Package 'HeatStressR'

November 2, 2026

Type Package

Title Calculate Heat Stress Indices

Version 3.0.0

Date 2026-10-09

Description Calculates heat-stress indices from meteorological observations using shared psychrometric calculations and consistent method-specific input handling. Methods include the physically based outdoor wet-bulb globe temperature model described by Liljegren et al. (2008) <doi:10.1080/15459620802310770>, wet-bulb temperature and heat-index calculations, and established closed-form heat-stress indices. Provides row-level diagnostics, configurable physical controls, vectorized batch processing, and optional parallel execution for the Liljegren method. It is an independently maintained fork of 'HeatStress'; its Liljegren implementation is not a bitwise-compatible port of the original program.

License GPL-3

URL https://github.com/zyf0717/HeatStressR

BugReports https://github.com/zyf0717/HeatStressR/issues

Depends R (>= 4.1.0)

Suggests pkgload, testthat (>= 3.2.0)

Config/testthat/edition 3

Imports stats, utils, parallel, heatindex (>= 0.0.2)

Encoding UTF-8

Config/roxygen2/version 8.1.0

Config/roxygen2/markdown true

Config/build/clean-inst-doc false

NeedsCompilation no

Author Yifei Zheng [aut, cre] (ORCID: <https://orcid.org/0009-0001-4858-2025>, Creator and maintainer of this fork),
Ana Casanueva [ctb] (Original HeatStress package author and R implementation)

Maintainer Yifei Zheng <zyf0717@gmail.com>

Repository CRAN

Date/Publication 2026-11-01 23:20:15 UTC

Contents

"data_obs" . . . 2
"data_wbgt.Bernard" . . . 3
"data_wbgt.Liljegren" . . . 4
"data_wbt.Stull" . . . 4
degToRad . . . 5
diffusivity . . . 5
emis_atm . . . 6
heat-stress-indices . . . 6
HeatStressR . . . 8
HeatStressR-deprecated . . . 9
heat_indices . . . 12
heat_methods . . . 14
humidity-conversions . . . 15
h_cylinder_in_air . . . 16
h_evap . . . 16
h_sphere_in_air . . . 17
radToDeg . . . 18
solar_zenith . . . 18
thermal_cond . . . 19
viscosity . . . 19
wbgt_bernard . . . 20
wbgt_liljegren . . . 21

Index **23**

`"data_obs"` *Meteorological input variables for the heat stress indices.*

Description

A dataset containing the input variables for the heat stress indices as examples. It contains observed daily data from the ECA&D dataset for summer 2003 and one station (Salamanca, Spain).

Usage

```
data(data_obs)
```

Format

A data frame with 92 rows (days) and 6 variables. The variables are as follows:

- Dates: Dates vector of dates for summer 2003.
- tasmean: numeric vector of daily mean temperature.
- dewp: numeric vector od daily mean dew point temperature.
- hurs: numeric vector of daily mean relative humidity.
- wind: numeric vector of daily mean wind speed.
- solar: numeric vector of daily mean solar radiation. These values come from the closest g grid box from the SARAH satellite dataset (DOI:10.5676/EUM_SAF_CM/SARAH/V001).

Source

https://www.ecad.eu/index.php and DOI:10.5676/EUM_SAF_CM/SARAH/V001

`"data_wbgt.Bernard"` *Example for Wet Bulb Globe Temperature in the shade (Bernard et al. 1999).*

Description

Dataset to check the result of `wbgt.Bernard` with the testing data `obs_data` for summer 2003 and one station (Salamanca, Spain).

Usage

```
data(data_wbgt.Bernard)
```

Format

List of two elements:

- data: numeric vector with the WBGT values in degC.
- Tpwb: numeric vector with the phychrometric wet bulb temperature (Tpwb) in degC.

"data_wbgt.Liljegren" *Example for Wet Bulb Globe Temperature in the sun (Liljegren et al. 2008).*

Description

Dataset to check the result of `wbgt.Liljegren` with the testing data `obs_data` for summer 2003 and one station (Salamanca, Spain).

