Package 'masscor'

October 13, 2022

Title Mass Measurement Corrections

Version 0.0.7.1

Date 2021-09-09

Description Mass measurement corrections and uncertainties using calibration data, as recommended by EURAMET's guideline No. 18 (2015) ISBN:978-3-942992-40-4 . The package provides classes, functions, and methods for storing information contained in calibration certificates and converting balance readings to both conventional mass and real mass. For the latter, the Magnitude of the Air Buoyancy Correction factor employs models (such as the CIMP-2007 formula revised by Picard, Davis, Gläser, and Fujii (2008) <doi:10.1088/0026-1394/45/2/004>) to estimate the local air density using measured environmental conditions.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

Imports metRology

Depends R (>= 3.5.0)

VignetteBuilder knitr

NeedsCompilation no

Author Cristhian Paredes [aut, cre] (<https://orcid.org/0000-0001-7049-9597>)

Maintainer Cristhian Paredes <craparedesca@unal.edu.co>

Repository CRAN

Date/Publication 2021-09-13 08:20:02 UTC

R topics documented:

`masscor-package` `masscor`*: Mass Measurement Corrections and Uncertainties.*

Description

The `R` package `masscor` provides functions, classes and methods to support mass measurements using non automatic balances as described in EURAMET's Calibration Guide No. 18 (2015). The new classes are objects that can store the calibration information for balances and mass standards. Those objects can be used to convert balance readings to both conventional mass and mass, and to perform routine balance verification (by using the normalized error function). Air buoyancy correction factors are calculated using local air density that can be calculated using environmental conditions and applying one of several models available in the package. The uncertainty of (corrected) mass measurements can also be evaluated allowing us to further assess the suitability of given mass measurement.

masscor functions

This package uses list objects of class 'calibCert' to store information of balance calibration certificates. The functions use the information of this object to convert balance reading indications to conventional mass and calculate mass uncertainties.

Several models for calculating air density are included and this information can be used to calculate the Magnitude of the Air Buoyancy Correction factor (MABC). Uncertainties calculations are made using Gauss Approximation according to the Guide to the Expression of Uncertainty in Measurement (GUM) implemented in R by the package metRology (Ellison, 2018).

Author(s)

Cristhian Paredes, <craparedesca@unal.edu.co>

References

EURAMET, Calibration Guide No. 18. 2015. Guidelines on the Calibration of Non-Automatic Weighing Instruments. https://www.euramet.org/Media/docs/Publications/calguides/I-CAL-GUI-018_Calibration_Guide_No._18_web.pdf.

Picard, A; Davis, R S; Gläser, M; Fujii, K (2008). Revised formula for the density of moist air (CIPM-2007). Metrologia, 45(2), 149–155. doi:10.1088/0026-1394/45/2/004

Harris, G. (2019). Selected Laboratory and Measurement Practices and Procedures to Support Basic Mass Calibrations. SOP 2 - Recommended Standard Operating Procedure for Applying Air Buoyancy Corrections. National Institute of Standards and Technology (NIST). doi:10.6028/NIST.IR.6969-2019

BIMP JCGM (2008) Evaluation of measurement data — Guide to the expression of uncertainty in measurement.

Stephen L R Ellison. (2018). metRology: Support for Metrological Applications. R package version 0.9-28-1. https://CRAN.R-project.org/package=metRology

---

airDensity *Models for calculating air density based on environmental conditions*

Description

The function uses environmental conditions information (barometric pressure, temperature and relative humidity.) to calculate a local air density value. If no parameter is defined, the air density at 20°C, 1013.25 hPa and 50% relative humidity is returned. The air density value value can later be used to calculate the Magnitude of Air Buoyancy Correction (MABC()). The uncertainty of the air density value can be calculated using the function uncertAirDensity().

Usage

```
airDensity(Temp = 20, p = 1013.25, h = 50, unitsENV = c("deg.C", "hPa",
  "%"), x_CO2 = 4e-04, model = "CIMP2007")
```

Arguments

`Temp`	ambient temperature in weighing room.
`p`	barometric pressure in weighing room.
`h`	relative humidity in weighing room.
`unitsENV`	character vector of length three with the units of `p`, `Temp` and `h`. Default is `c('deg.C', 'hPa', '%')`. See **unitsENV** below for more options.
`x_CO2`	molar fraction of carbon dioxide in the air inside weighing room.
`model`	model to use for air density calculation. Must be one of `'CIMP2007'` (default), `'CIMP.approx'` or `'Jones1978'`. See See Details for references.

Details

Local air density can be estimated using one of several methods. The most complete approach is the CIMP complete formula (the default, `method = 'CIMP2007'`) as described in Picard et al (2008). The CIMP approximated exponential formula (`method = 'CIMP.approx'`) and the method reported by Jones, (`method = 'Jones1978'`) (Harris, 2019) are also included.

