Package 'swfscMisc'

August 12, 2014

Type Package

Title Miscellaneous Functions for Southwest Fisheries Science Center

Version 1.0.1

Date 2014-08-11

Author Eric Archer

Maintainer Eric Archer <eric.archer@noaa.gov>

Description Collection of conversion, analytical, geodesic, mapping, and plotting functions. Used to support packages and code written by researchers at the Southwest Fisheries Science Center of the National Oceanic and Atmospheric Administration.

Imports maps

Depends mapdata

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2014-08-12 06:53:33

R topics documented:

`affin.prop` *Affinity Propagation*

Description

Runs the Affinity Propagation clustering algorithm of Frey and Dueck, 2007.

Usage

```
affin.prop(sim.mat, num.iter = 100, stable.iter = 10, shared.pref = "min",
  lambda = 0.5)
```

Arguments

`sim.mat`	a similarity matrix between individuals to be clustered.
`num.iter`	maximum number of iterations to attempt.
`stable.iter`	number of sequential iterations for which consistent clustering is considered acceptable.
`shared.pref`	type of shared preference to use. Can be one of "min", "median", or a numeric value.
`lambda`	damping factor.

Value

A matrix with one row per sample in 'sim.mat' and one column for each iteration. Values in columns indicate cluster assignment (arbitrary numbers) for each sample.

Author(s)

Eric Archer <eric.archer@noaa.gov>

References

Frey, B.J., and D. Dueck. 2007. Clustering by passing messages between data points. Science 315:972-976

Examples

```
data(iris)

# Take 75 random iris rows for example
iris <- iris[sample(1:nrow(iris), 75), ]
iris <- droplevels(iris)

iris.sim <- -dist(iris[, -5])

iris.affin <- affin.prop(iris.sim, stable.iter = 5)
table(iris$Species, iris.affin[, ncol(iris.affin)])
```

---

bearing *Calculate Bearing Between Two Positions*

Description

Calculates the bearing between two points, given each point's latitude and longitude coordinates

Usage

```
bearing(lat1, lon1, lat2, lon2)
```

Arguments

lat1,lon1	numeric. The latitude and longitude of the starting coordinate in decimal degrees.
lat2,lon2	numeric. The latitude and longitude of the ending coordinate in decimal degrees.

Value

vector with initial and final bearings.

Author(s)

Eric Archer <eric.archer@noaa.gov>

Examples

```
# What is the bearing from San Diego, CA to Honolulu, HI?
bearing(32.87, -117.25, 21.35, -157.98)
```

box.area *Area of a Box*

Description

Calculate the area of a square on the earth.

Usage

```
box.area(lat, lon, edge, units = "nm")
```

Arguments

lat,lon	The latitude and longitude of the lower right corner of the box in decimal degrees.
edge	The length of one side of the square in decimal degrees.
units	units of distance. Can be "km" (kilometers), "nm" (nautical miles), or "mi" (statute miles).

Author(s)

Eric Archer <eric.archer@noaa.gov>

Examples

```
#What is the area of a 5 degree grid off of San Diego, CA?
box.area(32.87, -117.25, edge = 1, units = "nm")
box.area(32.87, -117.25, edge = 1, units = "km")
box.area(32.87, -117.25, edge = 1, units = "mi")
```

central.quantile *Central Quantile*

Description

Upper and lower values of central quantile

Usage

```
central.quantile(x, pct = 0.95)
```

Arguments

x	numeric vector.
pct	central percentile desired.

Value

a two element vector giving the lower and upper quantiles.

Author(s)

Eric Archer <eric.archer@noaa.gov>

Examples

```
x <- runif(1000)
central.quantile(x)
central.quantile(x, pct = 0.75)
```

`circle.polygon` *Circle Polygon (on Earth)*

Description

Creates a circular polygon (optionally on the earth) centered at a given point with a constant radius.

