# Package ‘comf’

July 22, 2025

**Type** Package

**Title** Models and Equations for Human Comfort Research

**Version** 0.1.12

**Date** 2024-01-29

**Imports** plyr, graphics, reshape, jsonlite

**Maintainer** Marcel Schweiker <mschweiker@ukaachen.de>

**Description** Calculation of various common and less common comfort indices such as predicted mean vote or the two node model. Converts physical variables such as relative to absolute humidity and evaluates the performance of comfort indices.

**License** GPL-2

**LazyLoad** yes

**LazyData** true

**Suggests** R.rsp

**VignetteBuilder** R.rsp

**RoxygenNote** 7.2.3

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Marcel Schweiker [aut, cre] (ORCID:
<https://orcid.org/0000-0003-3906-4688>),
Sophia Mueller [aut],
Michael Kleber [ctr],
Boris Kingma [ctr] (ORCID: <https://orcid.org/0000-0001-5961-0215>),
Masanori Shukuya [ctr] (ORCID: <https://orcid.org/0000-0003-2777-4375>),
Shaomi Rahman [ctr],
Shoaib Sarwar [ctr]

**Repository** CRAN

**Date/Publication** 2024-01-29 17:40:02 UTC

# Contents

---

`comf-package` *Calculation and Evaluation of Common and Less Common Comfort Indices*

---

### Description

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

### Details

| | |
|---|---|
| Package: | comf |
| Type: | Package |
| Version: | 0.1.12 |
| Date: | 2024-01-29 |
| 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 for example 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 precentage 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 |
| PHS | Predicted heat strain |

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 and many others.

Contact: mschweiker@ukaachen.de

## References

See references in function descriptions.

## See Also

see also `createCond`, `calcComfInd`

---

`calc2Node` *Comfort Indices based on the 2-Node-Model*

---

## Description

`calc2Node` calculates Comfort Indices based on the 2-Node-Model by Gagge et al.

## Usage

```
calc2Node(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, sa = NULL, pb = 760,
ltime = 60, ht = 171, wt = 70, tu = 40, obj = "set", csw = 170, cdil = 120,
cstr = 0.5, varOut = "else", bodyPosition = 'sitting')
```

## 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] |
| `sa` | (optional)surface Area according to mosteller formula [m^2] |
| `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 |
| `bodyPosition` | a string representing body position, has to be 'sitting' or 'standing'. Default value is 'sitting' |

## 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

returns a data.frame with the following items:

`et` - **Effective temperature**

`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**

`pmvg` - **Gagge's version of Fanger's PMV**

`pmvstar` - **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 & Huizenga (1995) A thermal sensation model for use by the engineering profession ASHRAE RP-781 Final report.

Gagge, Fobelets & Berglund (1986) A standard predictive index of human response to the thermal environment, ASHRAE transactions, 92 (2B), 709-731.

## See Also

see also `calcComfInd`

## Examples

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

calcAD *PPD with Ankle Draft*

## Description

Function to calculate ankle draft using the predicted percentage of dissatisfied.

## Usage

```
calcAD(ta, tr, vel, rh, clo, met, vAnkle)
```

## 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] |
| `vAnkle` | air speed at the 0.1 m (4 in.) above the floor [m/s] |

## Details

Calculates the percentage of thermally dissatisfied people with the ankle draft (0.1 m) above floor level.This equation is only applicable for velocity < 0.2 m/s (40 fps)

## Value

Predicted Percentage of Dissatisfied occupants with ankle draft in [%]

Acceptability in [boolean]

## Author(s)

Code implemented in to R by Shoaib Sarwar. Further contribution by Marcel Schweiker.

## References

Original code in Python by Tartarini & Schiavon (2020) <doi:10.1016/j.softx.2020.100578>

## Examples

```
calcAD(25,25,0.2,50,0.5,1.2,0.3) # returns Ankle_draft_ppd:18.6, Acceptability:TRUE
```

`calcaPMV` *Adaptive Predicted Mean Votes*

## Description

Function to calculate adaptive Predicted Mean Vote (aPMV) adjusted through the adaptive coefficient.

## Usage

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

## 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 |

## Details

`apCoeff` can be derived using `calcapCoeff`.

## Value

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

## Note

In case one of `apCoeff` is not given, a standard value will be taken from a list (see `createCond` for details.

## Author(s)

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

## References

aPMV is based on Yao, Li and Liu (2009) <doi:10.1016/j.buildenv.2009.02.014>

## See Also

calcComfInd, calcapCoeff

## 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
```

---

calcATHBpmv — *PMV based on Adaptive Thermal Heat Balance Framework of multiple Models*

---

## Description

calcATHBpmv calculates the PMV based on adaptive thermal heat balance framework

based on the newest version (2022)

## Usage

```
calcATHBpmv(trm, ta, tr, vel, rh, met,
coolingStrategyBuilding = 'naturallyventilated',
buildingTypeSimple = 'office', seasonSimple = 'spring')
```

## Arguments

| | |
|---|---|
| trm | - Running mean outdoor temperature in [degree C] |
| 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] |

`coolingStrategyBuilding`
: - the process in which the building was ventilated. Value can be among Mixed Mode','Naturally Ventilated' as a String

`buildingTypeSimple`
: - simple building type. Value can be among Multifamily housing, Office as a string

`seasonSimple`
: - season. Value can be among 'Spring','Summer', 'Winter' as a String

## Details

aliases athb ATHB calcATHBPmvModel1, calcATHBPmvModel2, calcATHBPmvModel3

## Value

`calcATHBpmv` an array of PMV values of different models adapted through the ATHB appoach

## Author(s)

Code implemented in to R by Shaomi Rahman. Further contribution by Marcel Schweiker.

## References

Schweiker & Wagner (2015) <doi:10.1016/j.buildenv.2015.08.018> Schweiker (2022) <doi:10.1111/ina.13018>

## See Also

see also `calcComfInd`, `link{calcATHBpts}`, `link{calcATHBset}`,

`link{calcATHBpmv2015}`, `link{calcATHBPmvModel1}`, `link{calcATHBPmvModel2}`, `link{calcATHBPmvModel3}`

## Examples

```
calcATHBpmv(20, 25, 25, .1, 50, 1.1, 'naturallyventilated','office',
'spring')
```

---

# `calcATHBpmv2015` *PMV based on Adaptive Thermal Heat Balance Framework*

---

## Description

`calcATHBpmv2015` calculates the PMV based on adaptive thermal heat balance framework

based on the original method published 2015

## Usage

```
calcATHBpmv2015(trm, psych, ta, tr, vel, rh, met, wme = 0)
```

### 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] |

### Details

aliases athb2015 ATHB2015 athbOld ATHBOLD

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

### Value

`calcATHBpmv2015` PMV value adapted through the ATHB appoach

### Author(s)

Marcel Schweiker

### References

Schweiker & Wagner (2015) <doi:10.1016/j.buildenv.2015.08.018> Schweiker & Wagner (2016) 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`, `link{calcATHBpts}`, `link{calcATHBset}`

### Examples

```
calcATHBpmv2015(20, 0, 25, 25, .1, 50, 1.1)
```

calcATHBpts *Predicted Thermal Sensation by Gagge using Adaptive Thermal Heat Balance approach*

## Description

calcATHB calculates predicted thermal sensation based on the adaptive thermal heat balance approach using Gagge's 2 Node Model

## Usage

```
calcATHBpts(trm, psych, ta, tr, vel, rh, met, wme = 0, pb = 760,
                      ltime = 60, ht = 171, wt = 69.9)
```

## 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

calcATHBpts returns the predicted thermal sensation adapted through the ATHB approach

## Author(s)

Marcel Schweiker

### References

Schweiker & Wagner (2015) <doi:10.1016/j.buildenv.2015.08.018> Schweiker & Wagner (2016) 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`, `link{calcATHBpmv}`, `link{calcATHBset}`

### Examples