Value

Numeric value of air density in $g\ cm^{-3}$, according to chosen model.

unitsENV

Temperature units (`Temp`) can be either `'deg.C'` (for Celsius degrees) or `'K'`. Pressure units (`p`) can be any of `'mmHg'`, `'Pa'`, `'hPa'` or `'kPa'`. Relative humidity (`h`) can be expressed as fraction (`'frac'`) or as percentage (`'%'`). A typical arrangement for the parameter `unitsENV` would be `c('deg.C', 'hPa', '%')`.

References

Picard, A; Davis, R S; Gläser, M; Fujii, K (2008). Revised formula for the density of moist air (CIPM-2007). Metrologia, 45(2), 149–155. doi:10.1088/0026-1394/45/2/004

Harris, G. (2019). Selected Laboratory and Measurement Practices and Procedures to Support Basic Mass Calibrations. SOP 2 - Recommended Standard Operating Procedure for Applying Air Buoyancy Corrections. National Institute of Standards and Technology (NIST). doi:10.6028/NIST.IR.6969-2019

Preguntar a andres referencia de la f'ormula simplificada exponencial

See Also

`MABC()` to calculate the Magnitude of Air Buoyancy Correction and `uncertAirDensity()` to estimate the uncertainty of the calculated air density.

Examples

```
airDensity(Temp = 23.4, p = 612.3, h = 23,
           unitsENV = c('deg.C', 'mmHg', '%')) # [g/cm^3]
airDensity(Temp = 23.4, p = 612.3, h = 23,
           unitsENV = c('deg.C', 'mmHg', '%'), model = 'CIMP.approx') # [g/cm^3]
airDensity(Temp = 23.4, p = 612.3, h = 23,
           unitsENV = c('deg.C', 'mmHg', '%'), model = 'Jones1978')   # [g/cm^3]
```

---

airDensityHASL *Air density estimation using only height above sea level.*

Description

Calculates the approximated density of local air using a model that relies on height above sea level (HASL) information. More accurate alternatives are found in `airDensity()` but those require data form environmental conditions (temperature, barometric pressure and relative humidity).

Usage

```
airDensityHASL(HASL)
```

Arguments

HASL	height altitude above sea level in meters.

Value

Numeric value of an approximated air density in g cm^{-3}.

See Also

`airDensity()` for better models to predict air density.

Examples

```
airDensityHASL(HASL = 0)    # [g/cm^3]
airDensityHASL(HASL = 1600) # [g/cm^3]
```

Box.E2.MS.Kit *Calibration data of a weights kit in the interval 1 mg to 1 kg*

Description

An object of class "massStandardKit", with calibration information of mass standards with nominal masses in the interval 1 mg to 1 kg, performed by the Mass Laboratory at the Instituto Nacional de Metrologia de Colombia (2020-08-12).

Usage

```
Box.E2.MS.Kit
```

Format

A list with 25 objects of class massStandard()

Source

Mass Laboratory - Instituto Nacional de Metrologia de Colombia. CALIBRATION CERTIFICATE No. 4687.

calibCert *Information of balance calibration certificate*

Description

Creates an object of class calibCert that contains the information of a balance calibration certificate. The object can later be used to correct mass readings and calculate mass uncertainties. Mandatory arguments for this function are the balance division scale (d), the results of the indication error test (indError), the results of repeatability test (rep), and the results of the eccentricity test (eccen).

Usage

```
calibCert(balanceID = "BalanceID", serial = NULL, certificate = NULL, d,
  d.units = "mg", indError, indError.units = c("g", "mg", "mg"),
  expanded = TRUE, k = 2, rep, rep.units = c("g", "mg"), eccen,
  eccen.units = "mg", classSTD = NULL, traceability = NULL,
  Temp = NULL, p = NULL, h = NULL, unitsENV = c("deg.C", "hPa", "%"),
  institution = NULL, accreditation = NULL, date = NULL,
  add.info = NULL)
```

Arguments

balanceID	character with balance identification. May include balance model, brand or internal location.
serial	serial number of the balance.
certificate	character with the calibration certificate number and date of issue.
d	division scale of the balance.
d.units	character with the units of the division scale of the balance. Default value is 'mg'. See Details for more options.
indError	data.frame with the indication error test results in three columns containing balance reading, indication error and associated uncertainties, respectively, for at least two mass standards.
indError.units	character of length three with the units for each column in the data frame provided in indError. Default value is c('g', 'mg', 'mg').
expanded	if TRUE (the default), uncertainties provided in indError are assumed to be expanded uncertainties, instead of standard uncertainties.
k	coverage factor for the expanded uncertainties when expanded = TRUE.
rep	results of the repeatability test. If the test is performed in only one point, then rep is a numeric vector of length two with the balance load and standard deviation for the same object measured under repeatability conditions. If the test is performed at more than one point rep is a data frame with balance loadings in the first column and standard deviations in the second.
rep.units	character of length two with the units for balance loads and standard deviations provided in rep. Default value is c('g', 'mg').
eccen	numeric vector of length two with balance load and maximal reading difference obtained during eccentricity test.
eccen.units	character of length two with the units for balance loads and maximal reading difference provided in eccen. Default value is c('g', 'mg').
classSTD	character with the class of the mass standards used.
traceability	character with information regarding the traceability of the calibration.
Temp	ambient temperature at the moment of the calibration.
p	barometric pressure at the moment of the calibration.
h	relative humidity at the moment of the calibration.
unitsENV	character vector of length three with the units of p, Temp and h. Default is c('deg.C', 'hPa', '%'). See **unitsENV** below for more options.
institution	character with the identification of the calibration laboratory.
accreditation	character with the accreditation information of the calibration laboratory.
date	character with the date of the measurements.
add.info	named list or vector with any additional details included in the calibration certificate.

