

# Package ‘micromap’

February 8, 2018

**Version** 1.9.3

**Date** 2018-02-05

**Title** Linked Micromap Plots

**Depends** R (>= 2.10), maptools, RColorBrewer, rgdal, sp

**Imports** ggplot2 (>= 2.0.0), grid

**Description** This group of functions simplifies the creation of linked micromap plots.

**License** GPL (>= 2)

**URL** <http://www.jstatsoft.org/v63/i02/>

**Author** Quinn Payton [aut],  
Tony Olsen [aut],  
Marc Weber [ctb],  
Michael McManus [ctb],  
Tom Kincaid [ctb],  
Marcus W. Beck [cre, ctb]

**Maintainer** Marcus W. Beck <marcusb@sccwrp.org>

**NeedsCompilation** no

**Repository** CRAN

**Suggests** knitr, rgeos

**RoxygenNote** 6.0.1

**Date/Publication** 2018-02-08 18:53:05 UTC

## R topics documented:

alterForMedian . . . . .	2
assimilatePlot . . . . .	3
create_map_table . . . . .	3
edPov . . . . .	4
labels_att . . . . .	5
labels_build . . . . .	5
lmgroupedplot . . . . .	6

lungMort . . . . .	12
OrEcoLevel3 . . . . .	13
print.mm . . . . .	13
printLMPlot . . . . .	14
right . . . . .	14
sample_att . . . . .	15
statesFlatfile . . . . .	16
USstates . . . . .	17
vegCov . . . . .	17
WSA3 . . . . .	18

<b>Index</b>	<b>19</b>
--------------	-----------

---

alterForMedian	<i>Insert a Median Line</i>
----------------	-----------------------------

---

## Description

Adds a median line to the DF table.

## Usage

```
alterForMedian(DF, a)
```

## Arguments

DF	the DF table created by create_DF_rank.
a	the attributes list, which contains the user specified choice of whether to include a median row.

## Value

Returns the DF table.

## Note

See the Introduction Guide for a full list of the options available for altering micromaps.

## Author(s)

Quinn Payton <Payton.Quinn@epa.gov>

---

assimilatePlot	<i>Cleans a User Made Panel</i>
----------------	---------------------------------

---

**Description**

Sends a user made panel out to the graph "cleaning functions" in Implot in order to properly display a user made panel seamlessly into the rest of an Implot.

**Usage**

```
assimilatePlot(pl, i, a, limsx=NA, limsy=NA)
```

**Arguments**

pl	the Implot object.
i	the panel number.
a	the attribute list.
limsx	limits of the x axis if desired.
limsy	limits of the y axis if desired.

**Value**

Returns a cleaned plot object.

**Note**

See the Introduction Guide for a full list of the options available for altering micromaps.

**Author(s)**

Quinn Payton <Payton.Quinn@epa.gov>

---

create_map_table	<i>Prepares A Polygon Table</i>
------------------	---------------------------------

---

**Description**

Takes the user supplied polygon table and prepares it to be plotted correctly with the Implot function.

**Usage**

```
create_map_table(tmp.map, IDcolumn=NA, poly.thresh=.0001)
```

**Arguments**

<code>tmp.map</code>	a shapefile to be parsed into a flat table for use with <code>ggplot2</code> .
<code>IDcolumn</code>	specify which column in the data file to use as a unique identifier.
<code>poly.thresh</code>	specify the minimum polygon area which to keep. All polygons with less area will be dropped.

**Value**

A flat table of all the polygons in the shapefile with an entry for each vertice.

**Author(s)**

Quinn Payton <Payton.Quinn@epa.gov>

**Examples**

```
data("USstates")
head(USstates@data)
statePolys <- create_map_table(USstates, IDcolumn="ST")
head(statePolys)
```

---

edPov

*Example Dataset: Education and Poverty Levels*

---

**Description**

A dataset comparing education and poverty levels among the states.

