Package 'suncalc'

October 26, 2026

Version 0.5.4

Title Compute Sun Position, Sunlight Phases, Moon Position and Lunar Phase

Description Get sun position, sunlight phases (times for sunrise, sunset, dusk, etc.), moon position and lunar phase for the given location and time. Most calculations are based on the formulas given in Astronomy Answers articles about position of the sun and the planets : <https://www.aa.quae.nl/en/reken/zonpositie.html>.

Imports data.table, lubridate (>= 1.9.0), magrittr

Suggests testthat (>= 3.0.0)

License GPL-2 | file LICENSE

URL https://github.com/datastorm-open/suncalc

NeedsCompilation no

Encoding UTF-8

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Author Benoit Thieurmel [aut, cre] (R interface), Achraf Elmarhraoui [aut] (R interface)

Maintainer Benoit Thieurmel <bthieurmel@gmail.com>

Repository CRAN

Date/Publication 2026-10-26 15:00:02 UTC

Contents

getMoonIllumination 2
getMoonPosition 3
getMoonTimes 5
getSunlightPosition 6
getSunlightTimes 8

Index **10**

`getMoonIllumination` *Get Moon illumination*

Description

Get Moon illumination

Usage

```
getMoonIllumination(date = Sys.Date(), keep = c("fraction", "phase", "angle"))
```

Arguments

`date`	Single or multiple DateTime. Can be a `Date` (YYYY-MM-DD), a `character` in UTC (YYYY-MM-DD HH:mm:ss) or a `POSIXct`
`keep`	`character`. Vector of variables to keep. See `Details`

Details

Computes the following properties:

"fraction" illuminated fraction of the moon; varies from 0.0 (new moon) to 1.0 (full moon)

"phase" moon phase; varies from 0.0 to 1.0, described below

"angle" midpoint angle in radians of the illuminated limb of the moon reckoned eastward from the north point of the disk; the moon is waxing if the angle is negative, and waning if positive

Moon phase value should be interpreted like this:

0	New Moon
	Waxing Crescent
0.25	First Quarter
	Waxing Gibbous
0.5	Full Moon
	Waning Gibbous
0.75	Last Quarter
	Waning Crescent
1	New Moon

By subtracting the parallacticAngle from the angle one can get the zenith angle of the moons bright limb (anticlockwise). The zenith angle can be used do draw the moon shape from the observers perspective (e.g. moon lying on its back).

Value

`data.frame`

See Also

Other suncalc functions: `getMoonPosition()`, `getMoonTimes()`, `getSunlightPosition()`, `getSunlightTimes()`

Examples

```
# one date
getMoonIllumination(date = Sys.Date())

# in character
getMoonIllumination(date = c("2017-05-12", "2017-05-12 00:00:00"),
  keep = c("fraction", "phase"))

# in POSIXct
getMoonIllumination(date = as.POSIXct("2017-05-12 00:00:00", tz = "UTC"))
getMoonIllumination(date = as.POSIXct("2017-05-12 02:00:00", tz = "Europe/Paris"))

date <- seq(ISOdate(2009,1,1), ISOdate(2010,1,1), "hours")
date_cet <- date
attr(date_cet, "tzone") <- "Europe/Paris"
res <- getMoonIllumination(date = date_cet)
```

---

getMoonPosition *Get Moon position*

Description

Get Moon position

Usage

```
getMoonPosition(
  date = NULL,
  lat = NULL,
  lon = NULL,
  data = NULL,
  keep = c("altitude", "azimuth", "distance", "parallacticAngle")
)
```

Arguments

`date`	Single or multiple DateTime. Can be a `Date` (YYYY-MM-DD), a `character` in UTC (YYYY-MM-DD HH:mm:ss) or a `POSIXct`
`lat`	`numeric`. Single latitude
`lon`	`numeric`. Single longitude
`data`	`data.frame`. Alternative to use `date`, `lat`, `lon` for passing multiple coordinates
`keep`	`character`. Vector of variables to keep. See `Details`

Details

Returns an object with the following properties:

"altitude" moon altitude above the horizon in radians

"azimuth" moon azimuth in radians

"distance" distance to moon in kilometers

"parallacticAngle" parallactic angle of the moon in radians

Value

`data.frame`

See Also

Other suncalc functions: `getMoonIllumination()`, `getMoonTimes()`, `getSunlightPosition()`, `getSunlightTimes()`

