Package ‘micromapST’
December 8, 2016

Type Package
Version 1.1.1
Date 2016-12-07
Title Linked Micromap Plots for General U. S. and Other Geographic Areas
Author Jim Pearson <jpearson@statnetconsulting.com> and Dan Carr <dcarr@gmu.edu>, with contributions from Linda Pickle <linda@statnetconsulting.com>.
Maintainer Jim Pearson <jpearson@statnetconsulting.com>
Imports graphics, RColorBrewer, grDevices, stats, stringr, labeling, utils
Depends R (>= 3.1.0)
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 Micromaps are visualizations of georeferenced 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 micromaps 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 geographic area. By specifying a border group, the package create micromaps 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. Copyrighted 2013, 2014, 2015 and 2016 by Carr, Pearson and Pickle.
License GPL (>= 2)
LazyData no
LazyLoad no
BuildResaveData best
ByteCompile yes
NeedsCompilation no
Repository CRAN
Date/Publication 2016-12-08 01:17:59

R topics documented:

micromapST-package .................................................. 3
AfricaBG ................................................................. 7
AfricaPopData .......................................................... 9
bordGrp ................................................................. 10
ChinaBG ................................................................. 15
cnPopData .............................................................. 17
detailsVariables ....................................................... 18
Educ8thData ............................................................. 19
KansasBG ................................................................. 20
KansPopInc .............................................................. 23
MarylandBG .............................................................. 24
mdPopData .............................................................. 25
messages ................................................................. 26
micromapGDefaults ..................................................... 47
micromapGSetDefaults .................................................. 57
micromapGSetPanelDef ................................................ 58
micromapSEER .......................................................... 59
micromapST ............................................................. 60
NewYorkBG .............................................................. 79
nyPopData ............................................................... 81
panelDesc .............................................................. 82
Seer18Area ............................................................. 90
SeoulPopData ............................................................ 91
SeoulSKoreaBG ........................................................ 92
statePop2010 ........................................................... 94
TSData ................................................................. 95
UKIrelandBG ........................................................... 97
UKIrelandPopData ...................................................... 104
USSeerBG ............................................................... 105
USStatesBG ............................................................ 109
UtahBG ................................................................. 112
UtahPopData ............................................................ 113
wflung00and95 ......................................................... 116
wflung00and95US ....................................................... 117
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: 1.1.1
Date: 2016-12-07
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 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 (NORMBAR) or centered (CTRBAR)) bar charts, scatter plots and boxplots.

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 encouraged to 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.

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

```r
library(micromapST)
```

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

```r
statsDF <- 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

```r
panelDesc<-data.frame(...)
```

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

```r
micromapST(statsDF, 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 package 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 <jpearson@statnetconsulting.com>,  
with contributions from Linda Pickle  
Maintainer: "James B. Pearson Jr." <jpearson@statnetconsulting.com>  
Package compiled by "James B, Pearson, Jr." <jpearson@statnetconsulting.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


**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.
<table>
<thead>
<tr>
<th>name</th>
<th>ab</th>
<th>alt_ab</th>
<th>id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algeria</td>
<td>ALG</td>
<td>DZ</td>
<td>01</td>
</tr>
<tr>
<td>Angola</td>
<td>ANG</td>
<td>AO</td>
<td>02</td>
</tr>
<tr>
<td>Benin</td>
<td>BEN</td>
<td>BJ</td>
<td>03</td>
</tr>
<tr>
<td>Botswana</td>
<td>BOT</td>
<td>BW</td>
<td>04</td>
</tr>
<tr>
<td>Burkina Faso</td>
<td>BUF</td>
<td>BF</td>
<td>05</td>
</tr>
<tr>
<td>Burundi</td>
<td>BUR</td>
<td>BI</td>
<td>06</td>
</tr>
<tr>
<td>Cameroon</td>
<td>CAM</td>
<td>CM</td>
<td>07</td>
</tr>
<tr>
<td>Cape Verde</td>
<td>CAP</td>
<td>CV</td>
<td>08</td>
</tr>
<tr>
<td>Central African Republic</td>
<td>CAR</td>
<td>CF</td>
<td>09</td>
</tr>
<tr>
<td>Chad</td>
<td>CHA</td>
<td>TD</td>
<td>10</td>
</tr>
<tr>
<td>Comoros</td>
<td>COM</td>
<td>KM</td>
<td>11</td>
</tr>
<tr>
<td>Congo-Brazzaville</td>
<td>CNG</td>
<td>CG</td>
<td>12</td>
</tr>
<tr>
<td>Cote d'Ivoire</td>
<td>CDI</td>
<td>CI</td>
<td>13</td>
</tr>
<tr>
<td>Democratic Republic of Congo</td>
<td>ZAI</td>
<td>ZR</td>
<td>14</td>
</tr>
<tr>
<td>Djibouti</td>
<td>DJI</td>
<td>DJ</td>
<td>15</td>
</tr>
<tr>
<td>Egypt</td>
<td>EGY</td>
<td>EG</td>
<td>16</td>
</tr>
<tr>
<td>Equatorial Guinea</td>
<td>EQG</td>
<td>GQ</td>
<td>17</td>
</tr>
<tr>
<td>Eritrea</td>
<td>ERI</td>
<td>ER</td>
<td>18</td>
</tr>
<tr>
<td>Ethiopia</td>
<td>ETH</td>
<td>ET</td>
<td>19</td>
</tr>
<tr>
<td>Gabon</td>
<td>GAB</td>
<td>GA</td>
<td>20</td>
</tr>
<tr>
<td>Gambia</td>
<td>GAM</td>
<td>GM</td>
<td>21</td>
</tr>
<tr>
<td>Ghana</td>
<td>GHA</td>
<td>GH</td>
<td>22</td>
</tr>
<tr>
<td>Guinea</td>
<td>GIN</td>
<td>GN</td>
<td>23</td>
</tr>
<tr>
<td>Guinea-Bissau</td>
<td>GUB</td>
<td>GW</td>
<td>24</td>
</tr>
<tr>
<td>Kenya</td>
<td>KEN</td>
<td>KE</td>
<td>25</td>
</tr>
<tr>
<td>Lesotho</td>
<td>LES</td>
<td>LS</td>
<td>26</td>
</tr>
<tr>
<td>Liberia</td>
<td>LIB</td>
<td>LR</td>
<td>27</td>
</tr>
<tr>
<td>Libya</td>
<td>LAJ</td>
<td>LY</td>
<td>28</td>
</tr>
<tr>
<td>Madagascar</td>
<td>MAD</td>
<td>MG</td>
<td>29</td>
</tr>
<tr>
<td>Malawi</td>
<td>MAA</td>
<td>MW</td>
<td>30</td>
</tr>
<tr>
<td>Mali</td>
<td>MAL</td>
<td>ML</td>
<td>31</td>
</tr>
<tr>
<td>Mauritania</td>
<td>MAU</td>
<td>MR</td>
<td>32</td>
</tr>
<tr>
<td>Morocco</td>
<td>MOR</td>
<td>MA</td>
<td>33</td>
</tr>
<tr>
<td>Mozambique</td>
<td>MOZ</td>
<td>MZ</td>
<td>34</td>
</tr>
<tr>
<td>Namibia</td>
<td>NAM</td>
<td>NA</td>
<td>35</td>
</tr>
<tr>
<td>Niger</td>
<td>NIG</td>
<td>NE</td>
<td>36</td>
</tr>
<tr>
<td>Nigeria</td>
<td>NIR</td>
<td>NG</td>
<td>37</td>
</tr>
<tr>
<td>Rwanda</td>
<td>RWA</td>
<td>RW</td>
<td>38</td>
</tr>
<tr>
<td>Sao Tome and Principe</td>
<td>STP</td>
<td>ST</td>
<td>39</td>
</tr>
<tr>
<td>Senegal</td>
<td>SEN</td>
<td>SN</td>
<td>40</td>
</tr>
<tr>
<td>Sierra Leone</td>
<td>SIL</td>
<td>SL</td>
<td>41</td>
</tr>
<tr>
<td>Somalia</td>
<td>SOM</td>
<td>SO</td>
<td>42</td>
</tr>
<tr>
<td>South Africa</td>
<td>SOU</td>
<td>SA</td>
<td>43</td>
</tr>
<tr>
<td>Sudan</td>
<td>SUD</td>
<td>SD</td>
<td>44</td>
</tr>
<tr>
<td>Swaziland</td>
<td>SWA</td>
<td>SZ</td>
<td>45</td>
</tr>
<tr>
<td>Tanzania</td>
<td>TAN</td>
<td>TZ</td>
<td>46</td>
</tr>
<tr>
<td>Togo</td>
<td>TOG</td>
<td>TG</td>
<td>47</td>
</tr>
</tbody>
</table>
When this border group was created, there appeared to be no consistent 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 abbreviations 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**

```r
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 country’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 popu-
  lation of Africa.

**Details**

This dataset was pulled from wikipedia 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 many 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 abbreviations, 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 clarity, 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 area-
Parms$Map.L2Borders variable in the border group must be set to TRUE for the package
to draw the layer 2 boundaries. If a border group does not use an intermediate level out-
line, L2VisBorders should be set to L3VisBorders or NA and the areaParms$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 variable 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 `statsDF` or the row.names of the `statsDF` 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.Hdr1 - 1st title for the id type glyphics column.
ID.Hdr2 - 2nd title for the id type glyphics column.
Map.Hdr1 - 1st title for the map type glyphics column.
Map.Hdr2 - 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 draw 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 draw a regional boundary 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 abbreviation (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 specified. Used as the represented name of the sub-area in "ID" glyphics columns when ‘plotNames’ = "full" is specified (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 specified ‘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 creator 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). (areaVisBorders) The boundaries of an intermediate level for general orientation. (L2VisBorders) The boundaries of regional areas for highlight overlays and regional only mapping. (RegVisBorders) The boundaries of the an outline of the entire mapping area. (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, UKIrelandBG, 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 drawing 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 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 proceed 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 precede 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:

*USSatesBG USSeerBG UKIrelendBG ChinaBG* border groups.
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 prov-
- **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.

<table>
<thead>
<tr>
<th>name</th>
<th>ab</th>
<th>alt_ab</th>
<th>id</th>
<th>region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anhui</td>
<td>AH</td>
<td>CN.AH</td>
<td>34</td>
<td>Huadong</td>
</tr>
<tr>
<td>Beijing</td>
<td>BJ</td>
<td>CN.BJ</td>
<td>11</td>
<td>Huabei</td>
</tr>
<tr>
<td>Chongqing</td>
<td>CQ</td>
<td>CN.CQ</td>
<td>50</td>
<td>Xinan</td>
</tr>
<tr>
<td>Fujian</td>
<td>FJ</td>
<td>CN.FJ</td>
<td>35</td>
<td>Huadong</td>
</tr>
<tr>
<td>Gansu</td>
<td>GS</td>
<td>CN.GS</td>
<td>62</td>
<td>Xibei</td>
</tr>
<tr>
<td>Guangdong</td>
<td>GD</td>
<td>CN.GD</td>
<td>44</td>
<td>Zhongnan</td>
</tr>
<tr>
<td>Guangxi</td>
<td>GX</td>
<td>CN.GX</td>
<td>45</td>
<td>Zhongnan</td>
</tr>
<tr>
<td>County</td>
<td>Abbreviation</td>
<td>Code</td>
<td>Region</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>-------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>Guizhou</td>
<td>GZ</td>
<td>CN.GZ</td>
<td>Xinan</td>
<td></td>
</tr>
<tr>
<td>Hainan</td>
<td>HI</td>
<td>CN.HA</td>
<td>Zhongnan</td>
<td></td>
</tr>
<tr>
<td>Hebei</td>
<td>HE</td>
<td>CN.HB</td>
<td>Huabei</td>
<td></td>
</tr>
<tr>
<td>Heilongjiang</td>
<td>HL</td>
<td>CN.HL</td>
<td>Dongbei</td>
<td></td>
</tr>
<tr>
<td>Henan</td>
<td>HA</td>
<td>CN.HE</td>
<td>Zhongnan</td>
<td></td>
</tr>
<tr>
<td>Hubei</td>
<td>HB</td>
<td>CN.HU</td>
<td>Zhongnan</td>
<td></td>
</tr>
<tr>
<td>Hunan</td>
<td>HN</td>
<td>CN.HN</td>
<td>Zhongnan</td>
<td></td>
</tr>
<tr>
<td>Jiangsu</td>
<td>JS</td>
<td>CN.JS</td>
<td>Huadong</td>
<td></td>
</tr>
<tr>
<td>Jiangxi</td>
<td>JX</td>
<td>CN.JX</td>
<td>Huadong</td>
<td></td>
</tr>
<tr>
<td>Jilin</td>
<td>JL</td>
<td>CN.JL</td>
<td>Dongbei</td>
<td></td>
</tr>
<tr>
<td>Liaoning</td>
<td>LN</td>
<td>CN.LN</td>
<td>Dongbei</td>
<td></td>
</tr>
<tr>
<td>Nei Menggu</td>
<td>NM</td>
<td>CN.NM</td>
<td>Huadong</td>
<td></td>
</tr>
<tr>
<td>Ningxia Hui</td>
<td>NX</td>
<td>CN.NX</td>
<td>Xibei</td>
<td></td>
</tr>
<tr>
<td>Qinghai</td>
<td>QH</td>
<td>CN.QH</td>
<td>Xibei</td>
<td></td>
</tr>
<tr>
<td>Shaanxi</td>
<td>SN</td>
<td>CN.SA</td>
<td>Xibei</td>
<td></td>
</tr>
<tr>
<td>Shandong</td>
<td>SD</td>
<td>CN.SD</td>
<td>Huadong</td>
<td></td>
</tr>
<tr>
<td>Shanghai</td>
<td>SH</td>
<td>CN.SH</td>
<td>Huadong</td>
<td></td>
</tr>
<tr>
<td>Shanxi</td>
<td>SX</td>
<td>CN.SX</td>
<td>Huabei</td>
<td></td>
</tr>
<tr>
<td>Sichuan</td>
<td>SC</td>
<td>CN.SC</td>
<td>Xinan</td>
<td></td>
</tr>
<tr>
<td>Tianjin</td>
<td>TJ</td>
<td>CN.TJ</td>
<td>Huabei</td>
<td></td>
</tr>
<tr>
<td>Xinjiang Uygur</td>
<td>XJ</td>
<td>CN.XJ</td>
<td>Xibei</td>
<td></td>
</tr>
<tr>
<td>Xizang</td>
<td>XZ</td>
<td>CN.XZ</td>
<td>Xinan</td>
<td></td>
</tr>
<tr>
<td>Yunnan</td>
<td>YN</td>
<td>CN.YN</td>
<td>Xinan</td>
<td></td>
</tr>
<tr>
<td>Zhejiang</td>
<td>ZJ</td>
<td>CN.ZJ</td>
<td>Buadong</td>
<td></td>
</tr>
<tr>
<td>Hongkong</td>
<td>HK</td>
<td>CN.HK</td>
<td>Zhongnan</td>
<td></td>
</tr>
<tr>
<td>Macau</td>
<td>MC</td>
<td>CN.MC</td>
<td>Zhongnan</td>
<td></td>
</tr>
<tr>
<td>Taiwan</td>
<td>TW</td>
<td>CN.TW</td>
<td>Huadong</td>
<td></td>
</tr>
</tbody>
</table>

The ChinaBG supports two sets of abbreviations for each county (province or metro area). No consistent 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.

---

**Description**

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

**Usage**

