

# Package ‘dafishr’

July 23, 2024

**Title** Download, Wrangle, and Analyse Vessel Monitoring System Data

**Version** 1.0.1

**Description** Allows to download, clean and analyse raw Vessel Monitoring System, VMS, data from Mexican government. You can use the `vms_download()` function to download raw data, or you can use the `sample_dataset` provided within the package. You can follow the tutorial in the vignette available at <https://cbmc-gcmp.github.io/dafishr/index.html>.

**License** MIT + file LICENSE

**URL** <https://github.com/CBMC-GCMP/dafishr>,  
<https://cbmc-gcmp.github.io/dafishr/>

**BugReports** <https://github.com/CBMC-GCMP/dafishr/issues/>

**Depends** R (>= 3.5.0)

**Imports** dplyr, fst, ggplot2, lubridate, magrittr, mixtools, readr,  
readxl, rlang, sf, stringr, tibble, tidyr, tidyselect, utils,  
vroom

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Fabio Favoretto [aut, cre] (<<https://orcid.org/0000-0002-6493-4254>>),  
Eduardo Leon Solorzano [ctb]

**Maintainer** Fabio Favoretto <[fabio@gocmarineprogram.org](mailto:fabio@gocmarineprogram.org)>

**Repository** CRAN

**Date/Publication** 2024-07-22 22:10:09 UTC

## Contents

all_mpas . . . . .	2
clean_land_points . . . . .	3
join_mpa_data . . . . .	4
join_ports_locations . . . . .	5
model_vms . . . . .	6
mpas_buffers . . . . .	7
mx_coastline . . . . .	8
mx_coastline_buffer . . . . .	8
mx_eez . . . . .	9
mx_eez_pacific . . . . .	9
mx_inland . . . . .	10
mx_ports . . . . .	10
mx_shape . . . . .	11
pacific_landings . . . . .	11
pelagic_vessels_permits . . . . .	12
preprocessing_vms . . . . .	12
remote_mpas . . . . .	13
sample_dataset . . . . .	14
vms_clean . . . . .	14
vms_download . . . . .	15
<b>Index</b>	<b>17</b>

---

all\_mpas

*Marine Protected Areas (MPAs) of Mexico*

---

### Description

A sf object containing shapefiles of MPA polygons in Mexico

### Usage

all\_mpas

### Format

A simple feature collection with 24 features and 5 fields

**NOMBRE** Name of the MPA in Spanish

**CAT\_DECRET** Decree category, which define the type of MPA

**ESTADOS** State that have jurisdiction on the MPA

**MUNICIPIOS** Municipality that have jurisdiction on the MPA

**REGION** General regional localization of the MPA (in Spanish)

**geometry** column containing geometry details ...

**Source**

<http://sig.conanp.gob.mx/>

---

clean_land_points	<i>Clean points falling inland</i>
-------------------	------------------------------------

---

**Description**

This functions eliminates points falling inland by using `st_difference()` function from the `sf` package.

**Usage**

```
clean_land_points(x, mx_inland = mx_inland)
```

**Arguments**

<code>x</code>	A <code>data.frame</code> containing latitude and longitude coordinates of vessels tracks to be cleaned by land area
<code>mx_inland</code>	is a shapefile loaded with the packages representing inland Mexico area, it can be uploaded with <code>data("mx_inland")</code>

**Details**

Points falling inland in Vessel Monitoring System, VMS, dataset are obvious mistakes, thus need to be eliminated from the data. The function calls a stored shapefile `mx_inland` which is a custom `sf` object created using a coastline buffer to avoid eliminating points because of lack of precision within the shapefiles. The function works with any dataset containing coordinate points in `crs = 4326` and named `latitude` and `longitude`. See first example with a non-VMS dataset. A second example below shows the usage on VMS sample data.

**Value**

A `data.frame` object

**Warning**

This function takes a while!! To test you can use the `dplyr::sample_n()` function as it is shown in the example.

