Package 'geoelectrics'

December 30, 2015

Title 3D-Visualization of Geoelectric Resistivity Measurement Profiles

Version 0.1.5

Author Anja Kleebaum <anja.kleebaum@stmail.uni-bayreuth.de>

Maintainer Anja Kleebaum <anja.kleebaum@stmail.uni-bayreuth.de>

Description

Visualizes two-dimensional geoelectric resistivity measurement profiles in three dimensions.

Depends R (>= 2.10), lattice, rgl, fields

Imports methods

License GPL

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2015-12-30 16:54:37

R topics documented:

`calcRelativeCoords` *Calculate Relative Coordinates*

Description

Calculates relative coordinates (unity: meters) from GPS coordinates (either given in UTM or Gauss Krueger). This method is used when a profile set of many profiles is instantiated.

Usage

```
calcRelativeCoords(coords, minLat, minLon)
```

Arguments

`coords`	exact coordinates of a single Profile.
`minLat`	starting point (latititude).
`minLon`	starting point (longitude).

Value

data frame that contains the relative coordinates (latitude and longitude).

See Also

`ProfileSet-class`, `GpsCoordinates-class`

`GpsCoordinates-class` *GPS Coordinates Class*

Description

A class to handle gps coordinates.

Slots

`address` address of the gps ascii file

`exact` data frame that contains measured gps coordinates

`lm` linear model of the measured gps coordinates

`relative` relative coordinates

`lmRelative` linear model of relative coordinates

See Also

Profile-class, ProfileSet-class, heightAdjustment, calcRelativeCoords

Examples

```
# gpsCoordinates = new("GpsCoordinates",
#                      address = "../example/gps/p1.txt")
data(sinkhole)
sinkhole@profiles[[1]]@gpsCoordinates
sinkhole@profiles[[1]]@gpsCoordinates@address
sinkhole@profiles[[1]]@gpsCoordinates@exact
sinkhole@profiles[[1]]@gpsCoordinates@lm
sinkhole@profiles[[1]]@gpsCoordinates@relative
sinkhole@profiles[[1]]@gpsCoordinates@lmRelative
```

---

heightAdjustment *Adjust Profile Height*

Description

Adjusts the height of a single profile. GPS measurement heights might differ otherwise.

Usage

```
heightAdjustment(Profile, deltaMeter)
```

Arguments

Profile	a single Profile.
deltaMeter	positive or negative value.

See Also

GpsCoordinates-class, Profile-class

Examples

```
# p3 <- new("Profile",
#            title = "Profile 3",
#            xyzData =
#              new("XyzData",
#                  address = "../example/xyzFiles/p3_DipolDipol_S-N.xyz"),
#            rawData =
#              new("RawData",
#                  address = "../example/rawdata/p3_DipolDipol_S-N.dat"),
#            measurementType = "DipolDipol",
#            gpsCoordinates =
#              new("GpsCoordinates",
#                  address = "../example/gps/p3.txt"))
```

```
# p3 <- heightAdjustment(p3, -10)
```

`levelplotLegendLabel` *Levelplot Legend Label*

Description

Plots the label of the levelplot.

Usage

```
levelplotLegendLabel(legend.lab = "Resistivity",
  unit = expression(paste("[", Omega, "m]")))
```

Arguments

`legend.lab`	label (default: "Resistivity").
`unit`	unit (default: "Ohm*m").

See Also

`levelplotXyz`, `levelplotXyzHeight`, `levelplotRaw`

Examples

```
# data(sinkhole)

# levelplotRaw(sinkhole@profiles[[1]])
# levelplotLegendLabel()

# levelplotXyz(sinkhole@profiles[[2]])
# levelplotLegendLabel()

# levelplotXyzHeight(sinkhole@profiles[[3]])
# levelplotLegendLabel()
```

`levelplotRaw` *Levelplot of Raw Data*

Description

Plots raw data values without topography (height adjustment). The raw data values have not been inverted yet.