```r
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.

---

**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**

```r
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 provides a table to `micromapST` for verification of the details variables provided by a user to override the package's default values. The `varName` is the exact name of the variable. If the variable name provided by the user does not match this list, it is flagged as an error and ignored. The tests 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 checks 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 a glyphic and column basis. This allows each column to be uniquely tuned to the user's requirements. The variable names have also been simplified 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**

`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 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Abbreviation</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allen</td>
<td>AL</td>
<td>20001</td>
</tr>
<tr>
<td>Anderson</td>
<td>AN</td>
<td>20003</td>
</tr>
<tr>
<td>Atchison</td>
<td>AT</td>
<td>20005</td>
</tr>
<tr>
<td>Barber</td>
<td>BA</td>
<td>20007</td>
</tr>
<tr>
<td>Barton</td>
<td>BT</td>
<td>20009</td>
</tr>
<tr>
<td>Bourbon</td>
<td>BB</td>
<td>20011</td>
</tr>
<tr>
<td>Brown</td>
<td>BR</td>
<td>20013</td>
</tr>
<tr>
<td>Butler</td>
<td>BU</td>
<td>20015</td>
</tr>
<tr>
<td>Chase</td>
<td>CS</td>
<td>20017</td>
</tr>
<tr>
<td>Chautauqua</td>
<td>CQ</td>
<td>20019</td>
</tr>
<tr>
<td>Cherokee</td>
<td>CK</td>
<td>20021</td>
</tr>
<tr>
<td>Cheyenne</td>
<td>CN</td>
<td>20023</td>
</tr>
<tr>
<td>Clark</td>
<td>CA</td>
<td>20025</td>
</tr>
<tr>
<td>Clay</td>
<td>CY</td>
<td>20027</td>
</tr>
<tr>
<td>Cloud</td>
<td>CD</td>
<td>20029</td>
</tr>
<tr>
<td>Coffey</td>
<td>CF</td>
<td>20031</td>
</tr>
<tr>
<td>Comanche</td>
<td>CM</td>
<td>20033</td>
</tr>
<tr>
<td>Cowley</td>
<td>CL</td>
<td>20035</td>
</tr>
<tr>
<td>Crawford</td>
<td>CR</td>
<td>20037</td>
</tr>
<tr>
<td>Decatur</td>
<td>DC</td>
<td>20039</td>
</tr>
<tr>
<td>Dickinson</td>
<td>DK</td>
<td>20041</td>
</tr>
<tr>
<td>Doniphan</td>
<td>DP</td>
<td>20043</td>
</tr>
<tr>
<td>Douglas</td>
<td>DG</td>
<td>20045</td>
</tr>
<tr>
<td>Edwards</td>
<td>ED</td>
<td>20047</td>
</tr>
<tr>
<td>Elk</td>
<td>EK</td>
<td>20049</td>
</tr>
<tr>
<td>Ellis</td>
<td>EL</td>
<td>20051</td>
</tr>
<tr>
<td>Ellsworth</td>
<td>EW</td>
<td>20053</td>
</tr>
<tr>
<td>Finney</td>
<td>FI</td>
<td>20055</td>
</tr>
<tr>
<td>Ford</td>
<td>FO</td>
<td>20057</td>
</tr>
<tr>
<td>Franklin</td>
<td>FR</td>
<td>20059</td>
</tr>
<tr>
<td>Geary</td>
<td>GE</td>
<td>20061</td>
</tr>
<tr>
<td>Gove</td>
<td>GO</td>
<td>20063</td>
</tr>
<tr>
<td>Graham</td>
<td>GH</td>
<td>20065</td>
</tr>
<tr>
<td>Grant</td>
<td>GT</td>
<td>20067</td>
</tr>
<tr>
<td>Gray</td>
<td>GY</td>
<td>20069</td>
</tr>
<tr>
<td>Greeley</td>
<td>GL</td>
<td>20071</td>
</tr>
<tr>
<td>Greenwood</td>
<td>GW</td>
<td>20073</td>
</tr>
<tr>
<td>Hamilton</td>
<td>HM</td>
<td>20075</td>
</tr>
<tr>
<td>Harper</td>
<td>HP</td>
<td>20077</td>
</tr>
<tr>
<td>Harvey</td>
<td>HV</td>
<td>20079</td>
</tr>
<tr>
<td>Haskell</td>
<td>HS</td>
<td>20081</td>
</tr>
<tr>
<td>Hodgeman</td>
<td>HG</td>
<td>20083</td>
</tr>
<tr>
<td>Jackson</td>
<td>JA</td>
<td>20085</td>
</tr>
<tr>
<td>County</td>
<td>Code</td>
<td>Zip Code</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>Jefferson</td>
<td>JF</td>
<td>20087</td>
</tr>
<tr>
<td>Jewell</td>
<td>JW</td>
<td>20089</td>
</tr>
<tr>
<td>Johnson</td>
<td>JO</td>
<td>20091</td>
</tr>
<tr>
<td>Kearny</td>
<td>KE</td>
<td>20093</td>
</tr>
<tr>
<td>Kingman</td>
<td>KM</td>
<td>20095</td>
</tr>
<tr>
<td>Kiowa</td>
<td>KW</td>
<td>20097</td>
</tr>
<tr>
<td>Labette</td>
<td>LB</td>
<td>20099</td>
</tr>
<tr>
<td>Lane</td>
<td>LE</td>
<td>20101</td>
</tr>
<tr>
<td>Leavenworth</td>
<td>LV</td>
<td>20103</td>
</tr>
<tr>
<td>Lincoln</td>
<td>LC</td>
<td>20105</td>
</tr>
<tr>
<td>Linn</td>
<td>LN</td>
<td>20107</td>
</tr>
<tr>
<td>Logan</td>
<td>LG</td>
<td>20109</td>
</tr>
<tr>
<td>Lyon</td>
<td>LY</td>
<td>20111</td>
</tr>
<tr>
<td>Marion</td>
<td>MN</td>
<td>20115</td>
</tr>
<tr>
<td>Marshall</td>
<td>MS</td>
<td>20117</td>
</tr>
<tr>
<td>McPherson</td>
<td>MP</td>
<td>20113</td>
</tr>
<tr>
<td>Meade</td>
<td>ME</td>
<td>20119</td>
</tr>
<tr>
<td>Miami</td>
<td>MI</td>
<td>20121</td>
</tr>
<tr>
<td>Mitchell</td>
<td>MC</td>
<td>20123</td>
</tr>
<tr>
<td>Montgomery</td>
<td>MG</td>
<td>20125</td>
</tr>
<tr>
<td>Morris</td>
<td>MR</td>
<td>20127</td>
</tr>
<tr>
<td>Morton</td>
<td>MT</td>
<td>20129</td>
</tr>
<tr>
<td>Nemaha</td>
<td>NM</td>
<td>20131</td>
</tr>
<tr>
<td>Neosho</td>
<td>NO</td>
<td>20133</td>
</tr>
<tr>
<td>Ness</td>
<td>NS</td>
<td>20135</td>
</tr>
<tr>
<td>Norton</td>
<td>NT</td>
<td>20137</td>
</tr>
<tr>
<td>Osage</td>
<td>OS</td>
<td>20139</td>
</tr>
<tr>
<td>Osborne</td>
<td>OB</td>
<td>20141</td>
</tr>
<tr>
<td>Ottawa</td>
<td>OT</td>
<td>20143</td>
</tr>
<tr>
<td>Pawnee</td>
<td>PN</td>
<td>20145</td>
</tr>
<tr>
<td>Phillips</td>
<td>PL</td>
<td>20147</td>
</tr>
<tr>
<td>Pottawatomie</td>
<td>PT</td>
<td>20149</td>
</tr>
<tr>
<td>Pratt</td>
<td>PR</td>
<td>20151</td>
</tr>
<tr>
<td>Rawlins</td>
<td>RA</td>
<td>20153</td>
</tr>
<tr>
<td>Reno</td>
<td>RN</td>
<td>20155</td>
</tr>
<tr>
<td>Republic</td>
<td>RP</td>
<td>20157</td>
</tr>
<tr>
<td>Rice</td>
<td>RC</td>
<td>20159</td>
</tr>
<tr>
<td>Riley</td>
<td>RL</td>
<td>20161</td>
</tr>
<tr>
<td>Rooks</td>
<td>RO</td>
<td>20163</td>
</tr>
<tr>
<td>Rush</td>
<td>RH</td>
<td>20165</td>
</tr>
<tr>
<td>Russell</td>
<td>RS</td>
<td>20167</td>
</tr>
<tr>
<td>Saline</td>
<td>SA</td>
<td>20169</td>
</tr>
<tr>
<td>Scott</td>
<td>SC</td>
<td>20171</td>
</tr>
<tr>
<td>Sedgwick</td>
<td>SG</td>
<td>20173</td>
</tr>
<tr>
<td>Seward</td>
<td>SW</td>
<td>20175</td>
</tr>
<tr>
<td>Shawnee</td>
<td>SN</td>
<td>20177</td>
</tr>
<tr>
<td>Sheridan</td>
<td>SD</td>
<td>20179</td>
</tr>
<tr>
<td>Sherman</td>
<td>SH</td>
<td>20181</td>
</tr>
</tbody>
</table>
There are no alternate abbreviations or regions associated 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.

**Description**

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

**Usage**