```
calcATHBpts(20, 0, 25, 25, .1, 50, 1.1, 0, 760, 60, 171, 70)
```

---

## calcATHBset — *SET based on Adaptive Thermal Heat Balance Framework*

---

### Description

Calculation of SET based on Adaptive Thermal Heat Balance framework using Gagge's 2-node model

### Usage

```
calcATHBset(trm, psych, ta, tr, vel, rh, met, wme = 0, pb = 760,
                      ltime = 60, ht = 171, wt = 69.9)
```

### 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

ATHBset set value adapted through the ATHB appoach

## Author(s)

Marcel Schweiker

## References

Schweiker & Wagner (2015) <doi:10.1016/j.buildenv.2015.08.018> Schweiker & Wagner (2016) 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`, `link{calcATHBpmv}`, `link{calcATHBpts}`

## Examples

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

---

`calcATHBstandard` *PMV based on Adaptive Thermal Heat Balance Framework for the Standard Model*

---

## Description

`calcATHBstandard` calculates the PMV based on adaptive thermal heat balance framework

based on the newest version (2022)

## Usage

```
calcATHBstandard(trm, ta, tr, vel, rh, met)
```

## Arguments

| | |
|---|---|
| `trm` | - Running mean outdoor temperature in [degree C] |
| `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] |

## Details

aliases athb ATHB

### Value

calcATHBstandard PMV value adapted through the ATHB approach with standard model

### Author(s)

Code implemented in to R by Shaomi Rahman. Further contribution by Marcel Schweiker.

### References

Schweiker & Wagner (2015) <doi:10.1016/j.buildenv.2015.08.018> Schweiker (2022) <doi:10.1111/ina.13018>

### See Also

see also calcComfInd, link{calcATHBpts}, link{calcATHBset},

link{calcATHBpmv2015}

### Examples

```
calcATHBstandard(20, 25, 25, .1, 50, 1.1)
```

---

## calcATHBx — *PMV based on Adaptive Thermal Heat Balance Framework for the extended Model*

---

### Description

calcATHBx calculates the PMV based on adaptive thermal heat balance framework

based on the newest version (2022)

### Usage

```
calcATHBx(trm, ta, tr, vel, rh, met, buildingTypeSimple,
coolingStrategyBuilding, seasonSimple)
```

### Arguments

| | |
|---|---|
| trm | - Running mean outdoor temperature in [degree C] |
| 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] |
| buildingTypeSimple | - simple building type. Value can be among Multifamily housing, Office as a string |

`coolingStrategyBuilding`
: - the process in which the building was ventilated. Value can be among Mixed Mode','Naturally Ventilated' as a String

`seasonSimple`
: - season. Value can be among 'Spring','Summer', 'Winter' as a String

### Details

aliases athb ATHB

### Value

`calcATHBx` PMV value adapted through the ATHB approach with extended model

### Author(s)

Code implemented in to R by Shaomi Rahman. Further contribution by Marcel Schweiker.

### References

Schweiker & Wagner (2015) <doi:10.1016/j.buildenv.2015.08.018> Schweiker (2022) <doi:10.1111/ina.13018>

### See Also

see also `calcComfInd`, `link{calcATHBpts}`, `link{calcATHBset}`,

`link{calcATHBpmv2015}`

### Examples

```
calcATHBx(20, 25, 25, .1, 50, 1.1, 'Office', 'Mixed Mode',
'winter')
```

---

## calcAvgAcc — *Average Accuracy between Predicted and Actual Thermal Sensation Vote*

---

### Description

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

### Usage

```
calcAvgAcc(ref, pred)

calcavgacc(ref, pred)

AvgAcc(ref, pred)

avgacc(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 accuray 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. Further contribution by Shoaib Sarwar.

## References

Sokolova and Lapalme (2009) <doi:10.1016/j.ipm.2009.03.002>

## 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 *Bias between Predicted and Actual Thermal Sensation Vote*

## 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)

calcbias(ref, pred)

calcMeanBias(ref, pred)

MeanBias(ref, pred)

meanBias(ref, pred)

meanbias(ref, pred)

bias(ref, pred)

calcSdBias(ref, pred)

calcSeBias(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

`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. Further contribution by Shoaib Sarwar.

## References

Humphreys & Nicol (2002) <doi:10.1016/S0378-7788(02)00018-X>

Schweiker & Wagner (2016) 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

calcTPRTSV, calcAvgAcc

## Examples

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

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

calcBias(ref, pred)
```

---

calcCE *Cooling Effect*

---

## Description

Function to calculate cooling effect (CE) of elevated air velocities using the standard effective temperature (SET).

## Usage

```
calcCE(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0)
```

## 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] |

## Details

The CE of the elevated air velocity is the difference in SET between conditions with given air velocities and still air. The cooling effect should be calculated only for air velocities higher than 0.2 m/s.

## Value

ce - Cooling Effect in [degree C]

## Author(s)

Code implemented in to R by Shoaib Sarwar. Further contribution by Marcel Schweiker.

## References

Original code in Python by Tartarini & Schiavon (2020) <doi:10.1016/j.softx.2020.100578>

## Examples

```
calcCE(25,25,0.3,50,0.5,1) # returns Cooling Effect: 1.3
```

---

`calcComfInd` *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")

comfind(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) |
| tcr | 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 necess |

The vector `request` can contain the following elements:

| Element | Description |
|---|---|
| "all" | Calculation of all indices described below |
| "pmv" | Predicted mean vote |
| "ppd" | Predicted precentage 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. Further contribution by Shaomi Rahman.

### References

For references see individual functions.

### See Also

see also `calcPMVPPD`, `calc2Node`, `calcHbExSteady`, `calcATHBpmv2015`, `calcdTNZ`, `calcPMVadj`, `calcPtsa`, `calctAdapt`

### Examples

```
## Creating list with all values
lsCond <- createCond()
## Requesting all comfort indices
calcComfInd(lsCond, request="all")
## Requesting a single index
calcComfInd(lsCond, request="pmv")
## Requesting multiple indices
calcComfInd(lsCond, request=c("pmv", "ptse"))
```

---

`calcDisc` *Predicted Discomfort based on the 2-Node-Model*

---

### Description

`calcDisc` calculates Predicted Discomfort based on the 2-Node-Model by Gagge et al.

### Usage

```
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)
```

### 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 |

### 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

`calcDisc` returns the Predicted Discomfort

### 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 & Huizenga (1995) A thermal sensation model for use by the engineering profession ASHRAE RP-781 Final report.

Gagge, Fobelets & Berglund (1986) A standard predictive index of human response to the thermal environment, ASHRAE transactions, 92 (2B), 709-731.

### See Also

see also `calcComfInd`

### Examples

```
## 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))
Disc <- sapply(seq(maxLength), function(x) { calcDisc(ta[x], tr[x], vel[x], rh[x]) } )
```

---

## `calcdTNZ` *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 =.1,
fBasMet = "rosa", fSA = "duBois", percCov = 0, TcMin = 36, TcMax = 38,
plotZone = FALSE)
```

### 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 temperaturein [degree C] |
| met | a numeric value presenting metabolic rate (activity related) in [met] |
| rh | a numeric value presenting realtive humidity in [%] |
| deltaT | a numeric value presenting the resolution of the matrix to be used |
| fBasMet | a string presenting the method of calculating basal metbolic rate. Needs to be one of "rosa", "harris", "miflin", or "fixed". Fixed will result in the value of 58.2 W/m2. |
| fSA | a string presenting the method of calculating the surface area. Needs to be one of "duBois" or "mosteller". |
| percCov | a numeric value between 0 and 1 presenting the percentage of the body covered by clothes in [%] |
| TcMin | a numeric value presenting the minimum allowed core temperature in [degree C]. |
| TcMax | a numeric value presenting the maximum allowed core temperature in [degree C]. |
| plotZone | a boolean variable TRUE or FALSE stating, wether TNZ should be plotted or not. |

## Details

The percentage of the body covered by clothes can be estimated e.g. based on ISO 9920 Appendix H (Figure H.1). A typical winter case leads to a value of around .86, in the summer case this goes down to values around .68.

