
Nowon	NO	11110
Seocho	SE	11220
Seodaemun	SD	11130
Seongbuk	SN	11080
Seongdong	SG	11040
Songpa	SO	11240
Yangcheon	YA	11150
Yeongdeungpo	YE	11190
Yongsan	YO	11030

The *id* field value is the ISO numerical code for the district.

The 'rowNames' = "alias" or "alt_ab" and the 'regions' features are not supported in the *Seoul-SKoreaBG* border group.

---

statePop2010 *US State Population for 2010*

Description

US State 2010 population data by race and Hispanic ethnicity.

Usage

```
data(statePop2010)
```

Format

A data frame with 51 observations (one per state) on the following 6 variables per state:

Hisp an integer count of the Hispanic population

white an integer count of the white population

black an integer count of the black population

OtherWBH an integer count of the population other than white, black or Hispanic

pctHisp a numeric percentage of the Hispanic population to the total population

pctOtherWBH a numeric percentage of the population other than white, black, or Hispanic

Each row has the state 2 character abbreviation as its row name.

Details

The dataset contains 51 records, one for each state. The data represents the population count or percentage of the total population by race and Hispanic ethnicity within the state. This dataset is used by the micromapSEER examples with the "USStatesDF" border group.

Source

United States Census Bureau, Population Total by State, by Race, Combinations of Two Races, and not Hispanic or Latino, 2010 (Summary File 1, Table QT-P4), URL = http://factfinder2.census.gov/.

TSdata   *Time Series Example Dataset*

Description

Data for Time Series Examples

The data are age-adjusted (2000 U.S. standard) female lung cancer mortality rates (per 100,000 population) for each year from 1996 to 2010.

Usage

```
data(TSdata)
```

Format

This dataset is an array with dimensions of 51, 15, 4. The rownames of the array are the 51 state and DC abbreviations (2 characters). TSdata[,,1:4] contains the x (time) value, followed by the value for the line, then the lower 95% confidence limit, and finally the upper 95% confidence limit value.

Details

The first dimension [*st*,,] of 51 elements contains each state or DC. This dimension is referenced by the rownames of the array.

The second dimension [,*t*,] of *n* elements in this case are the time periods in the time series. Our example uses the years 1996 to 2010 as the time period values. A reasonable number of points is between 20 and 30.

The third dimension [,,*v*] of 2 or 4 elements is the x or y values during the time period. If no confidence data is provided, the third dimension is 2:

- data[,,*1*] is the X value

- data[,,*2*] is the mid-Y value (Y)

If a confidence band is being plotted in '\var{tsconf}' graphs then there are 4 elements.

- data[,,*1*] is the X value

- data[,,*2*] is the mid-Y value (Y)

- data[,,*3*] is the low-Y value

- data[..*4*] is the high-Y value

For example, the x,y coordinates for year=1996 (time period = 1) for the first state (*AK*) is TS-data[*1*,*1*,c(1,2)].

This approach was done to allow a data matrix built for the "tsconf" glyphs to be used for a *ts* glyphs.

This data is used by micromapSEER with the "USStatesDF" border group.

```
# how to create a new time series data set
tempTS <-read.table("...yourfilename.csv",sep=",",header=T)
yrmat  <-matrix(rep(1996:2010,51),nrow=51,ncol=15,byrow=T)  # year labels
ratemat<-as.matrix(
        tempTS[,c(8,13,18,23,28,33,38,43,48,53,58,63,68,73,78)]
        )
locimat<-as.matrix(
        tempTS[,c(9,14,19,24,29,34,39,44,49,54,59,64,69,74,79)]
        )
hicimat<-as.matrix(
        tempTS[,c(10,15,20,25,30,35,40,45,50,55,60,65,70,75,80)]
        )
workmat<-cbind(yrmat,ratemat,locimat,hicimat)
TSdata <-NULL
TSdata <-array(workmat,dim=c(51,15,4))
# change state ab from factors to characters.
rownames(TSdata)<-as.character(tempTS$stab)
```

Source

Surveillance, Epidemiology, and End Results (SEER) Program (www.seer.cancer.gov) SEER*Stat Database: Mortality - All COD, Aggregated With State, Total U.S. (1969-2010) (Katrina/Rita Population Adjustment), National Cancer Institute, DCCPS, Surveillance Research Program, Surveillance Systems Branch, released April 2013. Underlying mortality data provided by NCHS (www.cdc.gov/nchs).

| UKIrelandBG | *UKIrelandBG border group datasets contains the boundary informa-tion for the United Kingdom and Ireland.* |

Description

The *micromapST* function has the ability to generate linked micromaps for any geographical area. To specify the geographical area, the 'bordGrp' call argument is used to specify the border group dataset for the geographical area. The *UKIrelandBG* border group dataset supports creating linked micromaps for the all of the United Kingdom including Northern Ireland and the Isle of Man and Ireland. When the 'bordGrp' call argument is set to *UKIrelandBG*, the appropriate name table (county names and abbreviations) and the 219 sub-areas (counties, etc.) boundary data is loaded into *micromapST*. The user's data is then linked to the boundary data via the name, abbreviation, or alternate abbreviation for each sub-area (county, etc.).

The United Kingdom and Ireland information was pulled from the UK and Ireland public web sites in March of 2015.

The UKIreland border group was constructed to provide an area with more than 100 sub-areas for testing *micromapST* and enhancing it's ability to handle a large number of sub-areas and generate usable linked micromaps.

Details

The *UKIrelandBG* border group dataset contains the following data.frames:

areaParms - contains specific parameters for the border group

areaNamesAbbrsIDs - containing the names, abbreviations, alternate abbreviation, or numerical identifier for each of the United Kingdom or Ireland counties.

areaVisBorders - the boundary point lists for each sub-area.

L2VisBorders - the boundaries for an intermediate level. For the United Kingdom and Ireland border group, this boundary point list is not used and is set to equal L3VisBorders data.frame for the border group.

RegVisBorders - the boundaries for the 4 United Kingdom and Ireland regions or realms: England, Wales, Scotland, Northern Ireland, Ireland, and Isle of Man.

L3VisBorders - the boundary of the United Kingdom and Ireland area.