Usage

```
data(data_wbgt.Liljegren)
```

Format

List of three elements:

- data: numeric vector with the WBGT values in degC.
- Tpwb: natural wet bulb temperature (Tnwb) in degC.
- Tg: globe temperature in degC.

"data_wbt.Stull" *Example for Wet Bulb Temperature (Stull 2011).*

Description

Dataset to check the result of `wbt.Stull` with the testing data `obs_data` for summer 2003 and one station (Salamanca, Spain).

Usage

```
data(data_wbt.Stull)
```

Format

Numeric vector with 92 values (days).

`degToRad` *Convert degree angle to radians.*

Description

Convert degree angle to radians.

Usage

```
degToRad(angleDeg)
```

Arguments

`angleDeg`	angle in degree.

Value

Angle in radians.

Author(s)

Ana Casanueva (10.01.2017).

`diffusivity` *Compute the diffusivity of water vapor in air.*

Description

Compute the diffusivity of water vapor in air.

Usage

```
diffusivity(Tk, Pair)
```

Arguments

`Tk`	value of air temperature in Kelvin.
`Pair`	value of air pressure in hPa.

Details

Reference: BSL, page 505.

Value

Diffusivity of water vapor in air, m2/s.

Author(s)

Ana Casanueva (05.01.2017).

`emis_atm` *Calculate the atmospheric emissivity.*

Description

Calculate the atmospheric emissivity.

Usage

```
emis_atm(Tk, RH)
```

Arguments

`Tk`	value of air temperature in Kelvin.
`RH`	value of relative humidity in fraction.

Details

Reference: Oke (2nd edition), page 373.

Value

atmospheric emissivity.

Author(s)

Ana Casanueva (05.01.2017).

`heat-stress-indices` *Meteorological heat-stress indices*

Description

Calculate one index from aligned observations. Numeric scalars expand to the common observation length; other unequal lengths and matrices are rejected. Missing inputs propagate silently. Invalid observations return NA with one summary warning. Outside documented applicability bounds, a formula is evaluated where possible and flagged with a warning. Unknown bounds do not establish validity.

Usage

```
wet_bulb_stull(tas, hurs, diagnostics = FALSE)

wet_bulb_romps(tas, hurs, pressure = 1010, diagnostics = FALSE)

heat_index_rothfusz(tas, hurs, diagnostics = FALSE)

heat_index_lu(tas, hurs, diagnostics = FALSE)

wbgt_simplified_abm(tas, hurs, diagnostics = FALSE)

wbgt_simplified_indoor(tas, hurs, diagnostics = FALSE)

apparent_temperature(tas, hurs, wind, diagnostics = FALSE)

effective_temperature(tas, hurs, wind, diagnostics = FALSE)

humidex(tas, hurs, diagnostics = FALSE)

discomfort_index(tas, hurs, diagnostics = FALSE)
```

Arguments

tas	Air temperature in degrees Celsius.
hurs	Relative humidity in percent. The method catalog documents phase conventions. Methods using inherited Dosseger vapour pressure switch reference phase at 0 C; formulas using RH directly perform no conversion. Romps/Lu follow heatindex's water/ice convention at 273.16 K. Supply humidity on the chosen method's stated reference phase, especially below freezing.
diagnostics	Return values, component temperatures and diagnostics.
pressure	Atmospheric pressure in hPa, default 1010. This is an explicit near-sea-level assumption; use observed pressure at elevation.
wind	Wind speed in m/s measured at 10 m. Height adjustment is not performed for apparent or effective temperature.

Details

Romps calculates thermodynamic liquid-water wet bulb, not natural or aspirated wet bulb. Lu uses the simplified model published in 2026. Both call the external heatindex package; numerical reproducibility requires pinning its version. Stull assumes sea-level pressure and has a restricted domain. Cold observations within its rectangular bounds are marked unknown because its excluded cold/dry corner has no numeric boundary in the paper.

