

Package ‘comf’

September 28, 2016

Type Package

Title Functions for Thermal Comfort Research

Version 0.1.6

Date 2016-09-15

Imports plyr

Author Marcel Schweiker [aut, cre] and Sophia Mueller [aut] and Michael Kleber [ctr] and Boris Kingma [ctr] and Masanori Shukuya [ctr]

Maintainer Marcel Schweiker <marcel.schweiker@kit.edu>

Description Functions to calculate various common and less common thermal comfort indices, convert physical variables, and evaluate the performance of thermal comfort indices.

License GPL-2

LazyLoad yes

LazyData true

Suggests R.rsp

VignetteBuilder R.rsp

NeedsCompilation no

Repository CRAN

Date/Publication 2016-09-28 07:26:49

R topics documented:

comf-package	2
calc2Node	3
calcATHB	6
calcAvgAcc	7
calcBias	9
calcCOEFF	10
calcComfInd	11
calcdTNZ	14
calcHbExSteady	15
calcHbExUnsteady	18

calcHumidity	22
calcpmvadj	23
calcPMVPPD	25
calcPtsadj	27
calctadapt	29
calcTPRTSV	30
calcTroin	31
createCond	32
cutTSV	34
dfASHRAETableG11	35
dfField	36
dfISO7730AppE	37
dfISO7730TableD1	38

Index	40
--------------	-----------

comf-package	<i>Calculation and Evaluation of Common and Less Common Comfort Indices</i>
--------------	---

Description

This package contains several functions to calculate and evaluate a series of comfort indices.

Details

Package:	comf
Type:	Package
Version:	0.1.6
Date:	2016-09-15
License:	GPL-2
LazyLoad:	yes

To create input parameters, the function `createCond`, which creates a list of input parameters may be helpful. The main function of this packages is `calcComfInd`, which returns the desired comfort parameters. However, each index can be computed using its own function, e.g. to calculate only PMV the function `calcPMV` can be used.

The comfort indices calculated within this package are as follows. To get further information, go to the help page, which can be accessed using the index below e.g. `?pmv`:

Index	Description
pmv	Predicted mean vote (PMV)
ppd	Predicted percentage dissatisfied (PPD)
tnHumphreysNV	Neutral temperature in naturally ventilated buildings according to Humphreys 1978
tnHumphreysAC	Neutral temperature in climate-controlled buildings according to Humphreys 1978
tnAuliciems	Neutral temperature according to Auliciems 1981

tAdapt15251	Adaptive comfort temperature according to EN 15251
dTNZ	Distance to thermoneutral zone
ATHBpmv	Adaptive thermal heat balance vote based on pmv
ATHBset	Adaptive standard effective temperature
ATHBpts	Adaptive thermal heat balance vote based on set
apmv	Adaptive predicted mean vote according to Yao et al.
ptsa	Adaptive predicted thermal sensation vote according to Gao et al.
epmv	PMV adjusted with expectancy factor based on Fanger and toftum
ptse	Predicted thermal sensation vote based on set and adjusted with expectancy factor according to Gao et al.
set	standard effective temperature based on two node model by Gagge et al.
et	Effective temperature based on two node model by Gagge et al.
tsens	Predicted thermal sensation
disc	Predicted discomfort
ps	Predicted percentage satisfied with the level of air movement
pd	Predicted percentage dissatisfied due to draft
pts	Predicted thermal sensation vote based on set
HBxst	Human body exergy consumption rate using steady state method

The performance criteria included in this package are presented below. Again you can get further information on the corresponding help pages:

Index	Description
meanBias	Mean bias between predicted and observed thermal sensation vote
TPR	True positive rate
avgAcc	Average accuracy of predicted thermal sensation vote

Author(s)

Marcel Schweiker in cooperation with Sophia Mueller
 Contact: marcel.schweiker@kit.edu

References

See references in function descriptions.

See Also

see also [createCond](#), [calcComfInd](#)

calc2Node

Calculating Comfort Indices based on the 2-Node-Model

Description

calc2Node calculates ET, SET, TSENS, DISC, PD, PS and PTS based on the 2-Node-Model by Gagge et al.

Usage

```

calc2Node(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, pb = 760, ltime = 60, ht = 171,
  wt = 70, tu = 40, obj = "set", csw = 170, cdil = 120, cstr = 0.5, varOut = "else")
calcET(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, pb = 760, ltime = 60, ht = 171,
  wt = 70, tu = 40, obj = "set", csw = 170, cdil = 120, cstr = 0.5)
calcSET(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, pb = 760, ltime = 60, ht = 171,
  wt = 70, tu = 40, obj = "set", csw = 170, cdil = 120, cstr = 0.5)
calcTSens(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, pb = 760, ltime = 60, ht = 171,
  wt = 70, tu = 40, obj = "set", csw = 170, cdil = 120, cstr = 0.5)
calcDisc(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, pb = 760, ltime = 60, ht = 171,
  wt = 70, tu = 40, obj = "set", csw = 170, cdil = 120, cstr = 0.5)
calcPD(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, pb = 760, ltime = 60, ht = 171,
  wt = 70, tu = 40, obj = "set", csw = 170, cdil = 120, cstr = 0.5)
calcPS(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, pb = 760, ltime = 60, ht = 171,
  wt = 70, tu = 40, obj = "set", csw = 170, cdil = 120, cstr = 0.5)
calcPTS(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, pb = 760, ltime = 60, ht = 171,
  wt = 70, tu = 40, obj = "set", csw = 170, cdil = 120, cstr = 0.5)
calcPMVGagge(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, pb = 760, ltime = 60, ht = 171,
  wt = 70, tu = 40, obj = "set", csw = 170, cdil = 120, cstr = 0.5)
calcPMVStar(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, pb = 760, ltime = 60, ht = 171,
  wt = 70, tu = 40, obj = "set", csw = 170, cdil = 120, cstr = 0.5)
calcSkinWettedness(ta, tr, vel, rh, clo = .5, met = 1, wme = 0, pb = 760, ltime = 60,
  ht = 171, wt = 70, tu = 40, obj = "set", csw = 170, cdil = 120, cstr = .5,
  varOut="skinWet")

```

Arguments

ta	a numeric value presenting air temperature in [degree C]
tr	a numeric value presenting mean radiant temperature in [degree C]
vel	a numeric value presenting air velocity in [m/s]
rh	a numeric value presenting relative humidity [%]
clo	a numeric value presenting clothing insulation level in [clo]
met	a numeric value presenting metabolic rate in [met]
wme	a numeric value presenting external work in [met]
pb	a numeric value presenting barometric pressure in [torr] or [mmHg]
ltime	a numeric value presenting exposure time in [minutes]
ht	a numeric value presenting body height in [cm]
wt	a numeric value presenting body weight in [kg]
tu	a numeric value presenting turbulence intensity in [%]
obj	a character element, either "set" or "pmvadj".
csw	a numeric value presenting the driving coefficient for regulatory sweating
cdil	a numeric value presenting the driving coefficient for vasodilation
cstr	a numeric value presenting the driving coefficient for vasoconstriction
varOut	a string value either "else" for normal output of SET or "skinWet" to report value of skin wettedness.

Details

All variables must have the same length 1. For the calculation of several values use function `calcComfInd`.

The value of `obj` defines whether the function will use the version presented in ASHRAE 55-2013 for adjustment of `pmv` (`obj = "pmvadj"`), or the original code by Gagge to calculate `set` (`obj = "set"`). In the version presented in ASHRAE 55-2013, the lines of code related to self-generated convection is deleted. Therefore, a difference can only be seen at higher values of `met`.

Value

`calc2Node` returns a `data.frame` with the following items:

<code>et</code>	Effective temperature
<code>set</code>	standard effective temperature
<code>tsens</code>	Predicted thermal sensation
<code>disc</code>	Predicted discomfort
<code>ps</code>	Predicted percentage satisfied with the level of air movement
<code>pd</code>	Predicted percentage dissatisfied due to draft
<code>pts</code>	Predicted thermal sensation vote based on set
<code>pmvg</code>	Gagge's version of Fanger's PMV
<code>pmvstar</code>	Same as Fanger's PMV except that dry is calculated using SET* rather than the operative temperature

The other functions return a single index, e.g. `code(calcSET)` returns the standard effective temperature.

Note

In case one of the variables is not given, a standard value will be taken from a list (see [createCond](#) for details).

Author(s)

The code for `calc2Node` is based on the code in BASIC and C++ presented by Fountain and Huizenga (1995). The translation into R-language and comparison with ASHRAE 55-2013 conducted by Marcel Schweiker.