Usage

```
circle.polygon(x, y, radius, brng.limits = 0, sides = 1, by.length = TRUE,
  units = "nm", ellipsoid = datum(), dist.method = "lawofcosines",
  destination.type = "ellipsoid", poly.type = "cart.earth")
```

Arguments

`x,y`	number specifying the coordinates of the center of the circle in decimal degrees. If `poly.type` is "simple.earth" or "complex.earth", this will be longitude and latitude respectively.
`radius`	radius of sphere.
`brng.limits`	number, or vector of two numbers. If one value is given, it is used as the starting bearing in degrees for the first point of the circle. If a vector of two values is given, then they are used as the start and end bearings of arc.
`sides`	number that represents either length of sides or number of sides, as specified by the 'by.length' argument.
`by.length`	logical. If `TRUE`, then `sides` is the length of sides, if `FALSE`, then `sides` is number of sides.
`units`	character for units of distance: Can be "km" (kilometers), "nm" (nautical miles), "mi" (statute miles).
`ellipsoid`	ellipsoid model parameters as returned from a call to `datum`.
`dist.method`	character specifying method for calculating distance for `type` = "cart.earth". See `method` argument of `distance` for more information.

`destination.type`
: character specifying type of surface for `type = "gc.earth"`. See `type` argument of `destination` for more information.

`poly.type`
: character specifying the type of polygon calculation to use. Can be one of "cartesian" using basic cartesian coordinates, "cart.earth" for a simple polygon on the earth's surface treating latitude and longitude as cartesian coordinates, or "gc.earth" for a more precise calculation keeping a constant great-circle radius.

Value

A matrix representing the desired circle polygon centered at lat, lon of radius.

Author(s)

Eric Archer <eric.archer@noaa.gov>

Examples

```
cart.earth <- circle.polygon(-117.24, 32.86, 40, poly.type = "cart.earth")
gc.earth <- circle.polygon(-117.24, 32.86, 40, poly.type = "gc.earth")

lat.range <- c(32, 34)
lon.range <- c(-118.5, -116)

op <- par(mar = c(3, 5, 5, 5) + 0.1, oma = c(1, 1, 1, 1))

map("worldHires", fill = TRUE, col = "wheat3", xlim = lon.range, ylim = lat.range)
points(-117.24, 32.86, pch = 19, col = "red")
polygon(cart.earth, border = "red", lwd = 3)
lat.lon.axes(lon.range, lat.range, n = 3)
box(lwd = 2)
mtext("poly.type = 'cart.earth'", line = 3)

map("worldHires", fill = TRUE, col = "wheat3", xlim = lon.range, ylim = lat.range)
points(-117.24, 32.86, pch = 19, col = "red")
polygon(gc.earth, border = "red", lwd = 3)
lat.lon.axes(lon.range, lat.range, n = 3)
box(lwd = 2)
mtext("poly.type = 'gc.earth'", line = 3)

par(op)
```

---

color.name *Color Name*

Description

Return the name of a color listed given the number.

Usage

```
color.name(i)
```

Arguments

i	integer specifying color .

Value

character value of 'i' color.

Author(s)

Eric Archer <eric.archer@noaa.gov>

---

convert.angle *Angle Conversion*

Description

Converts angles between radians and degrees.

Usage

```
convert.angle(x, from = "degrees", to = "radians")
```

Arguments

x	numeric. The angle to be converted.
from,to	character. Units to convert from and to. Can be "radians" or "degrees" or any partial match.

Author(s)

Eric Archer <eric.archer@noaa.gov>

Examples

```
convert.angle(45, "deg", "rad")
convert.angle(4.5, "r", "d")
```

convert.distance *Distance Conversion*

Description

Convert distances between kilometers, nautical miles, and statute miles.

Usage

```
convert.distance(x, from = "nm", to = "km")
```

Arguments

x	numeric. The distance to be converted.
from,to	character. Units to convert from and to. Can be "km" (kilometers), "nm" (nautical miles), or "mi" (statute miles).

