

Package ‘sharpshootR’

July 2, 2014

Type Package

Title A collection of functions to support soil survey

Description A collection of functions to support soil survey

Version 0.6-3

Date 2014-05-05

Author USDA-NRCS Soil Survey Staff

Maintainer Dylan Beaudette <dylan.beaudette@ca.usda.gov>

License GPL (>= 2)

Repository CRAN

URL <http://aqp.r-forge.r-project.org/>

Suggests soilDB, MASS

Depends R (>= 2.15.0), aqp, ape

Imports igraph, cluster, lattice, latticeExtra, vegan, sp, reshape2,Hmisc, scales, circular, RColorBrewer, plyr

NeedsCompilation no

Date/Publication 2014-05-06 07:09:27

R topics documented:

sharpshootR-package	2
amador	2
aspect.plot	3
CDEC.snow.courses	4
CDECquery	5
CDECsnowQuery	6
component.adj.matrix	7

diagnosticPropertyPlot	8
dueling.dendrograms	10
hillslope.probability	11
plotSoilRelationGraph	12
sample.by.poly	14
SoilTaxonomyDendrogram	15

Index	17
--------------	-----------

sharpshootR-package *A collection of functions to support soil survey*

Description

A collection of functions to support soil survey

amador *SSURGO Data Associated with the Amador Soil Series*

Description

SSURGO Data Associated with the Amador Soil Series

Usage

data(amador)

Format

A subset of data taken from the "component" table of SSURGO

mukey map unit key

compname component name

compct_r component percentage

Source

USDA-NRCS SSURGO Database

 aspect.plot

Plot Aspect Data

Description

Plot a graphical summary of multiple aspect measurements on a circular diagram.

Usage

```
aspect.plot(p, p.bins = 60, p.bw = 30, stack=TRUE,
p.axis = seq(0, 350, by = 10), plot.title = NULL,
line.col='RoyalBlue', line.lwd=1, line.lty=2,
arrow.col=line.col, arrow.lwd=1, arrow.lty=1,
...)
```

Arguments

p	a vector of aspect angles in degrees, measured clock-wise from North
p.bins	number of bins to use for circular histogram
p.bw	bandwidth used for circular density estimation
stack	TRUE/FALSE, should the individual points be stacked into p.bins number of bins and plotted
p.axis	a sequence of integers (degrees) describing the circular axis
plot.title	an informative title
line.col	density line color
line.lwd	density line width
line.lty	density line line style
arrow.col	arrow color
arrow.lwd	arrow line width
arrow.lty	arrow line style
...	further arguments passed to plot.circular

Details

Spread and central tendency are depicted with a combination of circular histogram and kernel density estimate. The circular mean, and relative confidence in that mean are depicted with an arrow: longer arrow lengths correspond to greater confidence in the mean.

Note

Manual adjustment of p.bw may be required in order to get an optimal circular density plot. This function requires the package `circular`, version 0.4-7 or later.

Author(s)

D.E. Beaudette

Examples

```
# simulate some data
p.narrow <- runif(n=25, min=215, max=280)
p.wide <- runif(n=25, min=0, max=270)

# set figure margins to 0, 2-column plot
par(mar=c(0,0,0,0), mfcol=c(1,2))

# plot
aspect.plot(p.narrow, p.bw=10, plot.title='Soil A', pch=21, col='black', bg='RoyalBlue')
aspect.plot(p.wide, p.bw=10, plot.title='Soil B', pch=21, col='black', bg='RoyalBlue')
```

CDEC.snow.courses

CDEC Snow Course List

Description

The CDEC snow course list, updated 2014

Usage

```
data(CDEC.snow.courses)
```

Format

A data frame with 261 observations on the following 8 variables.

course_number course number

name connotative course label

id course ID

elev_feet course elevation in feet

latitude latitude

longitude longitude

april.1.Avg.inches average inches of snow as of April 1st

agency responsible agency

Source

Data were scraped from <http://cdec.water.ca.gov/misc/SnowCourses.html>, 2014.

Examples

```
data(CDEC.snow.courses)
head(CDEC.snow.courses)
```

CDECquery

Get water-related data (California only) from the CDEC website.