**Usage**

```
data(edPov)
```

**Format**

A data frame with 51 rows and the following 5 columns:

- state** full state name.
- ed** education rate.
- pov** poverty rate.
- region** major us region.
- StateAb** abbreviated state name.

**Note**

This is just an example data set.

**Examples**

```
data(edPov)
head(edPov)
```

---

labels_att	<i>Default Attribute Lists</i>
------------	--------------------------------

---

**Description**

Creates a list of default attribute lists for each panel type.

**Usage**

```
labels_att(show=FALSE)
```

**Arguments**

show                    If show equals FALSE, then the resulting list is returned invisibly.

**Value**

A list of defaults to be stored in the overall attribute list.

**Author(s)**

Quinn Payton <Payton.Quinn@epa.gov>

---

labels_build	<i>Builds panels</i>
--------------	----------------------

---

**Description**

Creates a panel of the user specified type using the attribute list to make adjustments.

**Usage**

```
labels_build(pl, p, DF, att)
```

**Arguments**

pl                    the lplot object being constructed.  
p                    number of the panel within the plot.  
DF                    the data table of statistics to be referenced.  
att                    the attribute list to specify visual and graphical characteristics.

**Value**

Returns a ggplot object to be printed later in the lmpplot function.

**Author(s)**

Quinn Payton <Payton.Quinn@epa.gov>

---

lmgroupedplot

*Linked Micromaps*

---

**Description**

Creates a linked micromap, displaying specified polygons and their associated statistical summary displays; differentiated by color.

**Usage**

```
lmgroupedplot(stat.data, map.data, panel.types, panel.data, map.link = NULL,
  nPanels = length(panel.types), grp.by, cat, colors = brewer.pal(10,
  "Spectral"), map.color = "lightyellow", map.all = FALSE,
  print.file = "no", print.res = NA, panel.att = vector("list", nPanels),
  plot.header = NA, plot.header.size = NA, plot.header.color = NA,
  plot.footer = NA, plot.footer.size = NA, plot.footer.color = NA,
  plot.width = 7, plot.height = 7, map.spacing = 1,
  plot.grp.spacing = 1, plot.panel.spacing = 1, plot.panel.margins = c(0,
  0, 1, 0), ...)
```

```
lmpplot(stat.data, map.data = NULL, panel.types, panel.data, map.link = NULL,
  nPanels = length(panel.types), ord.by, rev.ord = FALSE, grouping,
  median.row = FALSE, vertical.align = "top", median.color = gray(0.5),
  colors = brewer.pal(max(grouping), "Spectral"), map.all = FALSE,
  map.color2 = "lightgray", two.ended.maps = FALSE, print.file = "no",
  print.res = 300, panel.att = vector("list", nPanels), plot.header = NA,
  plot.header.size = NA, plot.header.color = NA, plot.footer = NA,
  plot.footer.size = NA, plot.footer.color = NA, plot.width = 7,
  plot.height = 7, map.spacing = 1, plot.pGrp.spacing = 1,
  plot.panel.spacing = 1, plot.panel.margins = c(0, 0, 1, 0), ...)
```

```
mmgroupedplot(stat.data, map.data, panel.types, panel.data, map.link = NULL,
  nPanels = length(panel.types), grp.by, cat, colors = brewer.pal(10,
  "Spectral"), map.color = "lightyellow", map.all = FALSE,
  print.file = "no", print.res = NA, panel.att = vector("list", nPanels),
  plot.header = NA, plot.header.size = NA, plot.header.color = NA,
  plot.footer = NA, plot.footer.size = NA, plot.footer.color = NA,
  plot.width = 7, plot.height = 7, map.spacing = 1,
  plot.grp.spacing = 1, plot.panel.spacing = 1, plot.panel.margins = c(0,
  0, 1, 0))
```

```

mmpplot(map.data, ...)

## S3 method for class 'SpatialPolygonsDataFrame'
mmpplot(map.data, ...)

## Default S3 method:
mmpplot(map.data, stat.data, panel.types, panel.data, map.link,
  nPanels = length(panel.types), ord.by, rev.ord = FALSE, grouping,
  median.row = FALSE, vertical.align = "top", median.color = gray(0.5),
  median.text.color = "black", median.text.size = 1,
  median.text.label = "Median", colors = brewer.pal(max(grouping),
  "Spectral"), map.all = FALSE, map.color2 = "lightgray",
  two.ended.maps = FALSE, trans = "identity", print.file = "no",
  print.res = 300, panel.att = vector("list", nPanels), plot.header = NA,
  plot.header.size = NA, plot.header.color = NA, plot.footer = NA,
  plot.footer.size = NA, plot.footer.color = NA, plot.width = 7,
  plot.height = 7, map.spacing = 1, plot.pGrp.spacing = 1,
  plot.panel.spacing = 1, plot.panel.margins = c(0, 0, 1, 0), ...)

