

Package ‘mardist’

October 1, 2026

Type Package

Title Calculation of Maritime Distances

Version 1.1.1

Encoding UTF-8

Date 2026-09-29

Description Tools for calculating and visualizing maritime distances and routes between geographic points. At its core, it implements a fast Haversine formula implemented in `data.table` to compute great circle distances across sea regions (i.e. avoiding land mass). The package builds a spatial network graph from port and cluster coordinates and uses a shortest path algorithm to identify optimal maritime routes between origin-destination pairs. For visualization, the package exports maps displaying individual routes, multi-destination networks, or continuous routes through specified waypoints. Utility functions identify the nearest network nodes to arbitrary coordinates and handle the antimeridian discontinuities common in Pacific maritime mapping. The package is particularly suited for analyzing shipping lanes, trade routes, and vessel trajectory data.

License EUPL

Depends R (>= 3.5)

Imports `data.table`, `dplyr`, `leaflet`, `igraph`

LazyData true

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Panayotis Christidis [cre, aut]

Maintainer Panayotis Christidis <Panayotis.Christidis@ec.europa.eu>

Repository CRAN

Date/Publication 2026-10-01 15:40:27 UTC

Contents

<code>cluster_coordinates</code>	2
<code>distance_route</code>	2
<code>dtHaversine</code>	3

multi_point_route	3
multi_route_map	4
multi_route_map_network	5
network	6
route_map	6

Index 7

cluster_coordinates	<i>Cluster coordinates</i>
---------------------	----------------------------

Description

Latitude and longitude of the maritime clusters used in the network.

Format

A data.frame with columns 'cluster', 'latitude', 'longitude'.

distance_route *Distance calculation*

Description

Calculates the maritime distance between two geographic points.

Usage

```
distance_route(lon1, lat1, lon2, lat2)
```

Arguments

lon1	numeric. Longitude of the origin (decimal degrees).
lat1	numeric. Latitude of the origin (decimal degrees).
lon2	numeric. Longitude of the destination (decimal degrees).
lat2	numeric. Latitude of the destination (decimal degrees).

Value

The distance in nautical miles (numeric).

Examples

```
distance_route(121.48, 31.2198, 79.85, 6.95)
```

dtHaversine	<i>Haversine great circle distance</i>
-------------	--

Description

Computes the great circle distance between two points on the Earth surface using the Haversine formula. The result is returned in the same linear unit as the radius *r* that you supply (the default is the WGS 84 Earth radius in metres).

Usage

```
dtHaversine(lat_from, lon_from, lat_to, lon_to, r = 6378137)
```

Arguments

<i>lat_from</i>	numeric. Latitude of the <i>*from*</i> point (decimal degrees).
<i>lon_from</i>	numeric. Longitude of the <i>*from*</i> point (decimal degrees).
<i>lat_to</i>	numeric. Latitude of the <i>*to*</i> point (decimal degrees).
<i>lon_to</i>	numeric. Longitude of the <i>*to*</i> point (decimal degrees).
<i>r</i>	numeric. Radius of the sphere used for the calculation. The default value 6378137 is the WGS84 equatorial radius in metres.

Value

The distance between the two points, expressed in the same unit as *r* (metres by default).

Examples

```
## Distance between Shanghai (121.48 E, 31.2198 N) and
## Goa (79.85 E, 6.95 N) the value is given in metres.
dtHaversine(31.2198, 121.48, 6.95, 79.85)
```

multi_point_route	<i>Map a continues route that passes through specific points</i>
-------------------	--

Description

Given a data frame / data.table ('clist') that contains a list of coordinates, the function draws a **leaflet** map with the shortest route line that connects the points in the list. The function is useful for visualising a full route that passes from specific points (e.g. all vessel port calls during a certain period).

Usage

```
multi_point_route(clist)
```

Arguments

clist A data.frame or data.table that must contain **two** columns with the exact names 'longitude' and 'latitude'. Any additional columns are ignored.

Value

An object of class **'leaflet'** that displays all of the shortest maritime routes contained in 'clist'.

Examples

```
## A multi-port route:
routes <- data.frame(cbind(
  longitude=c(120,122,90,60,30,25,0,8),
  latitude=c(30,31,40,32,40,34,34,55)))
multi_point_route(routes)
```

multi_route_map	<i>Map several maritime routes from a common origin</i>
-----------------	---

Description

Given a single origin ('lon1', 'lat1') and a data frame / data.table ('clist') that contains destination coordinates, the function draws a **'leaflet'** map with a separate shortest route line for each destination.

Usage

```
multi_route_map(lon1, lat1, clist)
```

Arguments

lon1 numeric. Longitude of the origin (decimal degrees).
lat1 numeric. Latitude of the origin (decimal degrees).
clist data.frame or data.table. Must contain columns 'lon' and 'lat'. Additional columns are ignored.

Value

A 'leaflet' map object displaying all routes.

Examples

```
destinations <- data.frame(lon = c(80, 85), lat = c(7, 10))
multi_route_map(121.48, 31.2198, destinations)
```

`multi_route_map_network`*Map the full maritime network between paired points*

Description

Given a data frame / data.table ('clist') that contains a collection of origin destination pairs, the function draws a **leaflet** map with a separate shortest route line for each pair. The function is useful for visualising an entire network of maritime connections (e.g. trade routes, shipping lanes, etc.).

Usage

```
multi_route_map_network(clist)
```

Arguments

<code>clist</code>	A data.frame or data.table that must contain four columns with the exact names 'longitude.x', 'latitude.x', 'longitude.y' and 'latitude.y'. The "*.x" columns represent the origins, the "*.y" columns represent the destinations. Any additional columns are ignored.
--------------------	---

Value

An object of class **"leaflet"** that displays all of the shortest maritime routes contained in 'clist'.

Examples

```
## Two dummy routes:
##   from (lon=120, lat=30) to (lon=80, lat=7)
##   from (lon=122, lat=31) to (lon=85, lat=10)
routes <- data.frame(
  longitude.x = c(120, 122),
  latitude.x  = c(30, 31),
  longitude.y = c(80, 85),
  latitude.y  = c(7, 10)
)
multi_route_map_network(routes)
```

network	<i>Maritime network</i>
---------	-------------------------

Description

An igraph edge list that describes the global maritime network. The object is a data frame with columns ‘cluster’, ‘lag_cluster’, and ‘distance’ (in metres).

Format

A data.frame with N rows and 3 columns.

Source

JRC

route_map	<i>Map a single maritime route (convenient short name)</i>
-----------	--

Description

Wrapper that forwards to ‘map_route’. It exists only so that the examples can use the short name ‘route_map’.

Usage

```
route_map(lon1, lat1, lon2, lat2)
```

Arguments

- lon1 longitude of the origin (decimal degrees)
- lat1 latitude of the origin (decimal degrees)
- lon2 longitude of the destination (decimal degrees)
- lat2 latitude of the destination (decimal degrees)

Value

A ‘leaflet’ map object.

Index

cluster_coordinates, [2](#)

distance_route, [2](#)

dtHaversine, [3](#)

multi_point_route, [3](#)

multi_route_map, [4](#)

multi_route_map_network, [5](#)

network, [6](#)

route_map, [6](#)