Package ‘vismeteor’

May 8, 2026

Type Package

Title Analysis of Visual Meteor Data

Version 2.1.0

Author Janko Richter [aut, cre]

Maintainer Janko Richter <janko@richtej.de>

Description Provides a suite of analytical functionalities to process and analyze visual meteor observations from the Visual Meteor Database of the International Meteor Organization <https://www.imo.net/>.

URL https://github.com/jankorichter/vismeteor

BugReports https://github.com/jankorichter/vismeteor/issues

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Imports methods, stats, httr2 (>= 1.0.0)

RoxygenNote 7.3.3

Suggests testthat (>= 3.2.0), httptest2, knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2026-04-27 10:00:02 UTC

Contents

`vismeteor-package` *vismeteor: Analysis of Visual Meteor Data*

Description

Provides a suite of analytical functionalities to process and analyze visual meteor observations from the Visual Meteor Database of the International Meteor Organization `https://www.imo.net/`.

Details

The data used in this package can created and provided by imo-vmdb.

Author(s)

Maintainer: Janko Richter `<janko@richtej.de>`

See Also

Useful links:

- `https://github.com/jankorichter/vismeteor`
- Report bugs at `https://github.com/jankorichter/vismeteor/issues`

`freq.quantile` *Quantiles with a minimum frequency*

Description

This function generates quantiles with a minimum frequency. These quantiles are formed from a vector `freq` of frequencies. Each quantile then has the minimum total frequency `min`.

Usage

```
freq.quantile(freq, min)
```

Arguments

`freq`	integer; A vector of frequencies.
`min`	integer; Minimum total frequency per quantile.

Details

The frequencies `freq` are grouped in the order in which they are passed as a vector. The minimum `min` must be greater than `0`.

Value

A factor of indices is returned. The index references the corresponding passed frequency `freq`.

Examples

```
freq <- c(1,2,3,4,5,6,7,8,9)
cumsum(freq)
(f <- freq.quantile(freq, 10))
tapply(freq, f, sum)
```

---

load_vmdb *Loading visual meteor observations via the imo-vmdb REST API*

Description

Loads visual meteor observations from an imo-vmdb web server via its REST API.

Usage

```
load_vmdb_rates(
  base_url,
  shower = NULL,
  period = NULL,
  sl = NULL,
  lim.magn = NULL,
  sun.alt.max = NULL,
  moon.alt.max = NULL,
  session.id = NULL,
  rate.id = NULL,
  withSessions = FALSE,
  withMagnitudes = FALSE
)

load_vmdb_magnitudes(
  base_url,
  shower = NULL,
  period = NULL,
```

```
    sl = NULL,
    lim.magn = NULL,
    session.id = NULL,
    magn.id = NULL,
    withSessions = FALSE,
    withMagnitudes = TRUE
)
```

Arguments

base_url	character; base URL of the imo-vmdb API, e.g. "http://localhost:8000/api/v1".
shower	character; selects by meteor shower codes. NA loads sporadic meteors.
period	time; selects a time range by minimum/maximum.
sl	numeric; selects a range of solar longitudes by minimum/maximum.
lim.magn	numeric; selects a range of limiting magnitudes by minimum/maximum.
sun.alt.max	numeric; selects the maximum altitude of the sun (rates only).
moon.alt.max	numeric; selects the maximum altitude of the moon (rates only).
session.id	integer; selects by session ids.
rate.id	integer; selects rate observations by ids.
withSessions	logical; if TRUE, also load the corresponding session data.
withMagnitudes	logical; if TRUE, also load the corresponding magnitude observations.
magn.id	integer; selects magnitude observations by ids.

Details

sl, period and lim.magn expect a vector with successive minimum and maximum values. sun.alt.max and moon.alt.max are expected to be scalar values.

Note: Unlike the previous DBI-based version, only a single range per filter parameter is supported. If you previously passed a matrix with multiple rows to period, sl, or lim.magn, flatten them to a single min/max pair or issue multiple calls and combine with rbind().