References

- ASHRAE Standard 55-2013. Thermal environmental conditions for human occupancy. American society of heating, Refrigerating and Air-Conditioning Engineering, Atlanta, USA, 2013.
- Fountain, M. & Huizenga, C. A thermal sensation model for use by the engineering profession ASHRAE RP-781 Final report, 1995
- Gagge, A. P., Fobelets, A. P. and Berglund, L. G. A standard predictive index of human response to the thermal environment, ASHRAE transactions, 1986, 92 (2B), 709-731.

See Also

see also [calcComfInd](#), [calcPtsadj](#)

Examples

```
## Calculation of a single set of values.
calc2Node(22, 25, .50, 50)

## Using several rows of data:
ta <- c(20,22,24)
tr <- ta
vel <- rep(.15,3)
rh <- rep(50,3)

maxLength <- max(sapply(list(ta, tr, vel, rh), length))
SET <- sapply(seq(maxLength), function(x) { calcSET(ta[x], tr[x], vel[x], rh[x]) } )
```

calcATHB

Calculation of Adaptive Thermal Heat Balance Indices

Description

calcATHB calculates three different indices related to the adaptive thermal heat balance framework

Usage

```
calcATHBpmv(trm, psych, ta, tr, vel, rh, met, wme)
calcATHBpts(trm, psych, ta, tr, vel, rh, met, wme, pb, ltime, ht, wt)
calcATHBset(trm, psych, ta, tr, vel, rh, met, wme, pb, ltime, ht, wt)
```

Arguments

trm	Running mean outdoor temperature in [degree C]
psych	factor related to fixed effect on perceived control
ta	a numeric value presenting air temperature in [degree C]
tr	a numeric value presenting mean radiant temperature in [degree C]
vel	a numeric value presenting air velocity in [m/s]
rh	a numeric value presenting relative humidity [%]
met	a numeric value presenting metabolic rate in [met]
wme	a numeric value presenting external work in [met]
pb	a numeric value presenting barometric pressure in [torr] or [mmHg]
ltime	a numeric value presenting exposure time in [minutes]
ht	a numeric value presenting body height in [cm]
wt	a numeric value presenting body weight in [kg]

Details

All variables must have the same length 1. For the calculation of several values use function calcComfInd.

Value

Depending on the type chosen calcATHB returns the following item:

ATHBpmv	pmv value adapted through the ATHB approach
ATHBset	set value adapted through the ATHB approach
ATHBpts	pts value adapted through the ATHB approach

Author(s)

Marcel Schweiker

References

Schweiker, M. & Wagner, A. A framework for an adaptive thermal heat balance model (ATHB), Building and Environment, 2015, 94, 252 - 262

Schweiker, M. & Wagner, A. Exploring potentials and limitations of the adaptive thermal heat balance framework Proceedings of 9th Windsor Conference: making comfort relevant Cumberland Lodge, Windsor, UK, 2016

See Also

see also [calcComfInd](#)

Examples

```
## Calculation of ATHBpmv
calcATHBpmv(20, 0, 25, 25, .1, 50, 1.1, 0)

## Calculation of ATHBpts
calcATHBpts(20, 0, 25, 25, .1, 50, 1.1, 0, 760, 60, 171, 70)

## Calculation of ATHBset
calcATHBset(20, 0, 25, 25, .1, 50, 1.1, 0, 760, 60, 171, 70)
```

calcAvgAcc	<i>Calculating Average Accuracy between Predicted and Actual Thermal Sensation Vote</i>
------------	---

Description

calcAvgAcc calculates the average accuracy between predicted thermal sensation votes and actual obtained sensation votes

Usage

```
calcAvgAcc(ref, pred)
```

Arguments

ref	a numeric item or vector containing categorical actual thermal sensation votes coded from -3 'cold' to +3 'hot'
pred	a numeric item or vector containing categorical predicted thermal sensation votes coded from -3 'cold' to +3 'hot'

Value

calcAvgAcc returns a single value presenting the average accuracy between actual and predicted thermal sensation votes.

Note

The outcome heavily depends on the distribution of actual votes, i.e. in case most of the actual votes are in the same category, e.g. 'neutral', the average accuracy is very high due to the fact that for the other categories the number of TRUE negative predicted votes is high as well.

Author(s)

Marcel Schweiker

References

Sokolova, M. and Lapalme, G. A systematic analysis of performance measures for classification tasks, Information Processing & Management, Elsevier, 2009, 45, 427-437

See Also

see also [calcTPRTSV](#), [calcMeanBias](#)

Examples

```
## Define data
ref <- rnorm(5) # actual thermal sensation votes
ref <- cutTSV(ref)

pred <- rnorm(5) # predicted thermal sensation votes
pred <- cutTSV(pred)

calcAvgAcc(ref, pred)
```

calcBias	<i>Calculating the Bias between Predicted and Actual Thermal Sensation Vote</i>
----------	---

Description

calcMeanBias calculates the mean bias and its standard deviation and standard error between predicted thermal sensation votes and actual obtained sensation votes

Usage

```
calcBias(ref, pred)
```

Arguments

ref	a numeric item or vector containing categorical or continuous actual thermal sensation votes coded from -3 'cold' to +3 'hot'
pred	a numeric item or vector containing categorical or continuous predicted thermal sensation votes coded from -3 'cold' to +3 'hot'

Value

calcMeanBias returns a dataframe with the following items:

meanBias	single value presenting the mean bias between actual and predicted thermal sensation votes
sdBias	single value presenting the standard deviation of the mean bias
seBias	single value presenting the standard error of the mean bias

Author(s)

Marcel Schweiker

References

Humphreys, M. A. and Nicol, J. F. The validity of ISO-PMV for predicting comfort votes in everyday thermal environments, *Energy and Buildings*, 2002, 34, 667-684

Schweiker, M. and Wagner, A. Exploring potentials and limitations of the adaptive thermal heat balance framework Proceedings of 9th Windsor Conference: Making Comfort Relevant Cumberland Lodge, Windsor, UK, 2016

See Also

see also [calcTPRTSV](#), [calcAvgAcc](#)

Examples

```
## Define data
ref <- rnorm(5) # actual thermal sensation votes

pred <- rnorm(5) # predicted thermal sensation votes

calcBias(ref, pred)
```

calcCOEFF

Calculation of Coefficients for aPMV, ePMV, aPTS, ePTS

Description

The functions calcCOEFF calculate the coefficients necessary for apmv, epmv, apts, and epts based on a given dataset with actual comfort votes. calcapCoeff calculates lambda the adaptive coefficient for apmv, calcepCoeff calculates e the expectancy factor for epmv, calcasCoeff calculates lambda the adaptive coefficient for apts, calcesCoeff calculates e the expectancy factor for epts.

Usage

```
calcapCoeff(lsCond)
calcepCoeff(lsCond)
calcasCoeff(lsCond)
calcesCoeff(lsCond)
```

Arguments

lsCond a list with vectors for the necessary variables (see details)

Details

For calcapCoeff and calcepCoeff, lsCond should contain the following variables: ta, tr, vel, rh, clo, met, wme, asv (see [createCond](#) for details). In case one or more of these variables are not included in the list, standard values will be used. For calcasCoeff and calcesCoeff, lsCond should contain the following variables: ta, tr, vel, rh, clo, met, wme, pb, ltime, ht, wt, asv (see [createCond](#) for details). In case one or more of these variables are not included in the list, standard values will be used.

Value

calcCOEFF returns the adaptive coefficient lambda or expectancy factor depending on its call.

Author(s)

Marcel Schweiker

References

Coefficients are calculated based on Gao, J.; Wang, Y. and Wargocki, P. Comparative analysis of modified PMV models and set models to predict human thermal sensation in naturally ventilated buildings *Building and Environment*, 2015, 92, 200-208.

The apmv concept was introduced by Yao, R.; Li, B. and Liu, J. A theoretical adaptive model of thermal comfort - Adaptive Predicted mean Vote (apmv) *Building and Environment*, 2009, 44, 2089-2096.

The epmv concept was introduced by Fanger, P. and Toftum, J. Extension of the PMV model to non-air-conditioned buildings in warm climates *Energy and Buildings*, 2002, 34, 533-536.

See Also

see also [calcaPMV](#), [calcePMV](#), [calcPtsa](#), [calcPtse](#)

Examples

```
## Note. Due to random generated asv values. The values for the coefficients will not be meaningful.
## Create sample data
ta <- 20:24      # vector with air temperature values
tr <- ta        # vector with radiant temperature values
vel <- rep(.1,5) # vector with air velocities
rh <- rep(50,5) # vector with relative humidity values
clo <- rep(1.0,5) # vector with clo values
met <- rep(1.1,5) # vector with metabolic rates
asv <- rnorm(5)  # vector with actual sensation votes

lsCond <- as.list(data.frame(ta, tr, vel, rh, clo, met, asv))

## Calculate coefficients

calcapCoeff(lsCond)
calcepCoeff(lsCond)
calcasCoeff(lsCond)
calcesCoeff(lsCond)

## use coefficients to calculate apmv
lsCond$apCoeff[1] <- calcapCoeff(lsCond)$apCoeff
calcComfInd(lsCond, request="apmv")
```

calcComfInd

Calculating one or more Thermal Comfort Indices using a List of Climatic Conditions

Description

calcComfInd calculates one or more thermal comfort indices using a list of climatic conditions.