## 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

This function was used in earlier versions of TNZ calculation (see references above). The newest version is `calcTNZPDF`.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, Schweiker, Wagner & van Marken Lichtenbelt 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 & van Marken Lichtenbelt (2015) <doi:10.1038/nclimate2741> Kingma, Frijns, Schellen & van Marken Lichtenbelt (2014) <doi:10.4161/temp.29702>

### See Also

see also `calcTNZPDF` and `calcComfInd`

### Examples

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

---

## calcepCoeff *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 coefficients for apmv, `calcepCoeff` calculates e the expectancy factor for epmv, `calcasCoeff` calculates lambda the adaptive coefficients 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) . |

### Value

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

## Note

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.

## 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, Li & Liu (2009) <doi:10.1016/j.buildenv.2009.02.014>

The epmv concept was introudced by Fanger & Toftum (2002) <doi:10.1016/S0378-7788(02)00003-8>

## 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”)
```

calcePMV *Adjusted Predicted Mean Votes with Expectancy Factor*

## Description

Function to calculate Predicted Mean Votes (PMV) adjusted by the expectancy factor.

## Usage

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

ePMV(ta, tr, vel, rh, clo = 0.5, met = 1, wme = 0, epCoeff)

epmv(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] |
| `epCoeff` | expectancy factor e |

## Details

`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

`calcePMV` returns the predicted mean vote adjusted by the expectancy factor.

## Note

In case one of `epCoeff` is not given, a standard value will be taken from a list (see `createCond` for details.

## Author(s)

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

## References

epmv is based on Fanger & Toftum (2002) <doi:10.1016/S0378-7788(02)00003-8>

## See Also

calcComfInd, 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 epCoeff for data set
epCoeff <- calcepCoeff(lsCond)
## calculate epmv
epmv <- NULL
for (i in 1:length(ta)){
 epmv[i] <- calcePMV(ta[i], tr[i], vel[i], rh[i], clo[i], met[i], epCoeff = epCoeff)$epmv}
epmv
```

---

calcET *Effective Temperature based on the 2-Node-Model*

---

## Description

calcET calculates Effective temperature based on the 2-Node-Model by Gagge et al.

## Usage

```
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)
```

## 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 |

## 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

`calcET` returns the 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 & Huizenga (1995) A thermal sensation model for use by the engineering profession ASHRAE RP-781 Final report.

Gagge, Fobelets & Berglund (1986) A standard predictive index of human response to the thermal environment, ASHRAE transactions, 92 (2B), 709-731.

## See Also

see also `calcComfInd`

### Examples

```
## 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))
ET <- sapply(seq(maxLength), function(x) { calcET(ta[x], tr[x], vel[x], rh[x]) } )
```

---

## calcHbExSteady — *Human Body Exergy Consumption Rate Using Steady State Method*

---

### 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] |
| rh | a numeric value presenting relative humidity [%] |
| vel | a numeric value presenting air velocity in [m/s] |
| 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 [W/m2]
xInmetwcs Label warm/ cold for exergy input through metabolism [W/m2]
xInAIRwcs Exergy input through inhaled humid air [W/m2]
xInAIRwcwcs Label warm/ cold for exergy input through inhaled humid air [W/m2]
xInAIRwds Exergy input through inhaled dry air [W/m2]
xInAIRwdwds Label wet/ dry for exergy input through inhaled dry air [W/m2]
xInLUNGwcs Exergy input through water lung [W/m2]
xInLUNGwcwcs Label warm/ cold for exergy input through water lung [W/m2]
xInLUNGwds Exergy input through water lung [W/m2]
xInLUNGwdwds Label wet/ dry for exergy input through water lung [W/m2]
xInsheLLwcs Exergy input through water from sweat [W/m2]
xInsheLLwcwcs Label warm/ cold for exergy input through water from sweat [W/m2]
xInsheLLwds Exergy input through water from sweat [W/m2]
xInsheLLwdwds Label wet/ dry for exergy input through water from sweat [W/m2]
xInraDs Exergy input through radiation [W/m2]
xInraDwcs Label warm/ cold for exergy input through radiation [W/m2]
xIntotaLs total exergy input [W/m2]

Exergy output
xoutstorecores Exergy stored in core [W/m2]
xoutstoreshels Exergy stored in shell [W/m2]
xoutaIRwcs Exergy output through exhaled humid air [W/m2]
xoutaIRwcwcs Label warm/ cold for exergy output through exhaled humid air [W/m2]
xoutaIRwds Exergy output through exhaled dry air [W/m2]
xoutaIRwdwds Label wet/ dry for exergy output through exhaled dry air [W/m2]
xoutswEATwcs Exergy output through water vapour from sweat [W/m2]
xoutswEATwcwcs Label warm/ cold for exergy output through water vapour from sweat [W/m2]
xoutswEATwds Exergy output through water vapour from sweat [W/m2]
xoutswEATwdwds Label wet/ dry for exergy output through water vapour from sweat [W/m2]
xoutraDs Exergy output through radiation [W/m2]
xoutraDwcs Label warm/ cold for exergy output through radiation [W/m2]
xoutCONVs Exergy output through convection [W/m2]
xoutCONVwcs Label warm/ cold for exergy output through convection [W/m2]
xouttotaLs total exergy output [W/m2]

Exergy balance
xconss total exergy consumption [W/m2]

`xConsumption` total exergy consumption [W/m2]

Additional values
`tsks` Calculated skin temperature [degree C]
`tcrs` Calculated core temperature [degree C]
`ws` Calculated skin wettedness [degree C]

## 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. (2016), 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, Kolarik, Dovjak & Shukuya (2016) <doi:10.1016/j.enbuild.2016.01.002>

Shukuya (2015) 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.

## 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 *Human Body Exergy Consumption Rate using Unsteady State Method*

### Description

calcHbExUnsteady Function 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 value presenting air temperature in [degree C] |
| tr | a numeric value presenting mean radiant temperature in [degree C] |
| rh | a numeric value presenting relative humidity [%] |
| vel | a numeric value presenting air velocity in [m/s] |
| clo | a numeric value presenting clothing insulation level in [clo] |
| met | a numeric value presenting metabolic rate in [met] |
| tao | a numeric vector presenting outdoor air temperature in [degree C] |
| rho | 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 |
| `xInAIRwcwcu` | 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 |
| `xInLUNGwcwcu` | 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 |
| `xInsheLLwcwcu` | 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 |
| `xoutaIRwcwcu` | 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 |
| `xoutswEATwcwcu` | 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, Kolarik, Dovjak & Shukuya (2016) <doi:10.1016/j.enbuild.2016.01.002>

Shukuya (2015) 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.

## See Also

see also `calcComfInd`

## 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
```

`calcIREQ` *IREQ and Dlim*

## Description

Calculate minimal and neutral values of REQUIRED CLOTHING INSULATION (IREQ) and DURATION LIMITED EXPOSURE (Dlim).

## Usage