The UKIreland border group contains 219 sub-areas (counties, etc.) Each registry has a row in the *areaNamesAbbrsIDs* data.frame and a set of polygons in the areaVisBorders data.frame datasets. Regions are defined in this border group as the 6 country and kingdom regions in the UK and Ireland. The regions feature is enable. The siz (6) regions are: England, Scotland, Wales, Northern Ireland, Ireland and Isle of Man. The names, abbreviations, alternate abbreviations and IDs for the counties in the UKIreland border group are:

Name	ab	id	alt_ab	region
Aberdeen	GB.ABE	826139	ABE	SCT
Aberdeenshire	GB.ABD	826140	ABD	SCT

c	GB.AGY	826171	AGY	WLS
Angus	GB.ANS	826141	ANS	SCT
Antrim	GB.ANT	826113	ANT	NIR
Ards	GB.ARD	826114	ARD	NIR
Argyll and Bute	GB.AGB	826142	AGB	SCT
Armagh	GB.ARM	826115	ARM	NIR
Ballymena	GB.BLA	826116	BLA	NIR
Ballymoney	GB.BLY	826117	BLY	NIR
Banbridge	GB.BNB	826118	BNB	NIR
Barking and Dagenham	GB.BDG	826001	BDG	ENG
Bath and North East Somerset	GB.BAS	826002	BAS	ENG
Bedfordshire	GB.BDF	826003	BDF	ENG
Belfast	GB.BFS	826119	BFS	NIR
Berkshire	GB.BRK	826004	BRK	ENG
Bexley	GB.BEX	826005	BEX	ENG
Blackburn with Darwen	GB.BBD	826006	BBD	ENG
Blaenau Gwent	GB.BGW	826172	BGW	WLS
Bournemouth	GB.BMH	826007	BMH	ENG
Brent	GB.BEN	826008	BEN	ENG
Bridgend	GB.BGE	826173	BGE	WLS
Brighton and Hove	GB.BNH	826009	BNH	ENG
Bristol	GB.BST	826010	BST	ENG
Bromley	GB.BRY	826011	BRY	ENG
Buckinghamshire	GB.BKM	826012	BKM	ENG
Caerphilly	GB.CAY	826174	CAY	WLS
Cambridgeshire	GB.CAM	826013	CAM	ENG
Camden	GB.CMD	826014	CMD	ENG
Cardiff	GB.CRF	826175	CRF	WLS
Carlow	IE.CW	372001	CW	IRE
Carmarthenshire	GB.CMN	826176	CMN	WLS
Carrickfergus	GB.CKF	826120	CKF	NIR
Castlereagh	GB.CSR	826121	CSR	NIR
Cavan	IE.CN	372002	CN	IRE
Ceredigion	GB.CGN	826177	CGN	WLS
Cheshire	GB.CHS	826015	CHS	ENG
Clackmannanshire	GB.CLK	826143	CLK	SCT
Clare	IE.CE	372003	CE	IRE
Coleraine	GB.CLR	826122	CLR	NIR
Conwy	GB.CWY	826178	CWY	WLS
Cookstown	GB.CKT	826123	CKT	NIR
Cork	IE.CO	372004	CO	IRE
Cornwall	GB.CON	826016	CON	ENG
Craigavon	GB.CGV	826124	CGV	NIR
Croydon	GB.CRY	826017	CRY	ENG
Cumbria	GB.CMA	826018	CMA	ENG
Darlington	GB.DAL	826019	DAL	ENG
Denbighshire	GB.DEN	826179	DEN	WLS
Derby	GB.DER	826020	DER	ENG

Derbyshire	GB.DBY	826021	DBY	ENG
Derry	GB.DRY	826125	DRY	NIR
Devon	GB.DEV	826022	DEV	ENG
Donegal	IE.DL	372005	DL	IRE
Dorset	GB.DOR	826023	DOR	ENG
Down	GB.DOW	826126	DOW	NIR
Dublin	IE.D	372006	D	IRE
Dumfries and Galloway	GB.DGY	826144	DGY	SCT
Dundee	GB.DND	826145	DND	SCT
Dungannon	GB.DGN	826127	DGN	NIR
Durham	GB.DUR	826024	DUR	ENG
Ealing	GB.EAL	826025	EAL	ENG
East Ayrshire	GB.EAY	826146	EAY	SCT
East Dunbartonshire	GB.EDU	826147	EDU	SCT
East Lothian	GB.ELN	826148	ELN	SCT
East Renfrewshire	GB.ERW	826149	ERW	SCT
East Riding of Yorkshire	GB.ERY	826026	ERY	ENG
East Sussex	GB.ESX	826027	ESX	ENG
Edinburgh	GB.EDH	826150	EDH	SCT
Eilean Siar	GB.ELS	826151	ELS	SCT
Enfield	GB.ENF	826028	ENF	ENG
Essex	GB.ESS	826029	ESS	ENG
Falkirk	GB.FAL	826152	FAL	SCT
Fermanagh	GB.FER	826128	FER	NIR
Fife	GB.FIF	826153	FIF	SCT
Flintshire	GB.FLN	826180	FLN	WLS
Galway	IE.G	372007	G	IRE
Glasgow	GB.GLG	826154	GLG	SCT
Gloucestershire	GB.GLS	826030	GLS	ENG
Greenwich	GB.GRE	826031	GRE	ENG
Gwynedd	GB.GWN	826181	GWN	WLS
Hackney	GB.HCK	826032	HCK	ENG
Halton	GB.HAL	826033	HAL	ENG
Hammersmith and Fulham	GB.HMF	826034	HMF	ENG
Hampshire	GB.HAM	826035	HAM	ENG
Haringey	GB.HRY	826036	HRY	ENG
Harrow	GB.HRW	826037	HRW	ENG
Hartlepool	GB.HPL	826038	HPL	ENG
Havering	GB.HAV	826039	HAV	ENG
Herefordshire	GB.HEF	826040	HEF	ENG
Hertfordshire	GB.HRT	826041	HRT	ENG
Highland	GB.HLD	826155	HLD	SCT
Hillingdon	GB.HIL	826042	HIL	ENG
Hounslow	GB.HNS	826043	HNS	ENG
Inverclyde	GB.IVC	826156	IVC	SCT
Isle of Wight	GB.IOW	826044	IOW	ENG
Islington	GB.ISL	826045	ISL	ENG
Kensington and Chelsea	GB.KEC	826046	KEC	ENG