Description

Get water-related data (California only) from the CDEC website.

Usage

```
CDECquery(id, sensor, interval = "D", start, end)
```

Arguments

id	station ID (e.g. 'spw'), see details
sensor	the sensor ID (e.g. 45), see details
interval	character, 'D' for daily, 'H' for hourly, 'M' for monthly
start	starting date, in the format 'YYYY-MM-DD'
end	ending date, in the format 'YYYY-MM-DD'

Details

1. Station IDs can be found here: <http://cdec.water.ca.gov/staInfo.html>
2. Sensor IDs can be found using this URL cdec.water.ca.gov/cgi-progs/queryCSV?station_id=, followed by the station ID.

Value

a `data.frame` object with the following fields: 'datetime', 'year', 'month', 'value'.

Author(s)

D.E. Beaudette

References

<http://cdec.water.ca.gov/queryCSV.html>

See Also

[CDECsnowQuery](#)

Examples

```

## Not run:
library(RColorBrewer)
library(plyr)
library(e1071)

# get daily precip totals from Stan Powerhouse
x <- CDECquery(id='spw', sensor=45, interval='D', start='1900-01-01', end='2015-01-01')

# compute total precip by year/month
a <- ddply(x, c('year', 'month'), summarize, s=sum(value, na.rm=TRUE))

# make a nice color ramp function, scaled by the skewness of the underlying distribution
cols <- colorRampPalette(brewer.pal(9, 'Spectral'),
space='Lab', interpolate='spline', bias=skewness(a$s, na.rm=TRUE))

# plot, each pixel is colored by the total precip by year/month
levelplot(s ~ year * month, data=a, col.regions=cols, xlab='',
ylab='', scales=list(x=list(tick.number=10)), main='Monthly Total Precipitation (inches) SPW')

# get pre-aggregated monthly data from Sonora RS
x <- CDECquery(id='sor', sensor=2, interval='M', start='1900-01-01', end='2015-01-01')

# make a nice color ramp function, scaled by the skewness of the underlying distribution
cols <- colorRampPalette(brewer.pal(9, 'Spectral'), space='Lab',
interpolate='spline', bias=skewness(x$value, na.rm=TRUE))

# plot
levelplot(value ~ year * month, data=x, col.regions=cols, xlab='',
ylab='', scales=list(x=list(tick.number=20)), main='Monthly Total Precipitation (inches) SOR')

## End(Not run)

```

CDECsnowQuery

Get snow survey data (California only) from the CDEC website.

Description

Get snow survey data (California only) from the CDEC website.

Usage

```
CDECsnowQuery(course, start_yr, end_yr)
```

Arguments

course	integer, course number (e.g. 129)
start_yr	integer, the starting year (e.g. 2010)
end_yr	integer, the ending year (e.g. 2013)

Details

This function downloads data from the CDEC website, therefore an internet connection is required. The 'SWE' column contains adjusted SWE if available ('Adjusted' column), otherwise the reported SWE is used ('Water' column).

Value

a data.frame object, see examples

Note

Snow course locations, ID numbers, and other information can be found here: <http://cdec.water.ca.gov/misc/SnowCourses.html>

Author(s)

D.E. Beaudette

References

<http://cdec.water.ca.gov/cgi-progs/snowQuery>

Examples

```
## Not run:  
# get data for course numbe 129  
x <- CDECsnowQuery(course=129, start_yr=2010, end_yr=2011)  
  
## End(Not run)
```

component.adj.matrix *Create an adjacency matrix from a data.frame of component data*

Description

Create an adjacency matrix from SSURGO component data

Usage

```
component.adj.matrix(d, mu='mukey', co='compname', wt='compct_r',  
method='community.matrix', standardization='max', metric='jaccard',  
rm.orphans=TRUE)
```

Arguments

d	a data.frame, typically of SSURGO data
mu	name of the column containing the map unit ID (typically 'mukey')
co	name of the column containing the component ID (typically 'comname')
wt	name of the column containing the component weight percent (typically 'compct_r')
method	one of either: 'community.matrix', or 'occurrence'; see details
standardization	community matrix standardization method, passed to decostand
metric	community matrix dissimilarity metric, passed to vegdist
rm.orphans	logical, should map units with a single component be omitted? (typically yes)

Details

Pending...