Usage

```
levelplotRaw(Profile, xlab = "Length [m]", ylab = "Depth [m]",
  main = paste(Profile@title, "without topography (raw data)"),
  col = colors, trafo = log, ...)
```

Arguments

`Profile`	profile.
`xlab`	label for x-axes.
`ylab`	label for y-axes.
`main`	title to be plotted.
`col`	vector of colors.
`trafo`	function to transform raw data values (default: log).
`...`	lattice levelplot arguments.

Examples

```
data(sinkhole)

levelplotRaw(sinkhole@profiles[[1]])
levelplotLegendLabel()

levelplotRaw(sinkhole@profiles[[2]])
levelplotLegendLabel()

levelplotRaw(sinkhole@profiles[[3]])
levelplotLegendLabel()
```

levelplotXyz *Levelplot of XYZ Data*

Description

Plots the interpolated resistance values of the xyz data without topography (height adjustment). The xyz values are obtained through inversion of the raw values.

Usage

```
levelplotXyz(Profile, xlab = "Length [m]", ylab = "Depth [m]",
  main = paste(Profile@title, "without topography"), col = colors,
  breaks = 18, trafo = log, backtrafo = exp, ...)
```

Arguments

Profile	profile.
xlab	label for x-axes.
ylab	label for y-axes.
main	title to be plotted.
col	vector of colors.
breaks	number of color breaks.
trafo	transformation to be done on data (default: log).
backtrafo	back transformation to plot correct labels (default: exp).
...	lattice levelplot arguments.

See Also

XyzData-class, levelplotXyzHeight, levelplotRaw

Examples

```
# data(sinkhole)

# levelplotXyz(sinkhole@profiles[[1]])
# levelplotLegendLabel()

# levelplotXyz(sinkhole@profiles[[2]])
# levelplotLegendLabel()

# levelplotXyz(sinkhole@profiles[[3]])
# levelplotLegendLabel()
```

levelplotXyzHeight *Levelplot of XYZ Data regarding Topography*

Description

Plots the interpolated resistance values of the xyz data after height adjustment. The xyz values are obtained through inversion of the raw values.

Usage

```
levelplotXyzHeight(Profile, xlab = "Length [m]", ylab = "Depth [m]",
  main = paste(Profile@title, "with topography"), col = colors,
  breaks = 18, trafo = log, backtrafo = exp, ...)
```

Arguments

Profile	profile.
xlab	label for x-axes.
ylab	label for y-axes.
main	title to be plotted.
col	vector of colors.
breaks	number of color breaks.
trafo	transformation to be done on data (default: log).
backtrafo	back transformation to plot correct labels (default: exp).
...	lattice levelplot arguments.

See Also

XyzData-class, levelplotXyz, levelplotRaw

Examples

```
# data(sinkhole)

# levelplotXyzHeight(sinkhole@profiles[[1]])
# levelplotLegendLabel()

# levelplotXyzHeight(sinkhole@profiles[[2]])
# levelplotLegendLabel()

# levelplotXyzHeight(sinkhole@profiles[[3]])
# levelplotLegendLabel()
```

myColorRamp *Maps color to resistivity value*

Description

Maps color to (resistivity) values. A minimum and maximum value can be specified.

Usage

```
myColorRamp(col, values, minData = min(values), maxData = max(values))
```

Arguments

col	Character vector of colors.
values	Numeric vector of values.
minData	Minimum value (default min(values)). All smaller values will assigned to the first color in vector col.
maxData	Maximum value (default max(values)). All higher values will assigned to the last color in vector col.

plot3dXyz *Plots Profiles 3D*

Description

Plots the interpolated resistance values of the xyz data for all profiles.