Kent	GB.KEN	826047	KEN	ENG
Kerry	IE.KY	372008	KY	IRE
Kildare	IE.KE	372009	KE	IRE
Kilkenny	IE.KK	372010	KK	IRE
Kingston upon Hull	GB.KHL	826048	KHL	ENG
Kingston upon Thames	GB.KTT	826049	KTT	ENG
Lambeth	GB.LBH	826050	LBH	ENG
Lancashire	GB.LAN	826051	LAN	ENG
Laoighis	IE.LS	372011	LS	IRE
Larne	GB.LRN	826129	LRN	NIR
Leicester	GB.LCE	826052	LCE	ENG
Leicestershire	GB.LEC	826053	LEC	ENG
Leitrim	IE.LM	372012	LM	IRE
Lewisham	GB.LEW	826054	LEW	ENG
Limavady	GB.LMV	826130	LMV	NIR
Limerick	IE.LK	372013	LK	IRE
Lincolnshire	GB.LIN	826055	LIN	ENG
Lisburn	GB.LSB	826131	LSB	NIR
London	GB.LND	826056	LND	ENG
Longford	IE.LD	372014	LD	IRE
Louth	IE.LH	372015	LH	IRE
Luton	GB.LUT	826057	LUT	ENG
Magherafelt	GB.MFT	826132	MFT	NIR
Manchester	GB.MAN	826058	MAN	ENG
Mayo	IE.MO	372016	MO	IRE
Meath	IE.MH	372017	MH	IRE
Medway	GB.MDW	826059	MDW	ENG
Merseyside	GB.MSY	826060	MSY	ENG
Merthyr Tydfil	GB.MTY	826182	MTY	WLS
Merton	GB.MRT	826061	MRT	ENG
Middlesbrough	GB.MDB	826062	MDB	ENG
Midlothian	GB.MLN	826157	MLN	SCT
Milton Keynes	GB.MIK	826063	MIK	ENG
Monaghan	IE.MN	372018	MN	IRE
Monmouthshire	GB.MON	826183	MON	WLS
Moray	GB.MRY	826158	MRY	SCT
Moyle	GB.MYL	826133	MYL	NIR
Neath Port Talbot	GB.NTL	826184	NTL	WLS
Newham	GB.NWM	826064	NWM	ENG
Newport	GB.NWP	826185	NWP	WLS
Newry and Mourne	GB.NYM	826134	NYM	NIR
Newtownabbey	GB.NTA	826135	NTA	NIR
Norfolk	GB.NFK	826065	NFK	ENG
North Ayrshire	GB.NAY	826159	NAY	SCT
North Down	GB.NDN	826136	NDN	NIR
North East Lincolnshire	GB.NEL	826066	NEL	ENG
North Lanarkshire	GB.NLK	826160	NLK	SCT
North Lincolnshire	GB.NLN	826067	NLN	ENG

North Somerset	GB.NSM	826068	NSM	ENG
North Yorkshire	GB.NYK	826069	NYK	ENG
Northamptonshire	GB.NTH	826070	NTH	ENG
Northumberland	GB.NBL	826071	NBL	ENG
Nottingham	GB.NGM	826072	NGM	ENG
Nottinghamshire	GB.NTT	826073	NTT	ENG
Offaly	IE.OY	372019	OY	IRE
Omagh	GB.OMH	826137	OMH	NIR
Orkney Islands	GB.ORK	826161	ORK	SCT
Oxfordshire	GB.OXF	826074	OXF	ENG
Pembrokeshire	GB.PEM	826186	PEM	WLS
Perthshire and Kinross	GB.PKN	826162	PKN	SCT
Peterborough	GB.PTE	826075	PTE	ENG
Plymouth	GB.PLY	826076	PLY	ENG
Poole	GB.POL	826077	POL	ENG
Portsmouth	GB.POR	826078	POR	ENG
Powys	GB.POW	826187	POW	WLS
Redbridge	GB.RDB	826079	RDB	ENG
Redcar and Cleveland	GB.RCC	826080	RCC	ENG
Renfrewshire	GB.RFW	826163	RFW	SCT
Rhondda, Cynon, Taff	GB.RCT	826188	RCT	WLS
Richmond upon Thames	GB.RIC	826081	RIC	ENG
Roscommon	IE.RN	372020	RN	IRE
Rutland	GB.RUT	826082	RUT	ENG
Scottish Borders	GB.SCB	826164	SCB	SCT
Shetland Islands	GB.ZET	826165	ZET	SCT
Shropshire	GB.SHR	826083	SHR	ENG
Sligo	IE.SO	372021	SO	IRE
Somerset	GB.SOM	826084	SOM	ENG
South Ayrshire	GB.SAY	826166	SAY	SCT
South Gloucestershire	GB.SGC	826085	SGC	ENG
South Lanarkshire	GB.SLK	826167	SLK	SCT
South Yorkshire	GB.SYK	826086	SYK	ENG
Southampton	GB.STH	826087	STH	ENG
Southend-on-Sea	GB.SOS	826088	SOS	ENG
Southwark	GB.SWK	826089	SWK	ENG
Staffordshire	GB.STS	826090	STS	ENG
Stirling	GB.STG	826168	STG	SCT
Stockton-on-Tees	GB.STT	826091	STT	ENG
Stoke-on-Trent	GB.STE	826092	STE	ENG
Strabane	GB.STB	826138	STB	NIR
Suffolk	GB.SFK	826093	SFK	ENG
Surrey	GB.SRY	826094	SRY	ENG
Sutton	GB.STN	826095	STN	ENG
Swansea	GB.SWA	826189	SWA	WLS
Swindon	GB.SWD	826096	SWD	ENG
Telford and Wrekin	GB.TFW	826097	TFW	ENG
Thurrock	GB.THR	826098	THR	ENG

Tipperary	IE.TA	372022	TA	IRE
Torbay	GB.TOB	826099	TOB	ENG
Torfaen	GB.TOF	826190	TOF	WLS
Tower Hamlets	GB.TWH	826100	TWH	ENG
Tyne and Wear	GB.TWR	826101	TWR	ENG
Vale of Glamorgan	GB.VGL	826191	VGL	WLS
Waltham Forest	GB.WFT	826102	WFT	ENG
Wandsworth	GB.WND	826103	WND	ENG
Warrington	GB.WRT	826104	WRT	ENG
Warwickshire	GB.WAR	826105	WAR	ENG
Waterford	IE.WD	372023	WD	IRE
West Dunbartonshire	GB.WDU	826169	WDU	SCT
West Lothian	GB.WLN	826170	WLN	SCT
West Midlands	GB.WMD	826106	WMD	ENG
West Sussex	GB.WSX	826107	WSX	ENG
West Yorkshire	GB.WYK	826108	WYK	ENG
Westmeath	IE.WH	372024	WH	IRE
Westminster	GB.WSM	826109	WSM	ENG
Wexford	IE.WX	372025	WX	IRE
Wicklow	IE.WW	372026	WW	IRE
Wiltshire	GB.WIL	826110	WIL	ENG
Worcestershire	GB.WOR	826111	WOR	ENG
Wrexham	GB.WRX	826192	WRX	WLS
York	GB.YOR	826112	YOR	ENG
Isle of Man	IM	833000	IMN	IMN

When compiling the abbreviations for this border group, multiple sets of abbreviations were found. The two most common sets are included in the border group as "ab" and "alt_ab" types of ‘rowNames’.

