

Package: artfishr (via r-universe)

July 10, 2026

Version 0.1.20260701

Date 2026-07-01

Title ArtFish for R

Maintainer Emmanuel Blondel <emmanuel.blondel1@gmail.com>

Depends R (>= 4.1)

Imports methods, dplyr, tidyr, readr, data.table, DT, lubridate,
vrule, ggplot2, shiny, shiny.i18n, shinyWidgets, bs4Dash,
fdishinyr (>= 0.1.20260415)

Remotes fdiwg/vrule, fdiwg/fdishinyr

Suggests testthat, knitr

Description Set of functions in support of ArtFish implementation.

License MIT

URL <https://github.com/fdiwg/artfishr>,
<https://fdiwig.github.io/artfishr/>

BugReports <https://github.com/fdiwg/artfishr/issues>

RoxygenNote 7.3.3

VignetteBuilder knitr

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev libx11-dev zlib1g-dev

Repository <https://fdiwig.r-universe.dev>

Date/Publication 2026-07-10 06:03:55 UTC

RemoteUrl <https://github.com/fdiwg/artfishr>

RemoteRef HEAD

RemoteSha 102e1ddbf2083969cfa464c9f2f087ebb284cacd

Contents

accuracy_plot	3
artfish_accuracy	3
artfish_estimates	4
artfish_estimates_by_fleet_segment	4
artfish_flouca_by_period	5
artfish_new	6
artfish_new_by_period	7
artfish_shiny_accuracy_server	7
artfish_shiny_accuracy_ui	8
artfish_shiny_computation_server	8
artfish_shiny_fishing_unit_server	9
artfish_shiny_fishing_unit_ui	10
artfish_shiny_overview_server	11
artfish_shiny_overview_ui	12
artfish_shiny_report_server	12
artfish_shiny_report_ui	13
artfish_shiny_species_server	14
artfish_shiny_species_ui	15
artfish_shiny_welcome_server	15
artfish_shiny_welcome_ui	16
artfishr	16
artfishr_shiny_server	17
artfishr_shiny_ui	17
complete_active_days	17
compute_accuracy	18
compute_catch_estimate	18
compute_catch_estimates_by_species	19
compute_cpue	20
compute_effort_activity_coefficient	20
compute_effort_estimate	21
compute_effort_estimate_with_FAO_Artfish	22
compute_effort_estimate_with_IRD_Pechart	23
compute_report	24
compute_sui	25
create_artfish_template	25
create_artfish_templates	26
decode_artfish_schema	27
ERROR	27
generate_active_days	28
generate_active_days_by_period	28
get_fdi_terms	29
get_vrule_validators	30
INFO	30
join_guess_by	30
logger	31
run_artfishr_shiny	31

<i>accuracy_plot</i>	3
----------------------	---

sample_size_plot	31
select_active_vessels	32
set_translation_language	33
set_vrule_validators	33
translator	34
unif_index	34
validate_artfish_datasets	34
validate_input_datasets	35
WARN	35

Index	36
--------------	-----------

accuracy_plot	<i>Produces a plot for the accuracy</i>
---------------	---

Description

Produces a plot for the accuracy

Usage

accuracy_plot(sample = NULL, population, method = "higher")

Arguments

sample	sample
population	population
method	method. Default is 'higher'

artfish_accuracy	<i>artfish_accuracy</i>
------------------	-------------------------

Description

artfish_accuracy

Usage

artfish_accuracy(n, N, method = "higher")

Arguments

n	n
N	N
method	method

artfish_estimates	<i>artfish_estimates</i>
-------------------	--------------------------

Description

artfish_estimates

Usage

```
artfish_estimates(  
  data_effort = NULL,  
  data_landing = NULL,  
  ref_fishingunits = NULL,  
  ref_species,  
  year = NULL,  
  month = NULL  
)
```

Arguments

- data_effort effort data
- data_landing landing data
- ref_fishingunits
 fishing units reference dataset
- ref_species species reference dataset

Value

the result of Artfish

artfish_estimates_by_fleet_segment	<i>artfish_estimates_by_fleet_segment</i>
------------------------------------	---

Description

artfish_estimates_by_fleet_segment

Usage

```
artfish_estimates_by_fleet_segment(
  data_effort = NULL,
  data_landing = NULL,
  ref_fishingunits = NULL,
  ref_species,
  year = NULL,
  month = NULL
)
```