Usage

```
calcComfInd(lsCond, request = "all")
```

Arguments

lsCond a list of climatic conditions and additional variables necessary for one or more of the indices (see details below).

request a vector with one or more comfort indices (see details below).

Details

The list lsCond could contain one or more of the following variables:

ta	Air temperature in (degree C)
tr	mean radiant temperature in (degree C)
vel	Air velocity in (m/s)
rh	Relative Humidity (%)
clo	clothing (clo)
met	metabolic rate (met)
wme	External work (met)
tu	turbulence intensity (%)
tmmo	mean monthly outdoor temperature in (degree C)
ltime	Exposure time (min)
pb	Barometric pressure (torr)
wt	weight (kg)
ht	height (cm)
trm	Running mean outdoor temperature in (degree C)
age	age (years)
gender	gender (female = 1)
tsk	mean skin temperature in (degree C)
psych	factor related to fixed effect on perceived control
apCoeff	adaptive coefficient for pmv
epCoeff	expectancy factor for pmv
asCoeff	adaptive coefficient for set
esCoeff	expectancy factor for set
asv	actual sensation vote (0 = neutral)
tao	outdoor air temperature
rho	outdoor relative humidity
frad	0.7(for seating), 0.73(for standing) [-]
eps	emissivity [-]
ic	1.084 (average permeability), 0.4 (low permeability)
ter	initial values for core temp
tsk	initial values for skin temperature
basMet	basal metabolic rate
warmUp	length of warm up period, i.e. number of times, loop is running for HBx calculation
cdil	value for cdil in 2-node model of Gagge (applied in calculation of HbEx)
sigmatr	value for cdil in 2-node model of Gagge (applied in calculation of HbEx)

In case a variable is not given, but necessary for the respective index, a standard value from a list of values is used.

The vector request can contain the following elements:

Element	Description
"all"	Calculation of all indices described below
"pmv"	Predicted mean vote
"ppd"	Predicted percentage dissatisfied
"tnhumphreys"	Neutral temperature according to Humphreys
"tAdapt15251"	Adaptive comfort temperature according to EN 15251
"dTNZ"	Distance to thermoneutral zone
"ATHBpmv"	Adaptive thermal heat balance vote based on pmv
"ATHBset"	Adaptive standard effective temperature
"ATHBpts"	Adaptive thermal heat balance vote based on set
"apmv"	Adaptive predicted mean vote according to Yao et al.
"ptsa"	Adaptive predicted thermal sensation vote according to Gao et al.
"epmv"	pmv adjusted with expectancy factor based on Fanger and toftum
"ptse"	Predicted thermal sensation vote based on set and adjusted with expectancy factor according to Gao et al.
"set"	standard effective temperature based on two node model by Gagge et al.
"et"	Effective temperature based on two node model by Gagge et al.
"tsens"	Predicted thermal sensation
"disc"	Predicted discomfort
"ps"	Predicted percentage satisfied with the level of air movement
"pd"	Predicted percentage dissatisfied due to draft
"pts"	Predicted thermal sensation vote based on set
"HBxst"	Human body exergy consumption rate using steady state method

Value

calcComfInd returns one or more rows with the comfort indices listed as request. For details see details above.

Note

In case one of the variables is not given, a standard value will be taken from a list (see [createCond](#) for details).

Author(s)

Sophia Mueller and Marcel Schweiker

References

For references see individual functions.

See Also

see also [calcPMVPPD](#), [calc2Node](#), [calcHbExSteady](#), [calcATHB](#), [calcdTNZ](#), [calcpmvadj](#), [calcPtsadj](#), [calctadapt](#)

Examples

```
## Creating list with all values
lsstrd <- createCond()

## Requesting all comfort indices
calcComfInd(lsstrd, request="all")

## Requesting a single index
calcComfInd(lsstrd, request="pmv")

## Requesting multiple indices
calcComfInd(lsstrd, request=c("pmv", "ptse"))
```

calcdTNZ

Calculating dTNZ, the Distance from the Thermoneutral Zone

Description

calcdTNZ calculates the distance from the thermoneutral zone, either skin temperature or room air related.

Usage

```
calcdTNZ(ht, wt, age, gender, clo, vel, tskObs, taObs, met, rh,
deltaT = 0.1, fBasMet = "rosa", fSA = "duBois")
```

Arguments

ht	a numeric value presenting body height in [cm]
wt	a numeric value presenting body weight in [kg]
age	a numeric value presenting the age in [years]
gender	a numeric value presenting sex (female = 1, male = 2)
clo	a numeric value presenting clothing insulation level in [clo]
vel	a numeric value presenting air velocity in [m/s]
tskObs	a numeric value presenting actual mean skin temperature in [degree C]
taObs	a numeric value presenting air temperature in [degree C]
met	a numeric value presenting metabolic rate (activity related) in [met]
rh	a numeric value presenting relative humidity in [
deltaT	a numeric value presenting the resolution of the matrix to be used
fBasMet	a string presenting the method of calculating basal metabolic rate. Needs to be one of "rosa", "harris", "miflin", or "fixed". Fixed will result in the value of 58.2 W/m ² .
fSA	a string presenting the method of calculating the surface area. Needs to be one of "duBois" or "mosteller".

Value

calcdTNZ returns a dataframe with the columns dTNZ, dTNZTs, dTNZTa. Thereby

dTNZ	The absolute distance to the centroid of the thermoneutral zone
dTNZTs	Relative value of distance assuming skin temperature to be dominant for sensation
dTNZTa	Relative value of distance assuming ambient temperature to be dominant for sensation

Note

In case one of the variables is not given, a standard value will be taken from a list (see [createCond](#) for details).

Author(s)

Marcel Schweiker and Boris Kingma

References

Kingma, B. R., Schweiker, M., Wagner, A. and van Marken Lichtenbelt, W. D. Exploring the potential of a biophysical model to understand thermal sensation Proceedings of 9th Windsor Conference: Making Comfort Relevant Cumberland Lodge, Windsor, UK, 2016.

Kingma, B. and van Marken Lichtenbelt, W. Energy consumption in buildings and female thermal demand Nature Clim. Change, 2015, 5, 1054 - 1056.

Kingma, B. R.; Frijns, A. J.; Schellen, L. and van Marken Lichtenbelt, W. D. Beyond the classic thermoneutral zone Temperature, 2014, 1, 142-149.

See Also

see also [calcComfInd](#)

Examples

```
## Calculate all values
calcdTNZ(171, 71, 45, 1, .6, .12, 37.8, 25.3, 1.1, 50)
```

calcHbExSteady	<i>Calculates Human Body Exergy Consumption Rate Using Steady State Method</i>
----------------	--

Description

calcHbExSteady calculates the human body exergy consumption rate in W/m² using steady state method based on a set of environmental variables.

Usage

```
calcHbExSteady(ta, tr, rh, vel, clo, met, tao, rho, frad = 0.7, eps = 0.95, ic = 1.085,
ht = 171, wt = 70, tcr = 37, tsk = 36, basMet = 58.2, warmUp = 60, cdil = 100,
sigmatr = 0.25)
```

Arguments

ta	a numeric value presenting air temperature in [degree C]
tr	a numeric value presenting mean radiant temperature in [degree C]
vel	a numeric value presenting air velocity in [m/s]
rh	a numeric value presenting relative humidity [%]
clo	a numeric value presenting clothing insulation level in [clo]
met	a numeric value presenting metabolic rate in [met]
tao	a numeric value presenting outdoor air temperature in [degree C]
rho	a numeric value presenting outdoor relative humidity [%]
frad	a numeric value presenting the fraction of body exposed to radiation 0.7(for seating), 0.73(for standing) [-]
eps	a numeric value presenting emissivity [-]
ic	a numeric value presenting permeability of clothing: 1.084 (average permeability), 0.4 (low permeability)
ht	a numeric value presenting body height in [cm]
wt	a numeric value presenting body weight in [kg]
tcr	a numeric value presenting initial value for core temperature in [degree C]
tsk	a numeric value presenting initial value for skin temperature in [degree C]
basMet	a numeric value presenting basal metabolic rate in [met]
warmUp	a numeric value presenting length of warm up period, i.e. number of times, loop is running for HBx calculation
cdil	a numeric value presenting value for cdil in 2-node model of Gagge
sigmatr	a numeric value presenting value for cdil in 2-node model of Gagge