Usage

```
plot3dXyz(.Object, title = "", sub = "", xlab = "", ylab = "",
  zlab = "", minData = 0, maxData = 9999999, col = colors,
  trafo = log, psize = pointsize)

## S4 method for signature 'ProfileSet'
plot3dXyz(.Object, title = .Object@title, sub = "",
  xlab = "", ylab = "", zlab = "", minData = .Object@minData,
  maxData = .Object@maxData, col = colors, trafo = log,
  psize = pointsize)

## S4 method for signature 'Profile'
plot3dXyz(.Object, title = "", sub = "", xlab = "",
  ylab = "", zlab = "", minData = .Object@xyzData@minData,
  maxData = .Object@xyzData@maxData, col = colors, trafo = log,
  psize = pointsize)
```

Arguments

.Object	either an object of a single Profile or a ProfileSet.
title	title to be plotted.
sub	subtitle to be plotted.
xlab	label of the x-axes, e.g. length [m].
ylab	label of the y-axes, e.g. height above sea level [m].
zlab	label of the z-axes, e.g. length [m].
minData	mimimum value to adjust color bar.
maxData	maximum value to adjust color bar.
col	vector of colors.
trafo	transformation to be done on data (default: log).
psize	size of value points (default: 10).

See Also

Profile-class, ProfileSet-class, plotXyz, levelplotXyz

Examples

```
# data(sinkhole)

# plot3dXyz(sinkhole@profiles[[1]])
# plot3dXyz(sinkhole)
```

---

plotIntersect *Plot Profile Intersection*

Description

Plots resistivity against depth on and next to the intersection line between two profiles.

Usage

```
plotIntersect(.Object1, .Object2 = NULL,
  xlab = "Height above sea level [m]",
  ylab = expression(paste("Resistivity [", Omega, "m]")), main = "",
  trafo = log, backtrafo = exp, col = colors, pch = c(20, 20),
  type = "p", legendLoc = "bottomleft")

## S4 method for signature 'ProfileSet,ANY'
plotIntersect(.Object1, xlab, ylab, main, trafo,
  backtrafo, col, pch, type, legendLoc)

## S4 method for signature 'Profile,Profile'
```

```
plotIntersect(.Object1, .Object2 = NULL,
  xlab = "Height above sea level [m]",
  ylab = expression(paste("Resistivity [", Omega, "m]")), main = "",
  trafo = log, backtrafo = exp, col = colors, pch = c(20, 20),
  type = "p", legendLoc = "bottomleft")
```

Arguments

.Object1	either a single Profile or a ProfileSet.
.Object2	either a second single Profile or NULL if .Object1 is of type ProfileSet.
xlab	label of the x-axes, e.g. length [m].
ylab	label of the y-axes, e.g. height above sea level [m].
main	title to be plotted.
trafo	transformation to be done on data (default: log).
backtrafo	back transformation to plot correct labels (default: exp).
col	character vector of colors.
pch	numeric vector of plotting symbols.
type	plot type (default "p" for points). "b" for both points and lines, "c" for empty points joined by lines, "o" for overplotted points and lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.
legendLoc	legendLocation (default "bottomleft").

See Also

ProfileSet-class

Examples

```
# data(sinkhole)

# plotIntersect(sinkhole)
# plotIntersect(sinkhole@profiles[[1]], sinkhole@profiles[[2]])
```

---

plotLegend *Plots Legend*

Description

Plots the legend for resistivity values.

Usage

```
plotLegend(.Object, legend.lab = expression(paste("Resistivity [", Omega,
  " m]")), minData = 0, maxData = 999999, breaks = NULL,
  legend.line = 2.2, nlevel = 18, lab.breaks = c(), horizontal = T,
  col = colors, trafo = log, backtrafo = exp, ...)

## S4 method for signature 'ProfileSet'
plotLegend(.Object, legend.lab,
  minData = .Object@minData, maxData = .Object@maxData)

## S4 method for signature 'Profile'
plotLegend(.Object, legend.lab,
  minData = .Object@xyzData@minData, maxData = .Object@xyzData@maxData)
```

Arguments

.Object	either a single Profile or a ProfileSet.
legend.lab	label of legend (default: expression(paste("Resistivity [", Omega, "]"))).
minData	minimum value.
maxData	maximum value.
breaks	Break points in sorted order to indicate the intervals for assigning the colors. Note that if there are nlevel colors there should be (nlevel+1) breakpoints. If breaks is not specified (nlevel+1) equally spaced breaks are created where the first and last bin have their midpoints at the minimum and maximum values in z or at zlim.
legend.line	distance in units of character height (as in mtext) of the legend label from the color bar. Make this larger if the label collides with the color axis labels.
nlevel	number of color levels.
lab.breaks	number of breaks.
horizontal	If false legend will be a vertical strip on the right side. If true (default) the legend strip will be along the bottom.
col	vector of colors.
trafo	transformation to be done on data (default: log). For linear scale: function(x) x.
backtrafo	back transformation to plot correct labels (default: exp). For linear scale: function(x) x.
...	image.plot arguments.