**Examples**

```

# with non VMS data
x <- data.frame(
  longitude = runif(1000, min = -150, max = -80),
  latitude = runif(1000, min = 15, max = 35)
)
data("mx_inland")
x <- clean_land_points(x, mx_inland)

# using sample_dataset

data("sample_dataset", "mx_inland")

vms_cleaned <- vms_clean(sample_dataset)
vms_no_land <- clean_land_points(vms_cleaned, mx_inland)

# You can check the results by plotting the data

vms_cleaned_sf <- sf::st_as_sf(vms_cleaned, coords = c("longitude", "latitude"), crs = 4326)
vms_no_land_sf <- sf::st_as_sf(vms_no_land, coords = c("longitude", "latitude"), crs = 4326)

library(ggplot2)
ggplot(vms_cleaned_sf) +
  geom_sf(col = "red") +
  geom_sf(data = vms_no_land_sf, col = "black")

# in the provided example only few inland points are eliminated.
# There are more evident one within historical data.

```

---

join_mpa_data	<i>Detect fishing vessel presence within Marine Protected Areas polygons in Mexico</i>
---------------	--

---

**Description**

The function spatially joins the Vessels Monitoring System, VMS, points with the Marine Protected Area, MPAs, polygons in Mexico.

**Usage**

```
join_mpa_data(x, all_mpas = all_mpas)
```

**Arguments**

x	A data.frame with VMS data that must contain columns longitude and latitude
all_mpas	A shape file that contains all MPA polygons in Mexico you can upload this using data("all_mpas")

## Details

It adds three columns zone, mpa\_decree, state, municipality, region, which are data from the MPAs polygon. zone contains the name of the MPA (in Spanish) and when the vessel is outside an MPA polygon is dubbed as open area, mpa\_decree contains the type of MPA (such as National Park, etc.), state contains the Mexican state with jurisdiction on the MPA, municipality contains the Mexican municipality with jurisdiction over the MPA, and region contains the overall location of the MPA (in Spanish)

## Value

A data.frame

## Examples

```
# Use sample_dataset
data("sample_dataset")
data("all_mpas")
vms_cleaned <- vms_clean(sample_dataset)
vms_mpas <- join_mpa_data(vms_cleaned, all_mpas)

# Plotting data
# Points NOT inside MPA are removed to reduce data size
vms_mpas_sub <- vms_mpas |>
  dplyr::filter(zone != "open area")

vms_mpas_sf <- sf::st_as_sf(vms_mpas_sub, coords = c("longitude", "latitude"), crs = 4326)

# Loading Mexico shapefile
data("mx_shape")

# Map
library(ggplot2)
ggplot(mx_shape, col = "gray90") +
  geom_sf(data = all_mpas, fill = "gray60") +
  geom_sf(data = vms_mpas_sf, aes(col = zone)) +
  theme_void() +
  theme(legend.position = "")
```

---

join\_ports\_locations *Label points when vessel is at port*

---

## Description

The function joins ports locations using data from ports buffers. mx\_ports data is used which is provided by INEGI <https://en.www.inegi.org.mx/>

**Usage**

```
join_ports_locations(x, mx_ports = mx_ports, buffer_size = 0.15)
```

**Arguments**

`x` a data.frame with latitude and longitude coordinates

`mx_ports` is a shapefile of point data storing coordinates of ports and marina in Mexico, you can upload this using `data("mx_ports")`

`buffer_size` a number (double) indicating the size of the buffer for the ports to implement

**Details**

The function adds a `location` column indicating if the vessel was at port or at sea.