Examples

```
# one date
getMoonPosition(date = Sys.Date(), lat = 50.1, lon = 1.83)

# in character
getMoonPosition(date = c("2017-05-12", "2017-05-12 00:00:00"),
    lat = 50.1, lon = 1.83)

# in POSIXct
getMoonPosition(date = as.POSIXct("2017-05-12 00:00:00", tz = "UTC"),
    lat = 50.1, lon = 1.83)
getMoonPosition(date = as.POSIXct("2017-05-12 02:00:00", tz = "Europe/Paris"),
    lat = 50.1, lon = 1.83)

# multiple date + subset
getMoonPosition(date = seq.Date(Sys.Date()-9, Sys.Date(), by = 1),
  keep = c("altitude", "azimuth"),
  lat = 50.1, lon = 1.83)

# multiple coordinates
data <- data.frame(date = seq.Date(Sys.Date()-9, Sys.Date(), by = 1),
    lat = c(rep(50.1, 10), rep(49, 10)),
    lon = c(rep(1.83, 10), rep(2, 10)))

getMoonPosition(data = data,
  keep = c("altitude", "azimuth"))
```

`getMoonTimes` *Get Moon times*

Description

Get Moon times

Usage

```
getMoonTimes(
  date = NULL,
  lat = NULL,
  lon = NULL,
  data = NULL,
  keep = c("rise", "set", "alwaysUp", "alwaysDown"),
  tz = "UTC",
  ...
)
```

Arguments

`date`	`Date`. Single or multiple Date. YYYY-MM-DD
`lat`	`numeric`. Single latitude
`lon`	`numeric`. Single longitude
`data`	`data.frame`. Alternative to use `date`, `lat`, `lon` for passing multiple coordinates
`keep`	`character`. Vector of variables to keep. See `Details`
`tz`	`character`. Timezone of results
`...`	Not used. (Maintenance only)

Details

Available variables are :

"rise" `Date`. moonrise time

"set" `Date`. moonset time

"alwaysUp" `Logical`. TRUE if the moon never rises or sets and is always above the horizon during the day

"alwaysDown" `Logical`. TRUE if the moon is always below the horizon

Times are given for the requested day in `tz` (from 00:00 to 24:00). `rise` / `set` are `NA` when the moon doesn't rise / set this day (about once a month, as moonrise and moonset are about 50 minutes later each day).

Value

`data.frame`

See Also

Other suncalc functions: `getMoonIllumination()`, `getMoonPosition()`, `getSunlightPosition()`, `getSunlightTimes()`

Examples

```
# one date
getMoonTimes(date = Sys.Date(), lat = 47.21, lon = -1.557, tz = "Europe/Paris")

# multiple date + subset
getMoonTimes(date = seq.Date(Sys.Date()-9, Sys.Date(), by = 1),
  keep = c("rise", "set", "alwaysUp"),
  lat = 47.21, lon = -1.557, tz = "Europe/Paris")

# multiple coordinates
data <- data.frame(date = seq.Date(Sys.Date()-9, Sys.Date(), by = 1),
    lat = c(rep(50.1, 10), rep(49, 10)),
    lon = c(rep(1.83, 10), rep(2, 10)))

getMoonTimes(data = data, tz = "Europe/Paris")
```

---

getSunlightPosition *Get Sunlight position*

Description

Get Sunlight position

Usage

```
getSunlightPosition(
  date = NULL,
  lat = NULL,
  lon = NULL,
  data = NULL,
  keep = c("altitude", "azimuth")
)
```

Arguments

`date`	Single or multiple DateTime. Can be a `Date` (YYYY-MM-DD), a `character` in UTC (YYYY-MM-DD HH:mm:ss) or a `POSIXct`
`lat`	`numeric`. Single latitude
`lon`	`numeric`. Single longitude
`data`	`data.frame`. Alternative to use `date`, `lat`, `lon` for passing multiple coordinates
`keep`	`character`. Vector of variables to keep. See `Details`

Details

Returns an object with the following properties:

"altitude" sun altitude above the horizon in radians, e.g. 0 at the horizon and PI/2 at the zenith (straight over your head)

"azimuth" sun azimuth in radians (direction along the horizon, measured from south to west), e.g. 0 is south and Math.PI * 3/4 is northwest

Value

`data.frame`

See Also

Other suncalc functions: `getMoonIllumination()`, `getMoonPosition()`, `getMoonTimes()`, `getSunlightTimes()`