Simplified WBGT uses vapour pressure e in hPa from the shared Dosseger kernel. ABM: $0.567 tas + 0.393 e + 3.94$; indoor: $0.567 tas + 0.216 e + 3.38$. These empirical approximations do not calculate radiation or wind effects.

Value

By default, an aligned numeric vector on the Celsius scale. Humidex uses a Celsius-equivalent index scale. With diagnostics=TRUE, a list with values, components, and diagnostics. diagnostics contains a rows data frame (input_status, input_reason, domain_status, adjusted, converged, failure_reason), optional method-specific solver detail, and metadata.

References

- Stull (2011), doi:10.1175/JAMCD110143.1.
- Romps (2026), doi:10.1175/JAMCD250130.1.
- Lu et al. (2026), doi:10.1175/JAMCD250067.1.
- Lemke and Kjellstrom (2012), doi:10.2486/indhealth.MS1352.
- NWS Heat Index procedure.

See Also

`heat_methods()`, `heat_indices()`, `wbgt_liljegren()`

Examples

```
heat_index_rothfusz(c(25, 35), c(50, 10))
wet_bulb_romps(30, 70, pressure = 950)
wet_bulb_stull(c(30, 55), 50, diagnostics = TRUE)
```

`HeatStressR` *Meteorological heat-stress metrics*

Description

Canonical single-index functions, registry-driven multi-index calculations, aligned input validation and optional diagnostics. The package includes the vectorized R Liljegren WBGT model and delegates Romps thermodynamic wet bulb and the 2026 Lu Heat Index to the authors' heatindex package.

Details

Available methods:

`wet_bulb_stull` Stull wet-bulb approximation; inputs: tas, hurs; units: degC.

`wet_bulb_romps` Romps thermodynamic liquid-water wet bulb; inputs: tas, hurs, pressure; units: degC.

`wbgt_simplified_abm` ABM simplified WBGT; inputs: tas, hurs; units: degC.

`wbgt_simplified_indoor` Indoor simplified WBGT; inputs: tas, hurs; units: degC.

`apparent_temperature` Apparent temperature; inputs: tas, hurs, wind_10m; units: degC.

`effective_temperature` Effective temperature; inputs: tas, hurs, wind_10m; units: degC.

`humidex` Humidex (Celsius-equivalent scale); inputs: tas, hurs; units: degC.

`discomfort_index` Discomfort index; inputs: tas, hurs; units: degC.

`heat_index_rothfusz` NWS Rothfusz Heat Index; inputs: tas, hurs; units: degC.

`heat_index_lu` Lu et al. (2026) simplified Heat Index; inputs: tas, hurs; units: degC.

`wbgt_bernard` Bernard indoor/shade WBGT; inputs: tas, dewp; units: degC.

`wbgt_liljegren` Liljegren outdoor WBGT; inputs: tas, dewp, wind_2m, radiation, time, lon, lat, pressure; units: degC.

Temperatures use Celsius, pressure hPa, humidity percent, wind m/s, and radiation W/m2. Consult `heat_methods()` for phase conventions, wind measurement height, applicability and scientific references. Missing inputs propagate silently; invalid rows become NA with summarized warnings. Scalar observations expand to the common input length.

`wbgt_liljegren()` evaluates instantaneous states, using batch roots, scalar fallback and optional PSOCK workers. Its independently maintained R model is not a bitwise port of the original Liljegren program. Validated component temperatures are retained when another component fails.

HeatStressR is a GPL-3 fork of HeatStress. Ana Casanueva authored the original package and R translation; Yifei Zheng maintains this fork. Historical contributions and attribution are retained. The fork is not affiliated with the original project or its authors.

