Package ‘RadioSonde’

February 19, 2015

Version 1.4

Date 2014-09-15

Title Tools for plotting skew-T diagrams and wind profiles

Author Doug Nychka, Eric Gilleland, Liangying Zhang, Tim Hoar

Maintainer Doug Nychka <nychka@ucar.edu>

Description RadioSonde is a collection of programs for reading and plotting SKEW-T,log p diagrams and wind profiles for data collected by radiosondes (the typical weather balloon-borne instrument), which we will call ``flights'', ``sondes'', or ``profiles'' throughout the associated documentation. The raw data files are in a common format that has a header followed by specific variables. Use ``help(ExampleSonde)'' for the full explanation of the data files.

License GPL (>= 2)

URL http://www.image.ucar.edu/Software/RadioSonde

Repository CRAN

Date/Publication 2014-09-17 19:37:44

NeedsCompilation no

R topics documented:

`ExampleSonde` *Example of radiosonde observations*

Description

The `ExampleSonde` data frame has 461 rows and 21 columns. The columns are different physical measurements made during the balloon's ascent. `ExampleSondeText` is an image of a text file that allows for testing the getsonde function and is an example of a format with metadata used to document a sounding.

These data are a sounding from a well known field experiment TOGA/COARE studying the larege scale dynamical relationship of the Tropical Pacific and the atmospshere. This particular sounding was launched January 17, 1993 17:12 GMT from Kavieng, New Guninea. To write out the text dataset to a sounding file just use:

```
data(ExampleSondeText) ;  write( ExampleSondeText, file="sondedata.txt")
```

Format

This data frame contains the following named components:

time	Time in seconds from the ballon's release, a numeric vector.
press	Pressure (mb), a numeric vector
temp	Dry-bulb Temperature (deg. C), a numeric vector
dewpt	Dew point temperaure (deg. C), a numeric vector
rh	Relative humidity (%), a numeric vector
uwind	East-West component of the wind, a numeric vector
vwind	North-South component of the wind, a numeric vector
wspd	wind speed (m/s), a numeric vector
dir	Wind direction, a numeric vector
dz	Ascension rate (m/s), a numeric vector
lon	Longitude, a numeric vector
lat	Latitude, a numeric vector
rng	range, a numeric vector
az	azimuth (angle along horizon), a numeric vector
alt	altitude (m), a numeric vector
qp	QC flag for pressure, a numeric vector
qt	QC flag for temperature, a numeric vector
qh	QC flag for humidity, a numeric vector
qu	QC flag for U component, a numeric vector
qv	QC flag for V component, a numeric vector
quv	QC flag for ascension rate, a numeric vector

Note

Quality Control (QC) flags are generated locally at JOSS based on either automated or visual checks made.

999.0 implies the QC information is missing,
1.0 implies datum seems physically reasonable (good),
2.0 implies datum questionable on a physical basis (maybe),
3.0 implies datum seems to be in error (bad),
4.0 implies datum interpolated (estimated),
9.0 implies datum missing in original file.

See Also

`getsonde` , `skewt.lines`, `skewt.points`, `plotsonde`

`getsonde` *read a radiosonde file to an R dataframe.*

Description

`getsonde` takes a common radiosonde data file and reads it in as an R dataframe.

Usage

```
getsonde(filename, datakey="------", varkey=" Time", unitkey="  sec")
```

Arguments

`filename` Full path and file name of radiosonde data.

`datakey` character string identifying the line **preceeding** the datastream. The string gets compared to the first N characters of each line, so all whitespace is important!

`varkey` character string identifying the line containing the variable names. The string gets compared to the first N characters of each line, so all whitespace is important!

`unitkey` character string identifying the line containing the units of the variables. The string gets compared to the first N characters of each line, so all whitespace is important!

Details

`getsonde` assumes the data is in a very common format for radiosonde data. The files are typically ASCII files with a header of unknown length stating such things as the location and time of the data collection etc. Appended to this is a table of data where each column is a particular quantity and each row corresponds to the time of the observation. This typically has a header identifying the quantity in each of the columns and is separated from the data by a particular character string. The location of this string in the file is crucial in deciphering the start of the datastream from the metadata. For much more about the file format, look at the example in '`RadioSonde/inst/exampleData/ExampleSonde.txt`'. To find the exact path and file name on your system try `system.file("exampleData", "ExampleSonde.txt", package = "RadioSonde")`.

Value

Returns a dataframe with the following items, if present in the input file. Note that each of these items can be present in any order in the input file.

