Package 'nixmass'

October 30, 2026

Title Snow Water Equivalent Modeling with the 'Delta.snow' and 'HS2SWE' Models and Empirical Regression Models

Version 1.4.0

Date 2026-10-29

Maintainer Harald Schellander <harald.schellander@geosphere.at>

Description Snow water equivalent is modeled with the process based models 'delta.snow' and 'HS2SWE' and empirical regression, which use relationships between density and diverse at-site parameters. The methods are described in Winkler et al. (2021) <doi:10.5194/hess-25-1165-2021>, Magnusson et al. (2025) <doi:10.1016/j.coldregions.2025.104435>, Guyennon et al. (2019) <doi:10.1016/j.coldregions.2019.102859>, Pistocchi (2016) <doi:10.1016/j.ejrh.2016.03.004>, Jonas et al. (2009) <doi:10.1016/j.jhydrol.2009.09.021> and Sturm et al. (2010) <doi:10.1175/2010JHM1202.1>.

License GPL-3

URL https://haraldschellander.github.io/nixmass/

Depends ggplot2, graphics, grDevices, R (>= 3.5.0), stats

Imports colorspace, dplyr, lubridate, tidyr, utils, zoo

Suggests spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Harald Schellander [aut, cre, cph],
Michael Winkler [ctb, cph],
Jakob Werkgarner [ctb]

Repository CRAN

Date/Publication 2026-10-30 18:20:02 UTC

Contents

hs2swe 2
hsdata 4
nixmass 5
plot.nixmass 7
summary.nixmass 8
swe.delta.snow 9
swe.gu19 12
swe.jo09 13
swe.pi16 14
swe.st10 15

Index **17**

`hs2swe` *A model which translates snow depth observations to snow water equivalents*

Description

The HS2SWE model is based on similar principles as the `delta.snow` model. It is described in the following publication: Magnusson, J., B. Cluzet., L. Quéno, R. Mott, M. Oberrauch, G. Mazzotti, C. Marty, T. Jonas; 2025; Evaluating methods to estimate the water equivalent of new snow from daily snow depth recordings; Cold Regions Science and Technology, 233; 10.1016/j.coldregions.2025.104435.

Usage

```
hs2swe(
  data,
  RhoNew = 113.7,
  RhoMax = 571.6,
  SnoTemp = -0,
  Visc = 60510000,
  DQMultInc = 0.1,
  DQMultMax = 5,
  HsAcc = 2,
  c1 = 2.8e-06,
  c2 = 0.042,
  c3 = 0.046,
  c4 = 0.081,
  c5 = 0.018,
  g = 9.81,
  dt = 86400
)
```

Arguments

data	A data.frame with at least two columns named date and hs. They should contain date and corresponding daily observations of snow depth $hs \geq 0$ measured at one site. The unit must be meters (m). No gaps or NA are allowed. Dates must be either of class 'character', 'Date' or 'POSIXct' and given in the format YYYY-MM-DD.
RhoNew	The density of new snow in kg/m³. Default is 113.7 kg/m³.
RhoMax	The maximum density of snow in kg/m³. Default is 571.6 kg/m³.
SnoTemp	The temperature of the snow in °C. Default is -0.000 °C.
Visc	The viscosity of the snow in kg/m²/s. Default is 6.051e7 kg/m²/s.
DQMultInc	The increment of the densification rate. Default is 0.1.
DQMultMax	The maximum densification rate. Default is 5.
HsAcc	The threshold for new snow fall in cm. Default is 2 cm.
c1	The first coefficient for the densification rate. Default is 2.8e-6.
c2	The second coefficient for the densification rate. Default is 0.042.
c3	The third coefficient for the densification rate. Default is 0.046.
c4	The fourth coefficient for the densification rate. Default is 0.081.
c5	The fifth coefficient for the densification rate. Default is 0.018.
g	The gravitational acceleration in m/s². Default is 9.81 m/s².
dt	The time step in seconds. Default is 86400 seconds (1 day).

Details

The model is also available as matlab function, python package or as R function that accepts a slightly different input from https://github.com/oshd-slf/HS2SWE

Value

A numeric vector of simulated SWE in mm.

