

Package ‘AISanalyze’

September 30, 2026

Title Processing and Analyzing AIS Vessel Tracking Data

Version 3.1.3

Description Processes Automatic Identification

System (AIS) vessel tracking data, including travel estimation, trajectory correction, interpolation, extraction, and summarising vessel information. The package is designed to facilitate reproducible analyses of maritime traffic in ecological, environmental, and marine spatial planning applications.

For more details see <<https://remip48.github.io/AISanalyze/>>.

License MIT + file LICENSE

Depends R(>= 4.2.0)

Imports dplyr, sf, purrr, stats, stringr, doParallel, parallel, foreach, assertthat, data.table, magrittr

Suggests knitr, units, lubridate, rmarkdown, testthat

URL <https://github.com/remip48/AISanalyze>,
<https://remip48.github.io/AISanalyze/>

BugReports <https://github.com/remip48/AISanalyze/issues>

Encoding UTF-8

Config/testthat/edition 3

LazyData true

Config/roxygen2/version 8.0.0

VignetteBuilder knitr

RoxygenNote 7.3.1

NeedsCompilation no

Author Rémi Pigeault [aut, cre] (ORCID:
<<https://orcid.org/0000-0003-2714-5122>>)

Maintainer Rémi Pigeault <remi.pigeault@tiho-hannover.de>

Repository CRAN

Date/Publication 2026-09-30 18:00:02 UTC

Contents

ais	2
AIScorrect_speed	3
AISextract	4
AISidentify_stations_aircraft	6
AISinfos	7
AISinterpolate	8
AIStravel	10
point_to_extract	11
Index	12

ais	<i>Example AIS dataset</i>
-----	----------------------------

Description

A subset of Automatic Identification System (AIS) messages collected in the North Sea on 1 November 2022. The dataset contains vessel positions together with static and dynamic AIS information, including vessel identity, navigational status, speed, heading, dimensions, draught and destination.

Format

A data frame with AIS messages for multiple vessels. The main variables include:

- datetime** Date and time of the AIS message (UTC).
- mmsi** Anonymized Maritime Mobile Service Identity (MMSI) of the vessel.
- lon** Longitude (decimal degrees, WGS84).
- lat** Latitude (decimal degrees, WGS84).
- navigational_status** Reported navigational status.
- SOG** Speed over ground (knots).
- Heading** True heading (degrees).
- imo** Reported International Maritime Organization (IMO) number.
- shiptype** Reported vessel type.
- length** Reported vessel length (m).
- name** Anonymized vessel name.
- width** Reported vessel width (m).
- draught** Reported vessel draught (m).

Details

The dataset is intended for demonstrating the main functions of **AISanalyze**, including travel distance estimation, GPS error correction, vessel characteristic estimation, interpolation and vessel extraction. Original MMSI identifiers and vessel names have been replaced with anonymized values to protect vessel confidentiality.

Source

Example subset extracted from AIS observations.

AIScorrect_speed	<i>Correct GPS errors in AIS tracks</i>
------------------	---

Description

Detects and corrects GPS errors and delayed AIS messages that generate unrealistic vessel speeds, travelled distances, and travel times. Note: For consecutive GPS errors, only the first point is removed to avoid overcorrection.

Usage

```
AIScorrect_speed(
  ais_data,
  crs_meters = 3035,
  threshold_speed_to_correct = 100,
  threshold_speed_to_correct_function = NULL,
  nb_cores = 1,
  outfile = tempfile()
)
```

Arguments

<code>ais_data</code>	AIS data frame containing timestamp, lon, lat, and mmsi. timestamp must be Unix time (seconds since 1970-01-01), while lon and lat must be numeric. Another vessel identifier may be used if the column is named mmsi.
<code>crs_meters</code>	CRS (metres) used to calculate distances in the study area (defaults to EPSG:3035, Europe). Tip: use <code>suggest_crs</code> function (<code>crsuggest</code> package) to find a suitable CRS for your study area.
<code>threshold_speed_to_correct</code>	Speed threshold (km/h) above which observations are corrected.
<code>threshold_speed_to_correct_function</code>	a function to estimate vessel-specific speed thresholds. The function can use all columns of <code>ais_data</code> as argument (such as <code>speed_kmh</code> , <code>distance_travelled</code> or <code>time_travelled</code>). If set to <code>NULL</code> , an internal function is used (see details).
<code>nb_cores</code>	Number of CPU cores used.
<code>outfile</code>	File used to save logs.