Details

The units of d, indError, rep and eccen shall be provided to the arguments d.units, indError.units, rep.units and eccen.units, respectively. The units can be any multiple or subdivision of the SI unit for mass, the kilogram. The greek letter μ used to represent a millionth part, is replaced by the vocal u. Remember that both R and the SI prefixes are case sensitive.

Value

Object of class calibCert with information of the calibration certificate for a balance.

unitsENV

Temperature units (Temp) can be either 'deg.C' (for Celsius degrees) or 'K'. Pressure units (p) can be any of 'mmHg', 'Pa', 'hPa' or 'kPa'. Relative humidity (h) can be expressed as fraction ('frac') or as percentage ('%'). A typical arrangement for the parameter unitsENV would be c('deg.C', 'hPa', '%').

See Also

S3 methods print.calibCert() and plot.calibCert() are available. See convertMassUnitsSI() for information about mass units.

Examples

```
massSTD  <- c(0.01, 0.5, 1, 10, 20, 50, 100, 120, 150, 200, 220)  ## [g]
indError <- c(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -0.2, -0.2) ## [mg]
uncert   <- c(0.1, 0.1, 0.1, 0.1, 0.1, 0.2, 0.2, 0.3, 0.3, 0.4, 0.5) ## [mg]
d <- 0.1 ## [mg]

Balance.D1 <- calibCert(balanceID = 'MT XPE 204', serial = 'B403223982',
                        d = d, d.units = 'mg',
                        indError = data.frame(massSTD, indError, uncert),
                        indError.units = c('g', 'mg', 'mg'),
                        rep = data.frame(load = c(0.1, 100, 220),
                                         sd = c(0.00, 0.04, 0.03)),
                        rep.units = c('g', 'mg'),
                        eccen = c(100, 0.1), eccen.units = c('g', 'mg'),
                        Temp = c(17.4, 17.9), ## [deg.C]
                        p = c(750.4, 751.0), ## [hPa]
                        h = c(70.5, 71.4), ## [%]
                        unitsENV = c('deg.C', 'hPa', '%'),
                        institution = 'Instituto Nacional de Metrologia de Colombia',
                        date = '2021-03-18')
print(Balance.D1)
```

convertMassUnitsSI *Conversion between mass units of The International System of Units*

Description

Mass values are converted from a SI unit to another SI unit according to the SI prefixes as shown in BIMP (2019). The greek letter μ is replaced by the vocal u.

Usage

```
convertMassUnitsSI(value, from, to)
```

Arguments

value	numeric vector with the values to be converted.
from	character with the unit of original values.
to	character with the desired unit for the conversion.

Value

Numeric vector of mass values converted from a SI unit to another mass unit.

References

BIMP, 2019. Bureau International des Poids et Mesures. Brochure of The International System of Units. 9th Edition.

Examples

```
convertMassUnitsSI(value = c(0.2, 0.4), from = 'mg', to = 'g')
```

convMass *Corrects a balance reading using balance calibration data.*

Description

Given a balance reading indication and the calibration information of the balance, the function interpolates error correction for the reading using the errors of indication for the two closest calibration points. The output is generally the mass measurement result under the conditions of calibration. If densities from the object and the local air are provided the conventional mass of the object can be calculated. See Details.

Usage

```
convMass(calibCert, reading, units = NULL, rho = NULL, rho_air = NULL)
```

Arguments

calibCert	object of class "calibCert" with the calibration data of the balance. See calibCert() for details.
reading	numeric with balance reading for the mass of the object.
units	character with the units of reading. Must be a SI unit. If not provided, the balance standard units are assumed. See calibCert() for details.
rho	density of the object in g cm^{-3}.
rho_air	density of local air in g cm^{-3}.

Details

The conventional mass value of a body is equal to the mass m_c of a mass standard that balances this body under conventionally chosen conditions: at a temperature $t_{ref} = 20^o$C, with mass standards of density $\rho_c = 8000$ kg m^{-3}, in normal air of density $\rho_0 = 1.2$ kg m^{-3} (OIML, 2004).

Value

Numeric value of conventional mass.

References

OIML, (2004). ORGANISATION INTERNATIONALE DE MÉTROLOGIE LÉGALE. International Document D 28: Conventional value of the result of weighing in air.