Value

a similarity matrix / adjacency matrix suitable for use with igraph functions or anything else that can accommodate a similarity matrix.

Author(s)

D.E. Beaudette

Examples

```
# load sample data set
data(amador)

# convert into adjacency matrix
m <- component.adj.matrix(amador)

# plot network diagram, with Amador soil highlighted
plotSoilRelationGraph(m, s='amador')
```

diagnosticPropertyPlot

Diagnostic Property Plot

Description

Generate a graphical description of the presence/absence of soil diagnostic properties.

Usage

```
diagnosticPropertyPlot(f, v, k, id = "pedon_id")
diagnosticPropertyPlot2(f, v, k, id = "pedon_id")
```

Arguments

f	a SoilProfileCollection object
v	a character vector of site-level attributes that are boolean (e.g. TRUE/FALSE) data
k	an integer, number of groups to highlight
id	the name of the site-level attribute containing an ID to include next next to leave of the diagnostic property dendrogram

Details

This function attempts to display several pieces of information within a single figure. First, soil profiles are sorted according to the presence/absence of diagnostic features named in v. Second, these diagnostic features are sorted according to their distribution among soil profiles. Third, a binary grid is established with row-ordering of profiles based on step 1 and column-ordering based on step 2. Blue cell represent the presence of a diagnostic feature. Soils with similar diagnostic features should 'clump' together. See examples below.

Value

a list is silently returned by this function, containing:

rd a data.frame containing IDs and grouping code
 profile.order a vector containing the order of soil profiles (row-order in figure), according to diagnostic property values
 var.order a vector containing the order of variables (column-order in figure), according to their distribution among profiles

Note

diagnosticPropertyPlot2 is a lattice-based version, and requires a unique value passed to id.

Author(s)

D.E. Beaudette and J.M. Skovlin

Examples

```
## Not run:
# sample data, an SPC
data(gopheridge, package='soilDB')

# get depth class
sdc <- getSoilDepthClass(gopheridge)
site(gopheridge) <- sdc
```

```
# diagnostic properties to consider, no need to convert to factors
v <- c('lithic.contact', 'paralithic.contact', 'argillic.horizon',
'cambic.horizon', 'ochric.epipedon', 'mollic.epipedon', 'very.shallow',
'shallow', 'mod.deep', 'deep', 'very.deep')

# base graphics
x <- diagnosticPropertyPlot(gopheridge, v, k=5)

# lattice graphics
x <- diagnosticPropertyPlot2(gopheridge, v, k=3)

# check output
str(x)

## End(Not run)
```

dueling.dendrograms *Dueling Dendrograms*

Description

Graphically compare two related dendrograms

Usage

```
dueling.dendrograms(p.1, p.2, lab.1 = "D1",
lab.2 = "D2", cex.nodelabels=0.75, arrow.length=0.05)
```

Arguments

p.1	left-hand phylo-class dendrogram
p.2	right-hand phylo-class dendrogram
lab.1	left-hand title
lab.2	right-hand title
cex.nodelabels	character expansion size for node labels
arrow.length	arrow head size

Details

Connector arrows are used to link nodes from the left-hand dendrogram to the right-hand dendrogram.

Author(s)

D. E. Beaudette

Examples

```

library(aqp)
library(cluster)
library(ape)

# load sample dataset from aqp package
data(sp3)

# promote to SoilProfileCollection
depths(sp3) <- id ~ top + bottom

# compute dissimilarity using different sets of variables
# note that these are rescaled to the interval [0,1]
d.1 <- profile_compare(sp3, vars=c('clay', 'cec'), k=0, max_d=100, rescale.result=TRUE)
d.2 <- profile_compare(sp3, vars=c('clay', 'L'), k=0, max_d=100, rescale.result=TRUE)

# cluster via divisive hierarchical algorithm
# convert to 'phylo' class
p.1 <- as.phylo(as.hclust(diana(d.1)))
p.2 <- as.phylo(as.hclust(diana(d.2)))

# graphically compare two dendrograms
dueling.dendrograms(p.1, p.2, lab.1='clay and CEC', lab.2='clay and L')

# graphically check the results of ladderize() from ape package
dueling.dendrograms(p.1, ladderize(p.1), lab.1='standard', lab.2='ladderized')

# sanity-check: compare something to itself
dueling.dendrograms(p.1, p.1, lab.1='same', lab.2='same')

# graphically compare diana() to agnes() using d.2
dueling.dendrograms(as.phylo(as.hclust(diana(d.2))),
as.phylo(as.hclust(agnes(d.2))), lab.1='diana', lab.2='agnes')