**Value**

A data.frame

**Examples**

```
# With sample data

data("sample_dataset")
data("mx_ports")
vms_cleaned <- vms_clean(sample_dataset)

# It is a good idea to subsample when testing... it takes a while on the full data!

vms_subset <- dplyr::sample_n(vms_cleaned, 1000)
with_ports <- join_ports_locations(vms_subset)
with_ports_sf <- sf::st_as_sf(with_ports, coords = c("longitude", "latitude"), crs = 4326)

data("mx_shape")
library(ggplot2)
ggplot(mx_shape) +
  geom_sf(col = "gray90") +
  geom_sf(data = with_ports_sf, aes(col = location)) +
  facet_wrap(~location) +
  theme_bw()
```

**Description**

This function uses `normalmixEM` from the `mixtools` package to model speed of vessels and estimates their behavior. Specifically, if the vessel was in a fishing activity or cruising

**Usage**

```
model_vms(df)
```

**Arguments**

**df** a data.frame preprocessed using the `preprocessing_vms()` function from this package

**Value**

a data.frame with a `vessel_state` column with the type of model implemented

**Examples**

```
preprocessing_vms(sample_dataset, destination.folder = tempdir())
df <- fst::read_fst(paste0(tempdir(), "/vms_2019_1_1_10_preprocessed.fst"))
model_vms(df)
```

---

mpas\_buffers

*Buffer around remote Marine Protected Areas, MPAs, of Mexico*


---

**Description**

A `sf` object containing shapefiles of buffers around remote MPAs in Mexico. The buffer equals the area inside each MPA polygon and was created to assess differences in fishing activity inside or outside each of the remote MPAs.

**Usage**

```
mpas_buffers
```

**Format**

A simple feature collection with 5 features and 2 fields

**Name** Name of the MPAs to which the buffer correspond

**Description** empty

**geometry** column containing geometry details ...

**Source**

this project

---

mx_coastline	<i>Mexican coastline</i>
--------------	--------------------------

---

**Description**

A sf object containing a the Mexican coastline shapefile

**Usage**

```
mx_coastline
```

**Format**

A simple feature collection with 177 features and 3 fields

**featurecla** Name of the object

**scalerank** resolution rank

**min\_zoom** zoom precision

**geometry** column containing geometry details ...

**Source**

<https://cran.r-project.org/package=rnaturalearth>

---

mx_coastline_buffer	<i>Buffer around the Mexican coastline</i>
---------------------	--

---

**Description**

A sf object containing a buffer around Mexican coastline that was used to create the inland shapefile available in this package.

**Usage**

```
mx_coastline_buffer
```

**Format**

A simple feature collection with 1 feature and 3 fields

**featurecla** Name of the object

**scalerank** resolution rank

**min\_zoom** zoom precision

**geometry** column containing geometry details ...

**Source**

<https://cran.r-project.org/package=rnaturalearth>



---

mx_eez	<i>Mexico shape</i>
--------	---------------------

---

**Description**

A sf object containing the shapefile representing Mexico

**Usage**

mx\_eez

**Format**

A simple feature collection with 1 features and 2 fields

**Name** empty

**Description** empty

**geometry** column containing geometry details ...

**Source**

<https://en.www.inegi.org.mx/>

---

mx_eez_pacific	<i>Economic Exclusive Zone (EEZ) of the Pacific side of Mexico</i>
----------------	--

---

**Description**

A sf object containing shapefiles of Mexican EEZ in the Pacific

**Usage**

mx\_eez\_pacific

**Format**

A simple feature collection with 1 feature and 1 field

**Name** Mexican Pacific Exclusive Economic Zone

**geometry** column containing geometry details ...

**Source**

<https://en.www.inegi.org.mx/>

---

mx_inland	<i>Area inland of Mexico</i>
-----------	------------------------------

---

**Description**

A sf object containing shapefiles of inland area in Mexico

**Usage**

```
mx_inland
```

**Format**

A simple feature collection with 1 feature and 2 fields

**Name** Mexico

**Description** empty

**geometry** column containing geometry details ...

**Source**

modified from Mexican shapefile

---

mx_ports	<i>Ports and Marinas of Mexico</i>
----------	------------------------------------

---

**Description**

A sf object containing points representing the locations of Ports and Marinas in Mexico