```r
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 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 associated 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**


**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 2040.

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:

```
***XXX NNNNNN text of message
```

or

```
***XXX 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 identifier 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.
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, 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/arguments. 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 entries 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-area 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 parameter ‘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 <DFName> 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, tconf, 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:

<table>
<thead>
<tr>
<th>callArgAbbr</th>
<th>callArg</th>
</tr>
</thead>
<tbody>
<tr>
<td>rowNamesCol</td>
<td>RNC</td>
</tr>
<tr>
<td>sortVar</td>
<td>SORT</td>
</tr>
</tbody>
</table>

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 argument 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 <DFName> 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 provided, 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
SeoulSKBG   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: <MissBGOObjects>
The border group has been located and loaded into micromapST. However, the <MissBGOObjects> 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 message and re-run. Execution was stopped.

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 be 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 than one value is provided, only the first value in the data will be used.

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 USSeeBG border group for more details and an example. At this time, only the USSeeBG 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 USSeeBG border group.

0192 CARG-RN rowNames=’seer’ is only supported for the USSeeBG 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 type/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 descending 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(5,5,5,5,1,5,5,5) would be valid, but c(4,4,4,4,4,1,4,4,4,4) 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 descending 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:
<table>
<thead>
<tr>
<th>before:</th>
<th>1000</th>
<th>2000</th>
<th>3000</th>
<th>4000</th>
</tr>
</thead>
<tbody>
<tr>
<td>after: in hundreds</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

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 postended to the label.

Example:

<table>
<thead>
<tr>
<th>before:</th>
<th>0</th>
<th>5000</th>
<th>10000</th>
<th>15000</th>
<th>20000</th>
</tr>
</thead>
<tbody>
<tr>
<td>after:</td>
<td>0</td>
<td>5K</td>
<td>10K</td>
<td>15K</td>
<td>20K</td>
</tr>
</tbody>
</table>

‘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 resub. ***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 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 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 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 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 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***
messages

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 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***

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 micromapSTSetDefaults 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 <lastVarName> 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 <lastVarName> 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 validation 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:

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

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:

1) If does not exist, create dummy vector to for variable
2) Verify type of vector is numeric, integer, or character. All other types not permitted.
3) 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:

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.

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

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

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 redundant.
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 panelDesc 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 <statsD-
Frame> 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 <statsD-
Frame> 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. <usage>
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. <usage>
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. <usage>
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 precede the last column in the data frame. That is must have a lower column number or a column name that precedes the last column name in the data frame.
messages 43

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**

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

02BF 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 statsDFrame. 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 bot-
               tom. 
               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***
micromapGDefaults

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

Back to the details section - Rest of the details section
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 glyphics. 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 highlighted 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 characteristics 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 descending 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(0,0,0,0,0,0,0,0,0,0,0,0), 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(7,7,7,7,1.6,7,7,7,7,7), 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(0,0,0,0,1,0,0,0,0,0). 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 number of 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.
\texttt{sc} = 1.08, x axis scale expansion factor. Applied to the data range to calculate the graph’s range.
\texttt{pad} = 0.67, y axis padding for integer plotting locations
\texttt{padex} = 0.34 inches, total panel padding (i.e., 0.17 at top and bottom of panel)
\texttt{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.
\texttt{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.
\texttt{rowSizeMn} = 0.5 inches. Minimum panel group height.
\texttt{rowSizeMx} = 1.25 inches. Maximum panel group height.
\texttt{rowSizeMin} = 1.65 units. Minor panel group height in units for single area panel.
\texttt{rowsizeMaj} = 7 units. Major panel group height for all panels, except single area panel.
\texttt{colSepGap} = 0.1 inches. Glyphics column separator space.
\texttt{colSizeMin} = 0.75 inches. Minimum Glyphics column size.
\texttt{colSizeMax} = 1.8 inches. Maximum Glyphics column size.
\texttt{rcRatioMin} = 0.25, minimum row size to col size ratio permitted. (not implemented)
\texttt{rcRatioMax} = 2, maximum row size to col size ratio permitted. (not implemented)
\texttt{XAxis.L.mcex} = 0.88888 Size used for large XAxis labels
\texttt{XAxis.M.mcex} = 0.77778 Size used for medium XAxis labels
\texttt{XAxis.S.mcex} = 0.66667 Size used for small XAxis labels
\texttt{XAxis.Sp.mcex} = 0.2 Size used for XAxis space between lines of labels
\texttt{XAxis.offset} = 0.0 inches. XAxis offset
\texttt{XAxis.indent} = 0.10 1000th inches. First and Last label indents from edge.
\texttt{XAxis.nGridpIn} = 3.4 labels per inch in XAxis - initial objective.
\texttt{XAxis.gapPC} = 0.75 (*100 for percentage). Percentage of the width of a space used to determine label overlaps.
\texttt{XAxis.staggered} = TRUE, enable staggered label feature - NOT USED
\texttt{YAxis.cex} = 0.33332 cex size of YAxis labels
\texttt{YAxis.offset} = 0.0 lines. Offset of YAxis labels from panel edge.
\texttt{YAxis.nGridpIn} = 5 labels. Number of labels per inch for YAxis - initial goal.
\texttt{YAxis.staggered} = TRUE. Enable staggered labels on YAxis - NOT USED.
\texttt{YAxis.width} = 0.2 inches. Extra column gap size required to support drawing an Y-Axis for TS and ScatDot glyphics

\texttt{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.

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 not used. Should never be greater than 0.90. Usable range is 0.333 to 0.90.

- **CSNBar.First.barht** = 0.50000, * 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.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.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.Fg.col = "white", * background of maps
Map.Bg.Line.col = "white", * 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 dot
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= parameter 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
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.

Details

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 highlight 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.
micromapGSetPanelDef

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 cumulative highlighting 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 highlight 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.
micromapSEER

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(statsFrame, panelDesc, ...)

Arguments

statsFrame - data frame of the data required for creating the graphical glyphs 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.
- 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,
micromapST

- 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 distributed with the package as .rda datasets.

Usage

```r
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,
  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 `scadot` 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
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 are 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, abbreviations, optional alternate abbreviations and numeric identifiers defined for each area in the particular border group in the areaNamesAbbrsIDs 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 rowNames information of the statsDFrame.

The linkage values are validated 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.)
rowNamesCol value must be a column number or column name within the statsDFrame 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 statsDFrame.

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 ‘full’.

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 contain at least one sub-area with data provided by the caller. If `FALSE`, the entire area is drawn. The default is `FALSE`. Relational boundaries are not required, but suggested. See border group documentation for more details.

regionsB when regional boundaries are provided, controls whether to overlay the boundaries 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 set 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 regional boundaries are drawn. See border group documentation for more details.

grpPattern The micromapST package generates the pattern of rows to panel groups automatically. The pattern is based on a maximum of 5 rows per group and number of rows can only descend from the edges toward the median row. The pattern generated by the package can be overridden 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 descending from the outside to the middle of the vector.

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 glyphs,
- 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 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
• 2 accent colors for "mapmedian" sub-area colors for above median and below 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., `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 highlighted state color: "light yellow" for "map", "mapcum", "maptail"
• 2 highlighted 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, structure 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 desirable 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:

```r
details=list(SBar.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 geographic 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` column in the `panelDesc` data.frame

**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

See Also
micromapST, micromapSEER

**Examples**

```r
###
# 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
data(wflung00and95,wflung00and95US,wflung00cnty,envir=environment())
wflboxlist = 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,NA,NA,"US Rate 2000-4",NA),
    panelData= c("","","",wflboxlist")
)
```
micromapST

# set up PDF output file, call package

df(file="Ex01-WFLung-2000-2004-State-Dot-County-Box.pdf", width=7.5, height=10)

micromapST(wflung00and95, panelDesc01, sortOrder=1, ascend=FALSE,
        title="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("","",""),
    lab3=c("mapcum",""),
    col1=c(NA,NA,"RATEWM_50","PERCENT"),
    col2=c(NA,NA,"RATEWM_70",NA)
)

colorsRrgb = matrix(t(c(
    213,  62,  79, #region 1: red     #D34E4F - Rust Red
    252, 141,  89, #region 2: orange   #FC8D59 - Brn/Org
    253, 225, 139, #region 3: green    #EE8B8B - 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 #E6F98 - YellowGreen
    255, 174, 185,  # alternate shape upper  #FFAE9 - Mauve
    191, 239, 255,  # alternate shape lower   #BF8F8F - Cyan
    242, 241, 242,  # lightest grey for non-referenced sub-areas  #F2F2F2
    234, 234, 234),  # lighter grey for background - non-active sub-areas. #EAEAE
    ncol=3, byrow=TRUE)

xcolors = c( grDevices::rgb(colorsRrgb[,1], colorsRrgb[,2], colorsRrgb[,3],
    maxColorValue=255),
# set solid colors
grDevices::rgb(colorsRgb[,1],colorsRgb[,2],colorsRgb[,3],64,
maxColorValue=255))

# set translucent colors for time series.

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_yellgreen","l_mauve",
"l_cyan","l_lightest grey","l_lighter grey")

pdf(file="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()

###

# 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])

# TSD ata 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="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="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="Ex05-Stkd-Bar-var-height.pdf",width=7.5,height=10)
micromapST(statePop2010, panelDesc05, sortVar="OtherWBH", ascend=FALSE,
```
title=c("Ex05-Stacked Bars: 2010 Census Pop by Race, Sorted by Count 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")
)

df <- read.csv("statePop2010.csv")

pdf(file="Ex06-Stkd-Bar-fixedheight-nodot.pdf",width=7.5,height=10)

micromapST(statePop2010,panelDesc06,sortVar=4,ascend=FALSE,
   title=c("Ex07-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, % < 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-Stacked Bars: Educational Progress (NAEP) in Math, 2011, 8th Grade", "Centered at Not-Prof vs. Prof")

df <- read.csv("example.csv")

pdf(file="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")

df <- read.csv("example.csv")

pdf(file="Ex08-DotSignificants.pdf", width=7.5, height=10)

micromapST(Seer18Area, panelDesc08, sortVar="Rate20022006", ascend=FALSE, title=ExTitle, rowNames="alias", rowNamesCol='Registry', bordGrp="USSeerBG")
micromapST

plotNames="ab"

dev.off()
#
# Both calls are effectively identical.
#
##### End of examples 08

#####
# micromapST - Example 09 - A linked micromap of the counties of the state
# of Kansas using the border group "KansasDF". The KansPopInc data is shown
# using two dot glypics - current population and average increase per county.
#####
data(KansPopInc,envir=environment())

# set up 4 column page layout

panelDesc09 = data.frame(
  type=c("mapcum","id","dot","dot"),
  lab1=c("","","Population","Average Inc."),
  lab2=c("","","in 2000","per year"),
  lab3=c("","","UNITS",""),
  col1=c(NA,NA,"Pop","AvgInc")
)

ExTitle <- c("Ex09 - Kansas Pop data 8.5x14",
"Current Pop and Average Inc-scaling=sn")

df(file="Ex09-Kansas Population and Inc-8.5x14-sn.pdf",width=7.5,height=13)
# legal size page (8.5 x 14) to handle 105 counties.