The L3VisBorders dataset contains the outline of the UK and Ireland.

The details on each of these data.frame structures can be found in the "bordGrp" section of this document. The *areaNamesAbbrsIDs* data.frame provides the linkages to the boundary data for each sub-area (registry) using the fullname, abbreviation, alternate abbreviation and numerical identifier for each county/provence to the <*statsDFrame*> data based on the setting of the ‘rowNames’ call argument.

A column or the data.frame row.names must match one of the types of names in the *areaNamesAbbrsIDs* data.frame name table. If the data row does not match a value in the name table, an warning is issued and the data is ignored. If no data is present for a sub-area in the name table, the sub-area is mapped but not colored.

The ‘dataRegionsOnly’ call parameter instructs the package to only map the regions with Seer registers with data. The regions are the four census regions: England, Scotland, Wales, Isle of Man, Northern Ireland and Ireland.

Source

NCI

References

???? Retrieved 2013-01-10.

UKIrelandPopData *Test data for the UK-Ireland border Group*

Description

The UKIrelandPopData and UKIrelandPopData2 datasets contain the county populations and area statistics for the UK and Ireland..

Usage

```
data(UKIrelandPopData)
```

Format

A data frame with 218 rows, 1 for each county, with 5 variables for each county:

Name a character vector containing the UK-Ireland county name.

Abbr a character vector containing the ISO 2 character abbreviation for the UK and Ireland country. Warning: the UKIreland border group uses the newer ISO 3 character notation for it\'s abbreviations.

Pop a numeric vector of the county's population.

Area.M a numeric vector of the county's area in square miles.

Area.Km a numeric vector of the county's area in square kilometers.

Details

This dataset was pulled from several UK and Ireland website in January, 2015. The difference between UKIrelandPopData and UKIrelandPopData2 is incorrect labeling of Anglesey as Anglesey, Isle of.

USSeerBG *USSeerBG border group datasets to support use with U.S. 20 Seer areas/registries*

Description

The *micromapST* function has the ability to generate linked micromaps for any geographical area. To specify the geographical area, the 'bordGrp' call argument is used to specify the border group dataset for the geographical area. The *USSeerBG* border group dataset supports creating linked micromaps for the 20 Seer registries in the U. S. When the 'bordGrp' call argument is set to *USSeerBG*, the appropriate name table (county names and abbreviations) and the 20 sub-areas (Seer registries) boundary data is loaded in *micromapST*. The user's data is then linked to the boundary data via the Seer registry's name, abbreviated, alias match or ID based on the table below.

The 20 U. S. Seer registries are the accepted registries as of January 2010 funded by NCI.

Details

The *USSeerBG* border group dataset contains the following data.frames:

areaParms - contains specific parameters for the border group

areaNamesAbbrsIDs - containing the names, abbreviations, numerical identifier and alias matching string for each of the 20 Seer registries.

areaVisBorders - the boundary point lists for each area.

L2VisBorders - the boundaries for an intermediate level. For Seer registry border group, L2VisBorders contains the boundaries for the 51 states and DC in the U. S to help provide a geographical reference of the registries to the states.

RegVisBorders - the boundaries for the 4 U. S. Census regions in the U. S in support of the region feature.

L3VisBorders - the boundary of the U. S.

The Seer Registries border group contains 20 Seer Registry sub-areas. Each registry has a row in the]varareaNamesAbbrsIDs data.frame and a set of polygons in the areaVisBorders data.frame datasets.

Regions are defined in this border group as the 4 census regions in the U. S. The regions feature is enable. The four census regions are: NorthEast, South, MidWest, and West. The states and Seer registries in each region are:

state	Seer Registries	region
Alabama	\<none\>	South
Alaska	Alaska Natives	West
Arizona	Arizona Natives	West
Arkansas	\<none\>	South
California	California-LA, California-Other, California-SF, California-SJ	West
Colorado	\<none\>	West
Connecticut	Connecticut	NorthEast
Delaware	\<none\>	South
District of Columbia	\<none\>	South
Florida	\<none\>	South
Georgia	Georgia-Atlanta, Georgia-Other, Georgia-Rural	South
Hawaii	Hawaii	West
Idaho	\<none\>	West
Illinois	\<none\>	MidWest
Indiana	\<none\>	MidWest
Iowa	Iowa	MidWest
Kansas	\<none\>	MidWest
Kentucky	Kentucky	South
Louisiana	Louisiana	South
Maine	\<none\>	NorthEast

Maryland	<none>	South
Massachusetts	<none>	NorthEast
Michigan	Michigan-Detroit	MidWest
Minnesota	<none>	MidWest
Mississippi	<none>	South
Missouri	<none>	MidWest
Montana	<none>	West
Nebraska	<none>	MidWest
Nevada	<none>	West
New Hampshire	<none>	NorthEast
New Jersey	New Jersey	NorthEast
New Mexico	New Mexico	West
New York	<none>	NorthEast
North Carolina	<none>	South
North Dakota	<none>	MidWest
Ohio	<none>	MidWest
Oklahoma	Oklahoma-Cherokee	South
Oregon	<none>	West
Pennsylvania	<none>	NorthEast
Rhode Island	<none>	NorthEast
South Carolina	<none>	South
South Dakota	<none>	MidWest
Tennessee	<none>	South
Texas	<none>	South
Utah	Utah	West
Vermont	<none>	NorthEast
Virginia	<none>	South
Washington	Washington-Seattle	South
West Virginia	<none>	South
Wisconsin	<none>	MidWest
Wyoming	<none>	West

The L3VisBorders dataset contains the outline of the United States.