```

## Arguments

<code>stat.data</code>	table of statistics for display
<code>map.data</code>	table of polygons to be associated with each item in <code>stat.data</code> .
<code>panel.types</code>	vector of panel types to specify the layout of the plot (e.g. <code>c('map', 'labels', 'dot.cl')</code> )
<code>panel.data</code>	a list (of lists) of data to be used with each panel (e.g. <code>list(NA, 'Names', list('lower.bound', 'estimate', 'upper.bound'))</code> ).
<code>map.link</code>	a vector with the name of the columns from <code>stat.data</code> and <code>map.data</code> , respectively, on which to join.
<code>nPanels</code>	the number of panels, which is not expected to be set by the user. The default is the length of <code>panel.types</code> .
<code>grp.by</code>	The column name from <code>stat.data</code> with which to order the lines of the output graphic for a standard <code>lmPLot</code> or identifier column on which to group the categorized <code>lmPLot</code> .
<code>cat</code>	category column within stats table for a categorization type <code>Implot</code> .
<code>colors</code>	a vector of colors for the perceptual groups. The default is <code>brewer.pal(max(grouping), 'Spectral')</code> for <code>Implot</code> and <code>brewer.pal(10, 'Spectral')</code> for <code>Imgroupedplot</code> . The colors are passed to <code>colorRampPalette</code> to create a continuous color vector equal in length to the groupings.
<code>map.color</code>	the color to fill in previously displayed polygons.
<code>map.all</code>	by default, <code>Implot</code> will only plot the polygons associated with data in the stats table; <code>map.all = TRUE</code> will show all the polygons in the polygon table regardless of whether they are actively referred to.
<code>print.file</code>	name of the file being created. The extension ( <code>.pdf</code> , <code>.tiff</code> , <code>.jpeg</code> , <code>.png</code> ) tells <code>Implot</code> which image creation tool to use.