```
calcIREQ(M,W,ta,tr,p,w,v,rh,clo)
```

## Arguments

| | |
|---|---|
| `M` | a numeric value presenting metabolic energy production (58 to 400 W/m2) in [W/m2] |
| `W` | a numeric value presenting Rate of mechanical work, (normally 0) in [W/m2] |
| `ta` | a numeric value presenting ambiant air temperature in [degree C] |
| `tr` | a numeric value presenting mean radiant temperature in [degree C] |
| `p` | a numeric value presenting air permeability (low < 5, medium 50, high > 100 l/m2s) in [l/m2s] |
| `w` | a numeric value presenting walking speed (or calculated work created air movements) in [m/s] |
| `v` | a numeric value presenting relative air velocity(0.4 to 18 m/s) in [m/s] |
| `rh` | a numeric value presenting relative humidity [%] |
| `clo` | a numeric value presenting clothing insulation level in [clo] |

## Details

The function gives IREQ Insulation of clothing required to maintain thermal equilibrium of the body at specified levels of physiological stress.

## Value

`calcIREQ` returns

| | |
|---|---|
| `IREQminimal` | Lower bound of insulation required in [clo] |
| `IREQneutral` | Upper bound of insulation required in [clo] |
| `ICLminimal` | Lower bound of REQUIRED basic clothing insulation (ISO 9920) in [clo] |
| `ICLneutral` | Upper bound of REQUIRED basic clothing insulation (ISO 9920) in [clo] |
| `DLEminimal` | Lower bound of duration limited exposure in [hours] |
| `DLEneutral` | upper bound of duration limited exposure in [hours] |

## Note

The authors disclaim all obligations and liabilities for damages arising from the use or attempted use of the information, including, but not limited to, direct, indirect, special and consequential damages, and attorneys' and experts' fees and court costs. Any use of the information will be at the risk of the user.

## Author(s)

Developed by Ingvar Holmer and Hakan O. Nilsson, 1990 in java and transferred to R by Shoaib Sarwar. Further contribution by Marcel Schweiker.

## References

ISO 11079, 2007-12-15, ERGONOMICS OF THE THERMAL ENVIRONMENT - DETERMINATION AND INTERPRETATION OF COLD STRESS WHEN USING REQUIRED CLOTHING INSULATION (IREQ) AND LOCAL COOLING EFFECTS

## Examples

```
calcIREQ(116,0,-15,-15,8,0.3,0.4,85,2.5)
```

---

`calcIso7933` *Heat Strain Indices based on ISO 7933*

---

## Description

`calcISO7933` calculates Tre, SWtotg, Dlimtre, Dlimloss50 and Dlimloss95 based on ISO 7933. It additionally provides intermediate results from the calculation: Cres, Eres, Ep, SWp, Texp, Tskeq, Tsk, wp

## Usage

```
calcIso7933(accl, posture, Ta, Pa, Tr, Va, Met, Icl, THETA, Walksp, Duration,
weight, height, DRINK, Adu, spHeat, SWp, Tre, Tcr, Tsk, Tcreq, Work, imst,
Ap, Fr, defspeed, defdir, HR, pb)
```

## Arguments

| | |
|---|---|
| `accl` | a numeric value presenting state of acclimation [100 if acclimatised subject, 0 otherwise] |
| `posture` | a numeric value presenting posture of person [sitting=1, standing=2, crouching=3] |
| `Ta` | a numeric value presenting air temperature in [degrees celsius] |
| `Pa` | a numeric value presenting partial water vapour pressure [kPa] |
| `Tr` | a numeric value presenting mean radiant temperature in [degrees celsius] |
| `Va` | a numeric value presenting air velocity in [m/s] |

| | |
|---|---|
| `Met` | a numeric value presenting metabolic rate in [W/(m*m)] |
| `Icl` | a numeric value presenting static thermal insulation of clothing [clo] |
| `THETA` | a numeric value presenting angle between walking direction and wind direction in [degrees] |
| `Walksp` | a numeric value presenting walking speed in [m/s] |
| `Duration` | a numeric value presenting the duration of the work sequence in [min] |
| `weight` | a numeric value presenting the body mass in [kg] |
| `height` | a numeric value presenting the body height in [m] |
| `DRINK` | a numeric value presenting if workers can drink as they want [1 if they can drink without restriction, 0 if restricted] |
| `Adu` | a numeric value presenting body surface area according to Du Bois [m*m] |
| `spHeat` | a numeric value presenting specific body heat [(W/(m*m))/K] |
| `SWp` | a numeric value presenting predicted sweat rate [W/(m*m)] |
| `Tre` | a numeric value presenting rectal temperature [degrees celsius] |
| `Tcr` | a numeric value presenting temperature of body core [degrees celsius] |
| `Tsk` | a numeric value presenting skin temperature at start [degrees celsius] |
| `Tcreq` | a numeric value presenting temperature of body core dependent on energy metabolism [degrees celsius] |
| `Work` | a numeric value presenting effective mechanical power [W/(m*m)] |
| `imst` | a numeric value presenting static moisture permeability index [-] |
| `Ap` | a numeric value presenting fraction of the body surface covered by the reflective clothing [-] |
| `Fr` | a numeric value presenting emissivity of the reflective clothing [-] |
| `defspeed` | a numeric value presenting if walking speed entered [1 if walking speed entered, 0 otherwise] |
| `defdir` | a numeric value presenting if walking direction entered [1 if walking direction entered, 0 otherwise] |
| `HR` | a numeric value presenting humidity ratio [g/kg] |
| `pb` | a numeric value presenting normal barometric pressure in [Pa] |

## Details

All variables must have the same length 1.

## Value

`calcISO7933` returns a data.frame with the following items:

`Tre final` rectal temperature [degrees Celsius]

`SWtotg` total water loss [g]

`Dlimtre` time when limit for rectal temperature is reached [min]

`Dlimloss50` time when limit for water loss Dmax50 (7.5 percent of body mass of an average person) is reached [min]

`Dlimloss95` time when limit for water loss Dmax95 (5 percent of body mass of 95 percent of the working people) is reached [min]

`Cres` convective heat flow at respiration [W/(m*m)]

`Eres` evaporative heat flow at respiration [W/(m*m)]

`Ep` predicted evaporative heat flow [W/(m*m)]

`SWp` predicted sweating rate [W/(m*m)]

`Texp` temperature of the exhaled air [degrees Celsius]

`Tskeq` skin Temperature in equilibrium [degrees Celsius]

`Tsk` skin Temperature at the minute [degrees Celsius]

`wp` predicted skin wettedness [-]

## Note

In case one of the variables is not given, a standard value according to ISO 7933 will be taken.

## Author(s)

The code for calcISO7933 is based on the code in BASIC presented in Addendum E of EN ISO 7933. The translation into R-language conducted by Michael Kleber.

## References

ISO 7933 (2004) Ergonomics of the thermal environment - Analytical determination and interpretation of heat stress using calculation of the predicted heat strain Malchaire, Piette, Kampmann, Mehnert, Gebhardt, Havenith, Den Hartog, Holmer, Parsons, Alfano, Griefahn (2000) <doi:10.1016/S0003-4878(00)00030-2> Malchaire, Kampmann, Havenith, Mehnert, Gebhardt (2000) <doi:10.1007/s004200050420>

## Examples

```
## Calculation of a single set of values.
calcIso7933(accl = 100, posture = 2, Ta = 35, Pa = 4, Tr = 35, Va = 0.3, Met = 150,
Icl = 0.5, THETA = 0, Walksp = 0, Duration = 480)
calcIso7933(100,2,35,4,35,0.3,150,0.5,0,0,480)
## Using several rows of data:
accl <- 100
posture <- 2
```

```
Ta <- c(40,35)
Pa <- c(2.5,4)
Tr <- c(40,35)
Va <- 0.3
Met <- 150
Icl <- 0.5
THETA <- 0
Walksp <- 0
Duration <- 480
maxLength <- max(sapply(list(accl, posture, Ta, Pa, Tr, Va, Met, Icl, THETA,
Walksp, Duration), length))
PHI <- sapply(seq(maxLength), function(x) {calcIso7933(accl, posture, Ta[x],
Pa[x], Tr[x], Va, Met, Icl, THETA, Walksp, Duration) } )
```

---

calcMixR *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)

calcVapourpressure(ta, rh)
```

## Arguments

| | |
|---|---|
| `ta` | a numeric value or vector presenting air temperature in [degree C]. |
| `rh` | a numeric value or vector presenting relative humidity in [%], except for `calcVapourpressure`, where it must be in decimal (e.g. 0.5). |
| `pb` | a numeric value or vector presenting barometric pressure in [torr]. |
| `mr` | a numeric value or vector presenting the mixIng ratio in [g/kg. |

### 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]

`calcVapourpressure` returns a single value or a vector of vapor pressure in [kpa]

### Author(s)

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

### References

Ranaa, Kusya, Jurdaka, Wallb & Hua (2013) <doi:10.1016/j.enbuild.2013.04.019>

Masterton & Richardson (1979) Humidex a method of quantifying humandiscomfort due to excessive heat and humidity, clI 1-79. Downsview, Ont: Environment Canada. Atmosheric Environment Service.