**Usage**

```
mx_ports
```

**Format**

A simple feature collection with 237 features and 2 fields

**class** Type of infrastructure it can be Puerto (Port), or Marina

**name** Name of the infrastructure (i.e. port or marina)

**geometry** column containing geometry details ...

**Source**

<https://en.www.inegi.org.mx/>

---

mx_shape	<i>Mexico mainland</i>
----------	------------------------

---

**Description**

A sf object containing a shapefile of Mexico

**Usage**

```
mx_shape
```

**Format**

A simple feature collection with 1 feature and 2 fields

**Name** Mexico

**Description** empty

**geometry** column containing geometry details ...

**Source**

<https://en.www.inegi.org.mx/>

---

pacific_landings	<i>Catch data from the vessels in Mexico</i>
------------------	--

---

**Description**

A data.frame object containing catch data per each vessel from 2008 to 2021. Vessels are only from the Pacific and are only Tuna, Sharks, and Marlin catches. The dataset was created by wrangling and filtering the raw data (available under request to the authors).

**Usage**

```
pacific_landings
```

**Format**

A data.frame with 23,231 rows and 5 columns

**date** Date of the catch report

**rnp\_activo** Vessel RNP unique ID code

**vessel\_name** Official name of the vessel

**catch** Final weight of the catch in tons

**days\_declared** Days at sea that were declared at port ...

**Source**

Data are available under request to CONAPESCA, a raw version of data is available under request to authors

---

pelagic\_vessels\_permits

*List of vessels with pelagic fishing permits*

---

**Description**

A data.frame object extracted from a raw dataset of permits available under request at dataMares (<https://datamares.org/>)

**Usage**

```
pelagic_vessels_permits
```

**Format**

A data.frame with 719 rows and 2 columns.

**RNP** Unique code identifying the vessel

**vessel\_name** Name of the vessel ...

**Source**

<https://www.datamares.org/>

---

preprocessing\_vms

*Preprocessing Vessel Monitoring System data*

---

**Description**

This functions bundles all the cleaning functions and allows them to be easily used in parallel processing to speed up the cleaning of all the Vessel Monitoring System, VMS, data .csv files. While it runs, it creates a folder called preprocessed that will store VMS data that underwent the preprocessing. If multiple files are used as input (see examples below) it will create multiple files. All the outputs are in .fst format, which allows fast upload of large files. See fst package documentation for further information <https://www.fstpackage.org/>.

**Usage**

```
preprocessing_vms(files.path, destination.folder)
```

**Arguments**

`files.path` it can be a path to the file downloaded or the data object itself. If function is used with a path it adds a file column to the returning data.frame object that stores the name of the file as a reference.

`destination.folder` it must record the path to a folder were all the preprocessed files will be stored.

**Value**

A .fst file saved within a directory chosen by the user, that is created automatically if does not exist, and that stores each of the files that are used as input to the function.

**Examples**

```
# An example with the `sample.dataset`
preprocessing_vms(sample_dataset, destination.folder = tempdir())
```

---

remote\_mpas

*Remote Marine Protected Areas (MPAs) of Mexico*


---

**Description**

A sf object containing shapefiles of remote MPA polygons in Mexico that are of particular conservation interest

**Usage**

```
remote_mpas
```

**Format**

A simple feature collection with 5 features and 2 fields

**Name** Name of the remote MPA in Spanish

**Description** empty

**geometry** column containing geometry details ...

**Source**

<http://sig.conanp.gob.mx/>

---

sample_dataset	<i>Vessel Monitoring System, VMS, sample dataset from Mexican fishery commission</i>
----------------	--

---

### Description

A data.frame object extracted from a raw dataset of Vessels Monitoring System, VMS, data from the year 2019.

### Usage

```
sample_dataset
```

### Format

A data.frame with 10,000 rows and 9 columns.

**Nombre** Name of the vessel

**RNP** Unique code identifying the vessel

**Puerto Base** Base port where the vessel is officially registered

**Permisionario o Concesionario** Owner of the vessel or partnership name

**FechaRecepcionUnitrac** Date as "%d/%m/%Y %H:%M"

**Latitud** Latitude degree in WGS84, crs = 4326, of the position of the vessel

**Longitud** Longitude degree in WGS84, crs = 4326, of the position of the vessel

**Velocidad** Speed in knots of the vessel at that specific time

**Rumbo** Direction in degrees of the vessel at that specific time ...

### Source

<https://www.datos.gob.mx/>

---

vms_clean	<i>Fixing dates and column names</i>
-----------	--------------------------------------

---

### Description

This function cleans raw Vessel Monitoring System, VMS, data column files, eliminate NULL values in coordinates, parse dates, and returns a data.frame.