<code>print.res</code>	the resolution of the image to use.
<code>panel.att</code>	a list of panel specific attributes to be altered (see <code>lmpplot</code> documentation).
<code>plot.header</code>	the overall title to be placed on the <code>lmPlot</code> .
<code>plot.header.size</code>	size of the overall title to be placed on the <code>lmPlot</code> .
<code>plot.header.color</code>	color of the overall title to be placed on the <code>lmPlot</code> .
<code>plot.footer</code>	the overall footer to be placed under the <code>lmPlot</code> .
<code>plot.footer.size</code>	size of the overall footer to be placed under the <code>lmPlot</code> .
<code>plot.footer.color</code>	color of the overall footer to be placed under the <code>lmPlot</code> .
<code>plot.width</code>	width of the overall plot in inches. Defaults to 7.
<code>plot.height</code>	height of the overall plot in inches. Defaults to 7.
<code>map.spacing</code>	the vertical spacing between maps measured in lines. Perceptual group spacing does not affect map spacing so as to leave the maps as large as possible. The user can increase map spacing using this argument. Defaults to 1.
<code>plot.grp.spacing</code>	the vertical spacing between groups measured in lines. Defaults to 1.
<code>plot.panel.spacing</code>	the vertical spacing between panels measured in lines. Defaults to 1.
<code>plot.panel.margins</code>	the horizontal spacing between panels measured in lines. THIS IS LEGACY CODE AND SHOULD NOT BE USED.
<code>...</code>	Additional arguments passed to or from other methods.
<code>ord.by</code>	The column name from <code>stat.data</code> with which to order the lines of the output graphic for a standard <code>lmPlot</code> or identifier column on which to group the categorized <code>lmPlot</code> .
<code>rev.ord</code>	specifies whether the plot should be displayed in reverse order of the ranking column. The default is <code>FALSE</code> .
<code>grouping</code>	the number of lines per perceptual group (for the standard <code>lmpplot</code> only). Can be a single number to have the same number in each group or a vector of numbers for unequal groupings.
<code>median.row</code>	specifies whether a median row should be included. If an odd number of data lines are supplied, a data line itself will be used as the median, otherwise median entries will be calculated from the supplied data. Note that without a median row maps are forced into proper size.
<code>vertical.align</code>	controls vertical alignment of the median row.
<code>median.color</code>	specifies color of the median row.
<code>map.color2</code>	the color to fill in previously displayed polygons.
<code>two.ended.maps</code>	the resulting micromaps will highlight previously referenced polygons (see <code>map.color2</code> ) up to the median perceptual group then switch to highlighting all polygons that are still to be referenced later.



`plot.pGrp.spacing` the vertical spacing between perceptual groups measured in lines. Defaults to 1.

`median.text.color` specifies color of text in the median row.

`median.text.size` specifies size of text in the median row.

`median.text.label` specifies the label for text in the median row.

`trans` chr string for axis transformations, passed to `scale_x_continuous`. Acceptable values are "asn", "atanh", "boxcox", "exp", "identity", "log", "log10", "log1p", "log2", "logit", "probability", "probit", "reciprocal", "reverse" or "sqrt". One value will be recycled to all panels as needed, otherwise one per panel can be used in a combined string. Applies only to panels with axes.

**Value**

A list of ggplot2 objects with entries for each individual panel.

**Note**

See the Introduction Guide for a full list of the options available for altering micromaps.

**Author(s)**

Quinn Payton <Payton.Quinn@epa.gov>

**Examples**

```
# initial example

data("USstates")
head(USstates@data)
statePolys <- create_map_table(USstates, 'ST')
head(statePolys)

data("edPov")

# basic figure 1
lmpplot(stat.data = edPov,
        map.data = statePolys,
        panel.types = c('labels', 'dot', 'dot', 'map'),
        panel.data = list('state', 'pov', 'ed', NA),
        ord.by = 'pov',
        grouping = 5, median.row = TRUE,
        map.link = c('StateAb', 'ID'))

## Not run:
# publication figure 1a
lmpplot(stat.data = edPov, map.data = statePolys ,
        panel.types = c('labels', 'dot', 'dot', 'map'),
        panel.data = list('state', 'pov', 'ed', NA),
```

```

ord.by = 'pov',
grouping = 5,
  median.row = TRUE,
map.link = c('StateAb','ID'),

  plot.height = 9,
colors = c('red','orange','green','blue','purple'),
map.color2 = 'lightgray',

panel.att = list(
  list(1, header = 'States', panel.width = .8, align = 'left',
    text.size = .9),
list(2, header = 'Percent Living Below \n Poverty Level',
graph.bgcolor = 'lightgray', point.size = 1.5,
xaxis.ticks = list(10,15,20), xaxis.labels = list(10,15,20),
xaxis.title = 'Percent'),
list(3, header = 'Percent Adults With\n4+ Years of College',
graph.bgcolor = 'lightgray', point.size = 1.5,
xaxis.ticks = list(0,20,30,40), xaxis.labels = list(0,20,30,40),
xaxis.title = 'Percent'),
list(4, header = 'Light Gray Means\nHighlighted Above',
inactive.border.color = gray(.7), inactive.border.size = 2,
panel.width = .8)))

edPov$points <- 0

# publication figure 1b
lmpplot (stat.data = edPov, map.data = statePolys,
panel.types = c('dot', 'labels', 'dot', 'dot', 'map'),
panel.data = list('points', 'state', 'pov', 'ed', NA),
map.link = c('StateAb','ID'),
ord.by = 'pov',
grouping = 5,
median.row = TRUE,

plot.height = 9,

colors = c('red','orange','green','blue','purple'),
map.color2 = 'lightgray',

panel.att = list(list(1, panel.width = .15, point.type = 20,
graph.border.color = 'white',
xaxis.text.display = FALSE, xaxis.line.display = FALSE,
graph.grid.major = FALSE),

list(2, header = 'States', panel.width = .8,
align = 'left', text.size = .9),

list(3, header = 'Percent Living Below\nPoverty Level',
graph.bgcolor = 'lightgray', point.size = 1.5,
xaxis.ticks = list(10,15,20),
xaxis.labels = list(10,15,20),
xaxis.title = 'Percent'),

