Package ‘vismeteor’

January 16, 2024

Type Package

Title Analysis of Visual Meteor Data

Version 1.8.5

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/>.

License MIT + file LICENSE

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

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

Encoding UTF-8

LazyData true

Depends R (>= 3.5)

Imports methods, stats, DBI

RoxygenNote 7.3.0

Suggests usethis (>= 2.1.1), testthat (>= 3.0.0), RSQLite (>= 2.2.1), RPostgreSQL (>= 0.7), RMySQL (>= 0.10.1), knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2024-01-16 14:10:02 UTC

R topics documented:

`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 from the data base*

Description

Loads the data of visual meteor observations from a data base created with imo-vmdb.

Usage

```
load_vmdb_rates(
  dbcon,
  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(
  dbcon,
  shower = NULL,
  period = NULL,
```

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

Arguments

dbcon	database connection.
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.
moon.alt.max	numeric; selects the maximum altitude of the moon.
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.

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

rate.id	unique identifier of the rate observation,
shower.code	IAU code of the shower. It is NA in case of sporadic meteors.
period.start	start of observation,
period.end	end of observation,
sl.start	solarlong at start of observation,
sl.end	solarlong at start of observation,
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). The zenith attraction is already applied.
radiant.az	azimuth of the radiant (optional),
magn.id	reference to the magnitude observations (optional).

load_vmdb_magnitudes returns a observations data frame, with

magn.id	unique identifier of the magnitude observation,
shower.code	IAU code of the shower. It is NA in case of sporadic meteors.
period.start	start of observation,
period.end	end of observation,
sl.start	solarlong at start of observation,
sl.end	solarlong at start of observation,
session.id	reference to the session,
freq	count of observed meteors,
magn.mean	mean of magnitudes,
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. The row names refer to the id of magnitude observations. The column names refer to the magnitude.

Note

Angle values are expected and returned in degrees.

References

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

Examples

```
## Not run:
# create a connection to the data base
con <- dbConnect(
    PostgreSQL(),
    dbname = "vmdb",
    host = "localhost",
    user = "vmdb"
)

# load rate observations including
# session data and magnitude observations
data <- load_vmdb_rates(
    con,
    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 including
# session data and magnitude observations
data <- load_vmdb_magnitudes(
    con,
    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
)

## End(Not run)
```

---

mideal *Ideal distributed meteor magnitudes*

Description

Density, distribution function, quantile function and random generation of ideal distributed 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 an ideal magnitude distribution 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 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(
    function(m) dmideal(m, psi, log = FALSE),
    -5, 10,
    main = paste0('density of ideal meteor magnitude\ndistribution (psi = ', psi, ')'),
    col = "blue",
    xlab = 'm',
    ylab = 'dp/dm'
)
abline(v=psi, col="red")

plot(
    function(m) dmideal(m, psi, log = TRUE),
    -5, 10,
    main = paste0('density of ideal meteor magnitude\ndistribution (psi = ', psi, ')'),
    col = "blue",
    xlab = 'm',
    ylab = 'log( dp/dm )'
)
abline(v=psi, col="red")

plot(
    function(m) pmideal(m, psi),
    -5, 10,
   main = paste0('probability of ideal meteor magnitude\ndistribution (psi = ', psi, ')'),
    col = "blue",
    xlab = 'm',
    ylab = 'p'
)
abline(v=psi, col="red")

plot(
    function(p) qmideal(p, psi),
    0.01, 0.99,
    main = paste('quantile of ideal meteor magnitude\n distribution (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.

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 and magnitude observations of the Perseid shower from 2015.

Details

`PER_2015_rates` are rate observations loaded with load_vmdb_rates.

See Also

load_vmdb

vmgeom *Visual magnitude distribution of geometric distributed meteor magnitudes*

Description

Density, distribution function, quantile function and random generation for the visual magnitude distribution of geometric distributed 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	numeric; the meteor magnitude.
lm	numeric; limiting magnitude.
r	numeric; the population index. It is the only parameter of the distribution.
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 observation, it is common to estimate meteor magnitudes in integer values. Hence, this distribution is discrete and has the density

$$P[X = x] \sim f(x)\,\mathrm{r}^{-x}\,,$$

where $x \geq -0.5$ is the difference between the limiting magnitude lm and the meteor magnitude m and $f(x)$ is the perception probability function. This distribution is thus a product of the perception probabilities and the actual geometric distribution of the meteor magnitudes. Therefore, the parameter p of the geometric distribution is p = 1 - 1/r.

The parameter lm indicate what the parameter m refers to. m must be an integer meteor magnitude. The length of the vector lm must then be equal to the length of the vector m or lm is a scalar value. In case of rvmgeom, the length of the vector lm must be n or lm is a scalar value.

If the perception probabilities function `perception.fun` is given, it must have the signature `function(x)` and must 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 the perception probability of `1.0`. If `log = TRUE` is given, the logarithm value of the perception probabilities must be returned. `perception.fun` is resolved using match.fun.

Value

`dvmgeom` gives the density, `pvmgeom` gives the distribution function, `qvmgeom` gives the quantile function, and `rvmgeom` generates random deviates.