Examples

```
# one date
getSunlightPosition(date = Sys.Date(), lat = 50.1, lon = 1.83)

# in character
getSunlightPosition(date = c("2017-05-12", "2017-05-12 00:00:00"),
    lat = 50.1, lon = 1.83)

# in POSIXct
getSunlightPosition(date = as.POSIXct("2017-05-12 00:00:00", tz = "UTC"),
    lat = 50.1, lon = 1.83)
getSunlightPosition(date = as.POSIXct("2017-05-12 02:00:00", tz = "Europe/Paris"),
    lat = 50.1, lon = 1.83)

# multiple date + subset
getSunlightPosition(date = seq.Date(Sys.Date()-9, Sys.Date(), by = 1),
  keep = c("altitude"),
  lat = 50.1, lon = 1.83)

# multiple coordinates
data <- data.frame(date = seq.Date(Sys.Date()-9, Sys.Date(), by = 1),
    lat = c(rep(50.1, 10), rep(49, 10)),
    lon = c(rep(1.83, 10), rep(2, 10)))

getSunlightPosition(data = data,
  keep = c("altitude", "azimuth"))
```

`getSunlightTimes` *Get Sunlight times*

Description

Get Sunlight times

Usage

```
getSunlightTimes(
  date = NULL,
  lat = NULL,
  lon = NULL,
  data = NULL,
 keep = c("solarNoon", "nadir", "sunrise", "sunset", "sunriseEnd", "sunsetStart",
  "dawn", "dusk", "nauticalDawn", "nauticalDusk", "nightEnd", "night", "goldenHourEnd",
    "goldenHour"),
  tz = "UTC"
)
```

Arguments

`date`	`Date`. Single or multiple Date. YYYY-MM-DD
`lat`	`numeric`. Single latitude
`lon`	`numeric`. Single longitude
`data`	`data.frame`. Alternative to use `date`, `lat`, `lon` for passing multiple coordinates
`keep`	`character`. Vector of variables to keep. See `Details`
`tz`	`character`. Timezone of results

Details

Available variables are :

"sunrise" sunrise (top edge of the sun appears on the horizon)

"sunriseEnd" sunrise ends (bottom edge of the sun touches the horizon)

"goldenHourEnd" morning golden hour (soft light, best time for photography) ends

"solarNoon" solar noon (sun is in the highest position)

"goldenHour" evening golden hour starts

"sunsetStart" sunset starts (bottom edge of the sun touches the horizon)

"sunset" sunset (sun disappears below the horizon, evening civil twilight starts)

"dusk" dusk (evening nautical twilight starts)

"nauticalDusk" nautical dusk (evening astronomical twilight starts)

"night" night starts (dark enough for astronomical observations)

"nadir" nadir (darkest moment of the night, sun is in the lowest position)

"nightEnd" night ends (morning astronomical twilight starts)

"nauticalDawn" nautical dawn (morning nautical twilight starts)

"dawn" dawn (morning nautical twilight ends, morning civil twilight starts)

Times are given for the requested day in `tz` (from 00:00 to 24:00). `NA` is returned when the event does not occur this day :

- the sun doesn't reach the corresponding altitude, e.g. no `night` / `nightEnd` (sun never 18 degrees below the horizon) during summer at high latitudes, no `sunrise` / `sunset` during polar day or polar night
- the event happens just before midnight the previous day and just after midnight the next day

Value

`data.frame`

See Also

Other suncalc functions: `getMoonIllumination()`, `getMoonPosition()`, `getMoonTimes()`, `getSunlightPosition()`

Examples

```
# one date
getSunlightTimes(date = Sys.Date(), lat = 50.1, lon = 1.83, tz = "Europe/Paris")

# multiple date + subset
getSunlightTimes(date = seq.Date(Sys.Date()-9, Sys.Date(), by = 1),
  keep = c("sunrise", "sunriseEnd", "sunset", "sunsetStart"),
  lat = 50.1, lon = 1.83, tz = "Europe/Paris")

# multiple coordinates
data <- data.frame(date = seq.Date(Sys.Date()-9, Sys.Date(), by = 1),
    lat = c(rep(50.1, 10), rep(49, 10)),
    lon = c(rep(1.83, 10), rep(2, 10)))

getSunlightTimes(data = data,
  keep = c("sunrise", "sunriseEnd", "sunset", "sunsetStart"), tz = "Europe/Paris")
```

Index

∗ **suncalc functions**
getMoonIllumination, 2
getMoonPosition, 3
getMoonTimes, 5
getSunlightPosition, 6
getSunlightTimes, 8

getMoonIllumination, 2
getMoonIllumination(), *4*, *6, 7*, *9*
getMoonPosition, 3
getMoonPosition(), *3*, *6, 7*, *9*
getMoonTimes, 5
getMoonTimes(), *3, 4*, *7*, *9*
getSunlightPosition, 6
getSunlightPosition(), *3, 4*, *6*, *9*
getSunlightTimes, 8
getSunlightTimes(), *3, 4*, *6, 7*