Value

Both functions return a list, with

observations	data frame, rate or magnitude observations,
sessions	data frame; session data of observations,
magnitudes	table; contingency table of meteor magnitude frequencies.

observations depends on the function call. load_vmdb_rates returns a data frame with columns:

rate.id	unique identifier of the rate observation,
shower.code	IAU code of the shower. NA for sporadic.
period.start	start of observation,
period.end	end of observation,

sl.start	solar longitude at start,
sl.end	solar longitude at end,
session.id	reference to the session,
freq	count of observed meteors,
lim.magn	limiting magnitude,
t.eff	net observed time in hours,
f	correction factor of cloud cover,
time.sidereal	sidereal time,
sun.alt	altitude of the sun,
sun.az	azimuth of the sun,
moon.alt	altitude of the moon,
moon.az	azimuth of the moon,
moon.illum	illumination of the moon (0.0 .. 1.0),
field.alt	altitude of the field of view (optional),
field.az	azimuth of the field of view (optional),
radiant.alt	altitude of the radiant (optional),
radiant.az	azimuth of the radiant (optional),
magn.id	reference to the magnitude observations (optional).

load_vmdb_magnitudes returns an observations data frame with:

magn.id	unique identifier of the magnitude observation,
shower.code	IAU code of the shower. NA for sporadic.
period.start	start of observation,
period.end	end of observation,
sl.start	solar longitude at start,
sl.end	solar longitude at end,
session.id	reference to the session,
freq	count of observed meteors,
magn.mean	mean magnitude,
lim.magn	limiting magnitude (optional).

The sessions data frame contains

session.id	unique identifier of the session,
longitude	location's longitude,
latitude	location's latitude,
elevation	height above mean sea level in km,
country	country name,
location.name	location name,
observer.id	observer id (optional),
observer.name	observer name (optional).

magnitudes is a contingency table of meteor magnitude frequencies. Row names are magnitude observation IDs; column names are magnitude classes.

Note

Angle values are expected and returned in degrees.

References

https://pypi.org/project/imo-vmdb/

Examples

```
## Not run:
# Load rate observations including session and magnitude data
data <- load_vmdb_rates(
    base_url       = "http://localhost:8000/api/v1",
    shower         = "PER",
    sl             = c(135.5, 145.5),
    period         = c("2015-08-01", "2015-08-31"),
    lim.magn       = c(5.3, 6.7),
    withMagnitudes = TRUE,
    withSessions   = TRUE
)

# Load magnitude observations
data <- load_vmdb_magnitudes(
    base_url     = "http://localhost:8000/api/v1",
    shower       = "PER",
    sl           = c(135.5, 145.5),
    period       = c("2015-08-01", "2015-08-31"),
    lim.magn     = c(5.3, 6.7),
    withSessions = TRUE
)

## End(Not run)
```

---

mideal *Ideal Distribution of Meteor Magnitudes*

Description

Density, distribution function, quantile function and random generation for the ideal distribution of meteor magnitudes.

Usage

```
dmideal(m, psi = 0, log = FALSE)

pmideal(m, psi = 0, lower.tail = TRUE, log = FALSE)

qmideal(p, psi = 0, lower.tail = TRUE)

rmideal(n, psi = 0)
```

Arguments

m	numeric; meteor magnitude.
psi	numeric; the location parameter of a probability distribution. It is the only parameter of the distribution.
log	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default) probabilities are $P[M \le m]$, otherwise, $P[M > m]$.
p	numeric; probability.
n	numeric; count of meteor magnitudes.

Details

The density of the ideal distribution of meteor magnitudes is

$$\frac{\mathrm{d}p}{\mathrm{d}m} = \frac{3}{2}\,\log(r)\sqrt{\frac{r^{3\,\psi+2\,m}}{(r^\psi + r^m)^5}}$$

where m is the continuous (real-valued) meteor magnitude, $r = 10^{0.4} \approx 2.51189\ldots$ is a constant and ψ is the only parameter of this magnitude distribution.

Value

dmideal gives the density, pmideal gives the distribution function, qmideal gives the quantile function and rmideal generates random deviates.

The length of the result is determined by n for rmideal, and is the maximum of the lengths of the numerical vector arguments for the other functions.

qmideal can return NaN value with a warning.