# use scaling "sn" to clean up x-axis labels.

micromapST(KansPopInc,panelDesc09,
  sortVar=c("AvgInc","Pop"),ascend=FALSE,
  title=ExTitle,
  axisScale="sn",
  rowNames="full",rowNamesCol='County',
  bordGrp="KansasBG",
  plotNames="ab")

dev.off()

#
##### Example # 09

#####
# Example # 10 - Counties in Kansas on an 11 x 17 page
#
# set up 4 column page layout

panelDesc10 = data.frame(
  type=c("map", "id", "dot", "dot"),
  lab1=c("", "", "Population", "Average Inc.")
)

text("Ex10 - Kansas Pop data 11x17,"
  "Current Pop and Average Inc - scaling=e")

pdf(file="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=text("Ex10 - Kansas Pop data 11x17,"
    "Current Pop and Average Inc - scaling=e")
)

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$Diff0_10 <- nypopData$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","Inc")
)
micromapST

,lab2=c("","","2010","2000","","2000to2010")
,lab3=c("","","","","",""")
,co11=c(NA,NA,"Pop_2010","Pop_2000","Pop_2000","Dif00_10")
,co12=c(NA,NA,"","","",""")
,co1Size=c(NA,NA,15,20,5,20)
)

ExTitle <- c("Ex11 - New York Population data",
"2010 Pop and since 2000-colSize,sn,stag")

df(file="Ex11-New York Pop 2010 and Change-sn colSize.pdf",
width=7.5,height=10.5)

micromapST(nypopData,panelDesc11,
sortBy="Pop_2000",ascend=FALSE,
title=ExTitle,
rowNames="full",rowNamesCol="Area",
axisScale="sn",staggerLab=TRUE,
bordGrp="NewYorkBG"
)

dev.off()
#
#
#### End of Examples 11


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("","","","",""")
  ,co11=c(NA,NA,"X1970","X2010","X1970")
  ,co12=c(NA,NA,"","",""")
)

ExTitle <- c("Ex12 - Maryland Population-map",
"1970 and 2010 Pop and Change,stag,sn")

df(file="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")


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.

```r
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")

dataEnv = data.frame(sapply(SeoulPopData, as.character))

dev.off()
```

---

micromapST(SeoulPopData, panelDesc14,
  sortVar = 3, ascend = FALSE, # sort based on the population
  title = ExTitle,
  rowNames = "full", rowNamesCol = "District",
  bordgrp = "SeoulSKoreaBG",
  plotNames = "full"
)

---

```
data(UtahPopData, envir = environment())

# 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"),
  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="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 = "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 = "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"),
coll = c(NA, NA,"Projection","PercOf", "Est2Time"))

pdf(file = "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

---

**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.

<table>
<thead>
<tr>
<th>name</th>
<th>ab</th>
<th>id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Albany</td>
<td>ALBA</td>
<td>36001</td>
</tr>
<tr>
<td>Allegany</td>
<td>ALLE</td>
<td>36003</td>
</tr>
<tr>
<td>Bronx</td>
<td>BRON</td>
<td>36005</td>
</tr>
<tr>
<td>Broome</td>
<td>BROO</td>
<td>36007</td>
</tr>
<tr>
<td>Cattaraugus</td>
<td>CATT</td>
<td>36009</td>
</tr>
<tr>
<td>Cayuga</td>
<td>CAYU</td>
<td>36011</td>
</tr>
<tr>
<td>Chautauqua</td>
<td>CHAU</td>
<td>36013</td>
</tr>
<tr>
<td>Chemung</td>
<td>CHEM</td>
<td>36015</td>
</tr>
<tr>
<td>Chenango</td>
<td>CHEN</td>
<td>36017</td>
</tr>
<tr>
<td>Clinton</td>
<td>CLIN</td>
<td>36019</td>
</tr>
<tr>
<td>Columbia</td>
<td>COLU</td>
<td>36021</td>
</tr>
<tr>
<td>Cortland</td>
<td>CORT</td>
<td>36023</td>
</tr>
<tr>
<td>Delaware</td>
<td>DELA</td>
<td>36025</td>
</tr>
<tr>
<td>Dutchess</td>
<td>DUTC</td>
<td>36027</td>
</tr>
<tr>
<td>Erie</td>
<td>ERIE</td>
<td>36029</td>
</tr>
<tr>
<td>Essex</td>
<td>ESSE</td>
<td>36031</td>
</tr>
<tr>
<td>Franklin</td>
<td>FRAN</td>
<td>36033</td>
</tr>
<tr>
<td>Fulton</td>
<td>FULT</td>
<td>36035</td>
</tr>
<tr>
<td>Genesee</td>
<td>GENE</td>
<td>36037</td>
</tr>
<tr>
<td>Greene</td>
<td>GREE</td>
<td>36039</td>
</tr>
<tr>
<td>Hamilton</td>
<td>HAMI</td>
<td>36041</td>
</tr>
<tr>
<td>Herkimer</td>
<td>HERK</td>
<td>36043</td>
</tr>
<tr>
<td>Jefferson</td>
<td>JEFF</td>
<td>36045</td>
</tr>
<tr>
<td>Kings</td>
<td>KING</td>
<td>36047</td>
</tr>
<tr>
<td>Lewis</td>
<td>LEWI</td>
<td>36049</td>
</tr>
<tr>
<td>Livingston</td>
<td>LIVI</td>
<td>36051</td>
</tr>
<tr>
<td>Madison</td>
<td>MADI</td>
<td>36053</td>
</tr>
<tr>
<td>County</td>
<td>Abbreviation</td>
<td>FIPS Code</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Monroe</td>
<td>MONR</td>
<td>36055</td>
</tr>
<tr>
<td>Montgomery</td>
<td>MONT</td>
<td>36057</td>
</tr>
<tr>
<td>Nassau</td>
<td>NASS</td>
<td>36059</td>
</tr>
<tr>
<td>New York</td>
<td>NEWY</td>
<td>36061</td>
</tr>
<tr>
<td>Niagara</td>
<td>NIAG</td>
<td>36063</td>
</tr>
<tr>
<td>Oneida</td>
<td>ONEI</td>
<td>36065</td>
</tr>
<tr>
<td>Onondaga</td>
<td>ONON</td>
<td>36067</td>
</tr>
<tr>
<td>Ontario</td>
<td>ONTA</td>
<td>36069</td>
</tr>
<tr>
<td>Orange</td>
<td>ORAN</td>
<td>36071</td>
</tr>
<tr>
<td>Orleans</td>
<td>ORLE</td>
<td>36073</td>
</tr>
<tr>
<td>Oswego</td>
<td>OSWE</td>
<td>36075</td>
</tr>
<tr>
<td>Otsego</td>
<td>OTSE</td>
<td>36077</td>
</tr>
<tr>
<td>Putnam</td>
<td>PUTN</td>
<td>36079</td>
</tr>
<tr>
<td>Queens</td>
<td>QUEE</td>
<td>36081</td>
</tr>
<tr>
<td>Rensselaer</td>
<td>RENS</td>
<td>36083</td>
</tr>
<tr>
<td>Richmond</td>
<td>RICH</td>
<td>36085</td>
</tr>
<tr>
<td>Rockland</td>
<td>ROCK</td>
<td>36087</td>
</tr>
<tr>
<td>St. Lawrence</td>
<td>STLA</td>
<td>36089</td>
</tr>
<tr>
<td>Saratoga</td>
<td>SARA</td>
<td>36091</td>
</tr>
<tr>
<td>Schenectady</td>
<td>SCHE</td>
<td>36093</td>
</tr>
<tr>
<td>Schoharie</td>
<td>SCHO</td>
<td>36095</td>
</tr>
<tr>
<td>Schuyler</td>
<td>SCHU</td>
<td>36097</td>
</tr>
<tr>
<td>Seneca</td>
<td>SENE</td>
<td>36099</td>
</tr>
<tr>
<td>Steuben</td>
<td>STEU</td>
<td>36101</td>
</tr>
<tr>
<td>Suffolk</td>
<td>SUFF</td>
<td>36103</td>
</tr>
<tr>
<td>Sullivan</td>
<td>SULL</td>
<td>36105</td>
</tr>
<tr>
<td>Tioga</td>
<td>TIOG</td>
<td>36107</td>
</tr>
<tr>
<td>Tompkins</td>
<td>TOMP</td>
<td>36109</td>
</tr>
<tr>
<td>Ulster</td>
<td>ULST</td>
<td>36111</td>
</tr>
<tr>
<td>Warren</td>
<td>WARR</td>
<td>36113</td>
</tr>
<tr>
<td>Washington</td>
<td>WASH</td>
<td>36115</td>
</tr>
<tr>
<td>Wayne</td>
<td>WAYN</td>
<td>36117</td>
</tr>
<tr>
<td>Westchester</td>
<td>WEST</td>
<td>36119</td>
</tr>
<tr>
<td>Wyoming</td>
<td>WYOM</td>
<td>36121</td>
</tr>
<tr>
<td>Yates</td>
<td>YATE</td>
<td>36123</td>
</tr>
</tbody>
</table>

There are no alternate abbreviations or regions associated 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 New York BG 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 borough name.


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_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.
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

```r
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 supplemental 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", "dottsignf", "bar", "arrow", "ts", "tsconf", "scatdot", "segbar", "normbar", "ctrlbar", "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 significant.

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:

```r
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 glyphic 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.

<table>
<thead>
<tr>
<th>Glyph Name</th>
<th>Meaning</th>
<th>col1</th>
<th>col2</th>
<th>col3</th>
<th>panelData</th>
</tr>
</thead>
<tbody>
<tr>
<td>arrow</td>
<td>Arrow</td>
<td>Beginning</td>
<td>Ending Values</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Values (arrow head)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bar</td>
<td>Horizontal bar</td>
<td>Bar end</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>values</td>
<td>(length)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>segbar</td>
<td>Horizontal stacked bar</td>
<td>Values for</td>
<td>Values for</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>first (left</td>
<td>the last (right</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-most) segment</td>
<td>-most) bar</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(length)</td>
<td>segment</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(length)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>normbar</td>
<td>Horizontal stacked bar, normalized to total 100%</td>
<td>Values for first (left-most) bar segment</td>
<td>Values for last (right-most) bar segment</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(length)</td>
<td>(length)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ctrbar</td>
<td>Horizontal stacked bar, centered on the middle bar</td>
<td>Values for first (left-most) bar segment</td>
<td>Values for last (right-most) bar segment</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(length)</td>
<td>(length)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>boxplot</td>
<td>Horizontal box plot</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>Name of output list from call to</td>
</tr>
<tr>
<td>Panel Description</td>
<td>Function</td>
<td>Values for dots</td>
<td>Values of lower limits</td>
<td>Values for upper limits tab</td>
<td>Notes</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------</td>
<td>-----------------</td>
<td>------------------------</td>
<td>-----------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Dot</td>
<td>boxplot(...plot=F)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>Dot with confidence interval line</td>
<td>dotconf</td>
<td>Values for dots</td>
<td>Values of lower limits</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>Dot with line length +/- standard error</td>
<td>dotse</td>
<td>Values for dots</td>
<td>Standard errors</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>Dot overprinted if not significant</td>
<td>dotsignf</td>
<td>Values for dots</td>
<td>P value associated with dot</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>Scater plot of dots</td>
<td>scatdot</td>
<td>Values on horizontal (x) axis</td>
<td>Values on vertical (y) axis</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>Time Series (line) plot</td>
<td>ts</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>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]</td>
</tr>
<tr>
<td>Time Series (line) plot with confidence limits</td>
<td>tsconf</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>Name of array with dimensions of c(51,t,4), as ts lower limit is [,3] and the upper limit is [,4]</td>
</tr>
</tbody>
</table>
The panelData vector 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 vector 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 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 arrow, 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.

**Usage**

panelDesc

**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":
  - **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 precede 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 identify 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 entries should be NA for the respective columns.
**dot** uses `col1` to specify a single data column in `statsFrame` data.frame to be plotted.

**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 precede `col2` in the `statsFrame` data.frame. The minimum number of data columns is 2 columns with a maximum of 9 columns.

**scadot** uses `col1` and `col2` to specify the x and y values respectively 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 `scadot` 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 abbreviations 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 a 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 areaNames/AbbrsIDs 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 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 areaNames/AbbrsIDs 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 areaNames/AbbrsIDs 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**

`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

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 200 and 300.

The third dimension \([.,v]\) of 2 or 4 elements is the x or y values during the time period. If the 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" glyphics to be used for a ts glyphics.

This data is used by micromapSEER with the "USStatesDF" border group.