Examples

```
data(hsdata, package = "nixmass")

swe_deltasnow <- swe.delta.snow(hsdata)
swe_hs2swe <- hs2swe(hsdata)
plot(seq_along(hsdata$date), swe_deltasnow, type = "l", ylab = "SWE (mm) / hs (cm)", xlab = "day")
lines(seq_along(hsdata$date), swe_hs2swe, type = "l", col = "red")
lines(seq_along(hsdata$date), hsdata$hs * 100, type = "l", lty = 2, col = "grey30")
legend("topleft", legend = c("deltaSNOW", "HS2SWE", "HS"),
 col = c("black", "red", "grey30"), lty = c(1, 1, 2))
```

hsdata *Daily snow depth data for a northern alpine station*

Description

Gapless daily snow depth observations for a winter season from 1.8. - 31.7. from a station situated in the northern Eastern Alps at an altitude of 600 m. For anonymization the years are intentionally set to 1900 - 1901. This data series is free of gaps with a minimum of 0 and a maximum of 1.3 meters. It is intended to be used as is as input data for the package `nixmass` to calculate snow water equivalent and bulk snow density with the delta.snow method and several empirical regression models from the literature.

Usage

```
data(hsdata)
```

Format

A 'data.frame' named `data` with columns `date` and `hs`.

date The date column contains character strings of the format "YYYY-MM-DD" and is of class `character`.

hs The hs column holds daily observed snow depths in meters and is of class `numeric`.

Examples

```
## Load example data
data("hsdata")

## explore dataset
head(hsdata)
plot(hsdata$hs, type="o")

## compute snow water equivalents
o <- nixmass(hsdata, model="delta.snow")
plot(o)

o1 <- nixmass(hsdata, alt=600, region.jo09 = 6, region.gu19 = "central",
snowclass.st10 = "alpine", verbose = FALSE)
plot(o1)
summary(o1)
```

nixmass *SWE modeling with the delta.snow process based model and several empirical regression models.*

Description

Snow Water Equivalent (SWE) is modeled either exclusively from daily snow depth changes or statistically, depending on snow depth, elevation, date and climate class.

Usage

```
nixmass(
  data,
 model = c("delta.snow", "delta.snow.dyn_rho_max", "hs2swe", "jo09", "pi16", "st10",
    "gu19"),
  alt,
  region.jo09,
  region.gu19,
  snowclass.st10,
  layers = FALSE,
  strict_mode = TRUE,
  verbose = FALSE
)
```

Arguments

data	A data.frame with at least two columns named date and hs. They should contain date and corresponding daily observations of snow depth $hs \geq 0$ measured at one site. The unit must be meters (m). No gaps or NA are allowed. Dates must be either of class 'character', 'Date' or 'POSIXct' and given in the format YYYY-MM-DD. No sub-daily resolution is allowed at the moment (see details).
model	Defines model for SWE computation. Can be one, several or all of 'delta.snow', 'delta.snow.dyn_rho_max', 'hs2swe', 'jo09', 'pi16', 'st10', 'gu19'. If no model is given, 'delta.snow' will be taken.
alt	Must be given in meters if one of model is 'jo09'. Ignored otherwise.
region.jo09	Must be given if one of model is 'jo09', ignored otherwise. This must be an integer number between 1 and 7 of the Swiss region where the station belongs to, according to Fig. 1 in the original reference.
region.gu19	If model contains 'gu19' this must be one of 'italy', 'southwest', 'central' or 'southeast' as described in the original reference. Ignored if model is not 'gu19'.
snowclass.st10	Must be given if one of model is 'st10'. Must be one of the following character strings: 'alpine', 'maritime', 'prairie', 'tundra', 'taiga' as outlined in the original reference. Ignored if model is not 'st10'.
layers	Logical. Should parameters snow depth, swe and age be returned layerwise?

strict_mode	Logical. If 'TRUE', the function checks if the data is strictly regular and if the snow depth series starts with zero.
verbose	Logical. Should additional information be given during runtime?

Details

This function is a wrapper for the simulation of SWE with different models. The process based model delta.snow can be chosen in its original formulation (Winkler et al., 2021) and with a dynamically increasing maximum bulk snow density (Schroeder et al., 2024; Werkgarner, 2026). The hs2swe model is an alternative formulation of the same physical concept used in delta.snow (Magnusson et al., 2025). For a detailed discussion of the differences between the delta.snow and HS2SWE models see Werkgarner (2026). Some empirical regression models can also be chosen: Jonas, Pistocchi, Sturm and Guyennon. For the 'delta.snow' and 'hs2swe' models and the ones of 'Pistocchi' and 'Guyennon', the needed parameters and coefficients from the original references are set as default. They can however be changed according to results from other datasets. For the other models of 'Jonas' and 'Sturm' regression coefficients are fixed.