`time`	time in seconds since weather balloon launched or recording begins.
`press`	Pressure in mb.
`temp`	Dry-bulb Temperature in degrees C.
`dewpt`	Dew point temperature in degrees C.
`rhum`	Relative Humidity (Percent).
`uwind`	East-West wind component (m/s).
`vwind`	North-South wind component (m/s).
`wspd`	Wind speed (m/s).
`wdir`	Wind direction (deg.).
`dz`	Ascension rate of weather balloon (m/s).
`lon`	Longitude of weather balloon.
`lat`	Latitude of weather balloon .
`rng`	Range (see **warning** below)
`az`	Azimuth of weather balloon from originating station (see **warning** below)
`alt`	Altitude of weather balloon (m).
`qp`	Quality Control (QC) flag for pressure (see note below)
`qt`	QC flag for temperature (see note below)
`qh`	QC flag for humidity (see note below)
`qu`	QC flag for U Component (see note below)
`qv`	QC flag for V Component (see note below)
`quv`	QC flag for Ascension rate (see note below)

If the `units` are avialable in the datafile, they are included in the dataframe as the attribute `units`.

Rather than throw away the original header information, which usually contains valuable metadata, it is included as another attribute: `metadata`.

Warning

The connotation of fields is not exactly standard. Different recording systems use these fields in different ways. Hence, they cannot be automatically interpreted without knowledge of the recording system.

Note

The Quality Control information (flags): qp, qt, qh, qu, qv, and quv are generated at JOSS are based on the automated or visual checks made. The JOSS QC flags are as follows:
99.0 means it is unchecked,
1.0 implies datum seems physically reasonable (good),
2.0 implies datum seems questionable on physical basis (maybe),
3.0 implies datum seems to be in error (bad),
4.0 implies datum is interpolated (estimated), and
9.0 implies datum was missing in original file (missing).

Author(s)

Doug Nychka,Eric Gilleland

See Also

plotsonde , skewt.points , skewt.lines

Examples

```
# Read a typical radiosonde (ASCII) datafile, headers and all.
# The datakey,varkey,and unitkey arguments come from examining
# the datafile. The whitespace is important.

# The Sonde text file is stored in this package as a text dataset (this is just
# to make it easier to include in the package.)
# First create a real text file to read in
  data( ExampleSondeText)
  write( ExampleSondeText, file="SondeFile.txt")

  sample.sonde <- getsonde("SondeFile.txt")
# NOTE this assumes that the default datakey, varkey and unitkey.
# it is best to check the file for these choices!
  attr(sample.sonde,"units")
  attr(sample.sonde,"metadata")

#  skewt plot of data
  plotsonde(sample.sonde,title="SondeFile.txt")

#
# Read the same radiosonde dataset, but ignore the units ...
#
  datakey  <- "------"
  varkey   <- " Time"
  sample.sonde <- getsonde("SondeFile.txt",datakey,varkey)
  attr(sample.sonde,"metadata")
```

plotsonde *Creates a SKEW-T, log p diagram for a radiosonde dataframe.*

Description

Creates a SKEW-T, log p diagram with dry-bulb temperature and dewpoint temperature traces versus (log) pressure. Optionally plots the vertical wind profile using wind barbs.

Usage

```
plotsonde(dataframe, skewT=TRUE, winds=FALSE, site="", title="",
          windplot=NULL, s=3, col=c(1, 2), ... )
```

Arguments

`dataframe`	R dataframe with at least the components `c( "press", "temp",  "dewpt")` corresponding to pressure (hPa), temperature (C) and dew point temperature (C).
`skewT`	Logical value. If false, will not plot the SKEW-T, log p diagram.
`winds`	Logical value. If false, will not plot the winds profile.
`site`	Optional character to add to plot title.
`title`	Title for plot (must be a character value).
`windplot`	4X1 numeric vector that gives the position of wind profile plot (next to SKEW-T diagram). Only used if it is desired to have this plot in a different position. Generally, the default NULL is used and the placement is chosen automatically.
`s`	Size of winds profile plot.
`col`	Usual plotting parameter.
`...`	Other optional plotting parameters.

Value

No value returned. Creates a skewt plot. Note that default is temperature horizontal scale is Farenheit.

Author(s)

Doug Nychka, Eric Gilleland,

References

1. Department of Defense, 1969, "USAF SKEW-T, log p DIAGRAM," DOD-WPC-9-16-1, Aeronautical Chart and Information Center, United States Air Force, St. Louis, Missouri 63118.