Deprecated 2.x adapters preserve signatures and return shapes while applying documented scientific corrections. Removal is planned no earlier than v4.0.0. See HeatStressR-deprecated and the installed migration guide. Package attachment performs no network access. Use `citation("HeatStressR")` for the package citation. Please also cite the original methodological reference for each heat-stress method used; see the corresponding function documentation.

See Also

`heat_methods()`, `heat_indices()`, `wbgt_liljegren()`

`HeatStressR-deprecated`

Deprecated HeatStressR 2.x endpoints

Description

Compatibility adapters preserve names, positional arguments and return shapes. Scientific corrections and invalid-row handling follow the canonical APIs. Each endpoint warns once per session and will be removed no earlier than v4.0.0. New code should use the replacements in `heat_methods()`.

Usage

```
apparentTemp(tas, hurs, wind)

effectiveTemp(tas, hurs, wind)

discomInd(tas, hurs)
```

```
hi(tas, hurs)

swbgt(tas, hurs)

wbt.Stull(tas, hurs)

dewp2hurs(tas, dewp)

tashurs2vap.pres(tas, hurs)

wbgt.Bernard(tas, dewp, tolerance = 1e-04, noNAs = TRUE, swap = FALSE)

wbgt.Liljegren(
  tas,
  dewp,
  wind,
  radiation,
  dates,
  lon,
  lat,
  tolerance = 1e-04,
  noNAs = TRUE,
  swap = FALSE,
  hour = FALSE,
  engine = c("batch", "scalar"),
  diagnostics = FALSE,
  root_tolerance = NULL,
  residual_tolerance = NULL,
  dewpoint_tolerance = NULL,
  pressure = 1010,
  surface_albedo = 0.45,
  globe_diameter = 0.0508,
  min_wind_speed = 0.13,
  workers = 1L,
  solar_time = "timestamp",
  direct_fraction = 0.8
)

calZenith(dates, lon, lat, hour = FALSE, solar_time = "timestamp")

fTg(
  tas,
  relh,
  Pair,
  wind,
  min.speed,
  radiation,
```

```
  propDirect,
  zenith,
  SurfAlbedo = 0.45,
  tolerance = 1e-04,
  globe_diameter = 0.0508
)

fTnwb(
  tas,
  dewp,
  relh,
  Pair,
  wind,
  min.speed,
  radiation,
  propDirect,
  zenith,
  irad = 1,
  SurfAlbedo = 0.45,
  tolerance = 1e-04
)

indexShow()
```

Arguments

`tas`, `dewp`, `hurs`	Air/dew-point temperature in C and RH in percent.
`wind`	Wind in m/s: 10 m for apparent/effective temperature, 2 m for WBGT.
`tolerance`	Legacy tolerance mapped to numerical controls.
`noNAs`, `swap`	Legacy dew-point cap/reject/swap selection. Swapping is confined to these adapters. Small supersaturated differences are capped consistently rather than retained within a tolerance.
`radiation`	Solar shortwave radiation in W/m2.
`dates`	Legacy timestamps or date strings. Unzoned strings use UTC.
`lon`, `lat`	Coordinates in degrees east/north.
`hour`, `solar_time`	Legacy timestamp/date-noon selection.
`engine`	Legacy batch or scalar reference selection.
`diagnostics`	Return legacy solver diagnostics where supported.
`root_tolerance`, `residual_tolerance`	Independent numerical tolerances.
`dewpoint_tolerance`	Deprecated validation tolerance; accepted and validated, while canonical dew-point policy is applied exactly.
`pressure`	Atmospheric pressure in hPa.

`surface_albedo`, `globe_diameter`, `min_wind_speed`
: Legacy physical controls.

`workers`
: Number of PSOCK workers.

`direct_fraction`
: Direct share of shortwave radiation.

`relh`
: Relative humidity in percent for scalar component solvers.

`Pair`
: Pressure in hPa for scalar component solvers.

`min.speed`
: Minimum wind speed in m/s for scalar solvers.

`propDirect`
: Direct shortwave fraction for scalar solvers.