Computation is quite fast and there does not exist any restriction regarding the length of the data. However, if many years have to be modeled at once, it is recommended to split the computation into single years.

Value

A list of class nixmass with components:

swe	Contains a list of numerical vectors. Each entry refers to SWE values computed with the selected model(s).
date	Vector of date strings in the input class of format YYYY-MM-DD.
hs	Vector of given snow depth values used to compute SWE.

Author(s)

Harald Schellander, Michael Winkler

References

Guyennon, N., Valt, M., Salerno, F., Petrangeli, A., Romano, E. (2019) 'Estimating the snow water equivalent from snow depth measurements in the Italian Alps', Cold Regions Science and Technology. Elsevier, 167 (August), p. 102859. doi: 10.1016/j.coldregions.2019.102859.

Jonas, T., Marty, C. and Magnusson, J. (2009) "Estimating the snow water equivalent from snow depth measurements in the Swiss Alps", Journal of Hydrology, 378(1 - 2), pp. 161 - 167. doi: 10.1016/j.jhydrol.2009.09.021.

Magnusson, J., Cluzet, B., Quéno, L., Mott, R., Oberrauch, M., Mazzotti, G., Marty, C., Jonas, T. (2025) "Evaluating methods to estimate the water equivalent of new snow from daily snow depth recordings", Cold Regions Science and Technology, 233, 104435. doi: 10.1016/j.coldregions.2025.104435.

Pistocchi, A. (2016) "Simple estimation of snow density in an Alpine region", Journal of Hydrology:

Regional Studies. Elsevier B.V., 6(Supplement C), pp. 82 - 89. doi: 10.1016/j.ejrh.2016.03.004.

Schroeder, M. et al. (2024) "CONTINUOUS SNOW WATER EQUIVALENT MONITORING ON GLACIERS USING COSMIC RAY NEUTRON SENSOR TECHNOLOGY A CASE STUDY ON HINTEREISFERNER, AUSTRIA", Proceedings: International Snow Science Workshop 2024, Tromsø, Norway, 1107 - 1114

Sturm, M. et al. (2010) "Estimating Snow Water Equivalent Using Snow Depth Data and Climate Classes", Journal of Hydrometeorology, 11(6), pp. 1380 - 1394. doi: 10.1175/2010JHM1202.1.

Werkgarner, J. (2026): Improving the delta.snow model, Master's Thesis, University of Innsbruck.

Winkler, M., Schellander, H., and Gruber, S. (2021): Snow water equivalents exclusively from snow depths and their temporal changes: the delta.snow model, Hydrol. Earth Syst. Sci., 25, 1165-1187, doi: 10.5194/hess-25-1165-2021.

Examples

```
# Load example data with realistic snow depth values
# from a station at 600 meters in the northern Alps
# Note that the winter season is set to an arbitrary date
# to mask its origin
data("hsdata")
o <- nixmass(hsdata, model="delta.snow",verbose=TRUE)
plot(o)

o1 <- nixmass(hsdata, alt=600, region.jo09=6, region.gu19 = "central",
              snowclass.st10 = "alpine", verbose = FALSE)
plot(o1)
summary(o1)

swe <- nixmass(hsdata, alt = 1000, region.jo09=1, snowclass.st10 = "tundra", region.gu19 = "italy")
summary(swe)
```

---

plot.nixmass *Plot modeled SWE values of a nixmass object.*

Description

Plot modeled SWE values of a nixmass object.

Usage

```
## S3 method for class 'nixmass'
plot(x, title = NULL, density = FALSE, ...)
```

Arguments

x	nixmass object.
title	Main plot title.
density	Shall a density plot be created?
...	Further graphical parameters may also be supplied as arguments. See plot.

Value

Does not return anything. A plot is produced.

Examples

```
data("hsdata")
plot(nixmass(hsdata, model = "delta.snow"))
plot(nixmass(hsdata, model = "delta.snow.dyn_rho_max", layers = TRUE), density = TRUE)
```

---

summary.nixmass *Print summary of a nixmass object.*

Description

Print summary of a nixmass object.