```r
# 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)
TSD <- NULL
TSD <- array(workmat, dim=c(51,15,4))
# change state ab from factors to characters.
rownames(TSD) <- as.character(tempTS$stab)
```
Source


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:
<table>
<thead>
<tr>
<th>Name</th>
<th>ab</th>
<th>id</th>
<th>alt_ab</th>
<th>region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aberdeen</td>
<td>GB.ABE</td>
<td>826139</td>
<td>ABE</td>
<td>SCT</td>
</tr>
<tr>
<td>Aberdeenshire</td>
<td>GB.ABD</td>
<td>826140</td>
<td>ABD</td>
<td>SCT</td>
</tr>
<tr>
<td>c</td>
<td>GB.AGY</td>
<td>826171</td>
<td>AGY</td>
<td>WLS</td>
</tr>
<tr>
<td>Angus</td>
<td>GB.ANS</td>
<td>826141</td>
<td>ANS</td>
<td>SCT</td>
</tr>
<tr>
<td>Antrim</td>
<td>GB.ANT</td>
<td>826113</td>
<td>ANT</td>
<td>NIR</td>
</tr>
<tr>
<td>Ards</td>
<td>GB.ARD</td>
<td>826114</td>
<td>ARD</td>
<td>NIR</td>
</tr>
<tr>
<td>Argyll and Bute</td>
<td>GB.AGB</td>
<td>826142</td>
<td>AGB</td>
<td>SCT</td>
</tr>
<tr>
<td>Armagh</td>
<td>GB.ARM</td>
<td>826115</td>
<td>ARM</td>
<td>NIR</td>
</tr>
<tr>
<td>Ballymena</td>
<td>GB.BLA</td>
<td>826116</td>
<td>BLA</td>
<td>NIR</td>
</tr>
<tr>
<td>Ballymoney</td>
<td>GB.BLY</td>
<td>826117</td>
<td>BLY</td>
<td>NIR</td>
</tr>
<tr>
<td>Banbridge</td>
<td>GB.BNB</td>
<td>826118</td>
<td>BNB</td>
<td>NIR</td>
</tr>
<tr>
<td>Barking and Dagenham</td>
<td>GB.BDG</td>
<td>826001</td>
<td>BDG</td>
<td>ENG</td>
</tr>
<tr>
<td>Bath and North East Somerset</td>
<td>GB.BAS</td>
<td>826002</td>
<td>BAS</td>
<td>ENG</td>
</tr>
<tr>
<td>Bedfordshire</td>
<td>GB.BDF</td>
<td>826003</td>
<td>BDF</td>
<td>ENG</td>
</tr>
<tr>
<td>Belfast</td>
<td>GB.BFS</td>
<td>826119</td>
<td>BFS</td>
<td>NIR</td>
</tr>
<tr>
<td>Berkshire</td>
<td>GB.BRK</td>
<td>826004</td>
<td>BRK</td>
<td>ENG</td>
</tr>
<tr>
<td>Bexley</td>
<td>GB.BEX</td>
<td>826005</td>
<td>BEX</td>
<td>ENG</td>
</tr>
<tr>
<td>Blackburn with Darwen</td>
<td>GB.BBD</td>
<td>826006</td>
<td>BBD</td>
<td>ENG</td>
</tr>
<tr>
<td>Blaenau Gwent</td>
<td>GB.BGW</td>
<td>826172</td>
<td>BGW</td>
<td>WLS</td>
</tr>
<tr>
<td>Bournemouth</td>
<td>GB.BMH</td>
<td>826007</td>
<td>BMH</td>
<td>ENG</td>
</tr>
<tr>
<td>Brent</td>
<td>GB.BEN</td>
<td>826008</td>
<td>BEN</td>
<td>ENG</td>
</tr>
<tr>
<td>Bridgend</td>
<td>GB.BGE</td>
<td>826173</td>
<td>BGE</td>
<td>WLS</td>
</tr>
<tr>
<td>Brighton and Hove</td>
<td>GB.BNH</td>
<td>826009</td>
<td>BNH</td>
<td>ENG</td>
</tr>
<tr>
<td>Bristol</td>
<td>GB.BST</td>
<td>826010</td>
<td>BST</td>
<td>ENG</td>
</tr>
<tr>
<td>Bromley</td>
<td>GB.BRY</td>
<td>826011</td>
<td>BRY</td>
<td>ENG</td>
</tr>
<tr>
<td>Buckinghamshire</td>
<td>GB.BKM</td>
<td>826012</td>
<td>BKM</td>
<td>ENG</td>
</tr>
<tr>
<td>Caerphilly</td>
<td>GB.CAY</td>
<td>826174</td>
<td>CAY</td>
<td>WLS</td>
</tr>
<tr>
<td>Cambridgeshire</td>
<td>GB.CAM</td>
<td>826013</td>
<td>CAM</td>
<td>ENG</td>
</tr>
<tr>
<td>Camden</td>
<td>GB.CMD</td>
<td>826014</td>
<td>CMD</td>
<td>ENG</td>
</tr>
<tr>
<td>Cardiff</td>
<td>GB.CRF</td>
<td>826175</td>
<td>CRF</td>
<td>WLS</td>
</tr>
<tr>
<td>Carlow</td>
<td>IE.CW</td>
<td>372001</td>
<td>CW</td>
<td>IRE</td>
</tr>
<tr>
<td>Carmarthenshire</td>
<td>GB.CMN</td>
<td>826176</td>
<td>CMN</td>
<td>WLS</td>
</tr>
<tr>
<td>Carrickfergus</td>
<td>GB.CKF</td>
<td>826120</td>
<td>CKF</td>
<td>NIR</td>
</tr>
<tr>
<td>Castlereagh</td>
<td>GB.CSR</td>
<td>826121</td>
<td>CSR</td>
<td>NIR</td>
</tr>
<tr>
<td>Cavan</td>
<td>IE.CN</td>
<td>372002</td>
<td>CN</td>
<td>IRE</td>
</tr>
<tr>
<td>Ceredigion</td>
<td>GB.CGN</td>
<td>826177</td>
<td>CGN</td>
<td>WLS</td>
</tr>
<tr>
<td>Cheshire</td>
<td>GB.CHS</td>
<td>826015</td>
<td>CHS</td>
<td>ENG</td>
</tr>
<tr>
<td>Clackmannanshire</td>
<td>GB.CLK</td>
<td>826143</td>
<td>CLK</td>
<td>SCT</td>
</tr>
<tr>
<td>Clare</td>
<td>IE.CE</td>
<td>372003</td>
<td>CE</td>
<td>IRE</td>
</tr>
<tr>
<td>Coleraine</td>
<td>GB.CLR</td>
<td>826122</td>
<td>CLR</td>
<td>NIR</td>
</tr>
<tr>
<td>Conwy</td>
<td>GB.CWY</td>
<td>826178</td>
<td>CWY</td>
<td>WLS</td>
</tr>
<tr>
<td>Cookstown</td>
<td>GB.CKT</td>
<td>826123</td>
<td>CKT</td>
<td>NIR</td>
</tr>
<tr>
<td>Cork</td>
<td>IE.CO</td>
<td>372004</td>
<td>CO</td>
<td>IRE</td>
</tr>
<tr>
<td>Cornwall</td>
<td>GB.CON</td>
<td>826016</td>
<td>CON</td>
<td>ENG</td>
</tr>
<tr>
<td>Craigavon</td>
<td>GB.CGV</td>
<td>826124</td>
<td>CGV</td>
<td>NIR</td>
</tr>
<tr>
<td>Croydon</td>
<td>GB.CRY</td>
<td>826017</td>
<td>CRY</td>
<td>ENG</td>
</tr>
<tr>
<td>Cumbria</td>
<td>GB.CMA</td>
<td>826018</td>
<td>CMA</td>
<td>ENG</td>
</tr>
<tr>
<td>Location</td>
<td>Code</td>
<td>Area Code</td>
<td>Region</td>
<td></td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------</td>
<td>-----------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>Darlington</td>
<td>GB</td>
<td>DAL</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Denbighshire</td>
<td>GB</td>
<td>DEN</td>
<td>WLS</td>
<td></td>
</tr>
<tr>
<td>Derby</td>
<td>GB</td>
<td>DER</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Derbyshire</td>
<td>GB</td>
<td>DBY</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Derry</td>
<td>GB</td>
<td>DRY</td>
<td>NIR</td>
<td></td>
</tr>
<tr>
<td>Devon</td>
<td>GB</td>
<td>DEV</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Donegal</td>
<td>IE</td>
<td>DIL</td>
<td>IRE</td>
<td></td>
</tr>
<tr>
<td>Dorset</td>
<td>GB</td>
<td>DOR</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Down</td>
<td>GB</td>
<td>DOW</td>
<td>NIR</td>
<td></td>
</tr>
<tr>
<td>Dublin</td>
<td>IE</td>
<td>D</td>
<td>IRE</td>
<td></td>
</tr>
<tr>
<td>Dumfries and Galloway</td>
<td>GB</td>
<td>DGY</td>
<td>SCT</td>
<td></td>
</tr>
<tr>
<td>Dundee</td>
<td>GB</td>
<td>DND</td>
<td>SCT</td>
<td></td>
</tr>
<tr>
<td>Dungannon</td>
<td>GB</td>
<td>DGN</td>
<td>NIR</td>
<td></td>
</tr>
<tr>
<td>Durham</td>
<td>GB</td>
<td>DUR</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Ealing</td>
<td>GB</td>
<td>EAL</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>East Ayrshire</td>
<td>GB</td>
<td>EAY</td>
<td>SCT</td>
<td></td>
</tr>
<tr>
<td>East Dunbartonshire</td>
<td>GB</td>
<td>EDU</td>
<td>SCT</td>
<td></td>
</tr>
<tr>
<td>East Lothian</td>
<td>GB</td>
<td>ELN</td>
<td>SCT</td>
<td></td>
</tr>
<tr>
<td>East Renfrewshire</td>
<td>GB</td>
<td>ERW</td>
<td>SCT</td>
<td></td>
</tr>
<tr>
<td>East Riding of Yorkshire</td>
<td>GB</td>
<td>ERY</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>East Sussex</td>
<td>GB</td>
<td>ESX</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Edinburgh</td>
<td>GB</td>
<td>EDH</td>
<td>SCT</td>
<td></td>
</tr>
<tr>
<td>Eilean Siar</td>
<td>GB</td>
<td>ELS</td>
<td>SCT</td>
<td></td>
</tr>
<tr>
<td>Enfield</td>
<td>GB</td>
<td>ENF</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Essex</td>
<td>GB</td>
<td>ESS</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Falkirk</td>
<td>GB</td>
<td>FAL</td>
<td>SCT</td>
<td></td>
</tr>
<tr>
<td>Fermanagh</td>
<td>GB</td>
<td>FER</td>
<td>NIR</td>
<td></td>
</tr>
<tr>
<td>Fife</td>
<td>GB</td>
<td>FIF</td>
<td>SCT</td>
<td></td>
</tr>
<tr>
<td>Flintshire</td>
<td>GB</td>
<td>FLN</td>
<td>WLS</td>
<td></td>
</tr>
<tr>
<td>Galway</td>
<td>IE</td>
<td>G</td>
<td>IRE</td>
<td></td>
</tr>
<tr>
<td>Glasgow</td>
<td>GB</td>
<td>GLG</td>
<td>SCT</td>
<td></td>
</tr>
<tr>
<td>Gloucestershire</td>
<td>GB</td>
<td>GLS</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Greenwich</td>
<td>GB</td>
<td>GRE</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Gwynedd</td>
<td>GB</td>
<td>GWN</td>
<td>WLS</td>
<td></td>
</tr>
<tr>
<td>Hackney</td>
<td>GB</td>
<td>HCK</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Halton</td>
<td>GB</td>
<td>HAL</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Hammersmith and Fulham</td>
<td>GB</td>
<td>HMF</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Hampshire</td>
<td>GB</td>
<td>HAM</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Harlingey</td>
<td>GB</td>
<td>HRY</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Harrow</td>
<td>GB</td>
<td>HRW</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Hartlepool</td>
<td>GB</td>
<td>HPL</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Havering</td>
<td>GB</td>
<td>HAV</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Herefordshire</td>
<td>GB</td>
<td>HEF</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Hertfordshire</td>
<td>GB</td>
<td>HRT</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Highland</td>
<td>GB</td>
<td>HLD</td>
<td>SCT</td>
<td></td>
</tr>
<tr>
<td>Hillingdon</td>
<td>GB</td>
<td>HIL</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Hounslow</td>
<td>GB</td>
<td>HNS</td>
<td>ENG</td>
<td></td>
</tr>
<tr>
<td>Inverclyde</td>
<td>GB</td>
<td>IVC</td>
<td>SCT</td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>Code</td>
<td>Postcode</td>
<td>Country</td>
<td>Region</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------</td>
<td>----------</td>
<td>---------</td>
<td>--------</td>
</tr>