Author(s)

Eric Archer <eric.archer@noaa.gov>

copy.tri *Copy Matrix Triangles*

Description

Copy between lower left and upper right triangles of a matrix.

Usage

```
copy.tri(x, from = "lower")
```

Arguments

x	a matrix.
from	triangle to copy from. Can be "lower" or "upper".

Value

a matrix.

Author(s)

Eric Archer <eric.archer@noaa.gov>

Examples

```
x <- matrix(1:9, nrow = 3)
print(x)
copy.tri(x)
```

`datum` *Datum*

Description

Return parameters specifying ellipsoid datum model.

Usage

```
datum(model = "wgs84")
```

Arguments

`model`	character, specifying which model to use for ellipsoid model. Options are: "WGS84", "GRS80", "Airy", "International", "Clarke", "GRS67", or partial matches thereof (case-insensitive).

Value

vector of a, b, and f parameters.

Note

model parameters are based on distances in km.

Author(s)

Eric Archer <eric.archer@noaa.gov>

`destination` *Destination on Sphere or Ellipsoid*

Description

Calculates latitude and longitude of the destination along a sphere or ellipsoid.

Usage

```
destination(lat, lon, brng, distance, units = "nm", ellipsoid = datum(),
  radius = convert.distance(6371, "km", "nm"), type = "ellipsoid")
```

Arguments

lat,lon	numeric. The latitude and longitude of the coordinate in decimal degrees.
brng	numeric. The bearing, ranging from 0 to 360 degrees.
distance	numeric. The distance travelled, in units specified by units.
units	units of distance. Can be "km" (kilometers), "nm" (nautical miles), or "mi" (statute miles).
ellipsoid	ellipsoid model parameters as returned from a call to datum.
radius	numeric. Define the radius for type = "sphere". In units of units.
type	Character defining type of surface. Can be "sphere", "ellipsoid", "Vincenty", or partial match thereof (case-insensitive).

Value

latitude and longitude of destination.

Author(s)

Eric Archer <eric.archer@noaa.gov>

References

Ellipsoid code adapted from JavaScript by Larry Bogan http://www.go.ednet.ns.ca/~larry/bsc/jslatlng.html.
Vincenty code adapted from JavaScript by Chris Veness http://www.movable-type.co.uk/scripts/latlong-vincenty-direct.html Vincenty, T. 1975. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. Survey Review 22(176):88-93 http://www.ngs.noaa.gov/PUBS_LIB/inverse.pdf.

Examples

```
destination(32.87, -117.25, 262, 4174, units = "km", type = "sphere")
destination(32.87, -117.25, 262, 4174, units = "km", type = "ellipsoid")
destination(32.87, -117.25, 262, 4174, units = "km", type = "vincenty")
```

---

distance *Distance Between Coordinates*

Description

Calculates the distance between two coordinates using the Law of Cosines, Haversine, or Vincenty methods.

Usage

```
distance(lat1, lon1, lat2, lon2, radius = convert.distance(6371, "km", "nm"),
  units = "nm", ellipsoid = datum(), iter.limit = 20,
  method = "lawofcosines")
```

Arguments

lat1,lon1,lat2,lon2	The latitude and longitude of the first and second points in decimal degrees.
radius	radius of sphere.
units	units of distance. Can be "km" (kilometers), "nm" (nautical miles), or "mi" (statute miles).
ellipsoid	ellipsoid model parameters as returned from a call to datum.
iter.limit	An integer value defining the limit of iterations for Vincenty method.
method	Character defining the distance method to use. Can be "lawofcosines", "haversine", "vincenty", or any partial match thereof (case insensitive).

Author(s)

Eric Archer <eric.archer@noaa.gov>

References

Code adapted from JavaScript by Chris Veness http://www.movable-type.co.uk/scripts/latlong.html Vincenty, T. 1975. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. Survey Review 22(176):88-93 http://www.ngs.noaa.gov/PUBS_LIB/inverse.pdf.