`zenith`
: Solar zenith in radians for scalar solvers.

`SurfAlbedo`
: Surface albedo for scalar solvers.

`irad`
: Include radiation (1) or omit it (0) in the scalar wet-bulb solver.

Details

swbgt maps to `wbgt_simplified_indoor()`, not the ABM formula. hi maps to corrected NWS Heat Index in Celsius. fTg/fTnwb remain scalar reference adapters; use the components from `wbgt_liljegren()` in new code.

Value

Simple indices and conversions return numeric vectors. Liljegren returns list(data, Tnwb, Tg), with diagnostics when requested. Bernard returns list(Tpwb, data). indexShow returns the legacy-shaped catalog.

Author(s)

Original package: Ana Casanueva, with contributions attributed in inherited source. Current fork: Yifei Zheng.

`heat_indices` *Calculate all heat indices supported by supplied observations*

Description

Uses the method registry to select and dispatch methods, including numerical Romps, Lu, Bernard and Liljegren calculations. Explicit indices are recommended for stable column sets and predictable workloads.

Usage

```
heat_indices(
  tas,
  hurs = NULL,
  wind = NULL,
  dewp = NULL,
  indices = NULL,
  wind_10m = NULL,
  wind_2m = NULL,
  pressure = 1010,
  radiation = NULL,
  time = NULL,
  lon = NULL,
  lat = NULL,
  diagnostics = FALSE,
  liljegren_options = list()
)
```

Arguments

tas	Air temperature in degrees Celsius.
hurs	Optional RH in percent; required by humidity-based methods.
wind	Deprecated alias for wind_10m.
dewp	Optional dew point in C for Bernard and Liljegren.
indices	Canonical method names or deprecated aliases. NULL selects all methods whose declared observation inputs were supplied, in registry order. Explicit selection retains requested order and column labels.
wind_10m, wind_2m	Wind in m/s at the indicated height. One is never substituted for the other.
pressure	Atmospheric pressure in hPa, default 1010. This is an explicit near-sea-level assumption; use observed pressure at elevation.
radiation, time, lon, lat	Optional Liljegren inputs; see wbgt_liljegren().
diagnostics	Return values, component temperatures and diagnostics.
liljegren_options	Named list containing direct_fraction, dewpoint_policy, workers, or control; see wbgt_liljegren().

Details

No absent meteorological inputs are inferred. pressure defaults to 1010 hPa, so Romps is selected when tas and hurs are supplied. An all-NA supplied input still satisfies availability, with NA rows in the result. All default column names are canonical. Humidity phase references differ between published methods below freezing; see heat_methods().

Available methods:

`wet_bulb_stull` Stull wet-bulb approximation; inputs: tas, hurs; units: degC.

`wet_bulb_romps` Romps thermodynamic liquid-water wet bulb; inputs: tas, hurs, pressure; units: degC.

`wbgt_simplified_abm` ABM simplified WBGT; inputs: tas, hurs; units: degC.

`wbgt_simplified_indoor` Indoor simplified WBGT; inputs: tas, hurs; units: degC.

`apparent_temperature` Apparent temperature; inputs: tas, hurs, wind_10m; units: degC.

`effective_temperature` Effective temperature; inputs: tas, hurs, wind_10m; units: degC.

`humidex` Humidex (Celsius-equivalent scale); inputs: tas, hurs; units: degC.

`discomfort_index` Discomfort index; inputs: tas, hurs; units: degC.

`heat_index_rothfusz` NWS Rothfusz Heat Index; inputs: tas, hurs; units: degC.

`heat_index_lu` Lu et al. (2026) simplified Heat Index; inputs: tas, hurs; units: degC.

`wbgt_bernard` Bernard indoor/shade WBGT; inputs: tas, dewp; units: degC.

`wbgt_liljegren` Liljegren outdoor WBGT; inputs: tas, dewp, wind_2m, radiation, time, lon, lat, pressure; units: degC.

