Package 'birk'

August 29, 2014

Type Package

Title MA Birk functions

Version 1.1

Date 2014-08-29

Author Matthew A Birk

Maintainer Matthew A Birk <matthewabirk@gmail.com>

Description

This is a compilation of functions that I found useful to make. It currently includes a unit of measurement conversion function and a standard error function that behaves complementary to sd().

License GPL-2

R topics documented:

`birk-package` *MA Birk functions*

Description

This is a compilation of functions that I found useful to make. It currently includes a unit of measurement conversion function and a standard error function that behaves complementary to sd().

Details

Package:	birk
Type:	Package
Version:	1.1
Date:	2014-08-29
License:	GPL-2

Author(s)

Matthew A. Birk <matthewabirk@gmail.com>

`conv_unit` *Convert Units of Measurement*

Description

This function converts common units of measurement for a variety of dimensions. See conv_unit_options for all options.

Usage

```
conv_unit(x, from, to)
```

Arguments

`x`	the measurement value in its orignial units
`from`	the unit in which the measurement was made
`to`	the unit to which the measurement is to be converted
	Acceleration: mm_per_sec2, cm_per_sec2, m_per_sec2, km_per_sec2, grav, in_per_sec2, ft_per_sec2, mi_per_sec2
	Area: nm2, um2, mm2, cm2, m2, hectacre, km2, in2, ft2, yd2, acre, mi2, naut_mi2
	Duration: nsec, usec, msec, sec, min, hr, day, wk, mon, yr, dec, cen, mil
	Energy: J, erg, cal, Cal, Wsec, kWh, MWh, BTU
	Flow: ml_per_sec, ml_per_min, ml_per_hr, l_per_sec, l_per_min, l_per_hr, m3_per_sec, m3_per_min, m3_per_hr, gal_per_sec, gal_per_min, gal_per_hr, ft3_per_sec, ft3_per_min, ft3_per_hr
	Length: nm, um, mm, cm, dm, m, km, inch, ft, yd, mi, naut_mi, light_yr
	Mass: ug, mg, g, kg, metric_ton, oz, lb, short_ton, long_ton, stone
	Power: uW, mW, W, kW, MW, GW, erg_per_sec, cal_per_sec, cal_per_hr, Cal_per_sec, Cal_per_hr, BTU_per_sec, BTU_per_hr, hp
	Pressure: uatm, atm, Pa, hPa, kPa, torr, mmHg, inHg, mbar, bar, dbar, psi
	Speed: mm_per_sec, cm_per_sec, m_per_sec, km_per_sec, inch_per_sec, ft_per_sec, kph, mph, knot
	Temperature: C, F, K
	Volume: ml, l, cm3, m3, us_tsp, us_tbsp, us_oz, us_cup, us_pint, us_quart, us_gal, inch3, ft3, imp_tsp, imp_tbsp, imp_oz, imp_pint, imp_quart, imp_gal

Details

The conversion values have been defined based primarily from international weight and measurement authorities (e.g. General Conference on Weights and Measures, International Committee for Weights and Measures, etc.). While much effort was made to make conversions as accurate as possible, you should check the accuracy of conversions to ensure that conversions are precise enough for your applications.

Note

Duration: Years are defined as 365.25 days and months are defined as 1/12 a year

Energy: cal is a thermochemical calorie (4.184 J) and Cal is 1000 cal (kcal or 4184 J)

Mass: All non-metric units are based on the avoirdupois system

Power: hp is mechanical horsepower, or 745.69 W

Author(s)

Matthew A. Birk <matthewabirk@gmail.com>>

See Also

conv_unit_options

Examples

```
conv_unit(2.54, cm, inch) # Result = 1 inch
conv_unit(seq(1, 10), kg, short_ton) # A vector of measurement values can be converted
```

---

conv_unit_options *Unit of Measurement Conversion Options*

Description

This dataset shows what units of measurement can be converted with the function conv_unit.

Usage

```
conv_unit_options
```

Details

Duration: Years are defined as 365.25 days and months are defined as 1/12 a year

Energy: cal is a thermochemical calorie (4.184 J) and Cal is 1000 cal (kcal or 4184 J)

Mass: All non-metric units are based on the avoirdupois system

Power: hp is mechanical horsepower, or 745.69 W

Source

The conversion values have been defined based primarily from international weight and measurement authorities (e.g. General Conference on Weights and Measures, International Committee for Weights and Measures, etc.). While much effort was made to make conversions as accurate as possible, you should check the accuracy of conversions to ensure that conversions are precise enough for your applications.

See Also

conv_unit

Examples

```
conv_unit_options
conv_unit_options[Pressure]
```

`se` *Standard Error*

Description

This function computes the standard error of the values in x. If na.rm is TRUE then missing values are removed before computation proceeds.

Usage

```
se(x, na.rm = FALSE)
```

Arguments

`x`	a numeric vector or an R object which is coercible to one by as.vector(x, "numeric").
`na.rm`	logical. Should missing values be removed?

Author(s)

Matthew A. Birk <matthewabirk@gmail.com>>

See Also

sd,var

Examples

```
se(1:10)
```

Index