<tr>
<td>Isle of Wight</td>
<td>GB.IOW</td>
<td>826044</td>
<td>IOW</td>
<td>ENG</td>
</tr>
<tr>
<td>Islington</td>
<td>GB.ISL</td>
<td>826045</td>
<td>ISL</td>
<td>ENG</td>
</tr>
<tr>
<td>Kensington and Chelsea</td>
<td>GB.KEC</td>
<td>826046</td>
<td>KEC</td>
<td>ENG</td>
</tr>
<tr>
<td>Kent</td>
<td>GB.KEN</td>
<td>826047</td>
<td>KEN</td>
<td>ENG</td>
</tr>
<tr>
<td>Kerry</td>
<td>IE.KY</td>
<td>372008</td>
<td>KY</td>
<td>IRE</td>
</tr>
<tr>
<td>Kildare</td>
<td>IE.KE</td>
<td>372009</td>
<td>KE</td>
<td>IRE</td>
</tr>
<tr>
<td>Kilkenny</td>
<td>IE.KK</td>
<td>372010</td>
<td>KK</td>
<td>IRE</td>
</tr>
<tr>
<td>Kingston upon Hull</td>
<td>GB.KHL</td>
<td>826048</td>
<td>KHL</td>
<td>ENG</td>
</tr>
<tr>
<td>Kingston upon Thames</td>
<td>GB.KTT</td>
<td>826049</td>
<td>KTT</td>
<td>ENG</td>
</tr>
<tr>
<td>Lambeth</td>
<td>GB.LBH</td>
<td>826050</td>
<td>LBH</td>
<td>ENG</td>
</tr>
<tr>
<td>Lancashire</td>
<td>GB.LAN</td>
<td>826051</td>
<td>LAN</td>
<td>ENG</td>
</tr>
<tr>
<td>Laoghis</td>
<td>IE.LS</td>
<td>372011</td>
<td>LS</td>
<td>IRE</td>
</tr>
<tr>
<td>Larne</td>
<td>GB.LRN</td>
<td>826129</td>
<td>LRN</td>
<td>NIR</td>
</tr>
<tr>
<td>Leicester</td>
<td>GB.LCE</td>
<td>826052</td>
<td>LCE</td>
<td>ENG</td>
</tr>
<tr>
<td>Leicestershire</td>
<td>GB.LEC</td>
<td>826053</td>
<td>LEC</td>
<td>ENG</td>
</tr>
<tr>
<td>Leitrim</td>
<td>IE.LM</td>
<td>372012</td>
<td>LM</td>
<td>IRE</td>
</tr>
<tr>
<td>Lewisham</td>
<td>GB.LEW</td>
<td>826054</td>
<td>LEW</td>
<td>ENG</td>
</tr>
<tr>
<td>Limavady</td>
<td>GB.LMV</td>
<td>826130</td>
<td>LMV</td>
<td>NIR</td>
</tr>
<tr>
<td>Limerick</td>
<td>IE.LK</td>
<td>372013</td>
<td>LK</td>
<td>IRE</td>
</tr>
<tr>
<td>Lincolnshire</td>
<td>GB.LIN</td>
<td>826055</td>
<td>LIN</td>
<td>ENG</td>
</tr>
<tr>
<td>Lisburn</td>
<td>GB.LSB</td>
<td>826131</td>
<td>LSB</td>
<td>NIR</td>
</tr>
<tr>
<td>London</td>
<td>GB.LND</td>
<td>826056</td>
<td>LND</td>
<td>ENG</td>
</tr>
<tr>
<td>Longford</td>
<td>IE.LD</td>
<td>372014</td>
<td>LD</td>
<td>IRE</td>
</tr>
<tr>
<td>Louth</td>
<td>IE.LH</td>
<td>372015</td>
<td>LH</td>
<td>IRE</td>
</tr>
<tr>
<td>Luton</td>
<td>GB.LUT</td>
<td>826057</td>
<td>LUT</td>
<td>ENG</td>
</tr>
<tr>
<td>Magherafelt</td>
<td>GB.MFT</td>
<td>826132</td>
<td>MFT</td>
<td>NIR</td>
</tr>
<tr>
<td>Manchester</td>
<td>GB.MAN</td>
<td>826058</td>
<td>MAN</td>
<td>ENG</td>
</tr>
<tr>
<td>Mayo</td>
<td>IE.MO</td>
<td>372016</td>
<td>MO</td>
<td>IRE</td>
</tr>
<tr>
<td>Meath</td>
<td>IE.MH</td>
<td>372017</td>
<td>MH</td>
<td>IRE</td>
</tr>
<tr>
<td>Medway</td>
<td>GB.MDW</td>
<td>826059</td>
<td>MDW</td>
<td>ENG</td>
</tr>
<tr>
<td>Merseyside</td>
<td>GB.MSY</td>
<td>826060</td>
<td>MSY</td>
<td>ENG</td>
</tr>
<tr>
<td>Merthyr Tydfil</td>
<td>GB.MTY</td>
<td>826182</td>
<td>MTY</td>
<td>WLS</td>
</tr>
<tr>
<td>Merton</td>
<td>GB.MRT</td>
<td>826061</td>
<td>MRT</td>
<td>ENG</td>
</tr>
<tr>
<td>Middlesbrough</td>
<td>GB.MDB</td>
<td>826062</td>
<td>MDB</td>
<td>ENG</td>
</tr>
<tr>
<td>Midlothian</td>
<td>GB.MLN</td>
<td>826157</td>
<td>MLN</td>
<td>SCT</td>
</tr>
<tr>
<td>Milton Keynes</td>
<td>GB.MIK</td>
<td>826063</td>
<td>MIK</td>
<td>ENG</td>
</tr>
<tr>
<td>Monaghan</td>
<td>IE.MN</td>
<td>372018</td>
<td>MN</td>
<td>IRE</td>
</tr>
<tr>
<td>Monmouthshire</td>
<td>GB.MON</td>
<td>826183</td>
<td>MON</td>
<td>WLS</td>
</tr>
<tr>
<td>Moray</td>
<td>GB.MRY</td>
<td>826158</td>
<td>MRY</td>
<td>SCT</td>
</tr>
<tr>
<td>Moyle</td>
<td>GB.MYL</td>
<td>826133</td>
<td>MYL</td>
<td>NIR</td>
</tr>
<tr>
<td>Neath Port Talbot</td>
<td>GB.NTL</td>
<td>826184</td>
<td>NTL</td>
<td>WLS</td>
</tr>
<tr>
<td>Newham</td>
<td>GB.NWM</td>
<td>826064</td>
<td>NWM</td>
<td>ENG</td>
</tr>
<tr>
<td>Newport</td>
<td>GB.NWP</td>
<td>826185</td>
<td>NWP</td>
<td>WLS</td>
</tr>
<tr>
<td>Newry and Mourne</td>
<td>GB.NYM</td>
<td>826134</td>
<td>NYM</td>
<td>NIR</td>
</tr>
<tr>
<td>Newtownabbey</td>
<td>GB.NTA</td>
<td>826135</td>
<td>NTA</td>
<td>NIR</td>
</tr>
<tr>
<td>Norfolk</td>
<td>GB.NFK</td>
<td>826065</td>
<td>NFK</td>
<td>ENG</td>
</tr>
<tr>
<td>North Ayrshire</td>
<td>GB.NAY</td>
<td>826159</td>
<td>NAY</td>
<td>SCT</td>
</tr>
<tr>
<td>North Down</td>
<td>GB.NDN</td>
<td>826136</td>
<td>NDN</td>
<td>NIR</td>
</tr>
<tr>
<td>Region</td>
<td>Code</td>
<td>Postal Code</td>
<td>Region Code</td>
<td>Country</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>------</td>
<td>-------------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>North East Lincolnshire</td>
<td>GB.NEL</td>
<td>826066</td>
<td>NEL</td>
<td>ENG</td>
</tr>
<tr>
<td>North Lanarkshire</td>
<td>GB.NLK</td>
<td>826160</td>
<td>NLK</td>
<td>SCT</td>
</tr>
<tr>
<td>North Lincolnshire</td>
<td>GB.NLN</td>
<td>826067</td>
<td>NLN</td>
<td>ENG</td>
</tr>
<tr>
<td>North Somerset</td>
<td>GB.NSM</td>
<td>826068</td>
<td>NSM</td>
<td>ENG</td>
</tr>
<tr>
<td>North Yorkshire</td>
<td>GB.NYK</td>
<td>826069</td>
<td>NYK</td>
<td>ENG</td>
</tr>
<tr>
<td>Northamptonshire</td>
<td>GB.NTH</td>
<td>826070</td>
<td>NTH</td>
<td>ENG</td>
</tr>
<tr>
<td>Northumberland</td>
<td>GB.NBL</td>
<td>826071</td>
<td>NBL</td>
<td>ENG</td>
</tr>
<tr>
<td>Nottingham</td>
<td>GB.NGM</td>
<td>826072</td>
<td>NGM</td>
<td>ENG</td>
</tr>
<tr>
<td>Nottinghamshire</td>
<td>GB.NTT</td>
<td>826073</td>
<td>NTT</td>
<td>ENG</td>
</tr>
<tr>
<td>Offaly</td>
<td>IE.OY</td>
<td>372019</td>
<td>OY</td>
<td>IRE</td>
</tr>
<tr>
<td>Omagh</td>
<td>GB.OMH</td>
<td>826137</td>
<td>OMH</td>
<td>NIR</td>
</tr>
<tr>
<td>Orkney Islands</td>
<td>GB.ORK</td>
<td>826161</td>
<td>ORK</td>
<td>SCT</td>
</tr>
<tr>
<td>Oxfordshire</td>
<td>GB.OXF</td>
<td>826074</td>
<td>OXF</td>
<td>ENG</td>
</tr>
<tr>
<td>Pembrokeshire</td>
<td>GB.PEM</td>
<td>826186</td>
<td>PEM</td>
<td>WLS</td>
</tr>
<tr>
<td>Perthshire and Kinross</td>
<td>GB.PKN</td>
<td>826162</td>
<td>PKN</td>
<td>SCT</td>
</tr>
<tr>
<td>Peterborough</td>
<td>GB.PTE</td>
<td>826075</td>
<td>PTE</td>
<td>ENG</td>
</tr>
<tr>
<td>Plymouth</td>
<td>GB.PLY</td>
<td>826076</td>
<td>PLY</td>
<td>ENG</td>
</tr>
<tr>
<td>Poole</td>
<td>GB.POL</td>
<td>826077</td>
<td>POL</td>
<td>ENG</td>
</tr>
<tr>
<td>Portsmouth</td>
<td>GB.POR</td>
<td>826078</td>
<td>POR</td>
<td>ENG</td>
</tr>
<tr>
<td>Powys</td>
<td>GB.POW</td>
<td>826187</td>
<td>POW</td>
<td>WLS</td>
</tr>
<tr>
<td>Redbridge</td>
<td>GB.RDB</td>
<td>826079</td>
<td>RDB</td>
<td>ENG</td>
</tr>
<tr>
<td>Redcar and Cleveland</td>
<td>GB.RCC</td>
<td>826080</td>
<td>RCC</td>
<td>ENG</td>
</tr>
<tr>
<td>Renfrewshire</td>
<td>GB.RFW</td>
<td>826163</td>
<td>RFW</td>
<td>SCT</td>
</tr>
<tr>
<td>Rhondda, Cynon, Taff</td>
<td>GB.RCT</td>
<td>826188</td>
<td>RCT</td>
<td>WLS</td>
</tr>
<tr>
<td>Richmond upon Thames</td>
<td>GB.RIC</td>
<td>826081</td>
<td>RIC</td>
<td>ENG</td>
</tr>
<tr>
<td>Roscommon</td>
<td>IE.RN</td>
<td>372020</td>
<td>RN</td>
<td>IRE</td>
</tr>
<tr>
<td>Rutland</td>
<td>GB.RUT</td>
<td>826082</td>
<td>RUT</td>
<td>ENG</td>
</tr>
<tr>
<td>Scottish Borders</td>
<td>GB.SCB</td>
<td>826164</td>
<td>SCB</td>
<td>SCT</td>
</tr>
<tr>
<td>Shetland Islands</td>
<td>GB.ZET</td>
<td>826165</td>
<td>ZET</td>
<td>SCT</td>
</tr>
<tr>
<td>Shropshire</td>
<td>GB.SHR</td>
<td>826083</td>
<td>SHR</td>
<td>ENG</td>
</tr>
<tr>
<td>Sligo</td>
<td>IE.SO</td>
<td>372021</td>
<td>SO</td>
<td>IRE</td>
</tr>
<tr>
<td>Somerset</td>
<td>GB.SOM</td>
<td>826084</td>
<td>SOM</td>
<td>ENG</td>
</tr>
<tr>
<td>South Ayrshire</td>
<td>GB.SAY</td>
<td>826166</td>
<td>SAY</td>
<td>SCT</td>
</tr>
<tr>
<td>South Gloucestershire</td>
<td>GB.SGC</td>
<td>826085</td>
<td>SGC</td>
<td>ENG</td>
</tr>
<tr>
<td>South Lanarkshire</td>
<td>GB.SLK</td>
<td>826167</td>
<td>SLK</td>
<td>SCT</td>
</tr>
<tr>
<td>South Yorkshire</td>
<td>GB.SYK</td>
<td>826086</td>
<td>SYK</td>
<td>ENG</td>
</tr>
<tr>
<td>Southampton</td>
<td>GB.STH</td>
<td>826087</td>
<td>STH</td>
<td>ENG</td>
</tr>
<tr>
<td>Southend-on-Sea</td>
<td>GB.SOS</td>
<td>826088</td>
<td>SOS</td>
<td>ENG</td>
</tr>
<tr>
<td>Southwark</td>
<td>GB.SWK</td>
<td>826089</td>
<td>SWK</td>
<td>ENG</td>
</tr>
<tr>
<td>Staffordshire</td>
<td>GB.STS</td>
<td>826090</td>
<td>STS</td>
<td>ENG</td>
</tr>
<tr>
<td>Stirling</td>
<td>GB.STG</td>
<td>826168</td>
<td>STG</td>
<td>SCT</td>
</tr>
<tr>
<td>Stockton-on-Tees</td>
<td>GB.STT</td>
<td>826091</td>
<td>STT</td>
<td>ENG</td>
</tr>
<tr>
<td>Stoke-on-Trent</td>
<td>GB.STE</td>
<td>826092</td>
<td>STE</td>
<td>ENG</td>
</tr>
<tr>
<td>Strabane</td>
<td>GB.STB</td>
<td>826138</td>
<td>STB</td>
<td>NIR</td>
</tr>
<tr>
<td>Suffolk</td>
<td>GB.SFK</td>
<td>826093</td>
<td>SFK</td>
<td>ENG</td>
</tr>
<tr>
<td>Surrey</td>
<td>GB.SRY</td>
<td>826094</td>
<td>SRY</td>
<td>ENG</td>
</tr>
<tr>
<td>Sutton</td>
<td>GB.STN</td>
<td>826095</td>
<td>STN</td>
<td>ENG</td>
</tr>
<tr>
<td>Swansea</td>
<td>GB.SWA</td>
<td>826189</td>
<td>SWA</td>
<td>WLS</td>
</tr>
</tbody>
</table>
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/province 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

---

**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**