```

```

list(4, header = 'Percent Adults With\n4+ Years of College',
graph.bgcolor = 'lightgray', point.size = 1.5,
axis.ticks = list(20,30,40),
axis.labels = list(20,30,40),
axis.title = 'Percent'),

list(5, header = 'Light Gray Means\nHighlighted Above',
inactive.border.color = gray(.7), inactive.border.size = 2,
panel.width = .8)))

# publication figure 1c
myPlot <- lplot(stat.data = edPov, map.data = statePolys,
panel.types = c('map', 'dot', 'labels', 'dot', 'dot'),
panel.data = list(NA, 'points', 'state', 'pov', 'ed'),
map.link = c('StateAb','ID'),
ord.by = 'pov',
grouping = 5,
median.row = TRUE,

plot.height = 9,

colors = c('red','orange','green','blue','purple'),
map.color2 = 'lightgray',

panel.att = list(list(2, panel.width = .15, point.type = 20,
graph.border.color = 'white',
axis.text.display = FALSE, axis.line.display = FALSE,
graph.grid.major = FALSE),

list(3, header = 'States', panel.width = .8,
align = 'left', text.size = .9),

list(4, header = 'Percent Living Below\nPoverty Level',
graph.bgcolor = 'lightgray', point.size = 1.5,
axis.ticks = list(10,15,20), axis.labels = list(10,15,20),
axis.title = 'Percent'),

list(5, header = 'Percent Adults With\n4+ Years of College',
graph.bgcolor = 'lightgray', point.size = 1.5,
axis.ticks = list(20,30,40),
axis.labels = list(20,30,40),
axis.title = 'Percent'),

list(1, header = 'Light Gray Means\nHighlighted Above',
inactive.border.color = gray(.7), inactive.border.size = 2,
panel.width = .8)))

print(myPlot, name = 'myExhibit.tiff', res = 300)

## End(Not run)

```

---

`lungMort`*Example Dataset: Lung Cancer Mortality*

---

**Description**

Lung mortality rates for all 50 states and Washington D.C.

**Usage**

```
data(lungMort)
```

**Format**

A data frame with 51 rows and the following 14 columns:

**StateAb** abbreviation of state name.

**Rate\_95** mortality rate estimate for 1995.

**Count\_95** estimated number of cases in 1995.

**Lower\_95** lower bound of a 95 percent confidence interval for 1995 mortality rate.

**Upper\_95** upper bound of a 95 percent confidence interval for 1995 mortality rate.

**Pop\_95** population in 1995.

**StdErr\_95** estimated standard error of the estimated mortality rate in 1995.

**Rate\_00** mortality rate estimate for 2000.

**Count\_00** estimated number of cases in 2000.

**Lower\_00** lower bound of a 95 percent confidence interval for 2000 mortality rate.

**Upper\_00** upper bound of a 95 percent confidence interval for 2000 mortality rate.

**Pop\_00** population in 2000.

**StdErr\_00** estimated standard error of the estimated mortality rate in 2000.

**State** full state name.

**Note**

This is just an example data set.