Value

Returns a data.frame with the following columns

Exergy input

xInmets	Exergy input through metabolism
xInmetwcs	Label warm/ cold for exergy input through metabolism
xInAIRwcs	Exergy input through inhaled humid air
xInAIRwcwcs	Label warm/ cold for exergy input through inhaled humid air
xInAIRwds	Exergy input through inhaled dry air
xInAIRwdwds	Label wet/ dry for exergy input through inhaled dry air

xInLUNGwcs	Exergy input through water lung
xInLUNGwcwcs	Label warm/ cold for exergy input through water lung
xInLUNGwds	Exergy input through water lung
xInLUNGwdwds	Label wet/ dry for exergy input through water lung
xInshellwcs	Exergy input through water from sweat
xInshellwcwcs	Label warm/ cold for exergy input through water from sweat
xInshellwds	Exergy input through water from sweat
xInshellwdwds	Label wet/ dry for exergy input through water from sweat
xInraDs	Exergy input through radiation
xInraDwcs	Label warm/ cold for exergy input through radiation
xIntotalS	total exergy input
Exergy output	
xoutstorecores	Exergy stored in core
xoutstoreshels	Exergy stored in shell
xoutaIRwcs	Exergy output through exhaled humid air
xoutaIRwcwcs	Label warm/ cold for exergy output through exhaled humid air
xoutaIRwds	Exergy output through exhaled dry air
xoutaIRwdwds	Label wet/ dry for exergy output through exhaled dry air
xoutswEATwcs	Exergy output through water vapour from sweat
xoutswEATwcwcs	Label warm/ cold for exergy output through water vapour from sweat
xoutswEATwds	Exergy output through water vapour from sweat
xoutswEATwdwds	Label wet/ dry for exergy output through water vapour from sweat
xoutraDs	Exergy output through radiation
xoutraDwcs	Label warm/ cold for exergy output through radiation
xoutCONVs	Exergy output through convection
xoutCONVwcs	Label warm/ cold for exergy output through convection
xouttotalS	total exergy output
Exergy balance	
xconss	total exergy consumption
xConsumption	total exergy consumption
Additional values	
tsks	Calculated skin temperature
tcrs	Calculated core temperature
ws	Calculated skin wettedness

Note

According to Gagge's paper (1973), the value of 'cdil' may vary between 75 and 225 and 'sigma-tr' between 0.25 and 0.75. There is a note in the appendix of his paper saying two things: 1) whatever the values taken for cdil and sigma-tr, there must be no significant change in resulting thermal equilibrium. But, the values taken for cdil and sigmaTr do affect time to equilibrium. According to the analysis of schweiker et al. (2015), the values of 100 and .25 lead to the best fit of calculated and observed skin temperature.

Author(s)

This function is based on a VBA code developed by Masanori Shukuya. transformation of VBA-code and Excel procedures into R syntax by Marcel Schweiker.

References

Schweiker, M., Kolarik, J., Dovjak, M. and Shukuya, M. Unsteady-state human-body exergy consumption rate and its relation to subjective assessment of dynamic thermal environments, *Energy and Buildings* , 2016, 116, 164 - 180

Shukuya, M. Calculation of human body-core and skin-layer temperatures under unsteady-state conditions-for unsteady-state human-body exergy analysis-, internal report of exergy-research group, Tech. rep., KIT/TCU, 2015.

See Also

see also [calcComfInd](#), [calcHbExUnsteady](#)

Examples

```
## Calculation of human body exergy consumption rate
calcHbExSteady(22, 24, 50, .1, .8, 1.2, 5, 80)

## Calculation of multiple values
dfData <- data.frame(ta=c(20:25), tr=c(20:25))
dfResult <- calcHbExSteady(22, 24, 50, .1, .8, 1.2, 5, 80)
for(i in 1:nrow(dfData)){
  dfResult[i,] <- calcHbExSteady(dfData$ta[i], dfData$tr[i], 50, .1, .5, 1.1, 5, 80)
}
```

calcHbExUnsteady

Calculates Human Body Exergy Consumption Rate using Unsteady State Method

Description

calcHbExUnsteady calculates the human body exergy consumption rate using unsteady state method based on a series of environmental variables.

Usage

```
calcHbExUnsteady(ta, tr, rh, vel, clo, met, tao, rho, frad = 0.7, eps = 0.95,
ic = 1.085, ht = 171, wt = 70, tcr = 37, tsk = 36, basMet = 58.2, warmUp = 60,
cdil = 100, sigmatr = 0.25, dateTime)
```

Arguments

ta	a numeric vector presenting air temperature in [degree C]
tr	a numeric vector presenting mean radiant temperature in [degree C]
vel	a numeric vector presenting air velocity in [m/s]
rh	a numeric vector presenting relative humidity [%]
clo	a numeric vector presenting clothing insulation level in [clo]
met	a numeric vector presenting metabolic rate in [met]
tao	a numeric vector presenting outdoor air temperature in [degree C]
rho	a numeric vector presenting outdoor relative humidity [%]
frad	a numeric vector presenting the fraction of body exposed to radiation 0.7(for seating), 0.73(for standing) [-]
eps	a numeric vector presenting emissivity [-]
ic	a numeric vector presenting permeability of clothing: 1.084 (average permeability), 0.4 (low permeability)
ht	a numeric vector presenting body height in [cm]
wt	a numeric vector presenting body weight in [kg]
tcr	a numeric vector presenting initial value for core temperature in [degree C]
tsk	a numeric vector presenting initial value for skin temperature in [degree C]
basMet	a numeric vector presenting basal metabolic rate in [met]
warmUp	a numeric vector presenting length of warm up period, i.e. number of times, loop is running for HBx calculation
cdil	a numeric vector presenting value for cdil in 2-node model of Gagge
sigmatr	a numeric vector presenting value for cdil in 2-node model of Gagge
dateTime	a POsIxct vector of the times of measurement

Details

This function requires vectors of data including the corresponding time stamp. In case the time between two measurements is more than a minute, intermediate values are interpolated.

Value

Returns a data.frame with the following columns

Exergy input

xInmetu Exergy input through metabolism

xInmetwcu	Label warm/ cold for exergy input through metabolism
xInAIRwcu	Exergy input through inhaled humid air
xInAIRwccwu	Label warm/ cold for exergy input through inhaled humid air
xInAIRwdu	Exergy input through inhaled dry air
xInAIRwdwdu	Label wet/ dry for exergy input through inhaled dry air
xInLUNGwcu	Exergy input through water lung
xInLUNGwccwu	Label warm/ cold for exergy input through water lung
xInLUNGwdu	Exergy input through water lung
xInLUNGwdwdu	Label wet/ dry for exergy input through water lung
xInshellwcu	Exergy input through water from sweat
xInshellwccwu	Label warm/ cold for exergy input through water from sweat
xInshellwdu	Exergy input through water from sweat
xInshellwdwdu	Label wet/ dry for exergy input through water from sweat
xInraDu	Exergy input through radiation
xInraDwcu	Label warm/ cold for exergy input through radiation
xIntotalu	total exergy input
Exergy output	
xoutstorecoreu	Exergy stored in core
xoutstoreshelu	Exergy stored in shell
xoutaIRwcu	Exergy output through exhaled humid air
xoutaIRwccwu	Label warm/ cold for exergy output through exhaled humid air
xoutaIRwdu	Exergy output through exhaled dry air
xoutaIRwdwdu	Label wet/ dry for exergy output through exhaled dry air
xoutswEATwcu	Exergy output through water vapour from sweat
xoutswEATwccwu	Label warm/ cold for exergy output through water vapour from sweat
xoutswEATwdu	Exergy output through water vapour from sweat
xoutswEATwdwdu	Label wet/ dry for exergy output through water vapour from sweat
xoutraDu	Exergy output through radiation
xoutraDwcu	Label warm/ cold for exergy output through radiation
xoutCONVu	Exergy output through convection
xoutCONVwcu	Label warm/ cold for exergy output through convection
xouttotalu	total exergy output
Exergy balance	

xconsu	total exergy consumPtion
Additional values	
tsku	Calculated skin temperature
tcru	Calculated core temperature
wu	Calculated skin wettedness

Note

According to Gagge's paper (1973), the value of 'cdil' may vary between 75 and 225 and 'sigma-tr' between 0.25 and 0.75. There is a note in the appendix of his paper saying two things: 1) whatever the values taken for cdil and sigma-tr, there must be no significant change in resulting thermal equilibrium. But, the values taken for cdil and sigmaTr do affect time to equilibrium. According to the analysis of schweiker et al. (2015), the values of 100 and .25 lead to the best fit of calculated and observed skin temperature.