```r
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 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 areaNamesAbbrsIDs 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:

<table>
<thead>
<tr>
<th>state</th>
<th>Seer Registries</th>
<th>region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alabama</td>
<td>&lt;none&gt;</td>
<td>South</td>
</tr>
<tr>
<td>Alaska</td>
<td>Alaska Natives</td>
<td>West</td>
</tr>
<tr>
<td>Arizona</td>
<td>Arizona Natives</td>
<td>West</td>
</tr>
<tr>
<td>Arkansas</td>
<td>&lt;none&gt;</td>
<td>South</td>
</tr>
<tr>
<td>California</td>
<td>California-LA,</td>
<td>West</td>
</tr>
<tr>
<td></td>
<td>California-Other</td>
<td></td>
</tr>
<tr>
<td></td>
<td>California-SF</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>Region</td>
<td>Region</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Colorado</td>
<td>&lt;none&gt;</td>
<td>West</td>
</tr>
<tr>
<td>Connecticut</td>
<td>Connecticut</td>
<td>NorthEast</td>
</tr>
<tr>
<td>Delaware</td>
<td>&lt;none&gt;</td>
<td>South</td>
</tr>
<tr>
<td>District of Columbia</td>
<td>&lt;none&gt;</td>
<td>South</td>
</tr>
<tr>
<td>Florida</td>
<td>&lt;none&gt;</td>
<td>South</td>
</tr>
<tr>
<td>Georgia</td>
<td>Georgia-Atlanta,</td>
<td>South</td>
</tr>
<tr>
<td></td>
<td>Georgia-Other,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Georgia-Rural</td>
<td></td>
</tr>
<tr>
<td>Hawaii</td>
<td>Hawaii</td>
<td>West</td>
</tr>
<tr>
<td>Idaho</td>
<td>&lt;none&gt;</td>
<td>West</td>
</tr>
<tr>
<td>Illinois</td>
<td>&lt;none&gt;</td>
<td>MidWest</td>
</tr>
<tr>
<td>Indiana</td>
<td>&lt;none&gt;</td>
<td>MidWest</td>
</tr>
<tr>
<td>Iowa</td>
<td>Iowa</td>
<td>MidWest</td>
</tr>
<tr>
<td>Kansas</td>
<td>&lt;none&gt;</td>
<td>MidWest</td>
</tr>
<tr>
<td>Kentucky</td>
<td>Kentucky</td>
<td>South</td>
</tr>
<tr>
<td>Louisiana</td>
<td>Louisiana</td>
<td>South</td>
</tr>
<tr>
<td>Maine</td>
<td>&lt;none&gt;</td>
<td>NorthEast</td>
</tr>
<tr>
<td>Maryland</td>
<td>&lt;none&gt;</td>
<td>South</td>
</tr>
<tr>
<td>Massachusetts</td>
<td>&lt;none&gt;</td>
<td>NorthEast</td>
</tr>
<tr>
<td>Michigan</td>
<td>Michigan-Detroit</td>
<td>MidWest</td>
</tr>
<tr>
<td>Minnesota</td>
<td>&lt;none&gt;</td>
<td>MidWest</td>
</tr>
<tr>
<td>Mississippi</td>
<td>&lt;none&gt;</td>
<td>South</td>
</tr>
<tr>
<td>Missouri</td>
<td>&lt;none&gt;</td>
<td>MidWest</td>
</tr>
<tr>
<td>Montana</td>
<td>&lt;none&gt;</td>
<td>West</td>
</tr>
<tr>
<td>Nebraska</td>
<td>&lt;none&gt;</td>
<td>MidWest</td>
</tr>
<tr>
<td>Nevada</td>
<td>&lt;none&gt;</td>
<td>West</td>
</tr>
<tr>
<td>New Hampshire</td>
<td>&lt;none&gt;</td>
<td>NorthEast</td>
</tr>
<tr>
<td>New Jersey</td>
<td>New Jersey</td>
<td>NorthEast</td>
</tr>
<tr>
<td>New Mexico</td>
<td>New Mexico</td>
<td>West</td>
</tr>
<tr>
<td>New York</td>
<td>&lt;none&gt;</td>
<td>NorthEast</td>
</tr>
<tr>
<td>North Carolina</td>
<td>&lt;none&gt;</td>
<td>South</td>
</tr>
<tr>
<td>North Dakota</td>
<td>&lt;none&gt;</td>
<td>MidWest</td>
</tr>
<tr>
<td>Ohio</td>
<td>&lt;none&gt;</td>
<td>MidWest</td>
</tr>
<tr>
<td>Oklahoma</td>
<td>Oklahoma-Cherokee</td>
<td>South</td>
</tr>
<tr>
<td>Oregon</td>
<td>&lt;none&gt;</td>
<td>West</td>
</tr>
<tr>
<td>Pennsylvania</td>
<td>&lt;none&gt;</td>
<td>NorthEast</td>
</tr>
<tr>
<td>Rhode Island</td>
<td>&lt;none&gt;</td>
<td>NorthEast</td>
</tr>
<tr>
<td>South Carolina</td>
<td>&lt;none&gt;</td>
<td>South</td>
</tr>
<tr>
<td>South Dakota</td>
<td>&lt;none&gt;</td>
<td>MidWest</td>
</tr>
<tr>
<td>Tennessee</td>
<td>&lt;none&gt;</td>
<td>South</td>
</tr>
<tr>
<td>Texas</td>
<td>&lt;none&gt;</td>
<td>South</td>
</tr>
<tr>
<td>Utah</td>
<td>Utah</td>
<td>West</td>
</tr>
<tr>
<td>Vermont</td>
<td>&lt;none&gt;</td>
<td>NorthEast</td>
</tr>
<tr>
<td>Virginia</td>
<td>&lt;none&gt;</td>
<td>South</td>
</tr>
<tr>
<td>Washington</td>
<td>Washington-Seattle</td>
<td>South</td>
</tr>
<tr>
<td>West Virginia</td>
<td>&lt;none&gt;</td>
<td>South</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>&lt;none&gt;</td>
<td>MidWest</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Name</th>
<th>ab</th>
<th>alias string</th>
<th>id</th>
<th>counties</th>
<th>region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alaska Natives</td>
<td>AK-NAT</td>
<td>ALASKA NATIVES</td>
<td>18</td>
<td>all</td>
<td>West</td>
</tr>
<tr>
<td>Arizona Natives</td>
<td>AZ-NAT</td>
<td>ARIZONA NATIVES</td>
<td>20</td>
<td>all</td>
<td>West</td>
</tr>
<tr>
<td>California-LA</td>
<td>CA-LA</td>
<td>LOS ANGELES</td>
<td>4</td>
<td>Los Angeles</td>
<td>West</td>
</tr>
<tr>
<td>California-SF</td>
<td>CA-SF</td>
<td>SAN FRANCISCO</td>
<td>2</td>
<td>Alameda, Contra Costa, Marin, San Francisco, San Mateo</td>
<td></td>
</tr>
<tr>
<td>California-SJ</td>
<td>CA-SJ</td>
<td>SAN JOSE</td>
<td>3</td>
<td>Monterey, San Benito, Santa Clara, Santa Cruz</td>
<td>West</td>
</tr>
<tr>
<td>California-Other</td>
<td>CA-OTH</td>
<td>CALIFORNIA EXCLUDING</td>
<td>5</td>
<td>all other counties</td>
<td>West</td>
</tr>
<tr>
<td>Connecticut</td>
<td>CT</td>
<td>CONNECTICUT</td>
<td>1</td>
<td>all</td>
<td>NorthEast</td>
</tr>
<tr>
<td>Georgia-Atlanta</td>
<td>GA-ATL</td>
<td>ATLANTA</td>
<td>6</td>
<td>Clayton, Cobb, DeKalb, Fulton, Gwinnett</td>
<td>South</td>
</tr>
<tr>
<td>Georgia-Rural</td>
<td>GA-RUR</td>
<td>RURAL GEORGIA</td>
<td>8</td>
<td>Glascock, Greene, Hancock, Jasper, Jefferson, Morgan, Putnam, Taliaferro, Warren, Washington</td>
<td>South</td>
</tr>
<tr>
<td>Georgia-Other</td>
<td>GA-OTH</td>
<td>GREATER GEORGIA</td>
<td>7</td>
<td>all other counties</td>
<td>South</td>
</tr>
<tr>
<td>Hawaii</td>
<td>HI</td>
<td>HAWAII</td>
<td>9</td>
<td>all</td>
<td>West</td>
</tr>
<tr>
<td>Iowa</td>
<td>IA</td>
<td>IOWA</td>
<td>10</td>
<td>all</td>
<td>MidWest</td>
</tr>
<tr>
<td>Kentucky</td>
<td>KY</td>
<td>KENTUCKY</td>
<td>14</td>
<td>all</td>
<td>South</td>
</tr>
<tr>
<td>Michigan-Detroit</td>
<td>MI-DET</td>
<td>DETROIT</td>
<td>15</td>
<td>Macomb, Oakland, Wayne</td>
<td>MidWest</td>
</tr>
<tr>
<td>New Jersey</td>
<td>NJ</td>
<td>NEW JERSEY</td>
<td>11</td>
<td>all</td>
<td>NorthEast</td>
</tr>
<tr>
<td>New Mexico</td>
<td>NM</td>
<td>NEW MEXICO</td>
<td>12</td>
<td>all</td>
<td>West</td>
</tr>
<tr>
<td>Oklahoma-Cherokee</td>
<td>OK-CHE</td>
<td>OKLAHOMA</td>
<td>19</td>
<td>Adair, Cherokee,</td>
<td>South</td>
</tr>
</tbody>
</table>
Craig, Delaware, Mayes, McIntosh, Muskogee, Nowata, Ottawa, Rogers, Sequoyah, Tulsa, Wagoner, Washington