Harris, G. (2019). Selected Laboratory and Measurement Practices and Procedures to Support Basic Mass Calibrations. SOP 2 - Recommended Standard Operating Procedure for Applying Air Buoyancy Corrections. National Institute of Standards and Technology (NIST). doi:10.6028/NIST.IR.6969-2019

See Also

uncertConvMass()

Examples

```
data(minimalCert)
convMass(reading = 12.4835, calibCert = minimalCert)
```

E2.MS.20g *Calibration data of a E2 class mass standard of nominal mass 20 g.*

Description

An object of class "massStandard", with calibration information of a mass standards with nominal mass of 20 g, performed by the Mass Laboratory at the Instituto Nacional de Metrologia de Colombia (2020-08-12).

Usage

```
E2.MS.20g
```

Format

A list with 25 objects of class `massStandard()`

Source

Mass Laboratory - Instituto Nacional de Metrologia de Colombia. CALIBRATION CERTIFICATE No. 4687.

MABC *Magnitude of the Air Buoyancy Correction*

Description

Calculates the Magnitude of the Air Buoyancy Correction (MABC). If no parameters are provided the function returns MABC for weighing water at standard conditions.

Usage

```
MABC(rho = 0.997, rho_w = 8, rho_air = airDensity())
```

Arguments

`rho`	density of the sample in $g\ cm^{-3}$
`rho_w`	density of the weigths in $g\ cm^{-3}$
`rho_air`	density of the air in $g\ cm^{-3}$. If not provided, the value returned by the function `airDensity()` with no parameters is used. See `airDensity()` for details.

Details

Comparing masses (weighing) in air produces results that are influenced by the objects densities due to their buoyancy in air. This air buoyancy effects are usually small but must be taken in account when high accuracy is required. The effect can be corrected by using the densities of the object, the mass standard and the air filling the room where the measurement process takes place (Harris, 2019).

The uncertainty associated to MABC can be calculated by the function `uncertMABC()`.

Value

Numeric value of the Magnitude of the Air Buoyancy Correction factor.

References

Harris, G. (2019). Selected Laboratory and Measurement Practices and Procedures to Support Basic Mass Calibrations. SOP 2 - Recommended Standard Operating Procedure for Applying Air Buoyancy Corrections. National Institute of Standards and Technology (NIST). doi:10.6028/NIST.IR.6969-2019

See Also

uncertMABC(), airDensity(), uncertMABC()

Examples

```
## Magnitude of the air buoyancy correction for some materials:
# Water
MABC()
# Zinc metal
MABC(rho = 7.133)
# Copper metal
MABC(rho = 8.96)
```

---

massStandard *Creates an object of class* "massStandard".

Description

The object of class "massStandard" contains the calibration information of a mass standard that is used in routine balance verification (e g. to calculate normalized error. See normalizedError()). A version to store information of several mass standards that belong to the same kit is massStandardKit().

Usage

```
massStandard(nominal, convMassCor, uncert, units = c("g", "mg", "mg"),
  serial = NULL, manufacturer = NULL, class = NULL, certificate = NULL,
  traceability = NULL, Temp = NULL, p = NULL, h = NULL,
  unitsENV = c("deg.C", "hPa", "%"), expanded = TRUE, k = 2, rho = 8,
  u_rho = 0.06, unitsrho = "g/cm^3", institution = NULL, date = NULL,
  add.info = NULL, partofakit = FALSE)
```

Arguments

nominal	nominal mass of the mass standard.
convMassCor	conventional mass correction for the mass standard.
uncert	standard uncertainty of the conventional mass correction.
units	character vector of length 3 with the units of nominal, convMass and uncert, respectively. Default is c('g', 'mg', 'mg').
serial	serial number of the mass standard or mass standards kit.

manufacturer	character with the manufacturer of the mass standard or mass standards kit.
class	character with the claimed class of the mass standard or mass standards kit, according to OIML (2004).
certificate	character with the calibration certificate number and date of issue.
traceability	character with information regarding the traceability of the calibration.
Temp	ambient temperature at the moment of the calibration.
p	barometric pressure at the moment of the calibration.
h	relative humidity at the moment of the calibration.
unitsENV	character vector of length three with the units of p, Temp and h. Default is c('deg.C', 'hPa', '%'). See **unitsENV** below for more options.
expanded	if TRUE (the default), uncertainties provided in indError are assumed to be expanded uncertainties, instead of standard uncertainties.
k	coverage factor for the expanded uncertainties when expanded = TRUE.
rho	density of the mass standard.
u_rho	Uncertainty in the density of the mass standard.
unitsrho	Units of the density of the mass standard. Default is 'g/cm^3'.
institution	character with the identification of the calibration laboratory.
date	character with the date of the measurements.
add.info	named list or vector with any additional details included in the calibration certificate.
partofakit	Logical. Is the mass standard part of kit?

Value

Object of class "massStandard" with the information of a calibrated mass standard.

unitsENV

Temperature units (Temp) can be either 'deg.C' (for Celsius degrees) or 'K'. Pressure units (p) can be any of 'mmHg', 'Pa', 'hPa' or 'kPa'. Relative humidity (h) can be expressed as fraction ('frac') or as percentage ('%'). A typical arrangement for the parameter unitsENV would be c('deg.C', 'hPa', '%').

