comf: An introduction

Marcel Schweiker

Version September 13, 2016

1 Introduction

The R-package *comf* was developed in order to facilitate the computation of thermal comfort indices and the preparation and usage of corresponding data.

This document describes the groups of functions included in this package. These can be grouped according to

1. The preparation of a dataset and conversion of physical variables

2. The calculation of one or more comfort indices

3. The comparison of the values calculated by this package with existing validation tables

4. The evaluation of the performance of one or more comfort indices

At this stage, 22 indices can be calculated. For a complete list, use:

```
install.packages("comf_0.1.5.tar.gz", repos=NULL, type="source")
library(comf)
?comf
```

Further information about each index together with corresponding references can be obtained by consulting the corresponding help page, e.g.

```
?pmv
?ATHBpmv
```

2 Preparation of a dataset and conversion of physical variables

Each of the thermal comfort indeces requires different input parameters. This pacakge offers two procedures in order to assure that all necessary input parameters of present in the dataset used to the calculate one or more thermal comfort indices.

2.1 Using a predefined list of parameters

The first procedure uses the function *createCond*, which creates a list of standard values of parameters required for **all** comfort indices included in this package, which could then be adjusted according to the existing data:

```
lsCond <- createCond()
lsCond$ta <- 21:30
lsCond$rh <- 51:60
lsCond$met <- 1.0
```

The length of vectors assigned to the elements of this list need to be either 1 or have the same length. In above example, it would not be possible to assign a vector with 15 items to *ta*, the indoor air temperature, and a vector with 9 items to *rh*, the relative humidity indoors.

2.2   Converting a dataframe with variables

For the second procedure, a dataframe containing all variables necessary for a calculation can be used. This procedure requires knowledge about the necessary parameters for a comfort index, which could be obtained through the corresponding help file. The dataframe can either be transferred into a list or used directly.

```
ta <- 21:30
rh <- 51:60
met <- 1.0
dfCond <- data.frame(ta, rh, met)
lsCond2 <- as.list(dfCond)
```

2.3   Additional functions

Additional small functions to convert variables from one type to another are among others

- *calcDewp*, which calculates the dew point temperature, given air temperature and relative humidity,

- *calcEnth*, which calculates the enthalpy of the air, given air temperature, relative humidity, and barometric pressure,

- *calcRH*, which calculates the relative humidity of air, given air temperature, mixing ratio, and barometric pressure,

- *calcTroin*, which calculates the operative and radiant temperature for standard globe measurements according to ISO 7726, given air temperature, globe temperature, air velocity, and metabolic rate.

3   Calculating one or more comfort indices

There are two possibilities to calculate one or more comfort indices

1. Calculating one or more indices using the function *calcComfInd* or

2. Calculating one index with its corresponding function.

3.1   Calculating one or more indices using the function *calcComfInd*

The function *calcComfInd* requires a list or data frame of variables together with a vector of comfort indices to be calculated, e.g. *request="all"* to calculate all indices or *request=c("ATHBpmv", "pmv")*. The list of variables can consist of one element per variable or several elements per variable, e.g. one value for each input parameter, or for some parameters 234 values and for the others one parameter. This was implemented , because variables such as age, gender, or metabolic rate often do not differ in a given dataset, while others like the indoor air temperature are different for each case. A complete list of indices to be calculated can be found through

```
?calcComfInd
# or
listOfRequests()
```

The function *calcComfInd* performs preliminary checks before computing the requested indices. In case one or more required variables do not exist in the list, pre-defined standard values for these variables are used. A warning is given to inform the user about the missing variable(s) and the standard value(s) used for the calculation.

```
# using lsCond from above does not produce a warning
calcComfInd(lsCond, request="all")


# using lsCond2 from above displays 31 warnings which report
# the corresponding standard values used
calcComfInd(lsCond2, request="all")
warnings()


# the results however are identical
```

3.2   Calculating an index with its corresponding function

Individual functions, e.g. *calcSET* to calculate SET, can be used to calculate one or more comfort indices. With this procedure, one can use a data frame or a list of vectors to calculate a specific thermal comfort index.

The following example illustrates the usage for multiple input lines:

```
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]) } )
```

4   Comparing the values calculated by this package with existing validation tables

Included in this package are two tables to compare the output of computer programs with standard values. These are Table D.1 from ISO 7730 and Table C.1 from ASHRAE 55-2013.

```
data(dfISO7730TableD1)
head(dfISO7730TableD1)


data(dfASHRAETableC1)
head(dfASHRAETableC1)
```

Note that one value differs from ISO 7730 Table D1 to the values in *dfISO7730TableD1*, as a comparison with the calculation done by this package and the values given by the CBE Thermal Comfort Tool (ASHRAE approved) showed a different value. The difference is:

- row 7, PMV must be .36 (not .5) and PPD 8 (not 10)

Also note that some values in *dfASHRAETableC1* differ from those in ASHRAE 55-2013 Table C1, as a comparison with the calculation done by this package and the values given by the CBE Thermal Comfort Tool (ASHRAE approved) showed slightly adjusted values. The differences are:

- row 6 should be 26.4 degree C not 26.2

- row 7 should be 34.3 degree C not 33.6

- row 9 should be 24.9 degree C not 24.4

- row 12 should be 20.3 degree C not 20.4

- row 15 should be 31.8 degree C not 31.9

- row 18 should be 32.5 degree C not 32.6

- row 19 should be 37.7 degree C not 38.0

No changes have been made to the Fahrenheit values, which might be different as well.

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

A comparison can look as follows

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

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

5 Evaluating the performance of one or more comfort indices

The comparison between the performance of different indices with respect to the prediction of thermal sensation or thermal comfort will be an important aspect in future studies. This package offers the functions

- *calcBias* to calculate the mean bias, its standard deviation, and standard error between the actual (observed) thermal sensation vote (ASV) and the predicted thermal sensation vote (PSV)

- *calcTPRTSV*, which calculates the true positive rate (TPR): the proportion of true predicted cases, where the categorical ASV is equal to the categorical PSV

- *calcAvgAcc*, which calculates the average accuracy between PSV and ASV.

Using the example data from above together with an additional column for the ASV, such evaluation can look like this:

```
ta <- c(20,22,24)
tr <- ta
vel <- rep(.15,3)
rh <- rep(50,3)
asv <- c(-1.5, -1, .5)

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

plot(PTS~asv, xlim=c(-2,1), ylim=c(-2,1))
abline(0,1)

calcBias(asv, PTS)
```

Such evaluation is underlying e.g., the following paper:

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.

A sample of the dataset used for that paper is included in this package and can be obtained through:

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