2. List, R.J. (editor), 1958, *Smithsonian Meteorological Tables*, Smithsonian Institute, Washington, D.C.

3. Nordquist, W.S., 1973, "Numerical Approximations of Selected Meteorological Parameters for Cloud Physics Problems," ECOM-5475, Atmospheric Sciences Laboratory, US Army Electronics Command, White Sands Missile Range, New Mexico 88002.

4. Stipanuk, G.S., 1973, "Algorithms for Generating a SKEW-T, log p Diagram and Computing Selected Meteorological Quantities," American Sciences Laboratory, US Army Electronics Command, White Sands Missile Range, New Mexico 88002.

5. `http://www.atd.ucar.edu/dir_off/tc_corr/index.html`

6. `http://weather.unisys.com/upper_air/skew/details.html`

See Also

`getsonde`, `skewt.axis` , `skewt.lines` , `skewt.points`

Examples

```
data(ExampleSonde)
plotsonde(ExampleSonde, winds=TRUE)
```

`plotwind` *Winds Profile Plot*

Description

Creates a wind profile in the standard atmospheric notation. Each full barb = 10 m/s, half barb = 5 m/s, and a filled triangle for 50 m/s. The maximum wind speed that can be plotted without amending the program is 65 m/s.

Usage

```
 plotwind(dataframe, size=5, ylim=c(1050, 100), legend=FALSE)
```

Arguments

`dataframe` Data frame for sounding data, must have components for wind speed `wspd`, wind direction `dir`, and pressure `press`.

`size` change the thickness of the plotted lines

`ylim` plot extent, in units of pressure

`legend` explains wind barb strategy

Details

the dataframe must have components for wind speed `wspd`, wind direction `dir`, and pressure `press`. Missing values may be coded as either NA or 999. and are not plotted.
The standard atmospheric wind sybmol is a vector of fixed length with barbs proportional to wind speed. A full barb for each 10 m/s, half barbs for 5 m/s and a triangular barb for 50 m/s.

Value

None – creates a plot.

Author(s)

Doug Nychka, Eric Gilleland

See Also

getsonde, plotsonde

Examples

```
# Example 1:
data(ExampleSonde)
plotwind(ExampleSonde)
# Example 2:
plotwind(ExampleSonde, size = 7, legend=TRUE)
```

PWV *Precipitable Water Vapor*

Description

Calculates precipitable water (in mm) up to minp (minimum pressure) using dew-point temperature (td) and temperature (temp) (both in degrees C).

Usage

```
PWV(sonde = NULL, press, td, temp, minp = 400)
```

Arguments

sonde	Sonde data read in using getsonde function.
press	Pressure (only used if sonde is NULL)
td	Dew-point temperature (in deg C) (only used if sonde is NULL)
temp	Temperature (in deg C) (only used if sonde is NULL)
minp	Minimum Pressure

Value

single numeric value is returned.

Author(s)

Junhong Wang

See Also

getsonde, `td.to.q`

Examples

```
# Example 1

data(ExampleSonde)
######
# OK, now find precipitable water vapor.

PWV(ExampleSonde)
```

`skewt.axis` *Draws a SKEW-T, log p axis.*

Description

Draws a SKEW-T, log p axis. This is the standard axis for interpreting atmospheric sounding profiles like those collected by the instrument carried aloft by a weather balloon (*radiosonde*). Use `skewt.lines` or `skewt.points` to layer information on top of the skewt axis.

Usage

```
skewt.axis(BROWN="brown", GREEN="green", redo=FALSE, ... )
```

Arguments

`BROWN` Color of lines for both Temperature (solid skewed) and Pressure (dashed horizontal)

`GREEN` Color of lines for dry adiabats (solid) and constant mixing ratio (dashed)

`redo` flag to generate the adiabats, should `skewt.data` become corrupt. The adiabats are the result of an iterative process and to make smooth curves, you need a lot of points. Hence this is time-consuming, so [FALSE] is the preferred value.

`...` The usual `plot` parameters.

Details

Radiosondes record temperature, humidity and winds. They can be lifted by weather balloons, dropped from aircraft, there is even something called a glidersonde. The data collected by radiosondes are plotted versus pressure/height to give details on the vertical structure of the atmosphere. The type of plot is called a SKEW-T, log p diagram.
'`skewt.axis`' creates the traditional axis for a SKEW-T, log p diagram, including moist and dry adiabats, etc ...
Generating the necessary plot data is time-consuming, so there is an option to read from a specific dataset that Doug will change.

Value

Returns the `par()$plt` values, which are used by `plotsonde` if you choose to plot the wind profile adjacent to the skewt axis.