The details on each of these data.frame structures can be found in the "bordGrp" section of this document. The *areaNamesAbbrsIDs* data.frame provides the linkages to the boundary data for each sub-area (registry) using the fullname, abbreviation, and numerical identifier for each country to the *<statsDFrame>* data based on the setting of the 'rowNames' call argument.

A column or the data.frame row.names must match one of the types of names in the *areaNamesAbbrsIDs* data.frame name table. If the data row does not match a value in the name table, an warning is issued and the data is ignored. If no data is present for a sub-area (registry) in the name table, the sub-area (registry) is mapped but not colored.

The following are a list of the names, abbreviations, alias and IDs for each country in the *USSeerBG* border group.

Name	ab	alias string	id	counties	region
Alaska Natives	AK-NAT	ALASKA NATIVES	18	all	West

Arizona Natives	AZ-NAT	ARIZONA NATIVES	20	all	West
California-LA	CA-LA	LOS ANGELES	4	Los Angeles	West
California-SF	CA-SF	SAN FRANCISCO	2	Alameda, Contra Costa, Marin, San Francisco, San Mateo	West
California-SJ	CA-SJ	SAN JOSE	3	Montersey San Benito, Santa Clara, Santa Cruz	West
California-Other	CA-OTH	CALIFORNIA EXCLUDING	5	all other counties	West
Connecticut	CT	CONNECTICUT	1	all	NorthEast
Georgia-Atlanta	GA-ATL	ATLANTA	6	Clayton, Cobb, DeKalb, Fulton, Gwinnett	South
Georgia-Rural	GA-RUR	RURAL GEORGIA	8	Glascock, Greene, Hancock, Jasper, Jefferson, Morgan, Putnam, Taliaferro, Warren, Washington	South
Georgia-Other	GA-OTH	GREATER GEORGIA	7	all other counties	South
Hawaii	HI	HAWAII	9	all	West
Iowa	IA	IOWA	10	all	MidWest
Kentucky	KY	KENTUCKY	14	all	South
Michigan-Detroit	MI-DET	DETROIT	15	Macomb, Oakland, Wayne	MidWest
New Jersey	NJ	NEW JERSEY	11	all	NorthEast
New Mexico	NM	NEW MEXICO	12	all	West
Oklahoma-Cherokee	OK-CHE	OKLAHOMA	19	Adair, Cherokee, Craig, Delaware, Mayes, McIntosh, Muskogee, Nowata, Ottawa, Rogers, Seqouyah, Tulsa, Wagnorer, Washington	South
Utah	UT	UTAH	16	all	West
Washington-Seattle	WA-SEA	SEATTLE	17	Clallam, Grays Harbor, Island, Jefferson, King,	South

                                                                          Kitsap,
                                                                          Mason,
                                                                          Pierce,
                                                                          San Juan,
                                                                          Skagit,
                                                                          Snohomish,
                                                                          Thurston,
                                                                          Whatcom

The 'rowNames' = *alias* and the 'regions' = *TRUE* features are enabled in the *USSeerBG* border group.

The *alias* option is designed to allow the package to match the registry labels created by the Seer Stat website when exporting Seer data for analysis. The alias match is a "contains" match, so the registry field in the user data must "contain" the "alias" values listed in the above table. To help generalize the match, the user's registry value is stripped of any punctuation, control characters and multiple spaces (blanks, tabs, cr, lf) are reduced to a single blank and the string is converted to all upper case. Then the wild card match is performed.

The 'dataRegionOnly' call parameter (when set to *TRUE*) instructs the package to only map the regions with Seer registers with data. The regions used are the four census regions: NorthEast, South, MidWest and West. The RegVisBorders data.frame contains the outline of each of these regions. For example: if Seer registry data is provided for the only the New Mexico, Utah and California Registries in the West region, then only the states and regional boundary for the West region are drawn.

The *USSeerBG* border group does not contain or support an alternate set of abbreviations. If 'rowNames' is set to *alt_ab*, an warning is generated and the standard Seer registry abbreviations are used.

The following steps should be used to export data for *micromapST*'s use from the SEER*Stat Website:

1. Log on to the SEER^Stat website.

2. Create the matrix of results you want in SEER*Stat.

3. Click on Matrix, Export, Results as Text File (if you created multiple matrices of results, make sure that the one you want to export is highlighted)

4. In the Matrix Export Options window, click on:

    (a) Output variables as Labels without quotes

    (b) Remove all thousands separators

    (c) Output variable names before data

    (d) Preserve matrix columns & rename fields

    (e) Leave defaults clicked for Line delimiter, Missing Character, and Field delimiter

5. Change names and locations of text and dictionary files from defaults to the appropriate name and directory location.

To read the resulting text file into R use the read.delim function with 'header' = *TRUE*. Follow the read.delim call with a str function to verify the data was read correctly.

```
dataT <- read.delim("c:\datadir\seerstat.txt",header=FALSE)
str(dataT)
```

Source

NCI

References

United States National Cancer Institute Seer Website at www.seer.cancer.gov; Seer Software at seer.cancer.gov/seerstat.; United States Census Bureau, Geography Division. "Census Regions and Divisions of the United States" (PDF). Retrieved 2013-01-10.

| USStatesBG | *USStatesBG border group datasets to support use with U.S. States amd D.C. Areas* |

Description

The *micromapST* function has the ability to generate linked micromaps for any geographical area. To specify the geographical area, the 'bordGrp' call argument is used to specify the border group dataset for the geographical area. When *micromapSt* function is used to micromap the U.S. States and DC areas, no border group needs to be specified. The *USStatesBG* border group is the same as the original sub-areas (states and DC) and boundaries used by the all previous versions of the *micromapST* package. By default *micromapST* loads the area fullnames, abbreviations, IDs and boundaries files for 50 U. S. states and the District of Columbia for the processing of user data and the creation of the requested linked micromap.

Details

The *USStatesBG* border group contains in the following data.frames:

areaParms - specific parameters associated with this border group

areaNamesAbbrsIDs - containing the names, abbreviations, numerical identifier and alias matching string for each of the 51 U. S. States and D.C.

areaVisBorders - the boundary point lists for each of the 51 States and D.C..

L2VisBorders - the boundaries for the U. S. states and DC. It is identical to *areaVisBorders* data.frame.