Arguments

data_effort	effort data
data_landing	landing data
ref_fishingunits	fishing units reference dataset
ref_species	species reference dataset

Value

the result of Artfish

artfish_flouca_by_period
<i>artfish_flouca_by_period</i>

Description

artfish_flouca_by_period

Usage

```
artfish_flouca_by_period(
  year = NULL,
  month = NULL,
  active_vessels,
  effort,
  effort_source = c("survey", "registry"),
  active_days = NULL,
  landings,
  minor_strata = NULL,
  validate = TRUE
)
```

Arguments

active_vessels	active vessels
effort	effort
effort_source	effort source whether it's derived from survey -B1- (fishers interviews) or registry -B2- (boat counting)
active_days	active days
landings	landings
minor_strata	minor_strata (to investigate further later)

Value

the result of Artfish for a given year/month

artfish_new	<i>artfish_new</i>
-------------	--------------------

Description

artfish_new

Usage

```
artfish_new(  
  active_vessels = NULL,  
  effort = NULL,  
  effort_source = c("fisher_interview", "boat_counting"),  
  active_days = NULL,  
  landings = NULL,  
  minor_strata = NULL  
)
```

Arguments

active_vessels	active vessels
effort	effort
effort_source	effort source whether it's derived from -B1- (fishers interviews) or -B2- (boat counting)
active_days	active days
landings	landings
minor_strata	minor_strata (to investigate further later)

Value

the result of Artfish

artfish_new_by_period	<i>artfish_new_by_period</i>
-----------------------	------------------------------

Description

artfish_new_by_period

Usage

```
artfish_new_by_period(  
  year = NULL,  
  month = NULL,  
  active_vessels,  
  effort,  
  effort_source = c("fisher_interview", "boat_counting"),  
  active_days = NULL,  
  landings,  
  minor_strata = NULL,  
  validate = TRUE  
)
```

Arguments

- active_vessels active vessels
- effort effort
- effort_source effort source whether it's derived from -B1- (fishers interviews) or -B2- (boat counting)
- active_days active days
- landings landings
- minor_strata minor_strata (to investigate further later)

Value

the result of Artfish for a given year/month

artfish_shiny_accuracy_server
<i>POC shiny module server</i>

Description

Accuracy toolbox shiny module server

Usage

```
artfish_shiny_accuracy_server(id, lang = NULL)
```

Arguments

id	id
lang	lang a reactive version of the language. Default is NULL (optional)

artfish_shiny_accuracy_ui
<i>POC shiny module UI</i>

Description

POC shiny module UI

Usage

```
artfish_shiny_accuracy_ui(id)
```

Arguments

id	id
----	----

artfish_shiny_computation_server
<i>ARTFISH Computation - Shiny Server Module</i>

Description

Server-side shiny logic to run Artfish computation as business service

Usage

```
artfish_shiny_computation_server(  
  id,  
  refresh,  
  effort,  
  effort_source,  
  active_vessels,  
  active_vessels_strategy,  
  active_days,  
  landings,  
  minor_strata = NULL,  
  progress_fn = NULL  
)
```


Arguments

id	Character string. Module namespace identifier.
refresh	refresher for computation
effort	effort data
effort_source	Character string indicating the type of effort source.
active_vessels	active vessels data
active_vessels_strategy	active vessels strategy
active_days	active days
landings	landings
minor_strata	Character string targeting a column name considered as minor strata.
progress_fn	A function to monitor progress

artfish_shiny_fishing_unit_server

ARTFISH Fishing Unit - Shiny Server Module

Description

Server-side logic for the ARTFISH Fishing Unit dashboard.

The `"Catch and effort - Detailed by fishing unit"` module provides a dedicated workspace for exploring key output metrics derived from `artfish_compute_report`, with a focus on one or multiple fishing units. It enables users to analyse the evolution of fishing activity over time, with filtering options for fishing units and time range.