Details

When `threshold_speed_to_correct_function` is set to `NULL`, a vessel-specific speed threshold is estimated from the observed vessel `speed_kmh`. The threshold is calculated as:

$$T = 15 + \text{median}(v_{>1}) + 5 \times \text{SD}(v_{1 < v < Q_{0.75}})$$

where $v_{>1}$ represents vessel speeds greater than 1 km/h, and $v_{1 < v < Q_{0.75}}$ represents vessel speeds between 1 km/h and the 75th percentile. The equation is designed to capture the typical travelling speed of the vessel while accounting for variation in its observed travelling speeds.

Value

The input data with corrected travel metrics and the following columns:

- `speed_kmh_corrected`: Whether the speed was corrected.
- `time_travelled`: Travel time (s).
- `distance_travelled`: Travelled distance (m).
- `speed_kmh`: Vessel speed (km/h).

Examples

```
data("ais")

# use only a sample for the example:
ais <- ais[ais$mmsi %in% ais$mmsi[1:5], ]

# Define the Unix time (seconds since 1970-01-01)
ais$timestamp <- as.numeric(lubridate::ymd_hms(ais$datetime))

# calculate the travelled distance, time, and speed:
ais <- AIStravel(ais_data = ais, crs_meters = 3035)

# Correct speed:
out <- AIScorrect_speed(ais_data = ais,
                        crs_meters = 3035)
```

AISextract

Extract AIS positions around target locations and times

Description

Returns either (depending on `return_all_vessel_locations`):

- the position of each vessel at the time closest to the target timestamps.
- or all vessel positions within a specified time window.

Usage

```

AISextract(
  ais_data,
  data,
  crs_meters = 3035,
  return_all_vessel_locations = TRUE,
  search_into_radius_m = 50000,
  search_shape = "circle",
  interval_time_before = 5 * 60,
  interval_time_after = 5 * 60,
  nb_cores = 1,
  outfile = tempfile()
)

```

Arguments

<code>ais_data</code>	AIS data frame containing timestamp, lon, lat, and mmsi. timestamp, lon, and lat must be numeric. Another vessel identifier may be used if the column is named mmsi.
<code>data</code>	Data frame containing timestamp, lon, and lat. timestamp must be Unix time (seconds since 1970-01-01), while lon and lat must be numeric.
<code>crs_meters</code>	CRS (metres) used to calculate distances in the study area (defaults to EPSG:3035, Europe). Tip: use <code>suggest_crs</code> function (<code>crsuggest</code> package) to find a suitable CRS for your study area.
<code>return_all_vessel_locations</code>	Logical. If TRUE, returns all vessel positions within the specified time window. Otherwise, returns only the closest position in time.
<code>search_into_radius_m</code>	Search radius (m).
<code>search_shape</code>	"circle" (default; selects vessels within <code>search_into_radius_m</code> of the target location) or "square" (selects vessels within <code>search_into_radius_m</code> in both the X and Y directions, useful for grid-based analyses).
<code>interval_time_before</code>	Time window (s) before each <code>data\$timestamp</code> .
<code>interval_time_after</code>	Time window (s) after each <code>data\$timestamp</code> .
<code>nb_cores</code>	Number of CPU cores used.
<code>outfile</code>	File used to save logs.

Value

data joined with matching AIS positions. Rows are duplicated when several vessel positions match a target location and time. If no vessel is found, AIS columns (including `mmsi`) are filled with NA. The output also includes `distance_vessel_to_location_m`, the distance (m) between the target location and vessel positions.

Examples

```
data("ais")
data("point_to_extract")

# use only a sample for the example:
ais <- ais[20000:30000, ]

# Define the Unix time (seconds since 1970-01-01)
point_to_extract$timestamp <- as.numeric(lubridate::ymd_hm(point_to_extract$datetime))
ais$timestamp <- as.numeric(lubridate::ymd_hms(ais$datetime))

# calculate the travelled distance, time, speed, and interpolate AIS data:
ais <- AIStravel(ais, crs_meters = 3035)

# Extract all vessel positions within the target time interval and radius:
out <- AISextract(ais_data = ais,
  data = point_to_extract,
  crs_meters = 3035,
  return_all_vessel_locations = TRUE, # set FALSE to only
  # extract the vessel position closest in time to the
  # target timestamps.
  search_into_radius_m = 50000,
  interval_time_before = 5 * 60,
  interval_time_after = 5 * 60)
```

AISidentify_stations_aircraft

Identify AIS base stations and high-speed craft

Description

Stations and aircraft are identified from speed, distance and time only. Other criteria (e.g. MMSIs with fewer than 9 digits) are not considered.