### 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)
```

## calcMRTglobe *MRT calculation based on standard and mixed convection*

### Description

`calcMRTglobe` calculates the mean radiant temperature considering mixed convection

### Usage

```
calcMRTglobe(tg, ta, vel, x = 0.15)
```

### Arguments

| | |
|---|---|
| `tg` | - a numeric value presenting globe temperature in [degree C] |
| `ta` | - a numeric value presenting air temperature in [degree C] |
| `vel` | - a numeric value presenting air velocity in [m/s] |
| `x` | - a numeric value presenting globe diameter in [m] |

### Details

aliases MRT globe

This model has only been validated from x = 0.040m (ping pong ball) to

x = 0.150m (standard globe thermometer) globes

### Value

`calcMRTglobe` MRT with standard and mixed correction

### Author(s)

code implemented into R by Shaomi Rahman and Marcel Schweiker.

### References

Teitelbaum et al. (2022) <10.1038/s41598-022-10172-5> Teitelbaum (2022) <https://github.com/eteitelb/MixedConvection>

### Examples

```
#Globe temperature [C]
tg <- 30
#Air temperature [C]
ta <- 24
#Air speed [m/s]
vel <- 0.0
calcMRTglobe(tg, ta, vel)
```

## calcPD — *Predicted Percentage Dissatisfied due to Draft based on the 2-Node-Model*

### Description

calcPD calculates Predicted Percentage Dissatisfied due to Draft based on the 2-Node-Model by Gagge et al.

### Usage

```
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)
```

### 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 |

### 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

`calcPD` returns the Predicted Percentage Dissatisfied due to Draft

### 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 & Huizenga (1995) A thermal sensation model for use by the engineering profession ASHRAE RP-781 Final report.

Gagge, Fobelets & Berglund (1986) A standard predictive index of human response to the thermal environment, ASHRAE transactions, 92 (2B), 709-731.

### See Also

see also `calcComfInd`

### Examples

```
## 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))
pd <- sapply(seq(maxLength), function(x) { calcPD(ta[x], tr[x], vel[x], rh[x]) } )
```

---

`calcPMV` *Predicted Mean Votes (PMV)*

---

### Description

Function to calculate Predicted Mean Vote (PMV).

### Usage

```
calcPMV(ta, tr, vel, rh, clo=.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

The PMV is an index that predicts the mean value of the thermal sensation of a large group of people on a sensation scale expressed from (-3) to (+3) corresponding to the categories cold, cool, slightly cool, neutral, slightly warm, warm and hot. PMV model is limited to air speeds below 0.20 m/s.

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

PMV - Predicted Mean Vote

### Author(s)

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

### References

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

ISO 7730 (2005) 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.

### Examples

```
calcPMV(25,25,0.3,50,0.5,1)
```

## calcPMVadj — *Predicted Mean Votes adjusted for elevated air speed*

### Description

Function to calculate Predicted Mean Votes (PMV) adjusted for cooling effect of elevated air speed.

### Usage

```
calcPMVadj(ta, tr, vel, rh, clo, met, wme = 0)
```

### 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] |

### Details

Calculated through function Two node model(`calc2Node`

### Value

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

### Author(s)

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

### 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.

### Examples

```
calcPMVadj(25,25,0.3,50,0.5,1)
```

calcPMVGagge *Gagge's Version of Fanger's PMV based on the 2-Node-Model*

## Description

calcPMVGagge calculates Gagge's Version of Fanger's PMV based on the 2-Node-Model by Gagge et al.

## Usage

```
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)
```

## 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 |

## 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

calcPMVGagge returns Gagge's Version of Fanger's PMV

### 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 & Huizenga (1995) A thermal sensation model for use by the engineering profession ASHRAE RP-781 Final report.

Gagge, Fobelets & Berglund (1986) A standard predictive index of human response to the thermal environment, ASHRAE transactions, 92 (2B), 709-731.

### See Also

see also calcComfInd

### Examples

```
## 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))
pmvg <- sapply(seq(maxLength), function(x) { calcPMVGagge(ta[x], tr[x], vel[x], rh[x]) } )
```

---

## calcPMVPPD *PMV and PPD*

---

### Description

Function to calculate Predicted Mean Vote (PMV) and Predicted Percentage of Dissatisfied (PPD).

### Usage

```
calcPMVPPD(ta, tr, vel, rh, clo=.5, met=1, wme=0, basMet=58.15, getLoad = FALSE)
```

### 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] |
| `getLoad` | a boolean value. Set to true to get thermal load as output instead of PMV/PPD |

### Details

The PMV is an index that predicts the mean value of the thermal sensation of a large group of people on a sensation scale expressed from (-3) to (+3) corresponding to the categories cold, cool, slightly cool, neutral, slightly warm, warm and hot. The PPD is an index that establishes a quantitative prediction of the percentage of thermally dissatisfied people determined from PMV.

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

PMV - Predicted Mean Vote

PPD - Predicted Percentage of Dissatisfied occupants in [%]

Lraw - thermal load (only when getLoad was set to TRUE)

### Author(s)

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

### References

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

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

```
calcPMVPPD(25,25,0.3,50,0.5,1)
```

calcPMVStar *Fanger's PMV using SET* for DRY*

## Description

calcPMVStar calculates Fanger's PMV based on the 2-Node-Model by Gagge et al. except that DRY is calculated using SET* rather than the operative temperature

## Usage

```
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)
```

## 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 |

## 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

`calcPMVStar` returns Fanger's PMV except that DRY is calculated using SET* rather than the operative 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 & Huizenga (1995) A thermal sensation model for use by the engineering profession ASHRAE RP-781 Final report.

Gagge, Fobelets & Berglund (1986) A standard predictive index of human response to the thermal environment, ASHRAE transactions, 92 (2B), 709-731.


### See Also

see also `calcComfInd`

### Examples

```
## 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))
pmvstar <- sapply(seq(maxLength), function(x) { calcPMVStar(ta[x], tr[x], vel[x], rh[x]) } )
```

---

## calcPPD — *Predicted Percentage of Dissatisfied (PPD)*

---

### Description

Function to calculate Predicted Percentage of Dissatisfied (PPD).

### Usage

```
calcPPD(ta, tr, vel, rh, clo=.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

The PPD is an index that establishes a quantitative prediction of the percentage of thermally dissatisfied people determined from PMV.

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

PPD - Predicted Percentage of Dissatisfied occupants in [%]

### Author(s)

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

### References

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

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

```
calcPPD(25,25,0.3,50,0.5,1)
```

calcPS *Predicted Percentage Satisfied with the Level of Air Movement based on the 2-Node-Model*

## Description

calcPS calculates Predicted Percentage Satisfied with the Level of Air Movement based on the 2-Node-Model by Gagge et al.

## Usage

```
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)
```

## 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 |

## 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

calcPS returns the Predicted Percentage Satisfied with the Level of Air Movement

## 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 & Huizenga (1995) A thermal sensation model for use by the engineering profession ASHRAE RP-781 Final report.

Gagge, Fobelets & Berglund (1986) A standard predictive index of human response to the thermal environment, ASHRAE transactions, 92 (2B), 709-731.

## See Also

see also calcComfInd

## Examples

```
## 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))
ps <- sapply(seq(maxLength), function(x) { calcPS(ta[x], tr[x], vel[x], rh[x]) } )
```

---

calcPTS *Predicted Thermal Sensation Vote based on SET*

---

## Description

calcPTS calculates Predicted Thermal Sensation Vote based on SET by the 2-Node-Model by Gagge et al.

### Usage

```
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)
```

### 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 |

### 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

`calcPTS` returns the Predicted Thermal Sensation Vote based on SET

### 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`

### Examples