References

Richter, J. (2018) *About the mass and magnitude distributions of meteor showers*. WGN, Journal of the International Meteor Organization, vol. 46, no. 1, p. 34-38

Examples

```
old_par <- par(mfrow = c(2,2))
psi <- 5.0
plot(
    \(m) dmideal(m, psi, log = FALSE),
    -5, 10,
   main = paste0('Density of the Ideal Meteor Magnitude\nDistribution (psi = ', psi, ')'),
    col = "blue",
    xlab = 'm',
    ylab = 'dp/dm'
)
abline(v=psi, col="red")

plot(
    \(m) dmideal(m, psi, log = TRUE),
```

```
    -5, 10,
    main = paste0('Density of the Ideal Meteor Magnitude\nDistribution (psi = ', psi, ')'),
    col = "blue",
    xlab = 'm',
    ylab = 'log( dp/dm )'
)
abline(v=psi, col="red")

plot(
    \(m) pmideal(m, psi),
    -5, 10,
    main = paste0('Probability of the Ideal Meteor Magnitude\nDistribution (psi = ', psi, ')'),
    col = "blue",
    xlab = 'm',
    ylab = 'p'
)
abline(v=psi, col="red")

plot(
    \(p) qmideal(p, psi),
    0.01, 0.99,
    main = paste('Quantile of the Ideal Meteor Magnitude\nDistribution (psi = ', psi, ')'),
    col = "blue",
    xlab = 'p',
    ylab = 'm'
)
abline(h=psi, col="red")

# generate random meteor magnitudes
m <- rmideal(1000, psi)

# log likelihood function
llr <- function(psi) {
    -sum(dmideal(m, psi, log=TRUE))
}

# maximum likelihood estimation (MLE) of psi
est <- optim(2, llr, method='Brent', lower=0, upper=8, hessian=TRUE)

# estimations
est$par # mean of psi
sqrt(1/est$hessian[1][1]) # standard deviation of psi

par(old_par)
```

PER_2015_magn   *Visual magnitude observations of Perseids from 2015*

Description

Visual magnitude observations of the Perseid shower from 2015.

Format

A list with the same structure as returned by load_vmdb_magnitudes.

Details

`PER_2015_magn` are magnitude observations loaded with load_vmdb_magnitudes.

See Also

load_vmdb

`PER_2015_rates` *Visual rate observations of Perseids from 2015*

Description

Visual rate observations of the Perseid shower from 2015.

Format

A list with the same structure as returned by load_vmdb_rates.

Details

`PER_2015_rates` are rate observations loaded with load_vmdb_rates.

See Also

load_vmdb

`vmgeom` *Geometric Model of Visual Meteor Magnitudes*

Description

Density, distribution function, quantile function, and random generation for the geometric model of visual meteor magnitudes.

Usage

```
dvmgeom(m, lm, r, log = FALSE, perception.fun = NULL)

pvmgeom(m, lm, r, lower.tail = TRUE, log = FALSE, perception.fun = NULL)

qvmgeom(p, lm, r, lower.tail = TRUE, perception.fun = NULL)

rvmgeom(n, lm, r, perception.fun = NULL)
```

Arguments

m	integer; the meteor magnitude.
lm	numeric; limiting magnitude.
r	numeric; the population index.
log	logical; if TRUE, probabilities p are given as log(p).
perception.fun	function; perception probability function (optional). Default is vmperception.
lower.tail	logical; if TRUE (default) probabilities are $P[M < m]$, otherwise, $P[M \geq m]$.
p	numeric; probability.
n	numeric; count of meteor magnitudes.

Details

In visual meteor observations, magnitudes are estimated as integer values. Consequently, the distribution of observed magnitudes is discrete, and its probability mass function is given by

$$P[M = m] \sim \begin{cases} f(m_{\lim} - m)\, r^m, & \text{if } m_{\lim} - m > -0.5, \\ 0 & \text{otherwise,} \end{cases}$$

where $m_{\lim}$ denotes the limiting (non-integer) magnitude of the observation, and m the integer meteor magnitude. The function $f(\cdot)$ denotes the perception probability function.

Thus, the distribution is the product of the perception probabilities and the underlying geometric distribution of meteor magnitudes. Therefore, the parameter p of the geometric distribution is given by p = 1 - 1/r.

The parameter lm specifies the limiting magnitude for the meteor magnitude m. m must be an integer meteor magnitude. The length of the vector lm must either equal the length of the vector m, or lm must be a scalar value. In the case of rvmgeom, the length of the vector lm must equal n, or lm must be a scalar value.