See Also

Profile-class, ProfileSet-class, plot3dXyz,

Examples

```
# data(sinkhole)

# plotLegend(sinkhole)

# for linear scale:
# plotLegend(sinkhole@profiles[[1]],
#            trafo=function(x) x,
#            backtrafo=function(x) x,
#            minData=100, maxData=50000)
```

---

plotRaw *Plot Raw Data Points*

Description

Plots raw data points of a single profile (e.g. to show measurement gaps).

Usage

```
plotRaw(Profile, xlab = "Length [m]", ylab = "Depth [m]",
  main = paste(Profile@title, "without topography"), ...)
```

Arguments

Profile	profile.
xlab	label for x-axes.
ylab	label for y-axes.
main	title to be plotted. This can either be a single character or an integer code for one of a set of graphics symbols.
...	plot arguments (like pch, cex, col,...).

See Also

plotRawHeight, RawData-class

Examples

```
data(sinkhole)
plotRaw(sinkhole@profiles[[1]])
plotRaw(sinkhole@profiles[[2]])
plotRaw(sinkhole@profiles[[3]])
```

plotRawHeight *Plot Raw Data Points considering Topography*

Description

Plots raw data points of a single profile (e.g. to show measurement gaps). The topography is considered, i.e., heights are added to the measurement depth.

Usage

```
plotRawHeight(Profile, height = Profile@rawData@height, spline = TRUE,
  xlab = "Length [m]", ylab = "Depth [m]", main = paste(Profile@title,
  "without topography"), ...)
```

Arguments

Profile	profile.
height	topo data frame of distances and height. Use "Profile@xyzData@height" instead.
spline	if TRUE spline interpolation is conducted.
xlab	label for x-axes.
ylab	label for y-axes.
main	title to be plotted.
...	plot arguments (like pch, cex, col,...).

See Also

plotRaw, RawData-class

Examples

```
data(sinkhole)
plotRawHeight(sinkhole@profiles[[2]])
plotRawHeight(sinkhole@profiles[[2]], sinkhole@profiles[[2]]@xyzData@height)
```

plotXyz *Plot XYZ Points*

Description

Plots the interpolated points of the xyz data.

Usage

```
plotXyz(Profile, xlab = "Length [m]", ylab = "Depth [m]",
  main = paste(Profile@title, "without topography"), asp = 1, ...)
```

Arguments

Profile	profile.
xlab	label for x-axes.
ylab	label for y-axes.
main	title to be plotted.
asp	the y/x aspect ratio (default: 1).
...	plot parameters

See Also

Profile-class, plotXyzHeight, plot3dXyz, levelplotXyz, levelplotXyzHeight

Examples

```
data(sinkhole)

plotXyz(sinkhole@profiles[[1]])
```

plotXyzHeight *Plot XYZ Points considering Topography*

Description

Plots the interpolated points of the xyz data after height adjustment.

Usage

```
plotXyzHeight(Profile, xlab = "Length [m]", ylab = "Depth [m]",
  main = paste(Profile@title, "with topography"), asp = 1, ...)
```

Arguments

Profile	profile.
xlab	label for x-axes.
ylab	label for y-axes.
main	title to be plotted.
asp	the y/x aspect ratio (default: 1).
...	plot parameters.

See Also

Profile-class, plotXyz, plot3dXyz, levelplotXyz, levelplotXyzHeight

Examples

```
data(sinkhole)

plotXyzHeight(sinkhole@profiles[[1]])
```

Profile-class *Profile Class*

Description

A class to handle a single profile.