The module is composed of:

- A selector panel to adjust the time range and fishing units to display
- A set of KPIs presenting aggregated metrics for the selected period
- Three boxed plots powered by `fdishinyr::generic_chart` allowing users to analyse ranked top items for:
 - (1) Species composition (by quantity)
 - (2) Fishing unit composition (by quantity)
 - (3) Fishing unit composition (by value)
- Six boxed plots powered by `fdishinyr::generic_chart` allowing users to analyse key Artfish metrics:
 - (1) Total catch estimation
 - (2) Catch per unit effort (CPUE)
 - (3) Total nominal effort
 - (4) Activity coefficient
 - (5) Number of active vessels
 - (6) Total value estimation

Usage

```
artfish_shiny_fishing_unit_server(
  id,
  lang = NULL,
  estimate,
  effort_source,
  minor_strata = NULL,
  opts = list()
)
```

Arguments

id	Character string. Module namespace identifier.
lang	Optional language parameter. Can be either: • a character string (static language), or • a reactive returning a character string (dynamic language) If NULL, the current global language is used. Default is NULL
estimate	A reactive data frame aggregating the output of artfish_compute, enriched with human-readable labels.
effort_source	Reactive haracter string indicating the type of effort source. Must be either "fisher_interview" or "boat_counting".
minor_strata	Reactive haracter string targeting a column name considered as minor strata. Not activated
opts	a named list of options. For now limited to: - refresh_ui that gives the capacity to inject a refresh UI button (for dynamic computation)

artfish_shiny_fishing_unit_ui

ARTFISH Fishing unit - Shiny UI Module

Description

User interface for the ARTFISH Fishing Unit Dashboard

Usage

```
artfish_shiny_fishing_unit_ui(id)
```

Arguments

id	Character string. Module namespace identifier.
----	--

artfish_shiny_overview_server

ARTFISH Overview - Shiny Server Module

Description

Server-side logic for the ARTFISH Overview dashboard.

The `"Total catch and effort - Statistics at country level"` module provides a high-level visualization of key output metrics derived from `artfish_compute_report`. It acts as a dashboard to track the overall evolution of fishing activity over time, with the ability to filter by fishing units and time range.

The module is composed of:

- A selector panel to adjust the time range and fishing units to display
- A set of KPIs presenting aggregated country-level metrics for the selected period
- Seven boxed plots powered by `fdishinyr::generic_chart` allowing users to analyse:
 - (1) Total catch estimation
 - (2) Species composition (by quantity)
 - (3) Total nominal effort
 - (4) Number of active vessels
 - (5) Total value estimation
 - (6) Species composition (by value)
 - (7) Catch per unit effort (CPUE)

Usage

```
artfish_shiny_overview_server(
  id,
  lang = NULL,
  estimate,
  effort_source = NULL,
  minor_strata = NULL,
  opts = list()
)
```

Arguments

<code>id</code>	Character string. Module namespace identifier.
<code>lang</code>	Optional language parameter. Can be either: • a character string (static language), or • a reactive returning a character string (dynamic language) If NULL, the current global language is used. Default is NULL
<code>estimate</code>	A reactive data frame aggregating the output of <code>artfish_compute</code> , enriched with human-readable labels.

effort_source	Reactive Character string indicating the type of effort source. Must be either "fisher_interview" or "boat_counting". Not activated
minor_strata	Reactive Character string targeting a column name considered as minor strata. Not activated
opts	a named list of options. For now limited to: - refresh_ui that gives the capacity to inject a refresh UI button (for dynamic computation)

artfish_shiny_overview_ui
<i>ARTFISH Overview - Shiny UI Module</i>

Description

User interface for the ARTFISH Overview Dashboard

Usage

artfish_shiny_overview_ui(id)

Arguments

id Character string. Module namespace identifier.

artfish_shiny_report_server
<i>ARTFISH Report - Shiny Server Module</i>

Description

Server-side logic for the ARTFISH Report dashboard.

The *"Detailed results of Artfish computation – by stratum"* module provides an ergonomic view of key intermediate results and calculation steps underlying the Artfish algorithm for a given temporal unit.

The initial view displays a selector inviting the user to choose a year. If no year is available, the user must first compute at least one indicator using `artfish_compute_report`. Successive dropdown lists then allow the user to select a month and a fishing unit before submitting the selection. A complete report is displayed on the right-hand side.