Author(s)

This function is based on a VBA code developed by masanori Shukuya. transformation of VBA-code and Excel procedures into R syntax by Marcel Schweiker.

References

- Schweiker, M., Kolarik, J., Dovjak, M. and Shukuya, M. Unsteady-state human-body exergy consumption rate and its relation to subjective assessment of dynamic thermal environments, *Energy and Buildings* , 2016, 116, 164 - 180
- Shukuya, M. Calculation of human body-core and skin-layer temperatures under unsteady-state conditions-for unsteady-state human-body exergy analysis-, internal report of exergy-research group, Tech. rep., KIT/TCU, 2015.

See Also

see also [calcComfInd](#), [calcHbExSteady](#)

Examples

```
## Define environmental parameters
ta <- seq(20,25,.1)
tr <- ta
rh <- rep(50, length(ta))
vel <- rep(.1, length(ta))
clo <- rep(.8, length(ta))
met <- rep(1.2, length(ta))
tao <- rep(5, length(ta))
rho <- rep(80, length(ta))
dateTime <- as.POSIXct(seq(0,by=60,length.out=length(ta)), origin="1970-01-01")

## Calculation of human body exergy consumPtion rate
calcHbExUnsteady(ta, tr, rh, vel, clo, met, tao, rho, dateTime = dateTime)$xconsu
```

`calcHumidity`*Calculating Various Humidity Related Values*

Description

This set of functions calculates different humidity related values based on the given entities.

Usage

```
calcDewp(ta, rh)
calcEnth(ta, rh, pb)
calcHumx(ta, rh)
calcMixR(ta, rh, pb)
calcRH(ta, mr, pb)
calcSVP(ta)
calcVP(ta, mr, pb)
```

Arguments

<code>ta</code>	a numeric value or vector presenting air temperature in [degree C]
<code>rh</code>	a numeric value or vector presenting relative humidity in [%]
<code>mr</code>	a numeric value or vector presenting the mixIng ratio in [g/kg]
<code>pb</code>	a numeric value or vector presenting barometric pressure in [torr]

Details

The length of the arguments must be either the same or they must have the length one and one common second length.

Value

`calcDewp` returns the dew point temperature in [degree C]
`calcEnth` returns a single value or a vector of values of enthalpy in [J]
`calcHumx` returns a single value or a vector of values of the humidex of air []
`calcMixR` returns a single value or a vector of mixIng ratio in [g/kg]
`calcRH` returns a single value or a vector of relative humidities in [%]
`calcSVP` returns a single value or a vector of saturation vapor pressure in [kpa]
`calcVP` returns a single value or a vector of vapor pressure in [kpa]

Author(s)

Michael Kleber (code and documentation), Marcel Schweiker (documentation)

References

For references related to humidex, see: Rajib Ranaa, Brano Kusya, Raja Jurdaka, Josh Wallb and Wen Hua, Feasibility analysis of using humidex as an indoor thermal comfort predictor, Energy and Buildings 64 (2013) 17-25.

Masterton, J. M., and Richardson, F. A., Humidex a method of quantifying human discomfort due to excessive heat and humidity, cII 1-79. Downsview, Ont: Environment Canada. Atmospheric Environment Service, 1979.

Examples

```
## Calc single value of absolute humidity
ta <- 25
rh <- 50
calcMixR(ta, rh, 760)

## Calc set of values of absolute humidity
ta <- 25:30
rh <- 50
calcMixR(ta, rh, 760)

## Calculating dew point temperature with single values for ta and rh
calcDewp(25, 50)

## Calculating dew point temperature with a vector of values for ta and a single value for rh
calcDewp(25:29, 50)

## Calc single value of enthalpy
ta <- 25
rh <- 50
calcEnth(ta, rh, 760)

## Calc set of values of enthalpy
ta <- 25:30
rh <- 50
calcEnth(ta, rh, 760)
```

calcpmvadj

Calculating Adjusted Predicted Mean Votes

Description

calcpmvadj calculates predicted mean votes (pmv) adjusted for cooling effect of elevated air speed, through the adaptive coefficient, or the expectancy factor.

Usage

```
calcpmvadj(ta, tr, vel, rh, clo, met, wme = 0)
calcaPMV(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, apCoeff)
calcePMV(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, epCoeff)
```

Arguments

ta	a numeric value presenting air temperature in [degree C]
tr	a numeric value presenting mean radiant temperature in [degree C]
vel	a numeric value presenting air velocity in [m/s]
rh	a numeric value presenting relative humidity [%]
clo	a numeric value presenting clothing insulation level in [clo]
met	a numeric value presenting metabolic rate in [met]
wme	a numeric value presenting external work in [met]
apCoeff	adaptive coefficient lambda
epCoeff	expectancy factor e

Details

apCoeff can be derived using calcapCoeff.

epCoeff can be derived using calcepCoeff.

calcePMV requires the actual sensation vote related to the physical data as it is required to alter the metabolic rate.

Value

calcpmvadj returns the predicted mean vote adjusted for the cooling effect of elevated air speed.

calcaPMV returns the predicted mean vote adjusted through the adaptive coefficients.

calcePMV returns the predicted mean vote adjusted through the expectancy factor.

Note

In case one of the variables is not given, a standard value will be taken from a list (see [createCond](#) for details).

Author(s)

Marcel Schweiker. The function used for bisection method is taken from a forum entry and with permission by ravi Varadhan.

References

pmvadj is based on ASHRAE standard 55-2013. Thermal environmental conditions for human occupancy. American society of heating, Refrigerating and Air-Conditioning Engineering, Atlanta, Usa, 2013

apmv is based on Yao, R., Li, B. and Liu, J. A theoretical adaptive model of thermal comfort - Adaptive Predicted mean Vote (aPMV) Building and Environment, 2009, 44, 2089-209

epmv is based on Fanger, P. and Toftum, J. Extension of the PMV model to non-air-conditioned buildings in warm climates Energy and Buildings, 2002, 34, 533-536

See Also

see also [calcComfInd](#), [calcapCoeff](#), [calcepCoeff](#)

Examples

```
## Note. Due to random generated asv values. The values for the coefficients will not be meaningful.
## Create sample data
ta <- 20:24 # vector with air temperature values
tr <- ta # vector with radiant temperature values
vel <- rep(.1,5) # vector with air velocities
rh <- rep(50,5) # vector with relative humidity values
clo <- rep(1.0,5) # vector with clo values
met <- rep(1.1,5) # vector with metabolic rates
asv <- rnorm(5) # vector with actual sensation votes

lsCond <- as.list(data.frame(ta, tr, vel, rh, clo, met, asv))

## Calculate coefficient apCoeff for data set
apCoeff <- calcapCoeff(lsCond)

## calculate apmv
apmv <- NULL
for (i in 1:length(ta)){
  apmv[i] <- calcaPMV(ta[i], tr[i], vel[i], rh[i], clo[i], met[i], apCoeff = apCoeff)$apmv
}
apmv
```

calcPMVPPD

Calculating PMV and PPD

Description

Functions to calculate PMV and/or PPD indices according to Fanger.

Usage

```
calcPMVPPD(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, basMet = 58.15)
calcPMV(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, basMet = 58.15)
calcPPD(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, basMet = 58.15)
```

Arguments

ta	a numeric value presenting air temperature in [degree C]
tr	a numeric value presenting mean radiant temperature in [degree C]
vel	a numeric value presenting air velocity in [m/s]
rh	a numeric value presenting relative humidity [%]

clo	a numeric value presenting clothing insulation level in [clo]
met	a numeric value presenting metabolic rate in [met]
wme	a numeric value presenting external work in [met]
basMet	a numeric value presenting basal metabolic rate [w/m2]

Details

All variables must have the same length 1. For the calculation of several values use function `calcComfInd` or a loop. Note that the adjustments in the value for `basMet` need to be made with great cautiousness as the PMV calculation is an empirical model and might not be valid for other values of `basMet` than the one commonly used.

Value

`calcPMVPPD` returns a data.frame with the following items:

pmv	Predicted mean vote
ppd	Predicted percentage dissatisfied

`calcPMV` and `calcPPD` return a single value of PMV or PPD.

Author(s)

Code implemented in to R by Marcel Schweiker. Further contribution by Sophia Mueller.

References

Fanger, P. O. Thermal Comfort Analysis and Applications in Environmental Engineering McGraw-Hill, New York, 1970.

ISO 7730 Ergonomics of the thermal environment analytical determination and interpretation of thermal comfort using calculation of the pmv and ppd indices and local thermal comfort criteria 2005.