Value

A data frame by default. With diagnostics, list(values, components, diagnostics), where components and diagnostics are named lists by requested index. Validation/forcing warnings are summarized once per call.

Examples

```
heat_indices(c(25, 30), c(60, 70), indices = c("humidex", "heat_index_lu"))
heat_indices(30, 70, wind_10m = 2)
```

`heat_methods` *Discover available heat-stress methods*

Description

Discover available heat-stress methods

Usage

```
heat_methods()
```

Value

A data frame containing canonical method names, aliases, required inputs, units, applicability descriptions, humidity conventions, wind heights and scientific references. An unknown boundary does not establish validity. Humidex is reported on its Celsius-equivalent scale.

Examples

```
heat_methods()
```

`humidity-conversions` *Meteorological humidity conversions*

Description

Meteorological humidity conversions

Usage

```
vapour_pressure(tas, hurs)

relative_humidity(tas, dewp, dewpoint_policy = c("cap", "na", "error"))
```

Arguments

`tas`	Air temperature in degrees Celsius.
`hurs`	Relative humidity in percent. The method catalog documents phase conventions. Methods using inherited Dosseger vapour pressure switch reference phase at 0 C; formulas using RH directly perform no conversion. Romps/Lu follow heatindex's water/ice convention at 273.16 K. Supply humidity on the chosen method's stated reference phase, especially below freezing.
`dewp`	Dew point in degrees Celsius.
`dewpoint_policy`	Handling of dew point above air temperature: cap, na, or error. Default cap.

Details

Uses the paired inherited Dosseger equations. Coefficients are selected by air temperature: water at or above 0 C, ice below 0 C. This preserves that convention when dew point and air temperature straddle 0 C.

Value

An aligned numeric vector: vapour pressure in hPa or RH in percent.

Examples

```
vapour_pressure(30, 70)
relative_humidity(30, 20)
```

h_cylinder_in_air — *Calculate the convective heat transfer coefficient for a long cylinder in cross flow.*

Description

Calculate the convective heat transfer coefficient for a long cylinder in cross flow.

Usage

```
h_cylinder_in_air(Tk, Pair, speed, min.speed, diam.wick)
```

Arguments

Tk	value of air temperature in Kelvin.
Pair	value of air pressure in hPa.
speed	value of wind speed in m/s.
min.speed	value of minimum wind speed in m/s.
diam.wick	diameter of the cylinder in m.

Details

Reference: Bedingfield and Drew, eqn 32.

Value

Convective heat transfer coefficient for a long cylinder, W/(m2 K).

Author(s)

Ana Casanueva (05.01.2017).

h_evap — *Calculate the heat of evaporation, J/(kg K).*

Description

Calculate the heat of evaporation, J/(kg K), for temperature in the range 283-313 K.

Usage

```
h_evap(Tk)
```

Arguments

Tk	value of air temperature in Kelvin.

Details

Reference: Van Wylen and Sonntag, Table A.1.1

Value

Heat of evaporation, J/(kg K).

Author(s)

Ana Casanueva (05.01.2017).

`h_sphere_in_air` *Calculate the convective heat transfer coefficient for flow around a sphere.*

Description

Calculate the convective heat transfer coefficient for flow around a sphere.

Usage

```
h_sphere_in_air(Tk, Pair, speed, min.speed, diam.globe)
```

Arguments

`Tk`	value of air temperature in Kelvin.
`Pair`	value of air pressure in hPa.
`speed`	value of wind speed in m/s.
`min.speed`	value of minimum wind speed in m/s.
`diam.globe`	diameter of the sphere in m.

Details

Reference: Bird, Stewart, and Lightfoot (BSL), page 409.

Value

Convective heat transfer coefficient for flow around a sphere, W/(m2 K).

Author(s)

Ana Casanueva (05.01.2017).

radToDeg *Convert radian angle to degrees.*

Description

Convert radian angle to degrees.

Usage

```
radToDeg(angleRad)
```

Arguments

angleRad	angle in radians.