RegVisBorders - the boundaries for the 4 U. S. census regions

L3VisBorders - the boundary of the U.S

Refer to the section on the border group data.frames for a detailed discussion on the formats and usage of each of the above data.frame.

In this border group, there are 51 areas (states and DC) and information and names in the areaNamesAbbrIDs data.frame and there boundaries in the areaVisBorders data.frame. The L2VisBorders

data.frame contains a copy of the areaVisBorders data.frame to allow heavier overlaying of the state and DC boundaries during mapping. The RegVisBorders data.frame contains the information and boundaries for the 4 U. S. census regions. The L3VisBorders dataset contains the information on the boundaries of the U.S.

Alaska and Hawaii are relocated to below California and Alaska is reduced in size by 50

The names, abbreviations, id, and assigned U. S. regions used in the areaNamesAbbrsIDs data.frame the the 50 U. S. States and DC are as follows:

name	ab	id	region
Alabama	AL	01	South
Alaska	AK	02	West
Arizona	AZ	04	West
Arkansas	AR	05	South
California	CA	06	West
Colorado	CO	08	West
Connecticut	CT	09	NorthEast
Delaware	DE	10	South
DC	DC	11	South
Florida	FL	12	South
Georgia	GA	13	South
Hawaii	HI	15	West
Idaho	ID	16	West
Illinois	IL	17	MidWest
Indiana	IN	18	MidWest
Iowa	IA	19	MidWest
Kansas	KS	20	MidWest
Kentucky	KY	21	South
Louisiana	LA	22	South
Maine	ME	23	NorthEast
Maryland	MD	24	South
Massachusetts	MA	25	NorthEast
Michigan	MI	26	MidWest
Minnesota	MN	27	MidWest
Mississippi	MI	28	South
Missouri	MO	29	MidWest
Montana	MT	30	West
Nebraska	NE	31	MidWest
Neveda	NV	32	West
New Hampshire	NH	33	NorthEast
New Jersey	NJ	34	NorthEast
New Mexico	NM	35	West
New York	NY	36	NorthEast
North Carolina	NC	37	South
North Dakota	ND	38	MidWest
Ohio	OH	39	MidWest
Oklahoma	OK	40	South
Oregon	OR	41	West
Pennsylvania	PA	42	NorthEast

Rhode Island	RI	44	NorthEast
South Carolina	SC	45	South
South Dakota	SD	46	MidWest
Tennessee	TN	47	South
Texas	TX	48	South
Utah	UT	49	West
Vermont	VT	50	NorthEast
Virginia	VA	51	South
Washington	WA	53	West
West Virginia	WV	54	South
Wisconsin	WI	55	MidWest
Wyoming	WY	56	West

All data must be tagged with the name, abbreviation or id strings to be able to find the associated boundaries information. With the *USStatesBG* only, the package will accept several variations on the full name for DC. They include: Washington, D.C., District of Columbia, and D. C. All can be with or without punctuation and upper and lower case. When detected, the name is translated to "DC".

The 'dataRegionsOnly' call parameter can be set to *TRUE* to request the package to limit the mapping to only states in regional areas where data is being mapped and omit states and regions that do not contain data. This allows the caller to focus the micromaps on one of the four (4) census regions: NorthEast, South, Midwest, or West.

The 'rowNames' = *alias*, 'rowNames' = *alt_ab* are not support in the 'USStatesBG' border group.

Source

NIST - Federal Information Processing Standards and U. S. Census website.

References

NIST FIPS 6-4 Standards

UtahBG	*UtahBG border group datasets to support creating micromaps for the counties in the state of Utah*

Description

The *micromapST* function has the ability to generate linked micromaps for any geographical area. To specify the geographical area, the 'bordGrp' call parameter is used to specify the border group dataset for the geographical area. The *UtahBG* border group dataset is contained within this package and supports creating linked micromaps for the 29 counties in the state of Utah. When the 'bordGrp' call parameter is set to *UtahBG*, the appropriate name table (county names and abbreviations) and the 29 sub-areas (countries) boundary data is loaded in *micromapST*. The user's data is then linked to the boundary data via the county's name, abbreviation, alternate abbreviation, or ID based on the table below.

Details

The *UtahBG* border group dataset contains the following data.frames:

areaParms - contains specific parameters for the border group

areaNamesAbbrsIDs - containing the names, abbreviations, and numerical identifier for the 29
         counties in the state of Utah.

areaVisBorders - the boundary point lists for each county area in Utah.

L2VisBorders - the boundaries for an intermediate level. For this border group, this boundary
         data.frame is not used and set to L3VisBorders as a place holder.

RegVisBorders - the boundaries for an intermediate level. For this border group, this boundary
         data.frame is not used and set to L3VisBorders as a place holder.

L3VisBorders - the boundary of the state of Utah

The Utah county border group contains 29 county sub-areas. Each county has a row in the are-
aNamesAbbrsIDs data.frame and a set of polygons in the areaVisBorders data.frame datasets. No
regions are defined in the Utah county border group, so the L2VisBorders dataset is not used and
the regions option is disabled. The L3VisBorders dataset contains the outline of the state of Utah.

The details on each of these data.frame structures can be found in the "bordGrp" section of this
document. The *areaNamesAbbrsIDs* data.frame provides the linkages to the boundary data for
each sub-area (county) using the fullname, abbreviation, and numerical identifier for each country
to the <statsDFrame> data based on the setting of the 'rowNames' call argument.

A identified column (*idCol*) or the data.frame row.names must match one of the types of names in
the border group's *areaNamesAbbrsIDs* data.frame name table. If the data row does not match a
value in the name table, an warning is issued and the data is ignored. If no data is present for a
sub-area (county) in the name table, the sub-area is mapped but not colored.

The following are a list of the names, abbreviations, alternate abbreviations and IDs for each country
in the *UtahBG* border group.

name	ab	alt_ab	id
Beaver	BV	BEA	49001
Box Elder	BE	BOX	49003
Cache	CH	CAC	49005
Carbon	CA	CAR	49007
Daggett	DAG	DAG	49009
Davis	DAV	DAV	49011
Duchesne	DU	DUC	49013
Emery	EM	EME	49015
Garfield	GA	GAR	49017
Grand	GR	GRA	49019
Iron	IR	IRO	49021
Juab	JB	JUA	49023
Kane	KN	KAN	49025
Millard	MI	MIL	49027
Morgan	MO	MOR	49029
Piute	PT	PIU	49031
Rich	RH	RIC	49033

Salt Lake	SL	SAL	49035
San Juan	SJ	SNJ	49037
Sanpete	SP	SNP	49039
Sevier	SV	SEV	49041
Summit	SU	SUM	49043
Tooele	TO	TOO	49045
Uintah	UI	UIN	49047
Utah	UT	UTA	49049
Wasatch	WS	WST	49051
Washington	WA	WSH	49053
Wayne	WN	WAY	49055
Weber	WB	WEB	49057

When compiling the information for the *UtahBG* border group, it was not clear what was the standard accepted abbreviation for each county. Therefore, two sets of abbreviations for each county are included. The first abbreviation set can be referenced by setting the 'rowNames' call parameter to "ab". The second (alternate) abbreviation set can be used by setting the 'rowNames' to "alt_ab".