**Examples**

```
data(lungMort)
head(lungMort)
```

---

OrEcoLevel3

*Example Dataset: Oregon Level 3 Ecoregion Shapefile*


---

**Description**

An object of class 'SpatialPolygonsDataFrame' that was created from a shapefile detailing boundaries of the level 3 ecoregion borders of Oregon.

**Usage**

```
data(OrEcoLevel3)
```

**Format**

The format is: Formal class 'SpatialPointsDataFrame' [package "sp"].

**Note**

This is just an example data set.

**Examples**

```
data(OrEcoLevel3)
spplot(OrEcoLevel3, zcol="US_L3NAME", col.regions=rainbow(9, s=0.75))
```

---

print.mm

*Print Implot Object to a File*


---

**Description**

Takes a stored Implot object of class mm and prints to a file whose type is implied by the file name.

**Usage**

```
## S3 method for class 'mm'
print(x, name, res, ...)
```

**Arguments**

x	name of the Implot object.
name	name of the file to be produced. tiff, png, jpeg, and pdf are recognized.
res	resolution in dpi.
...	Additional arguments. (Currently ignored.)

**Value**

Returns nothing.

**Author(s)**

Quinn Payton <Payton.Quinn@epa.gov>

---

printLMPlot	<i>Print Implot Object to a File</i>
-------------	--------------------------------------

---

**Description**

Takes a stored Implot object and uses the print method for objects of class mm to print to a file whose type is implied by the file name.

**Usage**

```
printLMPlot(pobject, name, res)
```

**Arguments**

pobject	name of the Implot object.
name	name of the file to be produced. Tiff, png, jpeg, and pdf are recognized.
res	resolution in dpi.

**Value**

Returns nothing.

**Author(s)**

Quinn Payton <Payton.Quinn@epa.gov>

---

right	<i>Substring Function</i>
-------	---------------------------

---

**Description**

Mimics the Excel Function of the Same Name

**Usage**

```
right(txt, i)
```

**Arguments**

txt                    string to be substringed.  
i                        number of characters to substring.

**Value**

A character value containing the substring.

**Author(s)**

Quinn Payton <Payton.Quinn@epa.gov>

---

sample\_att                    *Sample Attribute List Creator*

---

**Description**

Creates a sample attribute list for use in the creation of new panel types.

**Usage**

```
sample_att(size=1, type=rep("standard", size), ord.by=NA, grouping=5,
           colors=brewer.pal(max(grouping), "Spectral"), plot.pGrp.spacing=0.05,
           plot.panel.margins=c(0,1,0,0), panel.data=list(NA), median.row=FALSE,
           show=FALSE)
```

**Arguments**

size                    specify the number of attributes lists to be created.  
type                    match the attributes to which type of default panel.  
ord.by                  argument inherited from lmplot/lmGroupedplot.  
grouping                argument inherited from lmplot/lmGroupedplot.  
colors                  argument inherited from lmplot/lmGroupedplot.  
plot.pGrp.spacing      argument inherited from lmplot/lmGroupedplot.  
plot.panel.margins     argument inherited from lmplot/lmGroupedplot.  
panel.data              argument inherited from lmplot/lmGroupedplot.  
median.row              argument inherited from lmplot/lmGroupedplot.  
show                    controls whether to print the resulting attribute list.

**Value**

A list of attributes to be used in the creation of a panel

**Note**

See the Introduction Guide for a full list of the options available for altering micromaps.

**Author(s)**

Quinn Payton <Payton.Quinn@epa.gov>

---

statesFlatfile

*Example Dataset: A Table of State Polygons*

---

**Description**

Polygon vertices for each state.