Usage

```
## S3 method for class 'nixmass'
summary(object, ...)
```

Arguments

object	A nixmass object.
...	Additional arguments affecting the summary produced.

Value

Summary information of SWE values calculated with selected models is printed to the screen.

Examples

```
data("hsdata")
n <- nixmass(hsdata, model = c("delta.snow", "pi16"))
summary(n)
```

swe.delta.snow *SWE modeling from daily snow depth differences*

Description

Model daily values of Snow Water Equivalent solely from daily differences of snow depth.

swe.delta.snow computes SWE solely from daily changes of snow depth at an observation site. Compression of a snow layer without additional load on top is computed on the basis of Sturm and Holmgren (1998), who regard snow as a viscous fluid:

$$\rho_i(t_{i+1}) = \rho_i(t_i) * (1 + (SWE * g)/\eta_0 * exp^{-k_2 * \rho_i(t_i)})$$

with $\rho_i(t_{i+1}) and \rho_i(t_i)$ being tomorrow's and today's respective density of layer i, the gravitational acceleration $g = 9.8ms^{-2}$, viscosity η_0 (Pa) and factor $k2[m^3kg^{-1}]$, determining the importance of today's for tomorrow's density.

Usage

```
swe.delta.snow(
  data,
  model_opts = list(),
  dyn_rho_max = TRUE,
  layers = FALSE,
  strict_mode = TRUE,
  verbose = FALSE
)
```

Arguments

data	A data.frame with at least two columns named date and hs. They should contain date and corresponding daily observations of snow depth $hs \geq 0$ measured at one site. The unit must be meters (m). No gaps or NA are allowed. Dates must be either of class character, Date or POSIXct and given in the format YYYY-MM-DD. No sub-daily resolution is allowed at the moment (see details). Note that hs has to start with zero.
model_opts	An empty list which can be populated with model coefficients specific to the original delta.snow model (Winkler et al., 2021) or a version, where the maximum layer and bulk snow densities are allowed to vary with age (see details).
dyn_rho_max	Logical. If TRUE, the maximum bulk snow density is allowed to vary with layer age (see details). If FALSE, the original delta.snow model is used.
layers	Should parameters snow depth, swe and age be returned layerwise? Can be TRUE or FALSE.
strict_mode	Logical. If TRUE, the function checks if the data is strictly regular and if the snow depth series starts with zero.
verbose	Should additional information be given during runtime? Can be TRUE or FALSE.

Details

If dyn_rho_max=TRUE, the bulk snow density varies with layer age. As activation function, atan is used, where the S-curve symmetrically transitions from the lower to the upper density bound. In that case, model_opts are extended by a lower density bound rho_l, an upper density bound rho_h, a slope sigma and a midpoint mu, which together with the other parameters have been found via the improved optimization procedure of Werkgarner (2026). There, all ten parameters were optimized simultaneously (Nelder-Mead) against daily SWE simulated by SNOWPACK at eight alpine stations, using the Kling-Gupta efficiency of SWE as objective. Among several calibration datasets and objective functions, this set performed best in an independent validation against SWE and new-snow water equivalent observations. Altering one or more of the model parameters can be done with the option model_opts, which must provide a named list with the altered parameters. Be aware that altering only one parameter may have negative effects on model accuracy, as during an optimization procedure, all parameters are balanced to yield a minimum loss.

The following model coefficients must be provided:

dyn_rho_max=FALSE:

- rho.max Maximum density of an individual snow layer produced by the delta.snow model (kg/m3), $rho.max > 0$. Default is rho.max=401.2588 kg/m3.
- rho.null Fresh snow density for a newly created layer (kg/m3), $rho.null > 0$. Currently optimized for daily snow depth observations. Default is rho.null=81.19417 kg/m3.
- c.ov Overburden factor due to fresh snow (-), $c.ov > 0$. Default is c.ov=0.0005104722.
- k.ov Defines the impact of the individual layer density on the compaction due to overburden (-), $k.ov \in [0, 1]$. Default is k.ov=0.37856737.
- k Exponent of the exponential-law compaction (m3/kg), $k > 0$. Default is k=0.02993175 m3/kg.
- tau Uncertainty bound (m), $tau > 0$. Default is tau=0.02362476 m.
- eta.null Effective compactive viscosity of snow for "zero-density" (Pa s). Default is eta.null=8523356 (Pa s).
- timestep Timestep between snow depth observations in hours. Default is 24 hours, i.e. daily snow depth observations. No sub-daily values are allowed at the moment (see details).

dyn_rho_max=TRUE:

Instead of a constant coefficient for rho.max, these four parameters describe the smooth S-curve approximated by the atan trigonometric function.