The module is composed of:

- A selector panel to choose the report to display by selecting year, month, and fishing unit
- A button to validate the selection
- A set of KPIs presenting the report status and the type of effort source
- A gauge chart presenting the overall accuracy quality of the indicator

- An ergonomic report view including intermediate steps and final indicators for the estimation of:
 - (1) Effort
 - (2) Landings
 - (3) Estimated catch by species

Usage

```
artfish_shiny_report_server(
  id,
  lang = NULL,
  estimate,
  effort_source,
  minor_strata
)
```

Arguments

id	Character string. Module namespace identifier.
lang	Optional language parameter. Can be either: • A character string (static language), or • A reactive expression returning a character string (dynamic language) If NULL, the current global language is used. Default is NULL.
estimate	A data frame aggregating the output of <code>artfish_compute</code> , enriched with human-readable labels.
effort_source	Character string indicating the type of effort source. Must be either "fisher_interview" or "boat_counting".
minor_strata	Character string targeting a column name considered as minor strata.

artfish_shiny_report_ui

ARTFISH Report - Shiny UI Module

Description

User interface for the ARTFISH Report Dashboard

Usage

```
artfish_shiny_report_ui(id)
```

Arguments

id	Character string. Module namespace identifier.
----	--

artfish_shiny_species_server

ARTFISH Species - Shiny Server Module

Description

Server-side logic for the ARTFISH Species dashboard.

The `"Catch and effort - Detailed by species"` module provides a species-oriented workspace for exploring key output metrics derived from `artfish_compute_report`, with a focus on individual species caught.

The initial view displays a selector inviting the user to choose a species. The dropdown lists all species for which entries are available in the database. Users can then analyse the evolution of fishing activity over time, with filtering options for species, fishing units, and time range.

The module is composed of:

- A selector panel to choose the species to display
- A selector panel to adjust the time range and fishing units to display
- Two boxed plots powered by `fdishinyr::generic_chart` allowing users to analyse, at species level:
 - (1) Fishing unit composition (by quantity)
 - (2) Target species ranking (by quantity)
- A set of KPIs presenting aggregated metrics for the selected period
- Five boxed plots powered by `fdishinyr::generic_chart` allowing users to analyse key Art-fish metrics:
 - (1) Total catch estimation
 - (2) Catch per unit effort (CPUE)
 - (3) Total value estimation
 - (4) Average price
 - (5) Total nominal effort

Usage

```
artfish_shiny_species_server(
  id,
  lang = NULL,
  estimate,
  effort_source = NULL,
  minor_strata = NULL,
  opts = list()
)
```

Arguments

id	Character string. Module namespace identifier.
lang	Optional language parameter. Can be either: • a character string (static language), or • a reactive returning a character string (dynamic language) If NULL, the current global language is used. Default is NULL
estimate	A reactive data frame aggregating the output of artfish_compute, enriched with human-readable labels.
effort_source	Reactive character string indicating the type of effort source. Must be either "fisher_interview" or "boat_counting". Not activated
minor_strata	Reactive character string targeting a column name considered as minor strata. Not activated
opts	a named list of options. For now limited to: - refresh_ui that gives the capacity to inject a refresh UI button (for dynamic computation)

artfish_shiny_species_ui

ARTFISH Species - Shiny UI Module

Description

User interface for the ARTFISH Species Dashboard

Usage

```
artfish_shiny_species_ui(id)
```

Arguments

id	Character string. Module namespace identifier.
----	--

artfish_shiny_welcome_server

POC shiny module server

Description

POC shiny module server

Usage

```
artfish_shiny_welcome_server(id, lang = NULL)
```

Arguments

id	id
lang	lang a reactive version of the language. Default is NULL (optional)

artfish_shiny_welcome_ui	<i>POC shiny module UI</i>
--------------------------	----------------------------

Description

POC shiny module UI

Usage

```
artfish_shiny_welcome_ui(id)
```

Arguments

id	id
----	----

artfishr	<i>Tools for implementing ArtFish methodology</i>
----------	---

Description

artfishr provides a set of utility tools to implement ArtFish methodology

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

See Also

Useful links:

- <https://github.com/fdiwg/artfishr>
- <https://fdiwig.github.io/artfishr/>
- Report bugs at <https://github.com/fdiwg/artfishr/issues>

artfishr_shiny_server *Artfishr shiny server function*

Description

Artfishr shiny server function

Usage

```
artfishr_shiny_server(input, output, session)
```

artfishr_shiny_ui *Artfishr shiny ui function*

Description

Artfishr shiny ui function

Usage

```
artfishr_shiny_ui(id)
```

complete_active_days *Complete the active days.*

Description

If it is set to zero or or is NA, the nb of active days in the period will be autogenerated

Usage

```
complete_active_days(active_days)
```

Arguments

active_days active days object of class [tibble](#)

Value

the curated active days table

compute_accuracy	<i>Computes Accuracy</i>
------------------	--------------------------

Description

Compute spatial and temporal accuracy for activity coefficient and cpue. Overall accuracy return the minimal value of the 4 accuracy indicators.