References

OIML, (2004). ORGANISATION INTERNATIONALE DE MÉTROLOGIE LÉGALE. International Document OIML R 111: Weights of classes E1, E2, F1, F2, M1, M1–2, M2, M2–3 and M3.

See Also

massStandardKit(), normalizedError()

Examples

```
singleMS.E2.10g <- massStandard(nominal = 10, convMassCor = 0.001,
                                uncert = 0.1,
                                units = c('g', 'mg', 'mg'))
print(singleMS.E2.10g)
```

`massStandardKit` *Creates an object of class* "massStandardKit".

Description

The object of class "massStandardKit" is a wrapper for several objects of class "massStandard". The object of class "massStandard" contains the calibration information of a mass standard that is used in routine balance verification (e g. to calculate normalized error. See `normalizedError()`). When several mass standards are part of a kit their information can be in conveniently be stored together in a "massStandardKit" class object.

Usage

```
massStandardKit(nominal, convMassCor, uncert, units = c("g", "mg", "mg"),
  serial = NULL, manufacturer = NULL, class = NULL, certificate = NULL,
  traceability = NULL, Temp = NULL, p = NULL, h = NULL,
  unitsENV = c("deg.C", "hPa", "%"), expanded = TRUE, k = 2,
  rho = NULL, u_rho = NULL, unitsrho = "g/cm^3", institution = NULL,
  date = NULL, add.info = NULL)
```

Arguments

`nominal`	vector with nominal mass for each standard all in the same units. If no weights are duplicated in the kit the vector can be numeric. If there are duplicated standards some elements must be character type. See Details.
`convMassCor`	numeric vector with conventional mass corrections for each of the mass standard declared in `nominal`.
`uncert`	numeric vector with standard uncertainties of the conventional mass corrections for each mass standard declared in `nominal`.
`units`	character vector of length 3 with the units of `nominal`, `convMass` and `uncert`, respectively. Default is `c('g', 'mg', 'mg')`.
`serial`	serial number of the mass standard or mass standards kit.
`manufacturer`	character with the manufacturer of the mass standard or mass standards kit.
`class`	character with the claimed class of the mass standard or mass standards kit, according to OIML (2004).
`certificate`	character with the calibration certificate number and date of issue.
`traceability`	character with information regarding the traceability of the calibration.
`Temp`	ambient temperature at the moment of the calibration.

p	barometric pressure at the moment of the calibration.
h	relative humidity at the moment of the calibration.
unitsENV	character vector of length three with the units of p, Temp and h. Default is c('deg.C', 'hPa', '%'). See **unitsENV** below for more options.
expanded	if TRUE (the default), uncertainties provided in indError are assumed to be expanded uncertainties, instead of standard uncertainties.
k	coverage factor for the expanded uncertainties when expanded = TRUE.
rho	numeric vector with densities for each mass standard declared in nominal. If not provided the default value of 8.000 g cm^{-3} is used for all weights.
u_rho	numeric vector with uncertainties in the density for each mass standard declared in nominal. If not provided the default value of 0.060 g cm^{-3} is used for all weights.
unitsrho	Units of the density of the mass standards. Default is 'g/cm^3'.
institution	character with the identification of the calibration laboratory.
date	character with the date of the measurements.
add.info	named list or vector with any additional details included in the calibration certificate.

Details

When two mass standards of equal nominal mass are present in the same kit (typically those of nominal mass in the form 2×10^n), it is necessary to make a distinction between them because their real mass are not likely to be exactly the same. Physically this distinction is achieved by marking one of them, for example, with a dot in the head of the knot weights, or by bending the final part of the lifted extreme in wire weights. To differentiate those duplicated mass standards in the "massStandardKit" class object, its nominal mass value must be entered as a character including an asterisk after the value (ie. '200*' instead of just entering 200). The function returns error if the kit contains duplicated mass standards and no differentiation is indicated.

Value

Object of class "massStandard" with calibration information of a mass standards kit.

unitsENV

Temperature units (Temp) can be either 'deg.C' (for Celsius degrees) or 'K'. Pressure units (p) can be any of 'mmHg', 'Pa', 'hPa' or 'kPa'. Relative humidity (h) can be expressed as fraction ('frac') or as percentage ('%'). A typical arrangement for the parameter unitsENV would be c('deg.C', 'hPa', '%').

See Also

massStandard(), normalizedError()

Examples

```
nominal     <- c(1000, 500, 200, '200*', 100)   # [g]
convMassCor <- c(0.0, -0.03, 0.03, 0.06, 0.00)  # [mg]
uncert      <- c(0.50, 0.25, 0.10, 0.10, 0.05)  # [mg]
units       <- c('g', 'mg', 'mg')

rho         <- c(8012.217, 8008.640, 8011.126, 8010.722, 8010.935)# [kg/m^3]
u_rho       <- c(0.096, 0.090, 0.160, 0.160, 0.321)# [kg/m^3]
unitsrho    <- 'kg/m^3'

MS.Kit1 <- massStandardKit(nominal = nominal, convMassCor = convMassCor, uncert = uncert,
                           units = units, rho = rho, u_rho = u_rho, unitsrho = unitsrho)
print(MS.Kit1)
```

---

minimalCert *Minimal* calibCert() *object*

Description

An object of class calibCert() with the minimal calibration information of a hypothetical balance calibrated using only two points.