Usage

```
AISidentify_stations_aircraft(ais_data, crs_meters = 3035)
```

Arguments

<code>ais_data</code>	AIS data frame containing timestamp, lon, lat, and mmsi. timestamp must be Unix time (seconds since 1970-01-01), while lon and lat must be numeric. Another vessel identifier may be used if the column is named mmsi.
<code>crs_meters</code>	CRS (metres) used to calculate distances in the study area (defaults to EPSG:3035, Europe). Tip: use <code>suggest_crs</code> function (<code>crsuggest</code> package) to find a suitable CRS for your study area.

Value

The input AIS data with the following additional columns:

- `station`: Whether the MMSI is classified as a base station.
- `high_speed`: Whether the MMSI is classified as a high-speed craft.
- `n_point_mmsi_initial_data`: Number of valid AIS positions for the MMSI.
- `id_mmsi_point_initial`: Sequential identifier of each AIS position.

Examples

```
data("ais")

# Define the Unix time (seconds since 1970-01-01)
ais$timestamp <- as.numeric(lubridate::ymd_hms(ais$datetime))

# calculate the travelled distance, time, and speed:
ais <- AIStravel(ais_data = ais, crs_meters = 3035)

# Identify stations and aircrafts:
out <- AISidentify_stations_aircraft(ais_data = ais, crs_meters = 3035)
```

AISinfos

Estimate vessel characteristics

Description

Estimates the most likely vessel characteristics for each `mmsi` from AIS messages, including ship type, length, width, draught, IMO number, and vessel name. Estimates are based on the most frequent values, giving greater weight to records with more complete information. Warnings are printed if any value of length, draught, width, or IMO cannot be converted to numeric (and is therefore set to NA) or any value of ship type or name cannot be converted to character (set to NA).

Usage

```
AISinfos(
  ais_data,
  threshold_length = 475,
  threshold_draught = 30,
  threshold_width = 75,
  weight_complete_data = 10
)
```

Arguments

<code>ais_data</code>	AIS data frame containing the columns <code>mmsi</code> , <code>shiptype</code> , <code>length</code> , <code>width</code> , <code>draught</code> , <code>imo</code> , and <code>name</code> . Another vessel identifier may be used if the column is named <code>mmsi</code> .
<code>threshold_length</code>	Maximum valid vessel length (m). Larger values are set to NA.
<code>threshold_draught</code>	Maximum valid draught (m). Larger values are set to NA.
<code>threshold_width</code>	Maximum valid vessel width (m). Larger values are set to NA.
<code>weight_complete_data</code>	Weight assigned to records containing both vessel length and ship type.

Value

A list containing:

- `estimated_values`: Estimated vessel characteristics for each `mmsi`.
- `summary`: Summary statistics for each `mmsi`, including:
 - Number of AIS positions.
 - Number of non-missing values for `length`, `shiptype`, `width`, `draught`, `imo`, and `name`.
 - All valid values observed for each characteristic.
 - The most likely value for each characteristic.

Examples

```
data("ais")

out <- AISinfos(ais_data = ais)
```

AISinterpolate	<i>Interpolate AIS positions</i>
----------------	----------------------------------

Description

Interpolates vessel positions either: (depending on `type_interpolation`)

- to ensure time intervals do not exceed a specified maximum (= `maximum_gap_seconds`).
- at user-defined timestamps (= `exact_timestamp`). Interpolation can optionally be restricted to specific regions to reduce computation time with `locations_of_interest` and `radius` arguments.