- sigma Steepness or slope of atan at its midpoint mu, (-), $sigma > 0$. Default is sigma=0.0332.
- mu Central midpoint in days, where the steepest transition occurs (days), $mu > 0$. Default mu=86.2 days.
- rho_h Upper density bound for a single layer and the whole snow pack (kg/m3), $rho_h > 0$. Default is rho_h=688 kg/m3.
- rho_l Lower density bound for a single layer and the whole snow pack, where the transition begins (kg/m3), $rho_l > 0$. Default is rho_l=265 kg/m3.

All other coefficients are needed as well. Note that they differ from the constant rho_max version:

- `rho.null` Fresh snow density for a newly created layer (kg/m3), $rho.null > 0$. Default is rho.null=107 kg/m3.
- `c.ov` Overburden factor due to fresh snow (-), $c.ov > 0$. Default is c.ov=0.000585.
- `k.ov` Defines the impact of the individual layer density on the compaction due to overburden (-), $k.ov \in [0, 1]$. Default is k.ov=0.398.
- `k` Exponent of the exponential-law compaction (m3/kg), $k > 0$. Default is k=0.0285.
- `tau` Uncertainty bound (m), $tau > 0$. Default is tau=0.0201 m.
- `eta.null` Effective compactive viscosity of snow for "zero-density" (Pa s). Default is eta.null=8900212 (Pa s).
- `timestep` Timestep between snow depth observations in hours. Default is 24 hours, i.e. daily snow depth observations.

The easiest way to call the original delta.snow model is `swe.delta.snow(hsdata, dyn_rho_max = FALSE)`. Note that parameters intrinsic to the dynamic density model provided with the original model are silently ignored.

In principle, the model is able to cope with a sub-daily temporal resolution, e.g. hourly snow depth observations. However, the model was fitted to daily observations, and the model parameter `rho.null` reflects that. In other words, if the observation frequency changes, `rho.null` should change as well. Currently, no sub-daily resolution is allowed.

Value

If `layers=FALSE`, a vector with daily SWE values in mm. If `layers=TRUE`, a list with layerwise matrices of the parameters h (snow depth), swe and age is returned in addition to the SWE vector. The matrix rows correspond to `layers`, columns correspond to `dates`. swe is in mm, h in m and age in days.

Author(s)

Harald Schellander, Michael Winkler

References

Gruber, S. (2014) "Modelling snow water equivalent based on daily snow depths", Master's Thesis, Institute for Atmospheric and Cryospheric Sciences, University of Innsbruck.

Martinec, J., Rango, A. (1991) "Indirect evaluation of snow reserves in mountain basins". Snow, Hydrology and Forests in High Alpine Areas. pp. 111-120.

Schroeder, M. et al. (2024) "CONTINUOUS SNOW WATER EQUIVALENT MONITORING ON GLACIERS USING COSMIC RAY NEUTRON SENSOR TECHNOLOGY A CASE STUDY ON HINTEREISFERNER, AUSTRIA", Proceedings: International Snow Science Workshop 2024, Tromsø, Norway, 1107 - 1114

Sturm, M., Holmgren, J. (1998) "Differences in compaction behavior of three climate classes of snow". Annals of Glaciology 26, 125-130.

Werkgarner, J. (2026): Improving the delta.snow model, Master's Thesis, University of Innsbruck.

Winkler, M., Schellander, H., and Gruber, S. (2021): Snow water equivalents exclusively from snow depths and their temporal changes: the delta.snow model, Hydrol. Earth Syst. Sci., 25, 1165-1187, doi: 10.5194/hess-25-1165-2021.