Usage

```
minimalCert
```

Format

An object of class calibCert()

MT.XP.2002 *Calibration data for a balance Mettler Toledo XP 2002*

Description

An object of class calibCert() with the calibration information of a balance Mettler Toledo XP 2002, performed by the Mass Laboratory at the Instituto Nacional de Metrologia de Colombia (2021-03-12).

Usage

```
MT.XP.2002
```

Format

An object of class calibCert()

Source

Mass Laboratory - Instituto Nacional de Metrologia de Colombia. CALIBRATION CERTIFICATE No. 5142.

MT.XPE.204 *Calibration data for a balance Mettler Toledo XPE 204*

Description

An object of class `calibCert()` with the calibration information of a balance Mettler Toledo XPE 204, performed by the Mass Laboratory at the Instituto Nacional de Metrologia de Colombia (2021-03-18).

Usage

```
MT.XPE.204
```

Format

An object of class `calibCert()`

Source

Mass Laboratory - Instituto Nacional de Metrologia de Colombia. CALIBRATION CERTIFICATE No. 5143.

---

normalizedError *Calculates normalized in balance verification using a mass standard*

Description

Calculates normalized in balance verification using a mass standard

Usage

```
normalizedError(reading, standard, calibCert, u_massStandard = NULL)
```

Arguments

`reading`	balance reading for the standard mass.
`standard`	one of two options: an object of class `"massStandard"` (see `massStandard()`) or the numeric value of the conventional mass of the standard used.
`calibCert`	object of class `"calibCert"` with the calibration data of the balance. See `calibCert()` for details.
`u_massStandard`	uncertainty of the conventional mass of the standard used. Necessary only if `standard` is not an object of class `"massStandard"` (see `massStandard()`).

Value

Numeric value of normalized error for balance verification using a mass standard.

See Also

massStandard().

Examples

```
data(E2.MS.20g)
data(MT.XPE.204)
normalizedError(reading = 20.0000, standard = E2.MS.20g, calibCert = MT.XPE.204)
```

---

plot.calibCert *S3 method for plotting objects of class "calibCert"*

Description

The function plots the indication error or the conventional mass correction for a balance whose calibration data is in a object of class "calibCert".

Usage

```
## S3 method for class 'calibCert'
plot(x, error = TRUE, y0line = TRUE, ...)
```

Arguments

x	object of class "calibCert".
error	logical. If TRUE (the default), the indication error is plotted. If FALSE the conventional mass correction is plotted instead.
y0line	Logical. If TRUE (the default) a horizontal line is drawn at y = 0.
...	Other graphical parameters used in plot

Value

A base plot with calibration data of indication error or correction.

See Also

calibCert(), print.calibCert()

Examples

```
data(MT.XPE.204)
plot(MT.XPE.204)
```

print.calibCert *S3 method for printing objects of class "calibCert"*

Description

The function prints objects of class "calibCert".

Usage

```
## S3 method for class 'calibCert'
print(x, complete = FALSE, nudeCertificate = FALSE, ...)
```

Arguments

x	Object of class "massStandard".
complete	logical default to FALSE. If TRUE, all the information contained in the object of class calibCert is shown.
nudeCertificate	logical default to FALSE. If TRUE, the object of class calibCert is shown as a list.
...	Further arguments passed to or from other methods.

Value

No return value, called for side effects.

See Also

calibCert(), plot.calibCert()

Examples

```
data(MT.XPE.204)
print(MT.XPE.204)
```

print.massStandard *S3 method for printing objects of class "massStandard"*

Description

The function prints objects of class "massStandard".

Usage

```
## S3 method for class 'massStandard'
print(x, minimal = TRUE, description = TRUE,
  institution = TRUE, density = FALSE, envConditions = TRUE,
  addInfo = TRUE, ...)
```

Arguments

x	Object of class "massStandard".
minimal	logical default to TRUE. If TRUE, only minimal information regarding the calibration certificate is provided.
description	logical. If TRUE (the default) details of class, serial and manufacturer are printed. Ignored if minimal = TRUE.
institution	logical. If TRUE (the default) the calibrating institution information (including calibration traceability information) is printed. Ignored if minimal = TRUE.
density	logical. If TRUE the density information is printed.
envConditions	logical. If TRUE (the default) the environmental conditions at the place and the moment of calibration are printed. Ignored if minimal = TRUE.
addInfo	logical. If TRUE (the default) additional information of the calibration is printed. Ignored if minimal = TRUE.
...	further arguments passed to or from other methods.

Value

No return value, called for side effects.