```
## 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))
pts <- sapply(seq(maxLength), function(x) { calcPTS(ta[x], tr[x], vel[x], rh[x]) } )
```

---

## calcPtsa — *Predicted Thermal Sensation based on 2-Node Model adjusted for Adaptation*

---

### Description

`calcPtsa` calculates Predicted Thermal Sensation based on the 2-Node-Model by Gagge et al. and adjusts its output according to adaptive coefficient

### Usage

```
calcPtsa(ta, tr, vel, rh, clo = .5, met = 1, wme = 0, pb = 760,
                     ltime = 60, ht = 171, wt = 70, tu = 40, asCoeff)
```

### 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 [%] |
| `asCoeff` | a numeric values presenting adaptive coefficient [-] |

## 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

`calcPtsa` returns a dataframe containing the Predicted Thermal Sensation value

## 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 & Huizenga (1995) A thermal sensation model for use by the engineering profession ASHRAE RP-781 Final report.

Gagge, Fobelets & Berglund (1986) A standard predictive index of human response to the thermal environment, ASHRAE transactions, 92 (2B), 709-731. Coefficients are calculated based on Gao, Wang & Wargocki (2015) <doi:10.1016/j.buildenv.2015.04.030> The aPMV concept was introduced by Yao, Li & Liu (2009) <doi:10.1016/j.buildenv.2009.02.014> The ePMV concept was introudced by Fanger & Toftum (2002) <doi:10.1016/S0378-7788(02)00003-8>

## See Also

see also `calcComfInd` and `calc2Node`

## Examples

```
## Using several rows of data:
ta <- c(20,22,24)
tr <- ta
vel <- rep(.15,3)
rh <- rep(50,3)
asCoeff <- 0.5

maxLength <- max(sapply(list(ta, tr, vel, rh,asCoeff), length))
ptsa <- sapply(seq(maxLength), function(x) { calcPtsa(ta[x], tr[x], vel[x],
rh[x], asCoeff=asCoeff) } )
```

---

# calcPtse — *Predicted Thermal Sensation based on 2-Node Model adjusted for Expectancy*

---

## Description

`calcPtse` calculates Predicted Thermal Sensation based on the 2-Node-Model by Gagge et al. and adjusts its output according to expectancy factor

## Usage

```
calcPtse(ta, tr, vel, rh, clo = .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 turbulence intensity in [%] |
| `esCoeff` | a numeric values presenting expectancy factor [-] |

### 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

`calcPtse` returns a dataframe containing the Predicted Thermal Sensation value

### 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 & Huizenga (1995) A thermal sensation model for use by the engineering profession ASHRAE RP-781 Final report.

Gagge, Fobelets & Berglund (1986) A standard predictive index of human response to the thermal environment, ASHRAE transactions, 92 (2B), 709-731. Coefficients are calculated based on Gao, Wang & Wargocki (2015) <doi:10.1016/j.buildenv.2015.04.030> The aPMV concept was introduced by Yao, Li & Liu (2009) <doi:10.1016/j.buildenv.2009.02.014> The ePMV concept was introudced by Fanger & Toftum (2002) <doi:10.1016/S0378-7788(02)00003-8>

### See Also

see also `calcComfInd` and `calc2Node`

### Examples

```
## Using several rows of data:
ta <- c(20,22,24)
tr <- ta
vel <- rep(.15,3)
rh <- rep(50,3)
esCoeff <- 0.5

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

```
rh[x], esCoeff=esCoeff) } )
```

---

calcRT *Required recovery time*

---

## Description

Function to calculate Required recovery time, RT in hours.

## Usage

```
calcRT(M,W,ta,tr,p,w,v,rh,clo)
```

## Arguments

| | |
|---|---|
| `M` | a numeric value presenting metabolic energy production (58 to 400 W/m2) in [W/m2] |
| `W` | a numeric value presenting Rate of mechanical work, (normally 0) in [W/m2] |
| `ta` | a numeric value presenting ambiant air temperature in [degree C] |
| `tr` | a numeric value presenting mean radiant temperature in [degree C] |
| `p` | a numeric value presenting air permeability (low < 5, medium 50, high > 100 l/m2s) in [l/m2s] |
| `w` | a numeric value presenting walking speed (or calculated work created air movements) in [m/s] |
| `v` | a numeric value presenting relative air velocity(0.4 to 18 m/s) in [m/s] |
| `rh` | a numeric value presenting relative humidity [%] |
| `clo` | a numeric value presenting clothing insulation level in [clo] |

## Value

returns required recovery time in [hours]

## Note

The authors disclaim all obligations and liabilities for damages arising from the use or attempted use of the information, including, but not limited to, direct, indirect, special and consequential damages, and attorneys' and experts' fees and court costs. Any use of the information will be at the risk of the user.

## Author(s)

Developed by Ingvar Holmer and Hakan O. Nilsson, 1990 in java and transferred to R by Shoaib Sarwar. Further contribution by Marcel Schweiker.

## References

ISO 11079, 2007-12-15, ERGONOMICS OF THE THERMAL ENVIRONMENT - DETERMINATION AND INTERPRETATION OF COLD STRESS WHEN USING REQUIRED CLOTHING INSULATION (IREQ) AND LOCAL COOLING EFFECTS

## Examples

```
calcRT(90,0,25,25,8,0.2,0.4,50,1.5)
```

---

`calcSET` *Standard Effective Temperature (SET)*

---

## Description

`calcSET` calculates Standard Effective Temperature based on the 2-Node-Model by Gagge et al.

## Usage

```
calcSET(ta, tr, vel, rh, clo = .5, met = 1, wme = 0, sa = NULL, pb = 760,
ltime = 60, ht = 171, wt = 70, tu = 40, obj = "set", csw = 170, cdil = 120,
cstr = .5, bodyPosition = 'sitting')
```

## 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] |
| `sa` | (optional)surface Area according to mosteller formula [m^2] |
| `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 |
| `bodyPosition` | a string representing body position, has to be 'sitting' or 'standing'. Default value is 'sitting' |

## 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

`calcSET` returns the Standard Effective Temperature (SET)

## 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 & Huizenga (1995) A thermal sensation model for use by the engineering profession ASHRAE RP-781 Final report.

Gagge, Fobelets & Berglund (1986) A standard predictive index of human response to the thermal environment, ASHRAE transactions, 92 (2B), 709-731.

## See Also

see also `calcComfInd`

## Examples

```
## 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]) } )
```

calcSkinWettedness *Skin Wettedness based on the 2-Node-Model*

## Description

calcSkinWettedness calculates Skin Wettedness based on the 2-Node-Model by Gagge et al.

## Usage

```
calcSkinWettedness(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="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 "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

`calcSkinWettedness` returns the Skin Wettedness

### 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 & Huizenga (1995) A thermal sensation model for use by the engineering profession ASHRAE RP-781 Final report.

Gagge, Fobelets & Berglund (1986) A standard predictive index of human response to the thermal environment, ASHRAE transactions, 92 (2B), 709-731.

### See Also

see also `calcComfInd`

### Examples

```
## 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))
skinWet <- sapply(seq(maxLength), function(x) { calcSkinWettedness(ta[x], tr[x], vel[x], rh[x]) } )
```

---

## `calcSolarGain` *Solar Gain*

---

### Description

Function to calculate effective radiant field and delta mean radiant temperature.

### Usage

```
calcSolarGain(solAlt, solAzi, solRadDir, solTrans,
 fSvv, fBes, asw=0.7, posture="seated", floorRef=0.6)
```

### Arguments

| | |
|---|---|
| solAlt | a numeric value presenting solar altitude, degrees from horizontal in [degree C] |
| solAzi | a numeric value presenting solar azimuth, degrees clockwise from North in [degree C] |
| solRadDir | a numeric value presenting direct-beam solar radiation [W/m2] |
| solTrans | a numeric value presenting total solar transmittance. Ranges from 0 to 1. |
| fSvv | a numeric value presenting fraction of sky vault exposed to body. Ranges from 0 to 1. |
| fBes | a numeric value presenting fraction of the possible body surface exposed to sun. Ranges from 0 to 1. |
| asw | a numeric value presenting the average short-wave absorptivity of the occupant. |
| posture | a list of available options 'standing', 'supine' or 'seated'. |
| floorRef | a numeric value presenting floor reflectance. Usually assumed to be constant and equal to 0.6. |

### Value

An array of two values First values represents `erf` - Net energy flux to or from the human body using the Effective Radiant Field [W/m2] Second value represents `delMrt` - Delta mean radiant temperature, the increase in radiant temperature required without solar radiation [Degree C]

### Author(s)

Code implemented in to R by Shaomi Rahman. Further contribution by Marcel Schweiker.