Examples

```
data(hsdata, package = "nixmass")

swe_dyn <- swe.delta.snow(hsdata)
swe <- swe.delta.snow(hsdata, dyn_rho_max = FALSE)
plot(seq_along(hsdata$date), swe_dyn, type = "l", ylab = "SWE (mm) / hs (cm)", xlab = "day")
lines(seq_along(hsdata$date), swe, type = "l", col = "red")
lines(seq_along(hsdata$date), hsdata$hs * 100, type = "l", lty = 2, col = "grey30")
legend(title = "delta.snow", "topleft", legend = c("SWE dyn", "SWE", "HS"),
 col = c("black", "red", "grey30"), lty = c(1, 1, 2))
```

---

swe.gu19 *Statistical SWE modeling based on a quadratic dependance on the day-of-year*

Description

This model parameterizes bulk snow density with day-of-the-year as the only input similar to swe.pi16 but adds a quadratic dependance. It was calibrated for the regions of the whole Italian alps, and the subregions South-West, Central and South-East. By setting the coefficients of the empirical regression it can however be used with results from other datasets.

Usage

```
swe.gu19(data, region.gu19, n0 = NA, n1 = NA, n2 = NA)
```

Arguments

data	A data.frame of daily observations with two columns named *date* and *hs* referring to day and snow depth at that day. The date column can be of class 'character', 'Date' or 'POSIXct' with the format YYYY-MM-DD. The hs column must be snow depth values ≥ 0 in m.
region.gu19	Must be one of the italian subalpine regions *italy*, *southwest*, *central* or *southeast*, defined in the original reference (see details), or *myregion*, in which case the coefficients n0, n1 and n2 have to be set.
n0	Intercept of an empirical regression between densities and the day-of-year (see details).
n1	Slope of an empirical regression between densities and the day-of-year (see details).
n2	Quadratic dependence of an empirical regression between densities and the day-of-year (see details).

Details

swe.gu19 Similar to the model of Pistocchi (2016), this function uses only the day-of-year (DOY) as parameterization for bulk snow density and hence SWE. In contrast to the latter, here, a quadratic term for DOY was added, to reflect non-linearity in the snow bulk density variability. The datums in the input data.frame are converted to DOY as days spent since November 1st. Regression coefficients depend on regions defined in Guyennon et al. (2019), which are *italy* for the Italian Alps, *southwest* for the South-western Italian Alps, *central* for the Central Italian Alpes or *southeast* for the South-western Italian Alps.

If region.gu19 is set to *myregion*, the coefficients no, n1 and n2 must be set to values, obtained from a regression between densities and day-of-year from another dataset. It has to have the form density ~ DOY + DOY^2, where DOY is the day-of-year as defined in the original reference. Non computable values are returned as NA.

Value

A vector with daily SWE values in mm.

References

Guyennon, N., Valt, M., Salerno, F., Petrangeli, A., Romano, E. (2019) 'Estimating the snow water equivalent from snow depth measurements in the Italian Alps', Cold Regions Science and Technology. Elsevier, 167 (August), p. 102859. doi: 10.1016/j.coldregions.2019.102859.

Pistocchi, A. (2016) 'Simple estimation of snow density in an Alpine region', Journal of Hydrology: Regional Studies. Elsevier B.V., 6 (Supplement C), pp. 82 - 89. doi: 10.1016/j.ejrh.2016.03.004.

Examples

```
data(hsdata)
swe <- swe.gu19(hsdata, region = "italy")
summary(swe)
```

---

swe.jo09 *Statistical SWE modeling depending on month and climatic region in Switzerland*

Description

Snow Water Equivalent (SWE) is modeled statistically depending on snow depth, altitude, date and region in Switzerland.

Usage

```
swe.jo09(data, alt, region.jo09)
```

Arguments

data	A data.frame of daily observations with two columns named *date* and *hs* referring to day and snow depth at that day. The date column must be either of class 'character', 'Date' or 'POSIXct' with the format YYYY-MM-DD. The hs column must be snow depth values ≥ 0 in m.
alt	Station elevation in meters.
region.jo09	Integer number of the Swiss region where the station belongs to, according to Fig. 1 in the original reference. Must be one of 1,2,3,4,5,6,7.

Details

swe.jo09 This model parameterizes bulk snow density using snow depth, season (i.e. month), site altitude and site location. The location is implemented by a density offset according to the region in Switzerland, where the station belongs to. Non computable values are returned as NA.

Value

A numeric vector with SWE values for each region in mm.

References

Jonas, T., Marty, C. and Magnusson, J. (2009) 'Estimating the snow water equivalent from snow depth measurements in the Swiss Alps', Journal of Hydrology, 378(1 - 2), pp. 161 - 167. doi: 10.1016/j.jhydrol.2009.09.021.