See Also

massStandard(), print.massStandardKit()

Examples

```
data(E2.MS.20g)
print(E2.MS.20g)
print(E2.MS.20g, minimal = FALSE)
```

---

print.massStandardKit *S3 method for printing objects of class "massStandardKit"*

Description

The function prints objects of class "massStandardKit".

Usage

```
## S3 method for class 'massStandardKit'
print(x, minimal = FALSE, description = TRUE,
  institution = TRUE, density = FALSE, envConditions = TRUE,
  addInfo = TRUE, ...)
```

Arguments

x	object of class "massStandardKit".
minimal	logical default to TRUE. If TRUE, only minimal information regarding the calibration certificate is provided.
description	logical. If TRUE (the default) details of class, serial and manufacturer are printed. Ignored if minimal = TRUE.
institution	logical. If TRUE (the default) the calibrating institution information (including calibration traceability information) is printed. Ignored if minimal = TRUE.
density	logical. If TRUE the density information is printed.
envConditions	logical. If TRUE (the default) the environmental conditions at the place and the moment of calibration are printed. Ignored if minimal = TRUE.
addInfo	logical. If TRUE (the default) additional information of the calibration is printed. Ignored if minimal = TRUE.
...	further arguments passed to or from other methods.

Value

No return value, called for side effects.

See Also

massStandardKit(), print.massStandard()

Examples

```
data(Box.E2.MS.Kit)
print(Box.E2.MS.Kit, minimal = TRUE)
# We can print individual information of a single mass standard:
print(Box.E2.MS.Kit[['20']])
```

---

uncertAirDensity *Uncertainties in air density calculations*

Description

Propagates the uncertainty of environmental conditions measurements to estimated values of air densities calculated using the function airDensity() with models 'CIMP2007' y 'CIMP.approx' function. Uncertainty arising from to the chosen model itself is considered. Calculations are made according to the Guide to the Guide to the expression of uncertainty in measurement (GUM, JCGM, 2008) as implemented by the package metRology (Ellison, 2018).

Usage

```
uncertAirDensity(model = "CIMP2007", Temp = 20, p = 1013.25, h = 50,
  u_Temp = 2.9, u_p = 10.1, u_h = 11.3, unitsENV = c("deg.C", "hPa",
  "%"), plot = TRUE, printRelSD = TRUE)
```

Arguments

model	model to use for air density calculation. Must be one of 'CIMP2007' (default), 'CIMP.approx' or 'Jones1978'. See See Details for references.
Temp	ambient temperature in weighing room.
p	barometric pressure in weighing room.
h	relative humidity in weighing room.
u_Temp	standard uncertainty of temperature measurement
u_p	standard uncertainty of barometric pressure measurement
u_h	standard uncertainty of relative humidity
unitsENV	character vector of length three with the units of p, Temp and h. Default is c('deg.C', 'hPa', '%'). See **unitsENV** below for more options.
plot	logical. If TRUE (the default), the relative uncertainty contributions are shown in a barplot.
printRelSD	Logical. If TRUE (the default), a short statement indicating relative standard uncertainty of the air density estimation is printed.

Value

A numeric value of standard uncertainty of calculated air density in $g\ cm^{-3}$.

unitsENV

Temperature units (Temp) can be either 'deg.C' (for Celsius degrees) or 'K'. Pressure units (p) can be any of 'mmHg', 'Pa', 'hPa' or 'kPa'. Relative humidity (h) can be expressed as fraction ('frac') or as percentage ('%'). A typical arrangement for the parameter unitsENV would be c('deg.C', 'hPa', '%').

References

Picard, A; Davis, R S; Gläser, M; Fujii, K (2008). Revised formula for the density of moist air (CIPM-2007). Metrologia, 45(2), 149–155. doi:10.1088/0026-1394/45/2/004

Harris, G. (2019). Selected Laboratory and Measurement Practices and Procedures to Support Basic Mass Calibrations. SOP 2 - Recommended Standard Operating Procedure for Applying Air Buoyancy Corrections. National Institute of Standards and Technology (NIST). doi:10.6028/NIST.IR.6969-2019

BIMP JCGM (2008) Evaluation of measurement data — Guide to the expression of uncertainty in measurement.

Stephen L R Ellison. (2018). metRology: Support for Metrological Applications. R package version 0.9-28-1. https://CRAN.R-project.org/package=metRology

See Also

airDensity()

Examples

```
uncertAirDensity(model = 'CIMP2007',
                 Temp = 20, p = 1013.25, h = 50,
                 u_Temp = 0.29, u_p = 1.01, u_h = 11.3)
```

---

uncertConvMass *Uncertainty in conventional mass value*

Description

The function combines the uncertainty of the conventional mass correction (as obtained by uncertErrorCorr()) and the uncertainty in the balance reading (as obtained by uncertReading()), to produce the uncertainty of a conventional mass value.

