Package ‘NMEA’

September 14, 2018

Title NMEA parsing

Version 1.0.1

Description This package enables the processing of NMEA compliant log files to data.tables.

Depends R (>= 3.3.3),
data.table,
stringr

License Undecided

Encoding UTF-8

LazyData true

Date 2018-09-14

RoxygenNote 6.0.1

R topics documented:

`NMEA-package` *NMEA*

Description

The NMEA package provides functions for processing (reading) NMEA compliant GPS log files. It’s functions are used to convert one or more than one log file to a data.table containing all the NMEA information.

Compatibility

The function only reads the following NMEA lines: GNRMC, GNVTG, GNGGA, GNGSA, GPGSV, and GLGSV. Any other lines are (currently) omitted. The returned data.table is filtered for double information, as certain information is present in multiple lines (such as for example the time of fix in both GNRMC and GNGGA). This information will only be returned once.

Conversion

As the NMEA lines are textual, the returned columns in the data.table are converted from the original `character` type. Numbers will be converted to `numeric`. With many NMEA information containing numbers, many of the columns are also numeric.

NMEA column information

The columns of the data.table contain the following information. A lot of the NMEA information can also be found on http://www.gpsinformation.org/dale/nmea.htm.

DATETIME The date and time of the NMEA information in `POSIXct` format, in the GMT timezone. Transformed based on the `FIXTIME` and `FIXDATE` information.

FIXTIME The raw time of the GNRMC line.

FIXDATE The raw date of the GNRMC line.

FIX Status information, can either be A (active) or V (void). From GNRMC line.

FIX3D The 3D fix information, where 1 means no fix, 2 means 2D fix, and 3 means 3D fix. From GNGSA lines.

FIXQUALITY Quality of fix information, ranging from 0 to 8. For general usage, 0 means no fix, 1 means GPS fix, 6 means dead reckoning. More information to be found on the information link. From GNGGA line.

FIXSAT Amount of satellites being used for determining the position (fix). This field is calculated by counting the amount of satellites used for determining the position, as recorded in the GNGSA lines.

TRACKEDSAT Amount of satellites being tracked. Note that this number can be different from `FIXSAT`, as for certain receivers it only counts the amount of GPS satellites (and not those from other constellations). Using `FIXSAT` is more reliable. From GNGGA line.

VIEWSAT Amount of satellites in view. This field is calculated by counting the amount of satellites in view, as recorded in the GxGSV lines.

VIEWSAT_GP Amount of satellites in view, for GPS. This field is calculated by counting the amount of satellites in view, as recorded in the GPGSV lines.

VIEWSAT_GL Amount of satellites in view, for GLONASS. This field is calculated by counting the amount of satellites in view, as recorded in the GLGSV lines.

LATITUDE The latitude position, in decimal format. The original NMEA format is in degrees and multiplied by 100, but the division and conversion is already performed (for example, the NMEA line shows 4807.038, which means latitude 48 deg 07.038', the returned information shows latitude 48.1173 as this is the decimal verison).

LATITUDE_NS

LONGITUDE

LONGITUDE_NS

ALTITUDE

SPEED_KTS

SPEED_KMH

TRACKANGLE

TRUETRACK

MAGNETICTRACK

HEIGHTGEOID

PDOP

HDOP

VDOP

SATFIX_x

SATPRN_x

SATAZI_x

SATELV_x

SATSNR_x

Author(s)

Maintainer: Joost Kuckartz <joost.kuckartz@arcadis.com> [copyright holder]

---

nmea_calculateaveragesnr

Calculate average SNR

Description

Calculate the average SNR for all or a certain amount of satellites.

Usage

```
nmea_calculateaveragesnr(gpstable, sortedsatellites = NULL)
```

Arguments

`gpstable`	The data.table containing the GPS details as returned by `nmea_readfiles` or `nmea_parsetext`.
`sortedsatellites`	If an integer value is provided, it will sort the satellites it uses for determining the position by SNR descending, and calculate the average SNR of the top provided amount.

Value

Vector with average SNR. This vector can be appended as a new column using `cbind` to the provided `gpstable`.

`nmea_getpossibletypes` *Get possible types*

Description

Get a string vector of the possible NMEA line types that can be processed.

Usage

```
nmea_getpossibletypes()
```

Value

The string vector of the possible NMEA line types that can be processed.

`nmea_parsetext` *Parse NMEA text*

Description

Parse a vector of strings containing NMEA lines. Each vector element is a new NMEA line.

Usage

```
nmea_parsetext(nmeatextlines)
```

Arguments

`nmeatextlines`	The vector of strings containing NMEA lines.

Value

Data.table with all related NMEA information, with each row containing the GPS details for the single timestamp.

nmea_parsetext_multifile *Parse NMEA text*

Description

Parse a vector of strings containing NMEA lines. Each vector element is a new NMEA line. This function is used in loops where more than one file needs to be appended.

Usage

```
nmea_parsetext_multifile(nmeatextlines, previousfirstlines = NULL)
```

Arguments

nmeatextlines	The vector of strings containing NMEA lines.
previouslines	Data.table rows returned from a previous call to this function, to prepend to the current call. Use the `$firstlines` named item. See the example for how this works.

Value

Named list with the GPS data.table (`logtable`) and a data.table with the last rows of the data.table that should be prepended to a next one (`firstlines`), or `NULL` if the log did not contain a full set at all.

Examples

```
## Not run:
prependlines = NULL
gpstable = NULL
for (file in filelist) {
  nmealines = readLines(file)
  info = nmea_parsetext_multifile(file,prependlines)
  prependlines = info$firstlines
  gpstable = rbind(gpstable,info$logtable)
}

## End(Not run)
```

nmea_readfiles *Read NMEA files*

Description

Read NMEA compliant files. If more than one file path is provided, they are considered continuous and will be appended in the order provided.

Usage

```
nmea_readfiles(pathtofiles)
```

Arguments

`pathtofiles`	The full paths to the files.

Value

Data.table with all related NMEA information, with each row containing the GPS details for the single timestamp.

`nmea_tablestep1` *Execute step 1 in getting to a table.*

Description

Internal function, do not call directly. Step 1 transforms a vector string to a data.table, with prepending functionality, and returns a complete set that contains all necessary grouped GPS information, plus a set that can be prepended to the next call.

Usage

```
nmea_tablestep1(logtext, previouslines = NULL)
```

Arguments

`logtext`	Vector string with NMEA lines.
`previouslines`	Data.table rows (from an earlier executed step 1 on a previous vector string) to prepend to the lines of the current vector string.

Value

Named list with the first step NMEA data.table (`logtable`) and a data.table with the last rows of the data.table that should be prepended to a next one (last set beginning with GNRMC, `firstlines`), or `NULL` if the log did not contain a full set at all.

`nmea_tablestep2` *Execute step 2 in getting to a table.*

Description

Internal function, do not call directly. Step 2 transforms the data.table with `TYPE` and the other columns to the final resulting data.table.

Usage

```
nmea_tablestep2(logtable)
```

Arguments

`logtable`	The first step NMEA data.table as returned from `nmea_tablestep1`.

Value

The final data.table with GPS details.

Index