Usage

```
compute_accuracy(
  activity_coefficient,
  effort_estimate,
  cpue,
  sui,
  minor_strata = NULL
)
```

Arguments

activity_coefficient	activity_coefficient
effort_estimate	effort_estimate
cpue	cpue
minor_strata	minor_strata. Default is NULL

Value

a [tibble](#) object giving the different accuracy by strata

compute_catch_estimate	<i>Computes catch estimate</i>
------------------------	--------------------------------

Description

Computes catch estimate

Usage

```
compute_catch_estimate(effort_estimate, landings, minor_strata = NULL)
```

Arguments

effort_estimate	effort estimate computed with compute_effort_estimate
landings	landings
minor_strata	minor_strata. Default is NULL

Value

a [tibble](#) giving the estimated catch by strata

compute_catch_estimates_by_species
<i>Computes catch estimates by species</i>

Description

Computes catch estimates by species

Usage

```
compute_catch_estimates_by_species(  
  landings,  
  catch_estimate,  
  minor_strata = NULL  
)
```

Arguments

landings	landings
catch_estimate	result of catch estimate computed with compute_catch_estimate
minor_strata	minor_strata. Default is NULL

Value

a [tibble](#)

compute_cpue

Computes CPUE

Description

The CPUE (Catch Per Unit of Effort) is computed from the landings, as the ratio between nominal landed catches (`catch_nominal_landed`) and the effort fishing duration (`effort_fishing_duration`).

The computation is performed grouped by a strata compound by year, month and fishing_unit. This strata can be extended with additional columns with the `minor_strata` argument.

Since landings give details data on landed species, nominal landed catches are sum grouped by the strata and by fishing trip to get the total nominal landed catches by fishing trip. Both `catch_nominal_landed` and `effort_fishing_duration` are then sum by strata and the CPUE is computed as the ratio of these sum.

Note: Additional checks are performed to remove data with NAs.

Usage

```
compute_cpue(landings, minor_strata = NULL)
```

Arguments

`landings` `landings`
`minor_strata` `minor_strata`. Default is NULL

Value

a [tibble](#) object giving the CPUE by strata

compute_effort_activity_coefficient

Computes effort activity coefficient

Description

The activity coefficient is computed from the effort. Depending on the source of effort data, the source information used is different: - In the case of 'fisher_interview' the coefficient of activity is computed as the ratio between the `effort_fishing_duration` and the `effort_fishing_reference_period`. - In the case of 'boat_counting' the coefficient of activity is computed as the ratio between the `fleet_engagement_number` and the `fleet_engagement_max`

The computation is performed grouped by a strata compound at minimum by the year, month and fishing_unit. This strata can be extended by adding one or more columns with the `minor_strata` argument.

Note: Additional check are performed to remove data with NAs, and ensure data consistency

Usage

```
compute_effort_activity_coefficient(  
  effort,  
  effort_source = c("fisher_interview", "boat_counting"),  
  minor_strata = NULL  
)
```

Arguments

effort	effort data
effort_source	effort
minor_strata	minor_strata. Default is NULL

Value

a [tibble](#) object giving activity coefficient by strata

compute_effort_estimate

Computes nominal effort estimate

Description

Computes nominal effort estimate

Usage

```
compute_effort_estimate(  
  effort,  
  effort_source = c("fisher_interview", "boat_counting", "household_interview"),  
  landings,  
  active_days = NULL,  
  active_vessels = NULL,  
  active_vessels_strategy = c("latest", "closest"),  
  census_typology = NULL,  
  minor_strata = NULL,  
  progress_fn = NULL  
)
```

```
compute_effort_estimate_with_FAO_Artfish
```

Computes nominal effort estimate

Description

The effort estimate is computed as the product of the `fleet_engagement_number` (derived from the `active_vessels`), the `effort_fishable_duration` (derived from the `active_days`, and the `effort_activity_coefficient` computed with the [compute_effort_activity_coefficient](#) function.

The computation of the nominal effort estimate is done grouped by strata compound of year, month and fishing_unit. This strata can be extended with additional columns with the `minor_strata` argument.

Active vessels:

The `active_vessels` can be either time-dependent, ie given by year/month, year only or be atemporal, mainly depending on how and at which frequency the vessels census/survey is operated in the country.