Slots

title title of the profile (e.g. Profile 1).

number index of the profile.

xyzData object of Xyz Data Class (XyzData-class).

rawData object of Raw Data Class (RawData-class).

measurementType type of measurement (e.g. Dipole Dipole, Wenner, ...).

gpsCoordinates object of GpsCoordinates Class (GpsCoordinates-class).

See Also

XyzData-class, RawData-class, GpsCoordinates-class, plot3dXyz

Examples

```
# p1 <- new("Profile",
#           title = "Profile 1",
#           xyzData =
#             new("XyzData",
#           rawData =
#             new("RawData",
#                 address = "../example/rawdata/p1_DipolDipol_SW-NE.dat"),
#           measurementType = "DipolDipol",
#           gpsCoordinates =
#             new("GpsCoordinates",
#                 address = "../example/gps/p1.txt"))
#
# p1@title
# p1@xyzData
# p1@rawData
# p1@measurementType
# p1@gpsCoordinates
#
# plot3dXyz(p1)
```

`ProfileSet-class` *Profile Set Class*

Description

A class to handle a collection of many profiles.

Slots

`title` title to plot

`profiles` list that contains objects of class Profile (`Profile-class`)

`minLat` minimum latitude value of all profiles

`minLon` minimum longitude value of all profiles

`minData` minimum data value of all profiles

`maxData` maximum data value of all profiles

See Also

`Profile-class`, `plot3dXyz`

Examples

```
# sinkhole <- new("ProfileSet",
#                 profiles = list(p1, p2, p3),
#                 title="Sinkhole")

data(sinkhole)
plot3dXyz(sinkhole)
```

RawData-class *Raw Data Class*

Description

A class to handle geoelectrics raw data.

Slots

`address` address of the raw data ascii file.

`seaLevel` data frame that contains raw data positions and resitance values.

`height` data frame that contains topography information (distance and height).

See Also

Profile-class, ProfileSet-class

Examples

```
# rawData = new("RawData",
#               address = "../example/rawdata/p1_DipolDipol_SW-NE.dat")

data(sinkhole)
sinkhole@profiles[[2]]@rawData
sinkhole@profiles[[2]]@rawData@address
sinkhole@profiles[[2]]@rawData@height
sinkhole@profiles[[2]]@rawData@seaLevel
```

sinkhole *Filled Sinkhole*

Description

Geoelectrics profiles measured at a filled sinkhole. This data set contains an object of the ProfileSet class.

Usage

```
# data(sinkhole)
# plot3dXyz(sinkhole)
# plotLegend(sinkhole)

# plotIntersect(sinkhole)

# levelplotXyz(sinkhole@profiles[[1]])
# levelplotLegendLabel()
```

```
# plotRaw(sinkhole@profiles[[2]])
# plotRawHeight(sinkhole@profiles[[2]])

# levelplotRaw(sinkhole@profiles[[2]])
# levelplotLegendLabel()
```

Format

Object of Profile Set class including three Profiles.

`XyzData-class` *XYZ Data Class*

Description

A class to handle xyz data. The software Res2DInv produces .xyz-files that contain the inverted resistance values. The xyz class parses .xyz files.

Slots

`address` address of the xyz ascii file

`seaLevel` data frame that contains positions and values withouth height adjustment

`heightAdaption` data frame that contains positions and values after height adjustment

`height` data frame that contains topography information (distances and heights). It is reconstructed from .xyz-file.

`minData` minimum value

`maxData` maximum value

See Also

`Profile-class`, `ProfileSet-class`, `plotXyz`, `plotXyzHeight`, `plot3dXyz`

Examples

```
# xyzData = new("XyzData",
#               address = "../example/xyzFiles/p1_DipolDipol_SW-NE.xyz"),

data(sinkhole)
sinkhole@profiles[[1]]@xyzData
sinkhole@profiles[[1]]@xyzData@seaLevel
sinkhole@profiles[[1]]@xyzData@heightAdaption
sinkhole@profiles[[1]]@xyzData@height
sinkhole@profiles[[1]]@xyzData@minData
sinkhole@profiles[[1]]@xyzData@maxData
```

Index