**Usage**

```
data(statesFlatfile)
```

**Format**

A data frame with 434 rows and the following 4 columns:

**ID** abbreviation or state name.

**coordsx** x coordinates for each polygon vertice.

**coordsy** y coordinates for each polygon vertice.

**poly** individual polygon identifier within each state.

**Note**

This is just an example data set.

**Examples**

```
data(statesFlatfile)
head(statesFlatfile)
```



---

USstates

*Example Dataset: U.S. States Polygons*

---

### Description

An object of class 'SpatialPolygonsDataFrame' that was created from a shapefile of U.S. state borders.

### Usage

```
data(USstates)
```

### Format

The format is: Formal class 'SpatialPolygonsDataFrame' [package "sp"].

### Note

This is just an example data set.

### Examples

```
data(USstates)
plot(USstates)
```

---

vegCov

*Example Dataset: Vegetation Coverage Percentages*

---

### Description

A subset of a larger vegetation coverage analysis dataset.

### Usage

```
data(vegCov)
```

### Format

A data frame with 12 rows and the following 13 columns:

**Type** the population name.

**Subpopulation** the subpopulation name, either national or level 3 ecoregion.

**Indicator** the ecological variable (indicator)

**Category** category of disturbance.

**NResp** sample size.

**Estimate.P** estimated percentage of vegetation coverage with a given classification.

**StdError.P** standard error of the percentage estimate.

**LCB95Pct.P** lower bound of a 95 percent confidence interval for the estimated percentage.

**UCB95Pct.P** upper bound of a 95 percent confidence interval for the estimated percentage.

**Estimate.U** estimated area of vegetation coverage with a given classification.

**StdError.U** standard error of the area estimate.

**LCB95Pct.U** lower bound of a 95 percent confidence interval for the estimated area.

**UCB95Pct.U** upper bound of a 95 percent confidence interval for the estimated area.

### Note

This is just an example data set.

### Examples

```
data(vegCov)
head(vegCov)
```

---

WSA3

*Example Dataset: Major U.S. EcoRegions*

---

### Description

An object of class 'SpatialPolygonsDataFrame' that was created from a shapefile of the U.S. level 3 ecoregion boundaries.

### Usage

```
data(WSA3)
```

### Format

The format is: Formal class 'SpatialPointsDataFrame' [package "sp"].

### Note

This is just an example data set.

### Examples

```
data(WSA3)
spplot(WSA3, zcol="WSA_3_NM", col.regions=rainbow(3, s=0.75))
```

# Index

## \*Topic **datasets**

- edPov, 4
- lungMort, 12
- OrEcoLevel3, 13
- statesFlatfile, 16
- USstates, 17
- vegCov, 17
- WSA3, 18

## \*Topic **hplot**

- lmgroupedplot, 6

alterForMedian, 2

assimilatePlot, 3

bar\_att (labels\_att), 5

bar\_build (labels\_build), 5

bar\_cl\_att (labels\_att), 5

bar\_cl\_build (labels\_build), 5

box\_summary\_att (labels\_att), 5

box\_summary\_build (labels\_build), 5

colorRampPalette, 7

create\_map\_table, 3

dot\_att (labels\_att), 5

dot\_build (labels\_build), 5

dot\_cl\_att (labels\_att), 5

dot\_cl\_build (labels\_build), 5

edPov, 4

labels\_att, 5

labels\_build, 5

lmgroupedplot, 6

lmpplot (lmgroupedplot), 6

lungMort, 12

map\_att (labels\_att), 5

mmgroupedplot (lmgroupedplot), 6

mmpplot (lmgroupedplot), 6

OrEcoLevel3, 13

print.mm, 13

printLMPlot, 14

ranks\_att (labels\_att), 5

ranks\_build (labels\_build), 5

right, 14

sample\_att, 15

scale\_x\_continuous, 9

standard\_att (labels\_att), 5

statesFlatfile, 16

USstates, 17

vegCov, 17

WSA3, 18