Usage

```
uncertConvMass(calibCert, reading, units, sd, sd.units, d, d.units)
```

Arguments

calibCert	object of class "calibCert" with the calibration data of the balance. See calibCert() for details.
reading	numeric with balance reading for the mass of the object.
units	character with the units of reading. Must be a SI unit. If not provided, the balance standard units are assumed. See calibCert() for details.
sd	standard deviation when a balance reading is the result of averaging several individual measurements of same object. If not provided the information is taken from the calibration certificate of the balance
sd.units	character with the units of standard deviation. If not provided, the value stated at units or the balance standard units is used.
d	balance division scale. Useful when the balance is operated a a division scale different from that stated in the calibration certificate. This is the common case when the user is give up some readability in order to make faster mass measurements. If not provided. the functions uses the balance division scale stated in the calibration certificate.
d.units	character with the units of parameter d. If not provided, the value stated at units or the balance standard units is used.

Value

A numeric value of uncertainty for a conventional mass value.

See Also

convMass(), uncertReading(), uncertErrorCorr()

Examples

```
data(minimalCert)
uncertConvMass(reading = 12.4835, calibCert = minimalCert)
```

---

uncertErrorCorr *Uncertainty due to mass correction using calibration certificate*

Description

Given a balance reading indication and the calibration information of the balance, the function uses the conventional mass correction uncertainties of the two closest calibration points to the balance reading to interpolate the uncertainty due to the conventional mass correction.

Usage

```
uncertErrorCorr(calibCert, reading, units = NULL)
```

Arguments

calibCert	object of class "calibCert" with the calibration data of the balance. See calibCert() for details.
reading	numeric with balance reading for the mass of the object.
units	character with the units of reading. Must be a SI unit. If not provided, the balance standard units are assumed. See calibCert() for details.

Value

A numeric value of uncertainty for a conventional mass correction.

See Also

convMass(), uncertReading(), uncertConvMass()

Examples

```
data(minimalCert)
uncertErrorCorr(reading = 12.4835, calibCert = minimalCert)
```

uncertMABC *Uncertainty of the Magnitude of the Air Buoyancy Correction factor*

Description

Propagates density uncertainties in the calculation of the Magnitude of Air Buoyancy Correction (See `MABC()`).

Usage

```
uncertMABC(rho = 0.998, rho_w = 8, rho_air = NULL, u_rho = 1e-04,
  u_rho_w = 0.006, u_rho_air = NULL, plot = TRUE, printRelSD = TRUE)
```

Arguments

`rho`	density of the sample in $g\ cm^{-3}$
`rho_w`	density of the weigths in $g\ cm^{-3}$
`rho_air`	density of the air in $g\ cm^{-3}$. If not provided, the value returned by the function `airDensity()` with no parameters is used. See `airDensity()` for details.
`u_rho`	standard uncertainty of the sample density.
`u_rho_w`	standard uncertainty of the mass standard density.
`u_rho_air`	standard uncertainty of air density. See `uncertAirDensity()`.
`plot`	logical. If `TRUE` (the default), the relative uncertainty contributions are shown in a barplot.
`printRelSD`	Logical. If `TRUE` (the default), a short statement indicating relative standard uncertainty of the air density estimation is printed.

Details

Calculations are made according to the Guide to the Guide to the expression of uncertainty in measurement (GUM, JCGM, 2008) as implemented by the package metRology (Ellison, 2018). If air density and associated uncertainty are not provided the default output values of the functions `airDensity()` and `uncertAirDensity()`, respectively, are used.

Value

Numeric value of uncertainty for the Magnitude of the Air Buoyancy Correction factor.

References

BIMP JCGM (2008) Evaluation of measurement data — Guide to the expression of uncertainty in measurement.

Andrej-Nikolai Spiess (2018). propagate: Propagation of Uncertainty. R package version 1.0-6. https://CRAN.R-project.org/package=propagate

uncertReading *Uncertainty of balance readings*

Description

Uncertainty in a given balance reading considering the effects of rounding error, lack of repeatability, eccentricity and balance taring.

Usage

```
uncertReading(calibCert, reading, units = NULL, sd = NULL,
  sd.units = NULL, d = NULL, d.units = NULL)
```

Arguments

calibCert	object of class "calibCert" with the calibration data of the balance. See calibCert() for details.
reading	numeric with balance reading for the mass of the object.
units	character with the units of reading. Must be a SI unit. If not provided, the balance standard units are assumed. See calibCert() for details.
sd	standard deviation when a balance reading is the result of averaging several individual measurements of same object. If not provided the information is taken from the calibration certificate of the balance
sd.units	character with the units of standard deviation. If not provided, the value stated at units or the balance standard units is used.
d	balance division scale. Useful when the balance is operated a a division scale different from that stated in the calibration certificate. This is the common case when the user is give up some readability in order to make faster mass measurements. If not provided. the functions uses the balance division scale stated in the calibration certificate.
d.units	character with the units of parameter d. If not provided, the value stated at units or the balance standard units is used.

Value

A numeric value of uncertainty for a balance reading.

See Also

uncertErrorCorr(), uncertConvMass()

Examples

```
data(minimalCert)
uncertReading(calibCert = minimalCert, reading = 12.4835)
uncertReading(calibCert = minimalCert, reading = 12.484, d = 1, d.units = 'mg')
```

Index