If a different perception probability function perception.fun is provided, it must have the signature function(x) and return the perception probability of the difference x between the limiting magnitude and the meteor magnitude. If x >= 15.0, the function perception.fun should return a perception probability of 1.0. The argument perception.fun is resolved using match.fun.

Value

- dvmgeom: density
- pvmgeom: distribution function
- qvmgeom: quantile function
- rvmgeom: random generation

The length of the result is determined by n for rvmgeom, and by the maximum of the lengths of the numeric vector arguments for the other functions. All arguments are vectorized; standard R recycling rules apply.

Since the distribution is discrete, qvmgeom and rvmgeom always return integer values. qvmgeom may return NaN with a warning.

See Also

vmperception stats::Geometric

Examples

```
N <- 100
r <- 2.0
limmag <- 6.5
(m <- seq(6, -7))

# discrete density of `N` meteor magnitudes
(freq <- round(N * dvmgeom(m, limmag, r)))

# log likelihood function
lld <- function(r) {
    -sum(freq * dvmgeom(m, limmag, r, log=TRUE))
}

# maximum likelihood estimation (MLE) of r
est <- optim(2, lld, method='Brent', lower=1.1, upper=4)

# estimations
est$par # mean of r

# generate random meteor magnitudes
m <- rvmgeom(N, r, lm=limmag)

# log likelihood function
llr <- function(r) {
    -sum(dvmgeom(m, limmag, r, log=TRUE))
}

# maximum likelihood estimation (MLE) of r
est <- optim(2, llr, method='Brent', lower=1.1, upper=4, hessian=TRUE)

# estimations
est$par # mean of r
sqrt(1/est$hessian[1][1]) # standard deviation of r

m <- seq(6, -4, -1)
p <- vismeteor::dvmgeom(m, limmag, r)
barplot(
    p,
    names.arg = m,
    main = paste0('Density (r = ', r, ', limmag = ', limmag, ')'),
    col = "blue",
    xlab = 'm',
    ylab = 'p',
    border = "blue",
    space = 0.5
)
axis(side = 2, at = pretty(p))
```

vmgeomVst *Variance-Stabilizing Transformation for Geometric Visual Meteor Magnitudes*

Description

Applies a variance-stabilizing transformation to visual meteor magnitudes under the geometric model.

Usage

```
vmgeomVstFromMagn(m, lm)

vmgeomVstToR(tm, log = FALSE, deriv.degree = 0L)
```

Arguments

`m`	integer; meteor magnitude.
`lm`	numeric; limiting magnitude.
`tm`	numeric; transformed magnitude.
`log`	logical; if `TRUE`, the logarithm of the population index `r` is returned.
`deriv.degree`	integer; the order of the derivative at `tm` to return instead of `r` or `log(r)`. Must be `0`, `1`, or `2`.

Details

Many linear models require the variance of visual meteor magnitudes to be homoscedastic. The function `vmgeomVstFromMagn` applies a transformation that produces homoscedastic distributions of visual meteor magnitudes if the underlying distribution follows a geometric model.

The geometric model of visual meteor magnitudes depends on the population index `r` and the limiting magnitude `lm`, resulting in a two-parameter distribution. Without detection probabilities, the magnitude distribution is purely geometric, and for integer limiting magnitudes the variance depends only on the population index `r`. Since the limiting magnitude `lm` is a fixed parameter and never estimated statistically, the magnitudes can be transformed such that, for example, the mean of the transformed magnitudes directly provides an estimate of `r` using the function `vmgeomVstToR`.

A key advantage of this transformation is that the limiting magnitude `lm` is already incorporated into subsequent analyses. In this sense, the transformation acts as a normalization of meteor magnitudes and yields a variance close to `1.0`.

This transformation is valid for $1.4 \leq r \leq 3.5$. The numerical form of the transformation is version-specific and may change substantially in future releases. Do not rely on equality of transformed values across package versions.

Value

- `vmgeomVstFromMagn`: numeric value, the transformed meteor magnitude.
- `vmgeomVstToR`: numeric value of the population index `r`, derived from the mean of `tm`.

The argument `deriv.degree` can be used to apply the delta method. If `log = TRUE`, the logarithm of `r` is returned.

Note

The internal approximations used here are derived from the perception probabilities produced by vmperception. For details on the derivation, see the script `inst/derivation/vmgeom_vst.R` in the package's source code.