### References

Original code in Python by Tartarini & Schiavon (2020) <doi:10.1016/j.softx.2020.100578>

### See Also

see also `calcComfInd`

### Examples

```
calcSolarGain(0, 120, 800, 0.5, 0.5, 0.5, asw=0.7, posture="seated") # Returns [42.9, 10.3]
```

## calctAdapt15251 — *Adaptive Comfort 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

| | |
|---|---|
| `trm` | numerical value presenting the running mean outdoor temperature in [degree C]. |
| `ta` | numerical value presenting the indoor air 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)

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

### 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 & de Dear (2001) Climate, comfort, & natural ventilation: a new adaptive comfort standard for ASHRAE standard 55.

`calctnAuliciems` is based on Auliciems (1981) Psycho-physiological criteria for global thermal zones of building design.

`calctnHumphreysNV` and `calctnHumphreysAC` are based on Humphreys (1978) Outdoor temperatures and comfort indoors. Batiment International, Building Research and Practice, Taylor and Francis.

### See Also

see also `calcComfInd`

### Examples

```
## define variable
trm <- 21.2
## calculate adaptive comfort temperature
calctAdapt15251(trm)
```

---

## calcTNZPDF *Values related to TNZ approach*

---

### Description

`calcTNZPDF` calculates the distance from the thermoneutral zone, either skin temperature or room air related. Also calculates the probability function (PDF) of the thermoneutral zone.

### Usage

```
calcTNZPDF(ht, wt, age, gender, clo, vel, tskObs, taObs, met, rh,
fBasMet = "rosa", fSA = "duBois", percCov = 0, TcMin = 36, TcMax = 38,
plotZone = FALSE, gridTaMin = 20, gridTaMax = 30, gridTskMin = 30, gridTskMax = 42,
gridTa = 1000, gridTsk = 1000, sa = 1.86, IbMax = 0.124, IbMin = 0.03, alphaIn = 0.08,
metMin = 55.3, metMax = 57.3, metDiff = .1, forPDF = FALSE, metAdapt = "none",
trm = 15, TcPreAdapt = 37.2)
```

### 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 actual air temperature in [degree C]. |
| `met` | a numeric value presenting metabolic rate (activity related) in [met]. |
| `rh` | a numeric value presenting realtive humidity in [%]. |
| `fBasMet` | a string presenting the method of calculating basal metbolic rate. Needs to be one of "rosa", "harris", "miflin", "fixed", or "direct". Fixed will result in the value of 58.2 W/m2. Direct requires definition of metMin and metMax. |
| `fSA` | a string presenting the method of calculating the surface area. Needs to be one of "duBois", "mosteller", or "direct". |
| `percCov` | a numeric value between 0 and 1 presenting the percentage of the body covered by clothes in [%]. |
| `TcMin` | a numeric value presenting the minimum allowed core temperature in [degree C]. |
| `TcMax` | a numeric value presenting the maximum allowed core temperature in [degree C]. |
| `plotZone` | a boolean variable TRUE or FALSE stating, wether TNZ should be plotted or not. |
| `gridTaMin` | a numeric value defining the minimum grid value for Ta, ambient temperature, in [degree C]. |
| `gridTaMax` | a numeric value defining the maximum grid value for Ta, ambient temperature, in [degree C]. |
| `gridTskMin` | a numeric value defining the minimum grid value for Tsk, skin temperature, in [degree C]. |
| `gridTskMax` | a numeric value defining the maximum grid value for Tsk, skin temperature, in [degree C]. |
| `gridTa` | a numeric value defining the grid size in Ta dimension. |
| `gridTsk` | a numeric value defining the grid size in Tsk dimension. |
| `sa` | a numeric value for surface area (only used with method fSA: direct) in [m2]. |
| `IbMax` | a numeric value for maximum body tissue insulation in [m2K/W]. |
| `IbMin` | a numeric value for minimum body tissue insulation in [m2K/W]. |
| `alphaIn` | a numeric value for alpha (if 0, alpha will be calculated according to Fanger. |
| `metMin` | a numeric value for minimum metabolic rate (only used with method fBasMet:direct) in [W/m2]. |
| `metMax` | a numeric value for maximum metabolic rate (only used with method fBasMet:direct) in [W/m2]. |
| `metDiff` | a numeric value for difference between minimum and maximum metabolic rate (not used with method fBasMet:direct) in [W/m2]. |

| | |
|---|---|
| `forPDF` | a boolean value. If TRUE, matrix for drawing of PDF will be output, if FALSE, values for dTNZ and others will be output. |
| `metAdapt` | a string presenting the method of calculating the surface area. Needs to be one of 'Hori', 'Q10', 'ATHB', or 'none'. NOTE: all methods applied here still in development and need further validation. |
| `trm` | numerical value presenting the running mean outdoor temperature in [degree C]. Only used with metAdapt: Hori and ATHB. |
| `TcPreAdapt` | numerical value presenting the initial core temperature before adaptation in [degree C]. Only used with metAdapt: Q10. |

## Details

The percentage of the body covered by clothes can be estimated e.g. based on ISO 9920 Appendix H (Figure H.1). A typical winter case leads to a value of around .86, in the summer case this goes down to values around .68.

## Value

`calcTNZPDF` returns either a dataframe suitbale to draw the pdf of TNZ (by setting forPDF to TURE) or a dataframe with the columns dTNZ, dTNZTs, dTNZTa and others. 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

This function was used for the review paper by Schweiker et al. (2018) (see reference above). Some of the equations implemented are still to be validated further - therefore, use this function and its parameters with great care.

This function is not (yet) implemented in `calcComfInd`, `calcdTNZ` is applied there.

## Author(s)

Marcel Schweiker and Boris Kingma

## References

Schweiker, Huebner, Kingma, Kramer & Pallubinsky (2018) <doi:10.1080/23328940.2018.1534490> Kingma, Schweiker, Wagner & van Marken Lichtenbelt 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 & van Marken Lichtenbelt (2015) <doi:10.1038/nclimate2741> Kingma, Frijns, Schellen & van Marken Lichtenbelt (2014) <doi:10.4161/temp.29702>

## See Also

`calcdTNZ`

## Examples

```
## Calculate and draw pdf of TNZ for a young non-obese male
longTcYoungMale <- calcTNZPDF(ht = 178, wt = 70, age = 30, gender = 2, clo = 0.5,
           vel = 0.2, tskObs = 36.2, taObs = 26, met = 1,
           rh = 50, fBasMet = "rosa", fSA = "duBois", percCov = 0.6,
           TcMin = 36, TcMax = 38, plotZone = FALSE, gridTaMin = 20, gridTaMax = 30,
           gridTskMin = 20, gridTskMax = 42, gridTa = 1000, gridTsk = 1000,
           sa = 2.0335, IbMax = 0.124, IbMin = 0.03, alphaIn = 0, metMin = 55.3,
           metMax = 57.3, metDiff = 0.1, forPDF = TRUE)

plot(density(longTcYoungMale$X2), main="", xlim=c(14,36), ylim=c(0,.50),
    xlab="Operative temperature [degree C]")
```

---

calcTPRTSV *True Positive Rate between Predicted and Actual Thermal Sensation Vote*

---

## 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 & Wagner (2015) <doi:10.1016/j.buildenv.2015.08.018>

## 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 *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)

calctroin(vel, tg, ta, met)

Troin(vel, tg, ta, met)

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

### Arguments

| | |
|---|---|
| `vel` | a numeric value presenting air velocity in [m/s] |
| `tg` | a numeric value or vector presenting the globe temperature in [degree C] |
| `ta` | a numeric value presenting air temperature in [degree C] |
| `met` | a numeric value presenting 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. Further contribution by Shoaib Sarwar.