See Also

see also [calcComfInd](#)

Examples

```
## Calculating PMV and PPD
calcPMVPPD(25, 20, .2, 50)

## Using several rows of data:
ta <- c(20,22,24)
tr <- ta
vel <- rep(.15,3)
rh <- rep(50,3)

maxLength <- max(sapply(list(ta, tr, vel, rh), length))
PMV <- sapply(seq(maxLength), function(x) { calcPMV(ta[x], tr[x], vel[x], rh[x]) } )
```

calcPtsadj	<i>Calculating the Predicted Thermal Sensation based on 2-Node Model adjusted for Adaptation or Expectancy</i>
------------	--

Description

calcpts are two functions calculating predicted thermal sensation votes (pts) based on Gagges two-node model and adjusted either through the adaptive coefficient lambda or the expectancy factor e.

Usage

```
calcPtsa(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, pb = 760, ltime = 60, ht = 171,
wt = 70, tu = 40, asCoeff)
calcPtse(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, pb = 760, ltime = 60, ht = 171,
wt = 70, tu = 40, esCoeff)
```

Arguments

ta	a numeric value presenting air temperature in [degree C]
tr	a numeric value presenting mean radiant temperature in [degree C]
vel	a numeric value presenting air velocity in [m/s]
rh	a numeric value presenting relative humidity [%]
clo	a numeric value presenting clothing insulation level in [clo]
met	a numeric value presenting metabolic rate in [met]
wme	a numeric value presenting external work in [met]
pb	a numeric value presenting barometric pressure in [torr] or [mmHg]
ltime	a numeric value presenting exposure time in [minutes]
ht	a numeric value presenting body height in [cm]
wt	a numeric value presenting body weight in [kg]
tu	a numeric value presenting the turbulence intensity (value does not affect outcome)
asCoeff	adaptive coefficient lambda
esCoeff	expectancy factor e

Details

asCoeff can be derived using calcasCoeff.

esCoeff can be derived using calcesCoeff.

calcPtse requires the actual sensation vote related to the physical data as it is required to alter the metabolic rate.

Value

calcPtsa returns the predicted thermal sensation vote adjusted through the adaptive coefficients.

calcPtse returns the predicted thermal sensation vote adjusted through the expectancy factor.

Note

In case one of the variables is not given, a standard value will be taken from a list (see [createCond](#) for details).

Author(s)

Marcel Schweiker

References

Gao, J., Wang, Y. and Wargocki, P. Comparative analysis of modified pmv models and set models to predict human thermal sensation in naturally ventilated buildings, *Building and Environment*, 2015, 92, 200-208.

See Also

see also [calcComfInd](#), [calcasCoeff](#), [calcesCoeff](#)

Examples

```
## Note. Due to random generated asv values. The values for the coefficients will not be meaningful.
## Create sample data
ta <- 20:24      # vector with air temperature values
tr <- ta        # vector with radiant temperature values
vel <- rep(.1,5) # vector with air velocities
rh <- rep(50,5) # vector with relative humidity values
clo <- rep(1.0,5) # vector with clo values
met <- rep(1.1,5) # vector with metabolic rates
asv <- rnorm(5)  # vector with actual sensation votes

lsCond <- as.list(data.frame(ta, tr, vel, rh, clo, met, asv))

## Calculate coefficient asCoeff for data set
asCoeff <- calcasCoeff(lsCond)

## calculate ptsa
ptsa <- NULL
for (i in 1:length(ta)){
  ptsa[i] <- calcPtsa(ta[i], tr[i], vel[i], rh[i], clo[i], met[i], asCoeff = asCoeff)$ptsa
}
ptsa
```

calctadapt

Calculating Adaptive Comfort Temperatures or Neutral Temperatures

Description

calctadapt are three functions to calculate adaptive comfort or neutral temperatures based on a given outdoor temperature value.

Usage

```
calctAdapt15251(trm = 20)
calctAdaptASHRAE(tmmo)
calctnAuliciems(ta, tmmo)
calctnHumphreysNV(tmmo)
calctnHumphreysAC(tmmo)
```

Arguments

ta	numerical value presenting the indoor air temperature in [degree C]
trm	numerical value presenting the running mean outdoor temperature in [degree C]
tmmo	numerical value presenting the mean monthly outdoor temperature in [degree C]

Value

returns the adaptive comfort or neutral temperature with respect to the given outdoor temperature value

Note

The difference between calctnHumphreysNV and calctnHumphreysAC is that the former was found for natural ventilated buildings (NV), while the latter was found for climate-controlled buildings (AC).

Author(s)

Marcel Schweiker

References

calctAdapt15251 is based on DIN EN 15251 Indoor environmental input parameters for design and assessment of energy performance of buildings addressing indoor air quality, thermal environment, lighting and acoustics; German version EN 15251:2012 2012.

calcAdaptASHRAE is based on Brager, G. S. & de Dear, R. Climate, comfort, & natural ventilation: a new adaptive comfort standard for ASHRAE standard 55 Center for the Built Environment. UC Berkeley, 2001.

calctnAuliciems is based on Auliciems, A. Psycho-physiological criteria for global thermal zones of building design Int J Biometeorol, springer, 1981, 26, 69-86.

calctnHumphreysNV and calctnHumphreysAC are based on Humphreys, M. A., Outdoor temperatures and comfort indoors. Batiment International, Building Research and Practice, Taylor and Francis, 1978, 6, 92-92.

See Also

see also [calcComfInd](#)

Examples

```
## define variable
trm <- 21.2

## calculate adaptive comfort temperature
calctAdapt15251(trm)
```

calcTPRTSV	<i>Calculating the True Positive Rate between Predicted and Actual Thermal Sensation Vote</i>
------------	---

Description

calcTPRTSV calculates the true positive rate between predicted thermal sensation votes and actual obtained sensation votes

Usage

```
calcTPRTSV(ref, pred)
```

Arguments

ref	a numeric item or vector containing categorical actual thermal sensation votes coded from -3 'cold' to +3 'hot'
pred	a numeric item or vector containing categorical predicted thermal sensation votes coded from -3 'cold' to +3 'hot'

Value

calcTPRTSV returns a single value presenting the true positive rate between actual and predicted thermal sensation votes.

Author(s)

Marcel Schweiker

References

Schweiker, M. and Wagner, A. A framework for an adaptive thermal heat balance model (ATHB) Building and Environment, 2015, 94, 252-262

See Also

see also [calcMeanBias](#), [calcAvgAcc](#)

Examples

```
## Define data
ref <- rnorm(5) # actual thermal sensation votes
ref <- cutTSV(ref)

pred <- rnorm(5) # predicted thermal sensation votes
pred <- cutTSV(pred)

calcTPRTSV(ref, pred)
```

calcTroin

Calculation of Radiative and Operative Temperature

Description

The functions calcTroin calculates radiative and operative temperature based on air temperature, globe temperature, air velocity and metabolic rate. Globe temperature needs to be measured using a standard globe with a diameter of 0.15m and an emissivity of .95 (black coloured).

Usage

```
calcTroin(vel, tg, ta, met)
```

Arguments

vel	numeric value or vector presenting the air velocity in [m/s]
tg	numeric value or vector presenting the globe temperature in [degree C]
ta	numeric value or vector presenting the air temperature in [degree C]
met	numeric value or vector presenting the metabolic rate in [met]

Details

Calculation of the radiative temperature is based on ISO 7726:2001, equation (9). Calculation of operative temperature is based on ISO 7726:2001, Appendix G.3. The adjustment of air velocity to present relative air velocity based on metabolic rate is based on ISO 7730:2005 Appendix C.2.

Value

calcTroin returns a data.frame with radiative and operative temperature.

Author(s)

Marcel Schweiker

References

ISO 7726 Ergonomics of the Thermal Environment, Instruments for measuring Physical Quantities
Geneva: International standard Organization, 1998

ISO 7730 Ergonomics of the thermal environment - analytical determination and interpretation of thermal comfort using calculation of the pmv and ppd indices and local thermal comfort criteria
2005

Examples

```
## Note. Due to random generated asv values. The values for the coefficients will not be meaningful.
## Create sample data
ta <- 20:24      # vector with air temperature values
vel <- rep(.1,5) # vector with air velocities
met <- rep(1.1,5) # vector with metabolic rates
tg <- 25:29      # vector with globe temperature values

calcTroin(vel, tg, ta, met)
```

createCond

Creating a List with Standard Values

Description

createCond creates a list with standard variables to be used as an input parameter for calculating comfort indices using the function "calcComfInd".

Usage

```
createCond(a = TRUE)
```

Arguments

a logical. If a = TRUE, function returns a list of standard conditions. If a = FALSE, function returns a list of empty variables which may be edited manually. see details for further information.