```

hillslope.probability *Hillslope Probability via SDA*

Description

Hillslope position probability estimates from the SDA query service (SSURGO)

Usage

```
hillslope.probability(s)
```

Arguments

s a character vector of soil series names, in lower-case

Details

This function constructs and executes a query that is sent to the <http://sdmdataaccess.nrcs.usda.gov> webservice. Further information on the SDA webservice and query examples can be found at <http://sdmdataaccess.nrcs.usda.gov/QueryHelp.aspx>

Value

A data.frame object with rows representing soil series, and columns representing probability estimates of that series occurring along the standard 2D hillslope positions.

Note

Probability values are computed from SSURGO data.

Author(s)

D.E. Beaudette

Examples

```
## Not run:
library(soilDB)

# soil series of interest
s <- c('auburn', 'pardee', 'amador', 'pentz')

# generate hillslope probability table
hillslope.probability(s)

## End(Not run)
```

plotSoilRelationGraph *Plot a component relation graph*

Description

Plot a component relation graph based on an adjacency or similarity matrix.

Usage

```
plotSoilRelationGraph(m, s='', plot.style = "network", ...)
```

Arguments

m	adjacency matrix
s	central component; an empty character string is interpreted as no central component
plot.style	plot style ('network', or 'dendrogram')
...	further arguments passed to plotting function

Author(s)

D.E. Beaudette

Examples

```
# load sample data set
data(amador)

# create weighted adjacency matrix (see ?component.adj.matrix for details)
m <- component.adj.matrix(amador)

# plot network diagram, with Amador soil highlighted
plotSoilRelationGraph(m, s='amador')

# dendrogram representation
plotSoilRelationGraph(m, s='amador', plot.style='dendrogram')

# compare methods
m.o <- component.adj.matrix(amador, method='occurrence')

par(mfcol=c(1,2))
plotSoilRelationGraph(m, s='amador', plot.style='dendrogram')
title('community matrix')
plotSoilRelationGraph(m.o, s='amador', plot.style='dendrogram')
title('occurrence')

## Not run:
# get similar data from soilweb, for the Pardee series
s <- 'pardee'
u <- url(URLencode(paste(
'http://casoilresource.lawr.ucdavis.edu/soil_web/reflector_api/soils.php?',
'what=soil_series_component_query&q_string=', s, sep='')))

# fetch data
d <- read.table(u, sep='|', header=TRUE, stringsAsFactors=FALSE)

# normalize component names
d$compname <- tolower(d$compname)

# keep only major components
d <- subset(d, subset=compkind == 'Series')

# build adj. matrix and plot
m <- component.adj.matrix(d)
plotSoilRelationGraph(m, s=s, plot.style='dendrogram')

# alter plotting style, see ?plot.phylo
plotSoilRelationGraph(m, s=s, plot.style='dendrogram', type='fan')
plotSoilRelationGraph(m, s=s, plot.style='dendrogram', type='unrooted', use.edge.length=FALSE)

## End(Not run)
```

sample.by.poly *Sample by Polygon*

Description

Sample by polygon, according to a specified sampling density

Usage

```
sample.by.poly(p, n.pts.per.ac = 1, min.samples = 5,  
sampling.type = "hexagonal", iterations = 10, ...)
```

Arguments

p	a Polygon object, with coordinates from a projected CRS
n.pts.per.ac	requested sampling density in points per acre (results will be close)
min.samples	minimum requested samples / polygon
sampling.type	sampling type, see spsample
iterations	number of tries that spsample will attempt
...	additional arguments to spsample

Details

This function is typically used within the context of a loop or similar function that iterates over a list of Polygons associated with a SpatialPolygons object. See examples.