Value

Angle in degrees.

Author(s)

Ana Casanueva (10.01.2017).

solar_zenith *Solar zenith angle at supplied instants*

Description

Solar zenith angle at supplied instants

Usage

```
solar_zenith(time, lon, lat)
```

Arguments

time	POSIXct/POSIXlt instants or offset-bearing ISO 8601 strings. Observations are instantaneous; align interval data externally. Strings without an offset and date-only values are not accepted as instants.
lon, lat	Longitude and latitude in degrees east and north.

Value

Numeric solar zenith angles in degrees, aligned with time.

Examples

```
solar_zenith("2024-06-01T12:00:00Z", 0, 15)
```

thermal_cond *Calculate the thermal conductivity of air, W/(m K).*

Description

Calculate the thermal conductivity of air, W/(m K).

Usage

```
thermal_cond(Tk)
```

Arguments

Tk	value of air temperature in Kelvin.

Details

Reference: BSL, page 257.

Value

Thermal conductivity of air, W/(m K).

Author(s)

Ana Casanueva (05.01.2017).

viscosity *Compute the viscosity of air, kg/(m s).*

Description

Compute the viscosity of air, kg/(m s) given temperature (K).

Usage

```
viscosity(Tk)
```

Arguments

Tk	value of air temperature in Kelvin.

Details

Reference: BSL, page 23.

Value

viscosity of air, kg/(m s).

Author(s)

Ana Casanueva (05.01.2017).

`wbgt_bernard` *Bernard indoor or shade wet-bulb globe temperature*

Description

Bernard indoor or shade wet-bulb globe temperature

Usage

```
wbgt_bernard(
  tas,
  dewp,
  dewpoint_policy = c("cap", "na", "error"),
  diagnostics = FALSE,
  tolerance = 1e-04
)
```

Arguments

`tas`	Air temperature in degrees Celsius.
`dewp`	Dew point in degrees Celsius, using the inherited Dosseger humidity conversion documented in `heat_methods()`.
`dewpoint_policy`	Cap dew point at air temperature (default), return NA for supersaturated rows ("na"), or stop the call ("error").
`diagnostics`	Return values, component temperatures and diagnostics.
`tolerance`	Maximum final psychrometric root bracket width in C.

Value

Numeric WBGT in C, or the standardized diagnostic result with psychrometric wet-bulb component tpwb. Uses the inherited pressure assumption of 1010 hPa in Bernard's equation; it is not pressure configurable.

Author(s)

Original implementation: Ana Casanueva, P. Noti, J. Bhend.

References

Lemke and Kjellstrom (2012), doi:10.2486/indhealth.MS1352.

Examples

```
wbgt_bernard(30, 20, diagnostics = TRUE)
```

`wbgt_liljegren` *Liljegren outdoor wet-bulb globe temperature*

Description

Liljegren outdoor wet-bulb globe temperature

Usage

```
wbgt_liljegren(
  tas,
  dewp,
  wind,
  radiation,
  time,
  lon,
  lat,
  pressure = 1010,
  direct_fraction = 0.8,
  dewpoint_policy = c("cap", "na", "error"),
  diagnostics = FALSE,
  workers = 1L,
  control = list()
)
```

Arguments

`tas`	Air temperature in degrees Celsius.
`dewp`	Dew point in degrees Celsius, using the inherited Dosseger humidity conversion documented in `heat_methods()`.
`wind`	Wind speed in m/s at 2 m. Height adjustment is a caller responsibility.
`radiation`	Downwelling shortwave radiation in W/m2.
`time`	POSIXct/POSIXlt instants or offset-bearing ISO 8601 strings. Observations are instantaneous; align interval data externally. Strings without an offset and date-only values are not accepted as instants.
`lon, lat`	Longitude and latitude in degrees east and north.
`pressure`	Atmospheric pressure in hPa, default 1010. This is an explicit near-sea-level assumption; use observed pressure at elevation.