<table>
<thead>
<tr>
<th>Utah</th>
<th>UT</th>
<th>UTAH</th>
<th>16</th>
<th>all</th>
<th>West</th>
</tr>
</thead>
<tbody>
<tr>
<td>Washington-Seattle</td>
<td>WA-SEA</td>
<td>SEATTLE</td>
<td>17</td>
<td>Clallam, Grays Harbor, Island, Jefferson, King, Kitsap, Mason, Pierce, San Juan, Skagit, Snohomish, Thurston, Whatcom</td>
<td>South</td>
</tr>
</tbody>
</table>

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 `readNdelim` function with `header = TRUE`. Follow the `readNdelim` call with a `str` function to verify the data was read correctly.

```r
dataT <- readNdelim("c:\datadir\seerstat.txt",header=FALSE)
str(dataT)
```

**Source**

NCI

**References**


**USStatesBG**

**USStatesBG border group datasets to support use with U.S. States and 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 D.C. 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 areaNamesAbbrsIDs 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:

<table>
<thead>
<tr>
<th>name</th>
<th>ab</th>
<th>id</th>
<th>region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alabama</td>
<td>AL</td>
<td>01</td>
<td>South</td>
</tr>
<tr>
<td>Alaska</td>
<td>AK</td>
<td>02</td>
<td>West</td>
</tr>
<tr>
<td>Arizona</td>
<td>AZ</td>
<td>04</td>
<td>West</td>
</tr>
<tr>
<td>Arkansas</td>
<td>AR</td>
<td>05</td>
<td>South</td>
</tr>
<tr>
<td>California</td>
<td>CA</td>
<td>06</td>
<td>West</td>
</tr>
<tr>
<td>Colorado</td>
<td>CO</td>
<td>08</td>
<td>West</td>
</tr>
<tr>
<td>Connecticut</td>
<td>CT</td>
<td>09</td>
<td>NorthEast</td>
</tr>
<tr>
<td>Delaware</td>
<td>DE</td>
<td>10</td>
<td>South</td>
</tr>
<tr>
<td>DC</td>
<td>DC</td>
<td>11</td>
<td>South</td>
</tr>
<tr>
<td>Florida</td>
<td>FL</td>
<td>12</td>
<td>South</td>
</tr>
<tr>
<td>Georgia</td>
<td>GA</td>
<td>13</td>
<td>South</td>
</tr>
<tr>
<td>Hawaii</td>
<td>HI</td>
<td>15</td>
<td>West</td>
</tr>
<tr>
<td>Idaho</td>
<td>ID</td>
<td>16</td>
<td>West</td>
</tr>
<tr>
<td>Illinois</td>
<td>IL</td>
<td>17</td>
<td>MidWest</td>
</tr>
<tr>
<td>Indiana</td>
<td>IN</td>
<td>18</td>
<td>MidWest</td>
</tr>
<tr>
<td>Iowa</td>
<td>IA</td>
<td>19</td>
<td>MidWest</td>
</tr>
<tr>
<td>Kansas</td>
<td>KS</td>
<td>20</td>
<td>MidWest</td>
</tr>
<tr>
<td>Kentucky</td>
<td>KY</td>
<td>21</td>
<td>South</td>
</tr>
<tr>
<td>Louisiana</td>
<td>LA</td>
<td>22</td>
<td>South</td>
</tr>
<tr>
<td>Maine</td>
<td>ME</td>
<td>23</td>
<td>NorthEast</td>
</tr>
<tr>
<td>Maryland</td>
<td>MD</td>
<td>24</td>
<td>South</td>
</tr>
<tr>
<td>Massachusetts</td>
<td>MA</td>
<td>25</td>
<td>NorthEast</td>
</tr>
</tbody>
</table>
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

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 (counties) 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 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 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.
<table>
<thead>
<tr>
<th>name</th>
<th>ab</th>
<th>alt_ab</th>
<th>id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beaver</td>
<td>BV</td>
<td>BEA</td>
<td>49001</td>
</tr>
<tr>
<td>Box Elder</td>
<td>BE</td>
<td>BOX</td>
<td>49003</td>
</tr>
<tr>
<td>Cache</td>
<td>CH</td>
<td>CAC</td>
<td>49005</td>
</tr>
<tr>
<td>Carbon</td>
<td>CA</td>
<td>CAR</td>
<td>49007</td>
</tr>
<tr>
<td>Daggett</td>
<td>DAG</td>
<td>DAG</td>
<td>49009</td>
</tr>
<tr>
<td>Davis</td>
<td>DAV</td>
<td>DAV</td>
<td>49011</td>
</tr>
<tr>
<td>Duchesne</td>
<td>DU</td>
<td>DUC</td>
<td>49013</td>
</tr>
<tr>
<td>Emery</td>
<td>EM</td>
<td>EME</td>
<td>49015</td>
</tr>
<tr>
<td>Garfield</td>
<td>GA</td>
<td>GAR</td>
<td>49017</td>
</tr>
<tr>
<td>Grand</td>
<td>GR</td>
<td>GRA</td>
<td>49019</td>
</tr>
<tr>
<td>Iron</td>
<td>IR</td>
<td>IRO</td>
<td>49021</td>
</tr>
<tr>
<td>Juab</td>
<td>JB</td>
<td>JUA</td>
<td>49023</td>
</tr>
<tr>
<td>Kane</td>
<td>KN</td>
<td>KAN</td>
<td>49025</td>
</tr>
<tr>
<td>Millard</td>
<td>MI</td>
<td>MIL</td>
<td>49027</td>
</tr>
<tr>
<td>Morgan</td>
<td>MO</td>
<td>MOR</td>
<td>49029</td>
</tr>
<tr>
<td>Piute</td>
<td>PT</td>
<td>PIU</td>
<td>49031</td>
</tr>
<tr>
<td>Rich</td>
<td>RH</td>
<td>RIC</td>
<td>49033</td>
</tr>
<tr>
<td>Salt Lake</td>
<td>SL</td>
<td>SAL</td>
<td>49035</td>
</tr>
<tr>
<td>San Juan</td>
<td>SJ</td>
<td>SNJ</td>
<td>49037</td>
</tr>
<tr>
<td>Sanpete</td>
<td>SP</td>
<td>SNP</td>
<td>49039</td>
</tr>
<tr>
<td>Sevier</td>
<td>SV</td>
<td>SEV</td>
<td>49041</td>
</tr>
<tr>
<td>Summit</td>
<td>SU</td>
<td>SUM</td>
<td>49043</td>
</tr>
<tr>
<td>Tooele</td>
<td>TO</td>
<td>TOO</td>
<td>49045</td>
</tr>
<tr>
<td>Uintah</td>
<td>UI</td>
<td>UIN</td>
<td>49047</td>
</tr>
<tr>
<td>Utah</td>
<td>UT</td>
<td>UTA</td>
<td>49049</td>
</tr>
<tr>
<td>Wasatch</td>
<td>WS</td>
<td>WST</td>
<td>49051</td>
</tr>
<tr>
<td>Washington</td>
<td>WA</td>
<td>WSH</td>
<td>49053</td>
</tr>
<tr>
<td>Wayne</td>
<td>WN</td>
<td>WAY</td>
<td>49055</td>
</tr>
<tr>
<td>Weber</td>
<td>WB</td>
<td>WEB</td>
<td>49057</td>
</tr>
</tbody>
</table>

When compiling the information for the \textit{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 ‘\texttt{rownames}’ call parameter to "ab". The second (alternate) abbreviation set can be used by setting the ‘\texttt{rownames}’ to "alt_ab". The \textit{id} field value is the 5 digit U. S. state and county FIPS code. The ‘\texttt{rownames}’ = "alias" and the ‘\texttt{regions}’ features are not supported in the \textit{UtahBG} border group.

\textbf{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.
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.
X1990 a numeric vector of the county’s population in 1990.
X1993 a numeric vector of the county’s population in 1993.
X1996 a numeric vector of the county’s population in 1996.
X1997 a numeric vector of the county’s population in 1997.
X1999 a numeric vector of the county’s population in 1999.
X2000 a numeric vector of the county’s population in 2000.
X2002 a numeric vector of the county’s population in 2002.
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.

---

**wf1ung00and95**  
Lung cancer mortality data for white females, 2000-4 and 1995-9

---

Description


Usage

wf1ung00and95

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

---

Descripion

Usage
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 for 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


References

none

wflung@cnty

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

wflung@cnty

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 numeric 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

References
FIP 6-4 Codes

---

**wmlung5070**  
*Lung cancer mortality data for white males, 1950-69 and 1970-94*

Description

Usage
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


References


Description


Usage

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
- **COUNTWM_70** a numeric vector, US number of cases from 1970-1994

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

*Topic data structure
  micromapGDefaults, 47
  panelDesc, 82

*Topic datasets
  AfricaBG, 7
  AfricaPopData, 9
  bordGrp, 10
  ChinaBG, 15
cnPopData, 17
detailsVariables, 18
  Educ8thData, 19
  KansasBG, 20
  KansPopInc, 23
  MarylandBG, 24
  mdPopData, 25
  messages, 26
  NewYorkBG, 79
  nyPopData, 81
  Seer18Area, 90
  SeoulPopData, 91
  SeoulSKoreaBG, 92
  statePop2010, 94
  TSdata, 95
  UKIrelandBG, 97
  UKIrelandPopData, 104
  USSeerBG, 105
  USStatesBG, 109
  UtahBG, 112
  UtahPopData, 113
  wflung00and95, 116
  wflung00and95US, 117
  wflung00cnty, 118
  wmlung5070, 119
  wmlung5070US, 120

*Topic functions
  micromapGSetDefaults, 57
  micromapGSetPanelDef, 58
  micromapSEER, 59

*Topic panelDesc
  panelDesc, 82

AfricaBG, 7
AfricaPopData, 9
areaNamesAbbrsIDs (bordGrp), 10
areaParams (bordGrp), 10
areaVisBorders (bordGrp), 10
bordGrp, 10
ChinaBG, 15
cnPopData, 17
detailsVariables, 18
Educ8thData, 19
KansasBG, 20
KansPopInc, 23
L2VisBorders (bordGrp), 10
L3VisBorders (bordGrp), 10
MarylandBG, 24
mdPopData, 25
messages, 26
micromapGDefaults, 47, 58
micromapGSetDefaults, 57, 58
micromapGSetPanelDef, 57, 58, 59
micromapSEER, 59, 66
micromapST, 59, 60, 66, 90
micromapST-package, 3
NewYorkBG, 79
nyPopData, 81
panelDesc, 82
RegVisBorders (bordGrp), 10
Seer18Area, 90
SeoulPopData, 91
SeoulSKoreaBG, 92
statePop2010, 94
TSdata, 95
UKIrelandBG, 97
UKIrelandPopData, 104
UKIrelandPopData2(UKIrelandPopData),
104
USSeerBG, 105
USStatesBG, 109
UtahBG, 112
UtahPopData, 113
wflung00and95, 116
wflung00and95US, 117
wflung00cnty, 118
wmlung5070, 119
wmlung5070US, 120