See Also

vmgeom vmperception

Examples

```
N <- 100
r <- 2.0
limmag <- 6.3

# Simulate magnitudes
m <- rvmgeom(N, limmag, r)

# Variance-stabilizing transformation
tm <- vmgeomVstFromMagn(m, limmag)
tm.mean <- mean(tm)
tm.var  <- var(tm)

# Estimator for r from the transformed mean
r.hat  <- vmgeomVstToR(tm.mean)

# Derivative dr/d(tm) at tm.mean (needed for the delta method)
dr_dtm <- vmgeomVstToR(tm.mean, deriv.degree = 1L)

# Variance of the sample mean of tm
var_tm.mean <- tm.var / N

# Delta method: variance and standard error of r.hat
var_r.hat <- (dr_dtm^2) * var_tm.mean
se_r.hat  <- sqrt(var_r.hat)

# Results
print(r.hat)
print(se_r.hat)
```

vmideal *Ideal Distribution of Visual Meteor Magnitudes*

Description

Density, distribution function, quantile function, and random generation for the ideal distribution of visual meteor magnitudes.

Usage

```
dvmideal(m, lm, psi, log = FALSE, perception.fun = NULL)

pvmideal(m, lm, psi, lower.tail = TRUE, log = FALSE, perception.fun = NULL)

qvmideal(p, lm, psi, lower.tail = TRUE, perception.fun = NULL)

rvmideal(n, lm, psi, perception.fun = NULL)

cvmideal(lm, psi, log = FALSE, perception.fun = NULL)
```

Arguments

`m`	integer; visual meteor magnitude.
`lm`	numeric; limiting magnitude.
`psi`	numeric; the location parameter of the probability distribution.
`log`	logical; if `TRUE`, probabilities are returned as `log(p)`.
`perception.fun`	function; optional perception probability function. The default is vmperception.
`lower.tail`	logical; if `TRUE` (default), probabilities are $P[M < m]$; otherwise, $P[M \geq m]$.
`p`	numeric; probability.
`n`	numeric; count of meteor magnitudes.

Details

The density of the ideal distribution of meteor magnitudes is

$$f(m) = \frac{\mathrm{d}p}{\mathrm{d}m} = \frac{3}{2} \log(r) \sqrt{\frac{r^{3\,\psi + 2\,m}}{(r^{\psi} + r^{m})^5}}$$

where m denotes the continuous (real-valued) meteor magnitude, $r = 10^{0.4} \approx 2.51189\ldots$ is a constant, and ψ is the only parameter of this magnitude distribution.

In visual meteor observations, magnitudes are usually estimated as integer values. Hence, this distribution is discrete and its probability mass function is given by

$$P[M = m] \sim \begin{cases} g(m_{\mathrm{lim}} - m) \int\limits_{m-0.5}^{m+0.5} f(u)\,\mathrm{d}u, & \text{if } m_{\mathrm{lim}} - m > -0.5, \\ 0 & \text{otherwise,} \end{cases}$$

where $m_{\lim}$ denotes the limiting (non-integer) magnitude of the observation, and m the integer meteor magnitude. The function $f(\cdot)$ is the continuous density of the ideal magnitude distribution, and $g(\cdot)$ denotes the perception probability function.

If a different perception probability function `perception.fun` is supplied, it must have the signature `function(x)` and return the perception probabilities of the difference `x` between the limiting magnitude and the meteor magnitude. If `x >= 15.0`, the `perception.fun` function should return a perception probability of `1.0`. The argument `perception.fun` is resolved using match.fun.

Value

- `dvmideal`: density
- `pvmideal`: distribution function
- `qvmideal`: quantile function
- `rvmideal`: random generation
- `cvmideal`: partial convolution of the ideal distribution of meteor magnitudes with the perception probabilities.

The length of the result is determined by `n` for `rvmideal`, and is the maximum of the lengths of the numeric vector arguments for the other functions. All arguments are vectorized; standard R recycling rules apply.

Since the distribution is discrete, `qvmideal` and `rvmideal` always return integer values. `qvmideal` may return `NaN` with a warning.