Details

lsstrd and lsEmpty contain the following elements

Variable name	values in lsstrd	values in lsEmpty	description
ta	25	NA	Air temperature in (degree C)
tr	25	NA	mean radiant temperature in (degree C)
vel	.1	NA	Air velocity (m/s)
rh	50	NA	Relative Humidity (%)
clo	.5	NA	clothing (do)
met	1	NA	metabolic rate (met)

wme	0	NA	External work (met)
tu	40	NA	turbulence intensity (%)
tmmo	15	NA	mean monthly outdoor temperature in (degree C)
ltime	60	NA	Exposure time (min)
pb	760	NA	Barometric pressure (torr)
wt	70	NA	weight (Kg)
ht	171	NA	height (cm)
trm	15	NA	Running mean outdoor temperature in (degree C)
age	21	NA	age (years)
gender	1	NA	gender (female = 1)
tsk	35	NA	mean skin temperature in (degree C)
psych	-1.4	NA	factor related to fixed effect on perceived control
apCoeff	.293	NA	adaptive coefficient for pmv
epCoeff	.9	NA	expectancy factor for pmv
asCoeff	.2	NA	adaptive coefficient for set
esCoeff	1.3	NA	expectancy factor for set
asv	1.5	NA	actual sensation vote (0 = neutral)
tao	5	NA	outdoor air temperature
rho	70	NA	outdoor relative humidity
frad	.7	NA	0.7(for seating), 0.73(for standing) [-]
eps	.95	NA	emissivity [-]
ic	1.085	NA	1.084 (average permeability), 0.4 (low permeability)
tcr	37	NA	initial values for core temp
tsk	36	NA	initial values for skin temperature
basMet	58.2	NA	basal metabolic rate
warmUp	60	NA	length of warm up period, i.e. number of times, loop is running for H
cdil	100	NA	value for cdil in 2-node model of Gagge (applied in calculation of H
sigmatr	.25	NA	value for cdil in 2-node model of Gagge (applied in calculation of H

Value

lsstrd	List, which is created for a = TRUE; contains standard conditions.
lsEmpty	List, which is created for a = FALSE; contains empty variables to be modified manually.

Author(s)

Sophia Mueller and Marcel Schweiker

See Also

see also [calcComfInd](#)

Examples

```
## Creating list with standard variables
createCond()

## Creating list with empty values
```

```
createCond(a = FALSE)
```

cutTSV

Categorizing Thermal Sensation Votes

Description

cutTSV converts continuous thermal sensation votes to categorical ones.

Usage

```
cutTSV(pred)
```

Arguments

pred a numeric item or vector containing continuous thermal sensation votes coded from -3 'cold' to +3 'hot'

Details

Categorization is realized with intervals closed on the right, e.g. setting all values lower and equal then -2.5 to a value of -3, higher than -2.5 and lower or equal -1.5 to -2, and so on.

Value

cutTSV returns an item or a vector with categorical thermal sensation votes.

Author(s)

Marcel Schweiker

Examples

```
## define example data
pred <- rnorm(5)

## bin values
cutTSV(pred)
```

`dfASHRAETableG11`*Data from Table G1-1 of ASHRAE 55-2013*

Description

Calibration data for SET model (used for calculation of PMVadj) from ASHRAE 55-2013 Table G1-1

Usage

```
data(dfASHRAETableG11)
```

Format

A data frame with 22 rows on the following 11 variables.

`ta` a numeric vector of air temperature [degree C]

`taF` a numeric vector of air temperature [degree F]

`tr` a numeric vector of radiant temperature [degree C]

`trF` a numeric vector of radiant temperature [degree F]

`vel` a numeric vector of indoor air velocity [m/s]

`velFPM` a numeric vector of indoor air velocity [fpm]

`rh` a numeric vector of relative humidity [%]

`met` a numeric vector of metabolic rate [MET]

`clo` a numeric vector of clothing insulation level [CLO]

`set` a numeric vector of standard effective temperature (SET) [degree C]

`setF` a numeric vector of standard effective temperature (SET) [degree F]

Note

Note that rows 21 and 22 can only be compared to the function `calcSET` in case `obj="pmvadj"` is chosen. The value of `obj` defines whether the function will use the version presented in ASHRAE 55-2013 for adjustment of PMV (`obj = "pmvadj"`), or the original code by Gagge to calculate the standard effective temperature (SET) (`obj = "set"`). The reason is, that in the version presented in ASHRAE 55-2013, the lines of code related to self-generated convection are deleted. Therefore, a difference can be seen at higher values of `met`.

References

ASHRAE standard 55-2013. Thermal environmental conditions for human occupancy. American society of heating, Refrigerating and Air-Conditioning Engineering, Atlanta, Usa, 2013

Examples

```

data(dfASHRAETableG11)
head(dfASHRAETableG11)

# using option obj="pmvadj" in order to compare with ASHRAE Table G1-1
SET <- sapply(seq(nrow(dfASHRAETableG11)), function(x) { calcSET(dfASHRAETableG11$ta[x],
dfASHRAETableG11$tr[x], dfASHRAETableG11$vel[x], dfASHRAETableG11$rh[x], dfASHRAETableG11$clo[x],
dfASHRAETableG11$met[x], obj="pmvadj") } )
plot(SET~dfASHRAETableG11$set)
data.frame(set=dfASHRAETableG11$set, SET)

# using option obj="set" (standard) in order to calculate values for original SET values
SET2 <- sapply(seq(nrow(dfASHRAETableG11)), function(x) { calcSET(dfASHRAETableG11$ta[x],
dfASHRAETableG11$tr[x], dfASHRAETableG11$vel[x], dfASHRAETableG11$rh[x], dfASHRAETableG11$clo[x],
dfASHRAETableG11$met[x], ) } )
plot(SET2~dfASHRAETableG11$set)
data.frame(set=dfASHRAETableG11$set, SET2)

```

dfField

Data from Field Study Campaign

Description

Randomly sampled data from a field study campaign with data from 156 samples. For further description, see the reference given.

Usage

```
data(dfField)
```

Format

A data frame with 156 observations on the following 9 variables.

ta a numeric vector of air temperature [degree C]

tr a numeric vector of radiant temperature [degree C] - same as ta

rh a numeric vector of relative humidity [%]

trm a numeric vector of running mean outdoor temperature [degree C]

clo a numeric vector of clothing insulation level [CLO]

tao a numeric vector of outdoor air temperature [degree C]

vel a numeric vector of indoor air velocity [m/s]

met a numeric vector of metabolic rate [MET]

asv a numeric vector of actual thermal sensation vote on ASHRAE scale []

References

Schweiker, M. and Wagner, A. Exploring potentials and limitations of the adaptive thermal heat balance framework. Proceedings of 9th Windsor Conference: Making Comfort Relevant Cumberland Lodge, Windsor, UK, 2016.

Examples

```
data(dfField)
head(dfField)
```

dfISO7730AppE

Adjusted data from Appendix E of ISO 7730

Description

Calibration data for PMV/PPD model from ISO 7730 Appendix E

Usage

```
data(dfISO7730AppE)
```

Format

A data frame with 2963 rows on the following 6 variables.

`top` a numeric vector of operative temperature [degree C]

`vel` a numeric vector of indoor air velocity [m/s]

`rh` a numeric vector of relative humidity [%]

`met` a numeric vector of metabolic rate [MET]

`clo` a numeric vector of clothing insulation level [CLO]

`pmv` a numeric vector of Predicted mean vote (PMV)

Note

Several values differ from ISO 7730 Appendix E, as a comparison with the calculation done by this package, the values given by the CBE Thermal Comfort Tool (ASHRAE approved) and the values presented in the original source by Fanger showed a different value.

References

ISO 7730 Ergonomics of the thermal environment analytical determination and interpretation of thermal comfort using calculation of the pmv and ppd indices and local thermal comfort criteria 2005.

Fanger, P. O. Thermal Comfort Analysis and Applications in Environmental Engineering McGraw-Hill, New York, 1970.