Value

A SpatialPoints object.

Note

This function expects that the Polygon object has coordinates associated with a projected CRS—e.g. units of meters.

Author(s)

D.E. Beaudette

 SoilTaxonomyDendrogram

Soil Taxonomy Dendrogram

Description

Plot a dendrogram based on the first 4 levels of Soil Taxonomy, with soil profiles hanging below. A dissimilarity matrix is computed using Gower's distance metric for nominal-scale variables, based on order, sub order, great group, and subgroup level taxa. See the Details and Examples sections below for more information.

Usage

```
SoilTaxonomyDendrogram(spc, name = "hzname", max.depth = 150,
  n.depth.ticks = 6, scaling.factor = 0.015, cex.names = 0.75,
  cex.id = 0.75, axis.line.offset = -4, width = 0.1, y.offset = 0.5,
  cex.taxon.labels = 0.66)
```

Arguments

<code>spc</code>	a <code>SoilProfileCollection</code> object, see details
<code>name</code>	column name containing horizon names
<code>max.depth</code>	depth at which profiles are truncated for plotting
<code>n.depth.ticks</code>	suggested number of ticks on the depth axis
<code>scaling.factor</code>	scaling factor used to convert depth units into plotting units
<code>cex.names</code>	character scaling for horizon names
<code>cex.id</code>	character scaling for profile IDs
<code>axis.line.offset</code>	horizontal offset for depth axis
<code>width</code>	width of profiles
<code>y.offset</code>	vertical offset between dendrogram and profiles
<code>cex.taxon.labels</code>	character scaling for taxonomic information

Details

This function looks for specific site-level attributes named: `soilorder`, `suborder`, `greatgroup`, and `subgroup`.

Value

The pair-wise dissimilarity matrix is invisibly returned. See examples below for ideas on how to access and further use these data.

Author(s)

D.E. Beaudette

References

D.E. Beaudette, P. Roudier and A.T. O'Geen. 2012. Algorithms for Quantitative Pedology, a Toolkit for Soil Scientists. *Computers & Geosciences*: 52: 258–268. (doi: 10.1016/j.cageo.2012.10.020)

<http://aqp.r-forge.r-project.org/>

Examples

```
## Not run:
library(soilDB)

# soils of interest
s.list <- c('musick', 'cecil', 'drummer', 'amador', 'pentz', 'reiff',
           'san joaquin', 'montpellier', 'grangeville', 'pollasky', 'ramona')

# fetch and convert data into an SPC
h <- fetchOSD(s.list)

# plot dendrogram + profiles
SoilTaxonomyDendrogram(h)

# again, this time save the pair-wise dissimilarity matrix
# note that there isn't a lot of discrimination between soils
(d <- SoilTaxonomyDendrogram(h))

## End(Not run)
```


Index

*Topic **datasets**

amador, [2](#)

CDEC.snow.courses, [4](#)

*Topic **hplots**

aspect.plot, [3](#)

diagnosticPropertyPlot, [8](#)

dueling.dendrograms, [10](#)

SoilTaxonomyDendrogram, [15](#)

*Topic **hplot**

plotSoilRelationGraph, [12](#)

*Topic **manip**

component.adj.matrix, [7](#)

hillslope.probability, [11](#)

sample.by.poly, [14](#)

amador, [2](#)

aspect.plot, [3](#)

CDEC.snow.courses, [4](#)

CDECquery, [5](#)

CDECsnowQuery, [5](#), [6](#)

component.adj.matrix, [7](#)

diagnosticPropertyPlot, [8](#)

diagnosticPropertyPlot2

(diagnosticPropertyPlot), [8](#)

dueling.dendrograms, [10](#)

hillslope.probability, [11](#)

plotSoilRelationGraph, [12](#)

sample.by.poly, [14](#)

sharpshootR (sharpshootR-package), [2](#)

sharpshootR-package, [2](#)

SoilTaxonomyDendrogram, [15](#)