References

Richter, J. (2018) *About the mass and magnitude distributions of meteor showers*. WGN, Journal of the International Meteor Organization, vol. 46, no. 1, p. 34-38

See Also

mideal vmperception

Examples

```
N <- 100
psi <- 5.0
limmag <- 6.5
(m <- seq(6, -4))

# discrete density of `N` meteor magnitudes
(freq <- round(N * dvmideal(m, limmag, psi)))

# log likelihood function
lld <- function(psi) {
    -sum(freq * dvmideal(m, limmag, psi, log=TRUE))
}

# maximum likelihood estimation (MLE) of psi
est <- optim(2, lld, method='Brent', lower=0, upper=8, hessian=TRUE)
```

```
# estimations
est$par # mean of psi

# generate random meteor magnitudes
m <- rvmideal(N, limmag, psi)

# log likelihood function
llr <- function(psi) {
    -sum(dvmideal(m, limmag, psi, log=TRUE))
}

# maximum likelihood estimation (MLE) of psi
est <- optim(2, llr, method='Brent', lower=0, upper=8, hessian=TRUE)

# estimations
est$par # mean of psi
sqrt(1/est$hessian[1][1]) # standard deviation of psi

m <- seq(6, -4, -1)
p <- vismeteor::dvmideal(m, limmag, psi)
barplot(
    p,
    names.arg = m,
    main = paste0('Density (psi = ', psi, ', limmag = ', limmag, ')'),
    col = "blue",
    xlab = 'm',
    ylab = 'p',
    border = "blue",
    space = 0.5
)
axis(side = 2, at = pretty(p))

plot(
    \(lm) vismeteor::cvmideal(lm, psi, log = TRUE),
    -5, 10,
    main = paste0(
        'Partial convolution of the ideal meteor magnitude distribution\n',
        'with the perception probabilities (psi = ', psi, ')'
    ),
    col = "blue",
    xlab = 'lm',
    ylab = 'log(rate)'
)
```

`vmidealVst` *Variance-stabilizing Transformation for the Ideal Distribution of Visual Meteor Magnitudes*

Description

Applies a variance-stabilizing transformation to meteor magnitudes under the assumption of the ideal magnitude distribution.

Usage

```
vmidealVstFromMagn(m, lm)

vmidealVstToPsi(tm, lm, deriv.degree = 0L)
```

Arguments

`m`	integer; the meteor magnitude.
`lm`	numeric; limiting magnitude.
`tm`	numeric; transformed magnitude.
`deriv.degree`	integer; the degree of the derivative at `tm` to return instead of `psi`. Must be `0`, `1` or `2`.

Details

Many linear models require the variance of visual meteor magnitudes to be homoscedastic. The function `vmidealVstFromMagn` applies a transformation that produces homoscedastic distributions of visual meteor magnitudes if the underlying magnitudes follow the ideal magnitude distribution. In this sense, the transformation acts as a normalization of meteor magnitudes and yields a variance close to `1.0`.

The ideal distribution of visual meteor magnitudes depends on the parameter psi and the limiting magnitude `lm`, resulting in a two-parameter distribution. Without detection probabilities, the magnitude distribution reduces to a pure ideal magnitude distribution, which depends only on the parameter `psi`. Since the limiting magnitude `lm` is a fixed parameter and never estimated statistically, the magnitudes can be transformed such that, for example, the mean of the transformed magnitudes directly provides an estimate of `psi` using the function `vmidealVstToPsi`.

This transformation is valid for $-10 \leq$ `psi` ≤ 9. The numerical form of the transformation is version-specific and may change substantially in future releases. Do not rely on equality of transformed values across package versions.

Value

- `vmidealVstFromMagn`: a numeric value, the transformed meteor magnitude.
- `vmidealVstToPsi`: a numeric value of the parameter `psi`, derived from the mean of `tm`. The argument `deriv.degree` can be used to apply the delta method.

Note

The internal approximations used here are derived from the perception probabilities produced by vmperception. For details on the derivation, see the script `inst/derivation/vmideal_vst.R` in the package's source code.