Examples

```

data(dfISO7730AppE)
head(dfISO7730AppE)

PMV <- sapply(seq(nrow(dfISO7730AppE)), function(x) { calcPMV(dfISO7730AppE$top[x],
dfISO7730AppE$top[x], dfISO7730AppE$vel[x], dfISO7730AppE$rh[x], dfISO7730AppE$clo[x],
dfISO7730AppE$met[x]) } )

plot(PMV~dfISO7730AppE$pmv)

```

dfISO7730TableD1

Adjusted data from Table D1 of ISO 7730

Description

Calibration data for PMV / PPD model (used for calculation of PMVadj) from ISO 7730 Table D1

Usage

```
data(dfISO7730TableD1)
```

Format

A data frame with 13 rows on the following 8 variables.

ta a numeric vector of air temperature [degree C]

tr a numeric vector of radiant temperature [degree C]

vel a numeric vector of indoor air velocity [m/s]

rh a numeric vector of relative humidity [%]

met a numeric vector of metabolic rate [MET]

clo a numeric vector of clothing insulation level [CLO]

pmv a numeric vector of Predicted mean vote (PMV)

ppd a numeric vector of Predicted percentage dissatisfied (PPD)

Note

One value differs from ISO 7730 Table D1, as a comparison with the calculation done by this package and the values given by the CBE Thermal Comfort Tool (ASHRAE approved) showed a different value. The difference is: -row 7, PMV must be .36 (not .5) and PPD 8 (not 10)

References

ISO 7730 Ergonomics of the thermal environment analytical determination and interpretation of thermal comfort using calculation of the pmv and ppd indices and local thermal comfort criteria 2005.

Examples

```
data(dfISO7730TableD1)
head(dfISO7730TableD1)
```

```
PMV <- sapply(seq(nrow(dfISO7730TableD1)), function(x) { calcPMV(dfISO7730TableD1$ta[x],
dfISO7730TableD1$str[x], dfISO7730TableD1$vel[x], dfISO7730TableD1$rh[x], dfISO7730TableD1$clo[x],
dfISO7730TableD1$met[x]) } )
```

```
PPD <- sapply(seq(nrow(dfISO7730TableD1)), function(x) { calcPPD(dfISO7730TableD1$ta[x],
dfISO7730TableD1$str[x], dfISO7730TableD1$vel[x], dfISO7730TableD1$rh[x], dfISO7730TableD1$clo[x],
dfISO7730TableD1$met[x]) } )
```

```
plot(PMV~dfISO7730TableD1$pmv)
data.frame(pmv=dfISO7730TableD1$pmv, PMV)
```

```
plot(PPD~dfISO7730TableD1$ppd)
data.frame(ppd=dfISO7730TableD1$ppd, PPD)
```

Index

*Topic **datasets**

dfASHRAETableG11, 35
dfField, 36
dfISO7730AppE, 37
dfISO7730TableD1, 38

*Topic **manip**

calc2Node, 3
calcATHB, 6
calcAvgAcc, 7
calcBias, 9
calcCOEFF, 10
calcComfInd, 11
calcdTNZ, 14
calcHbExSteady, 15
calcHbExUnsteady, 18
calcHumidity, 22
calcpmvadj, 23
calcPMVPPD, 25
calcPtsadj, 27
calctadapt, 29
calcTPRTSV, 30
calcTroin, 31
createCond, 32
cutTSV, 34

*Topic **package**

comf-package, 2
2Node (calc2Node), 3

aPMV (calcpmvadj), 23
apmv (calcpmvadj), 23
ATHB (calcATHB), 6
athb (calcATHB), 6
ATHBpmv (calcATHB), 6
athbpmv (calcATHB), 6
ATHBpts (calcATHB), 6
athbpts (calcATHB), 6
ATHBset (calcATHB), 6
athbset (calcATHB), 6
AvgAcc (calcAvgAcc), 7
avgAcc (calcAvgAcc), 7

avgacc (calcAvgAcc), 7

bias (calcBias), 9

calc2Node, 3, 13

calcapCoeff, 25

calcapCoeff (calcCOEFF), 10

calcaPMV, 11

calcaPMV (calcpmvadj), 23

calcasCoeff, 28

calcasCoeff (calcCOEFF), 10

calcATHB, 6, 13

calcATHBpmv (calcATHB), 6

calcATHBpts (calcATHB), 6

calcATHBset (calcATHB), 6

calcAvgAcc, 7, 9, 31

calcBias, 9

calcCOEFF, 10

calcComfInd, 3, 6, 7, 11, 15, 18, 21, 25, 26,
28, 30, 33

calcDewp (calcHumidity), 22

calcDisc (calc2Node), 3

calcdTNZ, 13, 14

calcEnth (calcHumidity), 22

calcepCoeff, 25

calcepCoeff (calcCOEFF), 10

calcePMV, 11

calcePMV (calcpmvadj), 23

calcesCoeff, 28

calcesCoeff (calcCOEFF), 10

calcET (calc2Node), 3

calcHbExSteady, 13, 15, 21

calcHbExUnsteady, 18, 18

calcHumidity, 22

calcHumx (calcHumidity), 22

calcMeanBias, 8, 31

calcMeanBias (calcBias), 9

calcMixR (calcHumidity), 22

calcPD (calc2Node), 3

calcPMV (calcPMVPPD), 25

- calcpmvadj, [13](#), [23](#)
- calcPMVGagge (calc2Node), [3](#)
- calcPMVPPD, [13](#), [25](#)
- calcPMVStar (calc2Node), [3](#)
- calcPPD (calcPMVPPD), [25](#)
- calcPS (calc2Node), [3](#)
- calcPTS (calc2Node), [3](#)
- calcPtsa, [11](#)
- calcPtsa (calcPtsadj), [27](#)
- calcPtsadj, [6](#), [13](#), [27](#)
- calcPtse, [11](#)
- calcPtse (calcPtsadj), [27](#)
- calcRH (calcHumidity), [22](#)
- calcSdBias (calcBias), [9](#)
- calcSeBias (calcBias), [9](#)
- calcSET (calc2Node), [3](#)
- calcSkinWettedness (calc2Node), [3](#)
- calcSVP (calcHumidity), [22](#)
- calctadapt, [13](#), [29](#)
- calctAdapt15251 (calctadapt), [29](#)
- calctAdaptASHRAE (calctadapt), [29](#)
- calctnAuliciems (calctadapt), [29](#)
- calctnHumphreysAC (calctadapt), [29](#)
- calctnHumphreysNV (calctadapt), [29](#)
- calcTPRTSV, [8](#), [9](#), [30](#)
- calcTroin, [31](#)
- calcTSens (calc2Node), [3](#)
- calcVP (calcHumidity), [22](#)
- comf (comf-package), [2](#)
- comf-package, [2](#)
- createCond, [3](#), [5](#), [10](#), [13](#), [15](#), [24](#), [28](#), [32](#)
- cutTSV, [34](#)

- dfASHRAE (dfASHRAETableG11), [35](#)
- dfASHRAETableG11, [35](#)
- dfField, [36](#)
- dfISO7730 (dfISO7730TableD1), [38](#)
- dfISO7730AppE, [37](#)
- dfISO7730TableD1, [38](#)
- Disc (calc2Node), [3](#)
- disc (calc2Node), [3](#)
- dTNZ (calcdTNZ), [14](#)
- dtnz (calcdTNZ), [14](#)

- ePMV (calcpmvadj), [23](#)
- epmv (calcpmvadj), [23](#)
- ET (calc2Node), [3](#)
- et (calc2Node), [3](#)

- Fanger (calcPMVPPD), [25](#)

- Gagge (calc2Node), [3](#)

- HbExSteady (calcHbExSteady), [15](#)
- HBxst (calcHbExSteady), [15](#)
- hbxst (calcHbExSteady), [15](#)

- MeanBias (calcBias), [9](#)
- meanBias (calcBias), [9](#)

- PD (calc2Node), [3](#)
- pd (calc2Node), [3](#)
- PMV (calcPMVPPD), [25](#)
- pmv (calcPMVPPD), [25](#)
- pmvadj (calcpmvadj), [23](#)
- PPD (calcPMVPPD), [25](#)
- ppd (calcPMVPPD), [25](#)
- PS (calc2Node), [3](#)
- ps (calc2Node), [3](#)
- PTS (calc2Node), [3](#)
- pts (calc2Node), [3](#)
- Ptsa (calcPtsadj), [27](#)
- ptsa (calcPtsadj), [27](#)
- Ptsadj (calcPtsadj), [27](#)
- ptsadj (calcPtsadj), [27](#)
- Ptse (calcPtsadj), [27](#)
- ptse (calcPtsadj), [27](#)

- SET (calc2Node), [3](#)
- set (calc2Node), [3](#)

- tAdapt (calctadapt), [29](#)
- tadapt (calctadapt), [29](#)
- tAdapt15251 (calctadapt), [29](#)
- tAdaptASHRAE (calctadapt), [29](#)
- tnAuliciems (calctadapt), [29](#)
- tnHumphreysAC (calctadapt), [29](#)
- tnHumphreysNV (calctadapt), [29](#)
- TPR (calcTPRTSV), [30](#)
- tpr (calcTPRTSV), [30](#)
- Troin (calcTroin), [31](#)
- troin (calcTroin), [31](#)
- TSens (calc2Node), [3](#)
- tsens (calc2Node), [3](#)