The *id* field value is the 5 digit U. S. state and county FIPS code.

The 'rowNames' = "alias" and the 'regions' features are not supported in the *UtahBG* border group.

UtahPopData	*Test data for the Utah state border Group*

Description

This dataset contains the Utah county populations for each year from 1940 to 2011.

Usage

```
data(UtahPopData)
```

Format

A data frame with 29 rows, 1 for each county, with 73 variables for each county:

County a character vector containing the Utah county full name.

X1940 a numeric vector of the county's population in 1940.

X1941 a numeric vector of the county's population in 1941.

X1942 a numeric vector of the county's population in 1942.

X1943 a numeric vector of the county's population in 1943.

X1944 a numeric vector of the county's population in 1944.

X1945 a numeric vector of the county's population in 1945.

X1946 a numeric vector of the county's population in 1946.

X1947 a numeric vector of the county's population in 1947.

X1948 a numeric vector of the county's population in 1948.

X1949 a numeric vector of the county's population in 1949.

X1950 a numeric vector of the county's population in 1950.

X1951 a numeric vector of the county's population in 1951.

X1952 a numeric vector of the county's population in 1952.

X1953 a numeric vector of the county's population in 1953.

X1954 a numeric vector of the county's population in 1954.

X1955 a numeric vector of the county's population in 1955.

X1956 a numeric vector of the county's population in 1956.

X1957 a numeric vector of the county's population in 1957.

X1958 a numeric vector of the county's population in 1958.

X1959 a numeric vector of the county's population in 1959.

X1960 a numeric vector of the county's population in 1960.

X1961 a numeric vector of the county's population in 1961.

X1962 a numeric vector of the county's population in 1962.

X1963 a numeric vector of the county's population in 1963.

X1964 a numeric vector of the county's population in 1964.

X1965 a numeric vector of the county's population in 1965.

X1966 a numeric vector of the county's population in 1966.

X1967 a numeric vector of the county's population in 1967.

X1968 a numeric vector of the county's population in 1968.

X1969 a numeric vector of the county's population in 1969.

X1970 a numeric vector of the county's population in 1970.

X1971 a numeric vector of the county's population in 1971.

X1972 a numeric vector of the county's population in 1972.

X1973 a numeric vector of the county's population in 1973.

X1974 a numeric vector of the county's population in 1974.

X1975 a numeric vector of the county's population in 1975.

X1976 a numeric vector of the county's population in 1976.

X1977 a numeric vector of the county's population in 1977.

X1978 a numeric vector of the county's population in 1978.

X1979 a numeric vector of the county's population in 1979.

X1980 a numeric vector of the county's population in 1980.

X1981 a numeric vector of the county's population in 1981.

X1982 a numeric vector of the county's population in 1982.

X1983 a numeric vector of the county's population in 1983.

X1984 a numeric vector of the county's population in 1984.

X1985 a numeric vector of the county's population in 1985.

X1986 a numeric vector of the county's population in 1986.

X1987 a numeric vector of the county's population in 1987.

X1988 a numeric vector of the county's population in 1988.

X1989 a numeric vector of the county's population in 1989.

X1990 a numeric vector of the county's population in 1990.

X1991 a numeric vector of the county's population in 1991.

X1992 a numeric vector of the county's population in 1992.

X1993 a numeric vector of the county's population in 1993.

X1994 a numeric vector of the county's population in 1994.

X1995 a numeric vector of the county's population in 1995.

X1996 a numeric vector of the county's population in 1996.

X1997 a numeric vector of the county's population in 1997.

X1998 a numeric vector of the county's population in 1998.

X1999 a numeric vector of the county's population in 1999.

X2000 a numeric vector of the county's population in 2000.

X2001 a numeric vector of the county's population in 2001.

X2002 a numeric vector of the county's population in 2002.

X2003 a numeric vector of the county's population in 2003.

X2004 a numeric vector of the county's population in 2004.

X2005 a numeric vector of the county's population in 2005.

X2006 a numeric vector of the county's population in 2006.

X2007 a numeric vector of the county's population in 2007.

X2008 a numeric vector of the county's population in 2008.

X2009 a numeric vector of the county's population in 2009.

X2010 a numeric vector of the county's population in 2010.

X2011 a numeric vector of the county's population in 2011. .

Details

This dataset was pulled from the Utah government website in January, 2015.

| wflung00and95 | *Lung cancer mortality data for white females, 2000-4 and 1995-9* |

Description

Counts and rates of age-adjusted (2000 U.S. standard) lung cancer mortality data among white women, aggregated for 1995-9 and 2000-4.

Usage

```
data(wflung00and95)
```

Format

A data frame with 51 observations, 1 for each state + DC, on the following 12 variables.

Rate.00 a numeric vector of age-adjusted rates by state during 2000-4 for white females

Count.00 a numeric vector of the number of white female lung cancer deaths during 2000-4

Lower.00 a numeric vector of the 95% confidence interval lower bound for white female 2000-4 rates

Upper.00 a numeric vector of the 95% confidence interval upper bound for white female 2000-4 rates

Pop.00 a numeric vector of the white female population during 2000

StdErr.00 a numeric vector of the standard error of the white female 2000-4 rates

Rate.95 a numeric vector of age-adjusted rates by state during 1995-9 for white females

Count.95 a numeric vector of the number of white female lung cancer deaths during 1995-9

Lower.95 a numeric vector of the 95% confidence interval lower bound for white female 1995-9 rates

Upper.95 a numeric vector of the 95% confidence interval upper bound for white female 1995-9 rates

Pop.95 a numeric vector of the white female population estimates for 1995

StdErr.95 a numeric vector of the standard error of the white female 1995-9 rates

Details

The rates on this file are directly age adjusted to the US 2000 standard population and are expressed as the number of deaths per 100,000 person-years. The row names are the 2 character postal codes for the states. The data represents the rates for two periods of time: 2000 to 2004 and 1995 to 1999. This dataset is used in the *micromapSEER* examples using the border group of "USStatesDF".