### Usage

```
vms_clean(path_to_data)
```

**Arguments**

`path_to_data` it can be a path to the file downloaded or the data object itself. If function is used with a path it adds a `file` column to the returning `data.frame` object that stores the name of the file as a reference.

**Details**

It takes a raw data file downloaded using the `vms_download()` function by specifying directly its path or by referencing a `data.frame` already stored as an R object. If path is used, column with the name of the raw file is conveniently added as future reference. It also split date into three new columns `year`, `month`, `day`, and retains the original date column. This function can be used with apply functions over a list of files or it can be paralleled using `furrr` functions.

**Value**

A `data.frame`

**Examples**

```
# Using sample dataset, or a data.frame already stored as an object
# It is possible to use a path directly as argument
```

```
data("sample_dataset")
cleaned_vms <- vms_clean(sample_dataset)
head(cleaned_vms)
```

---

`vms_download`

*Download Vessel Monitoring System, VMS, raw data*

---

**Description**

This functions download data form the *Datos Abiertos* initiative

**Usage**

```
vms_download(
  year = lubridate::year((Sys.time())) - 1,
  destination.folder,
  check.url.certificate = TRUE
)
```

**Arguments**

`year` year of data that user wants to download are selected default to the last year. A vector of years can also be used.

`destination.folder`

can be set to a folder where user want the data to be downloaded into. Defaults to working directory.

`check.url.certificate`

logical. Under Ubuntu systems the function might draw a certificate error, you can deactivate the certificate check by setting this to FALSE and should work.

### Details

Data are downloaded from this link: <https://www.datos.gob.mx/busca/dataset/localizacion-y-monitoreo-satelital-de-embarcaciones-pesqueras/> Downloaded data will be downloaded and decompressed in a VMS-data folder in a location chosen by the user by specifying a path in `destination.folder`. If a location is not specified it downloads data by default to the current working directory. Within the main folder, data is organized in different folders by months (in Spanish names) and within each there are multiple .csv files each containing two weeks of data points.

### Value

saves downloaded data into a folder called VMS-data within the directory specified

### Examples

```
# Download single year
# in Ubuntu it draws a certificate error when downloading, testing in windows and MacOS
# does not draw that error and you can use default certificate checking.
```

```
vms_download(2019, destination.folder = tempdir(), check.url.certificate = FALSE)
```



# Index

## \* datasets

- all\_mpas, [2](#)
- mpas\_buffers, [7](#)
- mx\_coastline, [8](#)
- mx\_coastline\_buffer, [8](#)
- mx\_eez, [9](#)
- mx\_eez\_pacific, [9](#)
- mx\_inland, [10](#)
- mx\_ports, [10](#)
- mx\_shape, [11](#)
- pacific\_landings, [11](#)
- pelagic\_vessels\_permits, [12](#)
- remote\_mpas, [13](#)
- sample\_dataset, [14](#)

all\_mpas, [2](#)

clean\_land\_points, [3](#)

join\_mpa\_data, [4](#)

join\_ports\_locations, [5](#)

model\_vms, [6](#)

mpas\_buffers, [7](#)

mx\_coastline, [8](#)

mx\_coastline\_buffer, [8](#)

mx\_eez, [9](#)

mx\_eez\_pacific, [9](#)

mx\_inland, [10](#)

mx\_ports, [10](#)

mx\_shape, [11](#)

pacific\_landings, [11](#)

pelagic\_vessels\_permits, [12](#)

preprocessing\_vms, [12](#)

remote\_mpas, [13](#)

sample\_dataset, [14](#)

vms\_clean, [14](#)

vms\_download, [15](#)