### 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)
```

---

## calcTSens — *Predicted Thermal Sensation based on the 2-Node-Model*

---

### Description

`calcTSens` calculates Predicted Thermal Sensation based on the 2-Node-Model by Gagge et al.

### Usage

```
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)
```

### 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 |

## 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

`calcTSens` returns the Predicted Thermal 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)

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 & Huizenga (1995) A thermal sensation model for use by the engineering profession ASHRAE RP-781 Final report.

Gagge, Fobelets & Berglund (1986) A standard predictive index of human response to the thermal environment, ASHRAE transactions, 92 (2B), 709-731.

## See Also

see also `calcComfInd`

## Examples

```
## 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))
TSens <- sapply(seq(maxLength), function(x) { calcTSens(ta[x], tr[x], vel[x], rh[x]) } )
```

---

`calcTWC` *Windchill temperature (TWC)*

---

## Description

Function to calculate windchill temperature, TWC, in Degrees.

## Usage

```
calcTWC(v,ta)

windchill(v,ta)
```

## Arguments

| | |
|---|---|
| `v` | a numeric value presenting meteorological wind speed (at 10 m) in [km/h] |
| `ta` | a numeric value presenting ambient air temperature in [degree C] |

## Details

The function returns the temperature that considers the cooling effect on a localized skin segment.

## Value

returns (twc) Wind chill temperature in [Degree C]

## Note

The authors disclaim all obligations and liabilities for damages arising from the use or attempted use of the information, including, but not limited to, direct, indirect, special and consequential damages, and attorneys' and experts' fees and court costs. Any use of the information will be at the risk of the user.

## Author(s)

Developed by Ingvar Holmer and Hakan O. Nilsson, 1990 in java and transferred to R by Shoaib Sarwar. Further contribution by Marcel Schweiker.

## References

ISO 11079, 2007-12-15, ERGONOMICS OF THE THERMAL ENVIRONMENT - DETERMINATION AND INTERPRETATION OF COLD STRESS WHEN USING REQUIRED CLOTHING INSULATION (IREQ) AND LOCAL COOLING EFFECTS

### Examples

```
calcTWC(6.8,-25)
```

---

## calcUTCI — *Universal Thermal Comfort Index (UTCI)*

---

### Description

Functions to calculate UTCI.

### Usage

```
calcUTCI(ta, tr, vel, rh)
```

### 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 [%] |

### Details

air temperature and mean radiant temperature should be in Degree C unit. Air velocity has to be in m/s unit and relative humidity has to be put in percentage value.

### Value

the `utciValue` value rounded to one decimal

### Author(s)

Code implemented in to R by Shaomi Rahman. Further contribution by Marcel Schweiker.

### References

UTCI project page on http://www.utci.org/ Original code in Python by Tartarini & Schiavon (2020) <doi:10.1016/j.softx.2020.100578>

### See Also

see also `calcComfInd`

### Examples

```
calcUTCI(25, 25, 1.0, 50) # Returns 24.6
```

calcVTG *PPD with Vertical Air Temperature Gradient*

## Description

Function to calculate vertical air temperature gradient using the predicted percentage of dissatisfied.

## Usage

```
calcVTG(ta, tr, vel, rh, clo, met, v_tmp_grad)
```

## 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] |
| `v_tmp_grad` | vertical temperature gradient between the feet and the head [degree C/m] |

## Details

Calculates the percentage of thermally dissatisfied persons with a vertical temperature gradient between feet and head. Applicable only for velocity(vel) < 0.2 m/s

## Value

Predicted Percentage of Dissatisfied with vertical temperature gradient in [%]

Acceptability in [boolean]

## Author(s)

Code implemented in to R by Shoaib Sarwar. Further contribution by Marcel Schweiker.

## References

Original code in Python by Tartarini & Schiavon (2020) <doi:10.1016/j.softx.2020.100578>

## Examples

```
calcVTG(25,25,0.1,50,0.5,1.2,7)
# returns Vertical Air Temperature Gradient:12.4, Acceptability:FALSE
```

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)

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.

indices listed as `request`. For details see details above.

## Author(s)

Sophia Mueller and Marcel Schweiker.

## References

For references see individual functions.

## 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` *Calibration data for SET*

---

### Description

Data from ASHRAE 55-2013 to calibrate values given by SET model

### Usage

```
data(dfASHRAETableG11)
```

### Format

A data frame with 22 rows and 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]

## 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)
```

---

`dfField` *Field data example*

---

## 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 rows and 9 variables:

**ta** air temperature [degree C]

**tr** radiant temperature [degree C] - same as ta

**rh** relative humidity [%]

**trm** running mean outdoor temperature [degree C]

**clo** clothing insulation level [CLO]

**tao** outdoor air temperature [degree C]

**vel** indoor air velocity [m/s]

**met** metabolic rate [MET]

**asv** 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. <https://windsorconference.com/>

## Examples

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

---

`dfISO7730AppE` *Calibration data for PMV*

---

## Description

Data from ISO 7730 Appendix E to calibrate values given by PMV model

## Usage

```
data(dfISO7730AppE)
```

## Format

A data frame with 13 rows and 8 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)

## 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(dfISO7730AppE)
head(dfISO7730AppE)
```

`dfISO7730TableD1` *Calibration data for PMV and PPD*

### Description

Data from ISO 7730 to calibrate values given by PMV / PPD model

### Usage

```
data(dfISO7730TableD1)
```

### Format

A data frame with 13 rows and 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)

### 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)
```

`dfISO7933AppF` *Calibration data for Tre, SWtotg, Dlimtre, Dlimloss50, Dlimloss95*

## Description

Data from ISO 7933 Appendix F to calibrate values given by the proposed model

## Usage

```
data(dfISO7933AppF)
```

## Format

A data frame with 10 rows and 16 variables:

- `accl` a numeric vector of state of acclimatised subject, 100 if acclimatised, 0 otherwise [-]
- `posture` a numeric vector of posture of subject, posture = 1 sitting, =2 standing, =3 crouching [-]
- `Ta` a numeric vector of air temperature [degree C]
- `Pa` a numeric vector of partial water vapour pressure [kPa]
- `Tr` a numeric vector of mean radiant temperature [degree C]
- `Va` a numeric vector of air velocity (m/s)
- `Met` a numeric vector of metabolic rate (W/(m*m))
- `Icl` a numeric vector of static thermal insulation (clo)
- `THETA` a numeric vector of angle between walking direction and wind direction (degrees)
- `Walksp` a numeric vector of walking speed (m/s)
- `Duration` a numeric vector of the duration of the work sequence (min)
- `Tre` a numeric vector of final rectal temperature (degree C)
- `SWtotg` a numeric vector of total water loss (g)
- `Dlimtre` a numeric vector of maximum allowed exposition time for heat storage (min)
- `Dlimloss50` a numeric vector of maximum water loss for protection of an average person (g)
- `Dlimloss95` a numeric vector of maximum water loss for protection of 95% of the working people (g)

## References

ISO 7933 Ergonomics of the thermal environment - Analytical determination and interpretation of heat stress using calculation of the predicted heat strain 2004

## Examples

```
data(dfISO7933AppF)
head(dfISO7933AppF)
```

`dfUTCIValues` *Dataset with Different Combinations of Inputs to Calculate UTCI*

## Description

Dataset with Different Combinations of Inputs to Calculate UTCI

## Usage

```
data(dfUTCIValues)
```

## Format

A data frame with 81 rows and 5 variables:

- `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 [%]
- `utci` a numeric value presenting the UTCI value

## See Also

see also `calcUTCI`

## Examples

```
data(dfUTCIValues)
head(dfUTCIValues)
```

# Index