`direct_fraction`
: Direct / (direct + diffuse) shortwave fraction, from zero through one. Default 0.8; scalar or row-aligned.

`dewpoint_policy`
: Cap dew point at air temperature (default), return NA for supersaturated rows ("na"), or stop the call ("error").

`diagnostics`
: Return values, component temperatures and diagnostics.

`workers`
: Number of temporary PSOCK processes; default 1. Respect the detected CPU count and R check core limits. Avoid nested worker pools.

`control`
: Named list of advanced controls: root_tolerance (1e-6 K), residual_tolerance (1e-4 K, at most 0.01), surface_albedo (0.45), globe_diameter (0.0508 m), and min_wind_speed (0.13 m/s).

Details

Uses the existing vectorized R heat-balance model with scalar fallback. A validated component is retained if its partner fails. Radiation is zeroed below the solar horizon and this adjustment is recorded in solver diagnostics. This independently maintained R model is not a bitwise port of the original Liljegren C program. Match physical assumptions before comparing implementations.

Value

An aligned numeric WBGT vector in degrees Celsius. With diagnostics, a list of values, components (tnwb and tg in C), and diagnostics. Detailed solver records retain initial/final brackets, residuals and fallback reasons. Bracket temperatures are Kelvin; candidate roots are Celsius. Final residuals and root/residual tolerances are Kelvin. Globe endpoint residuals use the energy equation (K to the fourth power); wet-bulb endpoints use K.

Author(s)

Original R translation: Ana Casanueva. Fork: Yifei Zheng.

References

Liljegren et al. (2008), doi:10.1080/15459620802310770.

Examples

```
wbgt_liljegren(30, 22, 1.5, 700,
  as.POSIXct("2024-06-01 12:00:00", tz = "UTC"), 0, 15)
```

Index

apparent_temperature
(heat-stress-indices), 6
apparentTemp (HeatStressR-deprecated), 9

calZenith (HeatStressR-deprecated), 9

degToRad, 5
dewp2hurs (HeatStressR-deprecated), 9
diffusivity, 5
discomfort_index (heat-stress-indices), 6
discomInd (HeatStressR-deprecated), 9

effective_temperature
(heat-stress-indices), 6
effectiveTemp (HeatStressR-deprecated), 9
emis_atm, 6

fTg (HeatStressR-deprecated), 9
fTnwb (HeatStressR-deprecated), 9

h_cylinder_in_air, 16
h_evap, 16
h_sphere_in_air, 17
heat-stress-indices, 6
heat_index_lu (heat-stress-indices), 6
heat_index_rothfusz
(heat-stress-indices), 6
heat_indices, 12
heat_indices(), *8, 9*
heat_methods, 14
heat_methods(), *8, 9*, *13*, *20, 21*
HeatStressR, 8
HeatStressR-deprecated, *9*, 9
hi (HeatStressR-deprecated), 9
humidex (heat-stress-indices), 6
humidity-conversions, 15

indexShow (HeatStressR-deprecated), 9

radToDeg, 18
relative_humidity
(humidity-conversions), 15

solar_zenith, 18
swbgt (HeatStressR-deprecated), 9

tashurs2vap.pres
(HeatStressR-deprecated), 9
thermal_cond, 19

vapour_pressure (humidity-conversions), 15
viscosity, 19

wbgt.Bernard (HeatStressR-deprecated), 9
wbgt.Liljegren
(HeatStressR-deprecated), 9
wbgt_bernard, 20
wbgt_liljegren, 21
wbgt_liljegren(), *8, 9*, *12, 13*
wbgt_simplified_abm
(heat-stress-indices), 6
wbgt_simplified_indoor
(heat-stress-indices), 6
wbgt_simplified_indoor(), *12*
wbt.Stull (HeatStressR-deprecated), 9
wet_bulb_romps (heat-stress-indices), 6
wet_bulb_stull (heat-stress-indices), 6