Examples

```
# What is the distance from San Diego, CA to Honolulu, HI?
distance(32.87, -117.25, 21.35, -157.98, method = "lawofcosines")
distance(32.87, -117.25, 21.35, -157.98, method = "haversine")
distance(32.87, -117.25, 21.35, -157.98, method = "vincenty")
```

---

eiaMisc *Set of miscellaneous functions*

Description

Collection of miscellaneous utility and analytical functions

Details

Package:	eiaMisc
Type:	Package
Version:	1.0
Date:	2013-08-08
License:	GNU General Public License

Author(s)

Eric Archer<eric.archer@noaa.gov>

`geometric.mean` *Geometric Mean*

Description

Calculates the geometric mean of a vector.

Usage

```
geometric.mean(x)
```

Arguments

x	a numeric vector.

Author(s)

Eric Archer <eric.archer@noaa.gov>

Examples

```
x <- rlnorm(100)
mean(x)
median(x)
geometric.mean(x)
```

`harmonic.mean` *Harmonic Mean*

Description

Harmonic Mean

Usage

```
harmonic.mean(x)
```

Arguments

x	a numeric vector.

Author(s)

Eric Archer <eric.archer@noaa.gov>

Examples

```
x <- rlnorm(100)
mean(x)
median(x)
harmonic.mean(x)
```

---

invLogOdds *Odds Conversion*

Description

odds	converts probability to odds
logOdds	converts odds to log-odds
invOdds	converts odds to probability
invLogOdds	converts log-odds to odds

Usage

```
invLogOdds(x)

invOdds(x)

logOdds(x)

odds(x)
```

Arguments

x	a numeric vector of probabilities (0 to 1), odds (0 to Inf), or log.odds (-Inf to Inf).

Author(s)

Eric Archer <eric.archer@noaa.gov>

Examples

```
x <- sort(runif(10))
odds.df <- data.frame(x = x, odds = odds(x), logOdds = logOdds(x))
odds.df
```

```
invOdds(odds.df$odds)
invLogOdds(odds.df$logOdds)
```

`lat.lon.axes` *Latitude and Longitude axes*

Description

Add latitude and longitude axes to a map.

Usage

```
lat.lon.axes(lon.range, lat.range, n = 5, lon.n = n, lat.n = n)
```

Arguments

`lon.range,lat.range`	two-element vectors giving the minimum and maximum longitude and latitude.
`n,lon.n,lat.n`	the number of tick marks desired. Can be specified separately for longitude (`lon.n`) or latitude (`lat.n`). See `pretty` for more details.

Author(s)

Eric Archer <eric.archer@noaa.gov>

`mail` *Mail*

Description

Sends an email using the UNIX mail function.

Usage

```
mail(to, subject = "email from R", message = NULL, file = NULL)
```

Arguments

`to`	recipient of email.
`subject`	subject line of email.
`message`	email content.
`file`	file to be attached.

Author(s)

Eric Archer <eric.archer@noaa.gov>>

na.count *Count NAs*

Description

Counts NAs in an object.

Usage

```
na.count(x)
```

Arguments

x	a vector, data.frame, or matrix.

Author(s)

Eric Archer <eric.archer@noaa.gov>

Examples

```
x <- sample(c(1:10, NA), 30, replace = TRUE)
na.count(x)
x.df <- do.call(data.frame, lapply(1:4, function(i) sample(c(1:10, NA), 30, replace = TRUE)))
colnames(x.df) <- paste("X", 1:4, sep = "")
na.count(x.df)
```

normalize *Normalize a Numeric Vector*

Description

Normalize a numeric vector to have a mean of zero and a standard deviation of one.\

Usage

```
normalize(x)
```

Arguments

x	a numeric vector.

Value

a numeric vector of the same length as x.

Author(s)

Eric Archer <eric.archer@noaa.gov>

Examples

```
x <- runif(20, 50, 110)
x.norm <- normalize(x)
mean(x)
mean(x.norm)
sd(x)
sd(x.norm)
```

---

one.arg *One Argument*

Description

Does the function have just one argument?