Author(s)

Doug Nychka

See Also

`getsonde`, `plotsonde`, `skewt.lines`, `skewt.points`

Examples

```
# draw a background, then
#    draw the temperature (with a solid line) in color 6
#    draw the dewpoint in color 7
#    overlay the temperature observations in a different color
#    you get the point ...
#
data(ExampleSonde)
skewt.axis( mar=c(5.1, 1.1, 2.1, 5.1) )
skewt.lines( ExampleSonde$temp,  ExampleSonde$press, col = 6)
skewt.lines( ExampleSonde$dewpt, ExampleSonde$press, col = 7)
skewt.points(ExampleSonde$temp,  ExampleSonde$press, col = 3)
skewt.points(ExampleSonde$dewpt, ExampleSonde$press, col = 4)
#
# Changing the moist adiabats: you must edit the \code{skewt.axis} function
# directly and then capture the output in \code{skewt.data} to be used in
# subsequent calls.
skewt.data <- skewt.axis(redo=TRUE)
skewt.axis()
skewt.axis()
```

`skewt.lines` *Overlays data on a SKEW-T, log p axis*

Description

Overlays observations as lines on a SKEW-T, log p axis (as created by `skewt.axis`).

Usage

```
skewt.lines(temp, pressure, ...)
```

Arguments

`temp`	Temperature in degrees C.
`pressure`	Pressure in millibars
`...`	Any graphical arguments

Details

skewt.lines overlays observations on a SKEW-T, log p axis

See Also

skewt.points, skewt.axis, plotsonde

Examples

```
# draw a background, then
#    draw the temperature (with a solid line) in color 6
#    draw the dewpoint in color 7
#    overlay the temperature observations in a different color
#    you get the point ...

data(ExampleSonde)
skewt.axis()
skewt.lines( ExampleSonde$temp,  ExampleSonde$press, col = 6)
skewt.lines( ExampleSonde$dewpt, ExampleSonde$press, col = 7)
skewt.points(ExampleSonde$temp,  ExampleSonde$press, col = 3)
skewt.points(ExampleSonde$dewpt, ExampleSonde$press, col = 4)
```

skewt.points *Overlays data on a SKEW-T, log p axis*

Description

Overlays observations on a SKEW-T, log p axis (as created by skewt.axis).

Usage

```
skewt.points(temp, pressure, ...)
```

Arguments

temp	Temperature in degrees C.
pressure	Pressure in millibars
...	Any graphical arguments

Details

skewt.points overlays observations on a SKEW-T, log p axis

See Also

skewt.axis, skewt.lines, plotsonde

Examples

```
# draw a background, then
#    draw the temperature (with a solid line) in color 6
#    draw the dewpoint in color 7
#    overlay the temperature observations in a different color
#    you get the point ...
data(ExampleSonde)
skewt.axis()
skewt.lines( ExampleSonde$temp,  ExampleSonde$press, col = 6)
skewt.lines( ExampleSonde$dewpt, ExampleSonde$press, col = 7)
skewt.points(ExampleSonde$temp,  ExampleSonde$press, col = 3)
skewt.points(ExampleSonde$dewpt, ExampleSonde$press, col = 4)
```

station.symbol *Adds a meteorological surface station annotation to a plot.*

Description

Adds a meteorological surface station annotation at the given coordinates. The annotation includes speed and direction of the wind, temperature, pressure, dewpoint ... to a given plot.

Usage

```
station.symbol(cx, cy, direction=0, speed=NA, fill=rep(0,length(cx)), temp=NA,
                    press=NA, dewpt=NA, circle=TRUE, cex=1)
```

Arguments

`cx` x coordinate for location of the annotation.

`cy` y coordinate for the annotation.

`direction` Wind direction.

`speed` Wind speed.

`fill` Fill for visibility: 0 = clear skies, 1 = 25% obscured, 2 = 50% obscured, 3 = 75% obscured, 4 = no visibility

`temp` Temperature value to plot symbol (must be a scalar). If NULL then no temperature value will be annotated.

`press` Pressure value to plot symbol (must be a scalar). If NULL then no pressure value will be annotated.

`dewpt` Dewpoint Temperature value to plot symbol (must be a scalar). If NULL then no dewpoint value will be annotated.

`circle` If TRUE, will plot the usual station symbol with a circle at its base.

`cex` Usual plotting parameter.

Value

Adds to a plot.

Author(s)

Doug Nychka, Eric Gilleland

See Also

plotsonde, plotwind, skewt.axis

Examples

```
plot(0:1, 0:1, type="n")
station.symbol(0.5, 0.5, direction=130, speed=30, fill=3,
               temp=31, press=987, dewpt=26, cex=5)
```

Index