Author(s)

Linda W. Pickle and Jim Pearson of StatNet Consulting, LLC, Gaithersburg, MD

Source

Surveillance Research Program, National Cancer Institute SEER*Stat software (`https://www.seer.cancer.gov/seerstat`), November 2007 data submission, released April 2008. Data originally provided to NCI by the National Center for Health Statistics.

---

wflung00and95US	*wflung 2000 to 2004 and 1995 to 1999 US data*

Description

Counts and age-adjusted rate of white female lung cancer for the total U.S. for the aggregated periods 1995-9 and 2000-4.

Usage

```
data(wflung00and95US)
```

Format

A data frame with 1 observation on the following 13 variables.

Rate.00 a numeric vector, state rate for 2000-2004

Count.00 a numeric vector, state number of cases for 2000-2004

Lower.00 a numeric vector, lower end point for 95% confidence interval

Upper.00 a numeric vector, upper end point for 95% confidence interval

Pop.00 a numeric vector, state population fro 2000-2004

StdErr.00 a numeric vector, state standard error

Rate.95 a numeric vector, state rate for 1995-1999

Count.95 a numeric vector, state number of cases for 1995-1999

Lower.95 a numeric vector, lower end point for 95% confidence interval

Upper.95 a numeric vector, upper end point for 95% confidence interval

Pop.95 a numeric vector, state population for 2000-2004

StdErr.95 a numeric vector, state standard error

Details

See documentation for wflung00and95 for more details. The row name is the associated state abbreviation - 2 characters. This dataset is used in the *micromapSEER* examples using a border group of "USStatesDF"..

Author(s)

Linda W. Pickle and Jim Pearson of StatNet Consulting, LLC, Gaithersburg, MD

Source

Surveillance Research Program, National Cancer Institute SEER*Stat software (`https://www.seer.cancer.gov/seerstat`), November 2007 data submission, released April 2008. Data originally provided to NCI by the National Center for Health Statistics.

References

---

wflung00cnty	*Lung cancer mortality data for white females, by county, 2000-4*

Description

Counts and rates of lung cancer mortality data among white women, aggregated for 2000-4 by county.

Usage

```
data(wflung00cnty)
```

Format

A data frame with 2577 observations on the following 6 variables.

fips a numeric vector of 5 digit fips codes identifying the state and the county

rate a numeric vector of age-adjusted rates by county during 2000-4 for white females

count a numeric vector of the number of white female lung cancer deaths during 2000-4 by county

pop a numeric vector of the white female population in the county during 2000

stcode a numberic vector of the 2 digit state fips code

stabr a character vector of the 2 character state postal code

Details

The rates on this file are directly age adjusted to the US 2000 standard population and are expressed as the number of deaths per 100,000 person-years. Counties with from 1 to 9 deaths are suppressed (deleted from the file). This dataset is used by the *micromapSEER* examples using the border group of "USStatesDF".

Author(s)

Linda W. Pickle and Jim Pearson of StatNet Consulting, LLC, Gaithersburg, MD

Source

Surveillance Research Program, National Cancer Institute SEER*Stat software (http://www.seer.cancer.gov/seerstat), November 2007 data submission, released April 2008. Data originally provided to NCI by the National Center for Health Statistics.

References

FIP 6-4 Codes

---

wmlung5070 *Lung cancer mortality data for white males, 1950-69 and 1970-94*

Description

Counts and rates of lung cancer mortality data among white men by state, aggregated for 1950-1969 and 1970-1994

Usage

```
data(wmlung5070)
```

Format

A data frame with 51 observations, 1 for each state + DC, on the following 5 variables.

RATEWM_50 a numeric vector, state age-adjusted rates during 1950-69

COUNTWM_50 a numeric vector, the number of lung cancer deaths during 1950-69

RATEWM_70 a numeric vector, state age-adjusted rates during 1970-94

COUNTWM_70 a numeric vector, the number of lung cancer deaths during 1970-94

PERCENT a numeric vector of the percent change in rate from 1950-69 to 1970-94

Details

The rates on this file are directly age adjusted to the US 1970 standard population and are expressed as the number of deaths per 100,000 person-years. The row names are the 2 character postal codes for the states. Note that the data currently available on the NCI web site are from a later data submission and so may differ slightly (in first decimal place) from the rates provided here due to corrections to the dataset after its first publication. The name of each row is the state abbreviation - 2 characters. This dataset is used by the *micromapSEER* examples using the border group of "USStatesDF".

Author(s)

Linda W. Pickle and Jim Pearson of StatNet Consulting, LLC, Gaithersburg, MD

Source

Surveillance Research Program, National Cancer Institute SEER*Stat software (https://www.seer.cancer.gov/seerstat), November 2007 data submission, released April 2008. Data originally provided to NCI by the National Center for Health Statistics.

References

Devesa SS, Grauman DJ, Blot WJ, Pennello GA, Hoover RN, Fraumeni, JF Jr. Atlas of cancer mortality in the United States: 1950-94, NIH Publication 99-4564, Bethesda, MD: National Cancer Institute

---

wmlung5070US	*U.S. lung cancer mortality data for white males, 1950-1969 and 1970-1994*

Description

Count and age-adjusted rate of lung cancer mortality among white men for the total U.S., aggregated for 1950-69 and 1970-94.

Usage

```
data(wmlung5070US)
```

Format

A data frame with 1 observations on the following 5 variables.

RATEWM_50 a numeric vector, US age adjusted mortality rates for 1950-1969

COUNTWM_50 a numeric vector, US number of cases from 1950-1969

RATEWM_70 a numeric vector, US age adjusted mortality rates for 1970-1994

COUNTWM_70 a numeric vector, US number of cases from 1970-1994

PERCENT a numeric vector, change from 1950-1969 to 1970-1994 US rates.

Details

see wmlung5070 for further details. The row name is always *US* indicating US rates. This dataset is used by the *micromapSEER* examples using the border group of "USStatesDF".

Author(s)

Linda W. Pickle and Jim Pearson of StatNet Consulting, LLC, Gaithersburg, MD

References

None

Index