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

Since the distribution is discrete, `qvmgeom` and `rvmgeom` always return integer values. `qvmgeom` can return `NaN` value 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))
```

`vmideal` *Visual magnitude distribution of ideal distributed meteor magnitudes*

Description

Density, distribution function, quantile function and random generation for the visual magnitude distribution of ideal distributed 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 a probability distribution. It is the only parameter of the distribution.
`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

The density of an ideal magnitude distribution 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 is the 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 observation, it is common to estimate meteor magnitudes in integer values. Hence, this distribution is discrete and has the density

$$P[M = m] \sim g(m) \int_{m-0.5}^{m+0.5} f(m) \, \mathrm{d}m \,,$$

where $g(m)$ is the perception probability. This distribution is thus a product of the perception probabilities and the actual ideal distribution of the meteor magnitudes.

If the perception probabilities function `perception.fun` is given, it must have the signature `function(M)` and must return the perception probabilities of the difference `M` between the limiting magnitude and the meteor magnitude. If `m >= 15.0`, the `perception.fun` function should return the perception probability of `1.0`. If `log = TRUE` is given, the logarithm value of the perception probabilities must be returned. `perception.fun` is resolved using match.fun.

Value

`dvmideal` gives the density, `pvmideal` gives the distribution function, `qvmideal` gives the quantile function, and `rvmideal` generates random deviates. `cvmideal` gives the partial convolution of the ideal meteor magnitude distribution with the perception probabilities.

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

Since the distribution is discrete, `qvmideal` and `rvmideal` always return integer values. `qvmideal` 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

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(
    function(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)'
```

```
)
```

vmperception *Perception Probabilities of Visual Meteor Magnitudes*

Description

Provides the perception probability of visual meteor magnitudes and its first derivative.

Usage

```
vmperception(m, deriv.degree = 0L)
```

Arguments

`m`	numerical; difference between the limiting magnitude and the meteor magnitude.
`deriv.degree`	integer; degree of derivative of the perception probability. Currently, valid values of `deriv.degree` are `0`, `1` and `2`.

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

$$z(x) = 0.003\,x + 0.0056\,x^2 + 0.0014\,x^4$$

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

Value

This function returns the visual perception probabilities. If `deriv.degree` is specified, it will return the `deriv.degree`-th order derivative of the perception probability.

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'
)
plot(
    function(m) {
        vmperception(m, deriv.degree=1L)/vmperception(m)
    },
    -0.3, 8,
    main = paste(
        'q-values of',
        'visual meteor magnitudes'
    ),
    col = "blue",
    log = 'y',
    xlab = 'm',
    ylab = 'q'
)

par(old_par)
```

`vmperception.l` *Laplace-Transformed Perception Probabilities of Visual Meteor Magnitudes*

Description

Provides the Laplace-transformed perception probability of visual meteor magnitudes and its first derivative.

Usage

```
vmperception.l(s, deriv.degree = 0L)
```

Arguments

s	numerical; Real (non-complex) parameter for the Laplace transformation.

deriv.degree	integer; degree of derivative of the transformation. Currently, valid values of `deriv.degree` are `0`, `1` and `2`.

Details

The Laplace-transformed perception probabilities `F(s)`, given as

$$F(s) = \mathcal{L}\left\{p\right\}(s) = \int_{-0.5}^{\infty} f(m)\,\mathrm{e}^{-s\,m}\,\mathrm{d}m\,,$$

are approximately

$$P(s) = \begin{cases} s^{-1}\,\exp\left(-4.11\,s + 1.32\,s^2 - 0.15\,s^3\right) & \text{if } s >= 0.0,\\ \text{undefined} & \text{otherwise.}\end{cases}$$

Here, `m` is the difference between the limiting magnitude and the meteor magnitude, and `f(m)` denotes the perception probabilities as a function of `m`. The $\mathcal{L}$ recalls here the one-sided Laplace transform.

The Laplace transform is notably effective for determining the mean and variance of observed meteor magnitudes, which are measured relative to the limiting magnitude. This is just one example of its application. This approach is valid only when the actual magnitude distribution adheres to $p(m) \sim r^{-m}$, where $s = \log(r)$. In this scenario, the mean of the observable meteor magnitudes is given by $-\mathcal{L}'/\mathcal{L}$, and their variance is calculated as $\mathcal{L}''/\mathcal{L} - (\mathcal{L}'/\mathcal{L})^2$.

Value

returns the Laplace-transformed perception probabilities. If `deriv.degree` is specified, it will return the `deriv.degree`-th order derivative of these Laplace-transformed values.

See Also

vmperception vmgeom

Examples

```
r <- 2.0
s <- log(r)
F0 <- vmperception.l(s)
F1 <- vmperception.l(s, deriv.degree=1L)
# magnitude mean
-F1/F0
F2 <- vmperception.l(s, deriv.degree=2L)
# magnitude variance
F2/F0 - (F1/F0)^2
# plot the Laplace-transformed perception probabilities
old_par <- par(mfrow = c(1,1))
plot(
    vmperception.l,
```

```
    0.2, 1.1,
    main = paste(
        'Laplace-transformed perception',
        'probability of visual meteor magnitudes'
    ),
    col = "blue",
    log = 'y',
    xlab = 's',
    ylab = 'L'
)
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 asymptotic. This means that the more meteors the table contains, the more unbiased is the result of the rounding.

Value

A rounded contingency table of meteor magnitudes is returned.

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