Examples

```
data(hsdata)
swe <- swe.jo09(hsdata, 1500, 1)
summary(swe)
plot(swe)
```

---

swe.pi16 *Statistical SWE modeling depending on the day-of-year*

Description

This model parameterizes bulk snow density with day-of-the-year as the only input. It was calibrated for the region of South Tyrol, Italy, and is therefore called ST model in the original reference.

Usage

```
swe.pi16(data, rho_0 = 200, K = 1)
```

Arguments

data	A data.frame with at least two columns named date and hs. They should contain date and corresponding daily observations of snow depth $hs \geq 0$ measured at one site. The unit must be meters (m). No gaps or NA are allowed. Dates must be either of class 'character', 'Date' or 'POSIXct' and given in the format YYYY-MM-DD. No sub-daily resolution is allowed at the moment (see details).
rho_0	Intercept of the linear regression between observed snow depths and SWE values. rho_0 is set to 200 as default, which is the value from the original reference. It can however be set to any value according to regression modeling with other datasets.
K	Slope of the linear regression between observed densities and the day-of-year as defined in the original reference. K is set to 1 as default, which is the value from the original reference. It can however be set to any value according to regression modeling with other datasets.

Details

swe.pi16 This function uses only the day-of-year (DOY) as parameterization for bulk snow density and hence SWE. Here, the datums in the input data.frame are converted to DOY as defined in the original reference: negative values between 1.10. and 31.12. DOY=-92 at 1.10. In leap years 31.12. has DOY = 0, in non-leap years 31.12. has DOY = -1 with no day being 0. Non computable values are returned as NA.

Value

A vector with daily SWE values in mm.

References

Pistocchi, A. (2016) 'Simple estimation of snow density in an Alpine region', Journal of Hydrology: Regional Studies. Elsevier B.V., 6(Supplement C), pp. 82 - 89. doi: 10.1016/j.ejrh.2016.03.004.

Examples

```
data(hsdata)
swe <- swe.pi16(hsdata)
summary(swe)
```

---

swe.st10 *Statistical SWE modeling depending on day of year and a climatic region*

Description

The *Sturm* model parameterizes bulk snow density with day of the year and a *snowclass.st10*. It was trained on historical snow depth - density - SWE data from the United States, Canada, and Switzerland.

Usage

```
swe.st10(
  data,
  snowclass.st10 = c("alpine", "maritime", "prairie", "tundra", "taiga")
)
```

Arguments

`data`	A data.frame with at least two columns named `date` and `hs`. They should contain date and corresponding daily observations of snow depth $hs \geq 0$ measured at one site. The unit must be meters (m). No gaps or NA are allowed. Dates must be either of class 'character', 'Date' or 'POSIXct' and given in the format `YYYY-MM-DD`. No sub-daily resolution is allowed at the moment (see details).
`snowclass.st10`	Must be one of the following character strings: "alpine","maritime","prairie","tundra","taiga".

Details

This model converts snow depth to SWE using snow depth, day of year and station location (from which a climate class of snow can be inferred. The day of year (DOY) is the day-number of in the season 1.10. - 30.6. The 1.10. refers to DOY = -92. The 1.2. would be DOY = 32, while 15.11. would be DOY = -47. The *snowclass.st10* must be one out of the character strings "alpine","maritime","prairie","tundra" and "taiga". For the Alps probably "alpine" would be the most appropriate climate classification. Non computable values are returned as NA.

Value

A vector with daily SWE values in mm.

References

Sturm, M. et al. (2010) 'Estimating Snow Water Equivalent Using Snow Depth Data and Climate Classes', Journal of Hydrometeorology, 11(6), pp. 1380 - 1394. doi: 10.1175/2010JHM1202.1.

Examples

```
data(hsdata)
swe <- swe.st10(hsdata)
summary(swe)
```

Index

∗ **datasets**
hsdata, 4

delta.snow, *2*, *6*

Guyennon, *6*

hs2swe, 2, *6*
hsdata, 4

Jonas, *6*

nixmass, *4*, 5

Pistocchi, *6*
plot, *8*
plot.nixmass, 7

Sturm, *6*
summary.nixmass, 8
swe.delta.snow, 9
swe.gu19, 12
swe.jo09, 13
swe.pi16, *12*, 14
swe.st10, 15