Usage

```

AISinterpolate(
  ais_data,
  type_interpolation,
  maximum_gap_seconds,
  exact_timestamp = list(timestamp_to_interpolate, locations_of_interest, radius),
  crs_meters = 3035,
  nb_cores = 1,
  outfile = tempfile()
)

```

Arguments

<code>ais_data</code>	AIS data frame containing timestamp, lon, lat, and mmsi. timestamp must be Unix time (seconds since 1970-01-01), while lon and lat must be numeric.
<code>type_interpolation</code>	Interpolation mode: "maximum_gap_seconds" or "exact_timestamp".
<code>maximum_gap_seconds</code>	used when <code>type_interpolation = "maximum_gap_seconds"</code> : threshold above which AIS signals are interpolated.
<code>exact_timestamp</code>	List used when <code>type_interpolation = "exact_timestamp"</code> , containing: • <code>timestamp_to_interpolate</code> • <code>locations_of_interest</code>: (optional) data frame with lon and lat columns corresponding to each <code>timestamp_to_interpolate</code> • <code>radius</code>: (optional) a search radius (m) around target locations
<code>crs_meters</code>	CRS (metres) used to calculate distances in the study area (defaults to EPSG:3035, Europe). Tip: use <code>suggest_crs</code> function (<code>crsuggest</code> package) to find a suitable CRS for your study area.
<code>nb_cores</code>	Number of CPU cores used.
<code>outfile</code>	File used to save logs.

Value

The interpolated AIS data with an additional column:

- `interpolated`: Whether the position was interpolated.

Examples

```

data("ais")
data("point_to_extract")

# use only a sample for the example:
ais <- ais[20000:30000, ]

# Define the Unix time (seconds since 1970-01-01)

```

```

point_to_extract$timestamp <- as.numeric(lubridate::ymd_hm(point_to_extract$datetime))
ais$timestamp <- as.numeric(lubridate::ymd_hms(ais$datetime))

# calculate the travelled distance, time, and speed:
ais <- AIStravel(ais_data = ais, crs_meters = 3035)

# Interpolate all AIS signals further than > 120 seconds:
out <- AISinterpolate(ais_data = ais,
                      type_interpolation = "maximum_gap_seconds",
                      maximum_gap_seconds = 120, ## Alternatively, you can
                      ## interpolate at target timestamps with:
                      # exact_timestamp = list(
                      #   timestamp_to_interpolate = point_to_extract$timestamp,
                      #   locations_of_interest = data.frame(lon = point_to_extract$lon,
                      #                                     lat = point_to_extract$lat),
                      #   radius = 200000),
                      crs_meters = 3035)

```

AIStravel

Calculate vessel travel metrics

Description

Calculates the distance travelled (m), travel time (s), and speed (km/h) between consecutive AIS positions for each vessel (mmsi).

Usage

```
AIStravel(ais_data, crs_meters = 3035, nb_cores = 1, outfile = tempfile())
```

Arguments

<code>ais_data</code>	AIS data frame containing timestamp, lon, lat, and mmsi. timestamp must be Unix time (seconds since 1970-01-01), while lon and lat must be numeric. Another vessel identifier may be used if the column is named mmsi.
<code>crs_meters</code>	CRS (metres) used to calculate distances in the study area (defaults to EPSG:3035, Europe). Tip: use <code>suggest_crs</code> function (<code>crsuggest</code> package) to find a suitable CRS for your study area.
<code>nb_cores</code>	Number of CPU cores used.
<code>outfile</code>	File used to save logs.

Value

The input AIS data with the following additional columns:

- `time_travelled`: Travel time (s) since the previous AIS position.
- `distance_travelled`: Distance travelled (m) since the previous AIS position.
- `speed_kmh`: Vessel speed (km/h).
- X, Y: coordinates in `crs_meters`.

Examples

```
data("ais")

# Define the Unix time (seconds since 1970-01-01)
ais$timestamp <- as.numeric(lubridate::ymd_hms(ais$datetime))

# calculate the travelled distance, time, and speed:
out <- AIStravel(ais_data = ais, crs_meters = 3035)
```

point_to_extract	<i>Example extraction locations</i>
------------------	-------------------------------------

Description

Example locations and timestamps used to demonstrate [AISextract\(\)](#) and interpolation at exact timestamps with [AISinterpolate\(\)](#). Each row defines a target location and time for which nearby vessel positions can be extracted or interpolated.

Format

A data frame with the following variables:

point Unique identifier of the target location.
lon, lat Longitude and latitude (WGS84).
datetime Target date and time (UTC).

See Also

[AISextract\(\)](#), [AISinterpolate\(\)](#)

Index

ais, [2](#)
AIScorrect_speed, [3](#)
AISextract, [4](#)
AISextract(), [11](#)
AISidentify_stations_aircraft, [6](#)
AISinfos, [7](#)
AISinterpolate, [8](#)
AISinterpolate(), [11](#)
AIStravel, [10](#)

point_to_extract, [11](#)