See Also

vmideal mideal vmperception

Examples

```
N <- 100
psi <- 5.0
limmag <- 6.3

# Simulate magnitudes
m <- rvmideal(N, limmag, psi)

# Variance-stabilizing transformation
tm <- vmidealVstFromMagn(m, limmag)
tm.mean <- mean(tm)
tm.var  <- var(tm)

# Estimator for psi from the transformed mean
psi.hat  <- vmidealVstToPsi(tm.mean, limmag)

# Derivative d(psi)/d(tm) at tm.mean (needed for the delta method)
dpsi_dtm <- vmidealVstToPsi(tm.mean, limmag, deriv.degree = 1L)

# Variance of the sample mean of tm
var_tm.mean <- tm.var / N

# Delta method: variance and standard error of psi.hat
var_psi.hat <- (dpsi_dtm^2) * var_tm.mean
se_psi.hat  <- sqrt(var_psi.hat)

# Results
print(psi.hat)
print(se_psi.hat)
```

---

vmperception *Perception Probabilities of Visual Meteor Magnitudes*

Description

Provides the perception probability of visual meteor magnitudes.

Usage

```
vmperception(m)
```

Arguments

m	numeric; difference between the limiting magnitude and the meteor magnitude.

Details

The perception probabilities of *Koschack R., Rendtel J., 1990b* are estimated with the formula

$$p(m) = \begin{cases} 1.0 - \exp\left(-z(m + 0.5)\right) & \text{if } m > -0.5, \\ 0.0 & \text{otherwise,} \end{cases}$$

where

$$z(x) = 0.0037\, x + 0.0019\, x^2 + 0.00271\, x^3 + 0.0009\, x^4$$

and `m` is the difference between the limiting magnitude and the meteor magnitude.

Value

This function returns the visual perception probabilities.

References

Koschack R., Rendtel J., 1990b *Determination of spatial number density and mass index from visual meteor observations (II).* WGN 18, 119–140.

Examples

```
# Perception probability of visually estimated meteor of magnitude 3.0
# with a limiting magnitude of 5.6.
vmperception(5.6 - 3.0)

# plot
old_par <- par(mfrow = c(1,1))
plot(
    vmperception,
    -0.5, 8,
    main = paste(
        'perception probability of',
        'visual meteor magnitudes'
    ),
    col = "blue",
    xlab = 'm',
    ylab = 'p'
)

par(old_par)
```

`vmtable` *Rounds a contingency table of meteor magnitude frequencies*

Description

The meteor magnitude contingency table of VMDB contains half meteor counts (e.g. `3.5`). This function converts these frequencies to integer values.

Usage

```
vmtable(mt)
```

Arguments

`mt`	table; A two-dimensional contingency table of meteor magnitude frequencies.

Details

The contingency table of meteor magnitudes `mt` must be two-dimensional. The row names refer to the magnitude observations. Column names must be integer meteor magnitude values. Also, the columns must be sorted in ascending or descending order of meteor magnitude.

A sum-preserving algorithm is used for rounding. It ensures that the total frequency of meteors per observation is preserved. The marginal frequencies of the magnitudes are also preserved with the restriction that the deviation is at most ± 0.5. If the total sum of a meteor magnitude is integer, then the deviation is ± 0.

The algorithm is unbiased: for a fixed observation order it preserves the original totals without introducing systematic drift, even though each run follows the deterministic sequence dictated by the observed counts and their ordering.

Value

A rounded contingency table of meteor magnitudes is returned.

Note

Internally the counts are doubled to half-meteor units, leftover halves are alternated between rows so column margins stay within ± 0.5, and when the grand total is odd the matrix is temporarily mirrored so the unavoidable surplus meteor originates from the opposite end of the magnitude scale rather than always favouring the faintest bin. The mirroring is only the initial condition; the loop then processes the table cell by cell so the rounding direction alternates between bright and faint magnitudes depending on the current row and column state.

Examples

```
# For example, create a contingency table of meteor magnitudes
mt <- as.table(matrix(
    c(
        0.0, 0.0, 2.5, 0.5, 0.0, 1.0,
        0.0, 1.5, 2.0, 0.5, 0.0, 0.0,
        1.0, 0.0, 0.0, 3.0, 2.5, 0.5
    ), nrow = 3, ncol = 6, byrow = TRUE
))
colnames(mt) <- seq(6)
rownames(mt) <- c('A', 'B', 'C')
mt
margin.table(mt, 1)
margin.table(mt, 2)

# contingency table with integer values
```

```
(mt.int <- vmtable(mt))
margin.table(mt.int, 1)
margin.table(mt.int, 2)
```

Index