Usage

```
one.arg(f)
```

Arguments

f	a function.

Author(s)

Eric Archer <eric.archer@noaa.gov>

Examples

```
one.arg(mean)
one.arg(one.arg)
```

row.col.page.fit *Number of Rows and Columns on Page*

Description

Return the number of rows and columns for n that best fits on a page of size width x height.

Usage

```
row.col.page.fit(n, width = 8.5, height = 11)
```

Arguments

n	number of items (e.g., plots) to fit on page.
width,height	dimensions of page.

Value

A vector listing the number of rows and columns to use.

Author(s)

Eric Archer <eric.archer@noaa.gov>

Examples

```
# 9 frames on US letter paper
row.col.page.fit(9)

# 9 frames on a square
row.col.page.fit(9, width = 10, height = 10)
```

sample.map *Plot Samples on World Map*

Description

Plot a set of samples on a world map.

Usage

```
sample.map(lat, lon, lat.range, lon.range, main = NULL, pch = 19,
  pt.cex = 1, col = "black", bg = col, n = 5, lon.n = n, lat.n = n)
```

Arguments

lon,lat	vectors giving the longitude and latitude of points to plot.
lon.range,lat.range	vectors giving the minimum and maximum longitude and latitude of the map.
main	main title for the plot.
pch	point symbol to use.
pt.cex	point size to use.
col	point color.
bg	background color of point.
n,lon.n,lat.n	the number of tick marks desired. Can be specified separately for longitude (lon.n) or latitude (lat.n). See pretty for more details.

Value

original par settings for mar and oma.

Author(s)

Eric Archer <eric.archer@noaa.gov>

See Also

lat.lon.axes

Examples

```
# Some random points around San Diego, CA
lat <- runif(30, 32.5, 33.3)
lon <- runif(30, -118.3, -117.5)
lat.range <- c(32.4, 33.6)
lon.range <- c(-118.6, -117)
sample.map(lat, lon, lat.range, lon.range)
```

---

scatterdens *Scatter Plot with Density Margins*

Description

Scatter Plot with Density Margins

Usage

```
scatterdens(x, y, dens.frac = 1/5, ...)

scatterhist(x, y, xlab = "", ylab = "", dens.frac = 1/5, ...)
```

Arguments

x,y	vectors of points to plot.
xlab,ylab	labels for x and y axes.
dens.frac	fraction of screen to be taken up by density plots on margins.
...	Arguments to be passed to plot.

Author(s)

Eric Archer <eric.archer@noaa.gov>

References

Original code by Ken Kleiman: http://sas-and-r.blogspot.co.uk/2011/06/example-841-scatterplot-with-marg
html

Examples

```
x <- rnorm(100)
y <- rlnorm(100)
op <- par(ask = TRUE)
scatterdens(x, y, xlab = "x", ylab = "y")
par(op)
```

---

uniform.test *Uniform Distribution Test*

Description

Tests whether a histogram is significantly different from a uniform distribution.

Usage

```
uniform.test(hist.output, B = NULL)
```

Arguments

hist.output	output from a call to hist.
B	number of replicates for chi-squared permutation.

Value

result of chi-squared test.

Author(s)

Eric Archer <eric.archer@noaa.gov>

Examples

```
x.unif <- runif(100)
uniform.test(hist(x.unif), B = 1000)
x.lnorm <- rlnorm(100)
uniform.test(hist(x.lnorm), B = 1000)
```

`zero.pad` *Zero Pad Integers*

Description

Zero Pad Integers

Usage

```
zero.pad(x)
```

Arguments

x	a vector of integers.

Author(s)

Eric Archer <eric.archer@noaa.gov>

Examples

```
x <- c(0, 1, 3, 4, 10)
zero.pad(x)
x <- c(x, 11, 12, 100, 1000)
zero.pad(x)
```

Index