An additional argument called `active_vessels_strategy` controls how to select the `active_vessels` data in time, when this data is time-dependent. The `latest` strategy will select the latest data acquired in time, while the `closest` will select the closest data acquired in time, which could be data acquired after the effort data considered.

As example, a vessel census is performed each five years, and data is available for 2007 and 2012. Effort data for 2011 needs to be computed. Which `active_vessels` data should be used? In case of a `latest` strategy, in 2011, the latest data available is 2007, so this one will be used. In the case of a `closest` strategy, in 2011, we used the closest vessel data (ie 2012), assuming it better characterizes the fleet engagement for the year considered.

Active days:

In case of 'fisher_interview' (as source of the effort data), the `active_days` data should be ignored, as the days of the month will be used (and generated automatically by the function)

Usage

```
compute_effort_estimate_with_FAO_Artfish(
  effort,
  effort_source = c("fisher_interview", "boat_counting"),
  landings,
  active_days = NULL,
  active_vessels,
  active_vessels_strategy = c("latest", "closest"),
  minor_strata = NULL,
  progress_fn = NULL
)
```

Arguments

effort	effort data
effort_source	effort_source (register_interview / boat_counting)
landings	landings
active_days	active_days. Default is NULL (auto-generated)
active_vessels	active vessels
active_vessels_strategy	The strategy to associate the active vessels to the effort based on time. Active vessels period does not match necessarily the periods of data (effort, landings), and can be reported either by year or by year/month. This parameter let decide which methodology should be used to select the active vessels based on time. It can be either "latest" (taking the latest period), "closest" ie the closest active vessels in time, after or before the data period. In case 2 periods before/after are equally closer, the latest in time before the data period will be taken.
minor_strata	minor_strata. Default is NULL
progress_fn	a progress function with args (i, n, label). Default is NULL

```
compute_effort_estimate_with_IRD_Pechart
```

Computes nominal effort estimate

Description

TBD

Usage

```
compute_effort_estimate_with_IRD_Pechart(
  effort,
  effort_source = c("household_interview"),
  landings,
  active_days,
  census_typology,
  minor_stratum = NULL,
  progress_fn = NULL
)
```

Author(s)

Johanna Herfaut

compute_report	<i>compute_report</i>
----------------	-----------------------

Description

compute_report

Usage

```
compute_report(  
  effort,  
  effort_source = c("fisher_interview", "boat_counting", "household_interview"),  
  active_vessels = NULL,  
  active_vessels_strategy = NULL,  
  active_days,  
  landings,  
  minor_strata = NULL,  
  validate = FALSE,  
  progress_fn = NULL  
)
```

Arguments

effort	effort
effort_source	effort_source
active_vessels	active_vessels
active_vessels_strategy	active_vessels_strategy
active_days	active_days
landings	landings
minor_strata	minor_strata. Default is NULL
validate	validate
progress_fn	a progress function with args (label, p). Default is NULL

Value

the result of Artfish

compute_sui	<i>Computes sui</i>
-------------	---------------------

Description

Compute the sufficient uniformity index for effort and catch corresponding to the uniformity of sampling over the sampled days.

Usage

```
compute_sui(effort, landings, minor_strata = NULL)
```

Arguments

effort	effort
landings	landings
minor_strata	minor_strata. Default is NULL

Value

a [tibble](#) object giving the different accuracy by strata

create_artfish_template	<i>Create an empty artfishr data template</i>
-------------------------	---

Description

Generates an empty data frame following one of the artfishr format specifications.

Usage

```
create_artfish_template(
  format = c("artfish_A_active_vessels", "artfish_B1_effort", "artfish_B2_effort",
    "artfish_C_active_days", "artfish_D_landings"),
  include_meta = FALSE,
  save_as = NULL
)
```

Arguments

format	Character string indicating which template to create.
include_meta	Logical; if TRUE, include schema metadata.
save_as	Optional file path to save the template as CSV.

Value

A zero-row data frame matching the schema definition.

See Also

[decode_artfish_schema()]

create_artfish_templates

Wrapper functions for creating ArtFishR templates

Description

Family of functions to create empty templates for artfishr dataset types.

Usage

```
create_active_vessels_template(include_meta = FALSE, save_as = NULL)

create_active_days_template(include_meta = FALSE, save_as = NULL)

create_landings_template(include_meta = FALSE, save_as = NULL)

create_effort_template(
  effort_source = c("boat_counting", "fisher_interview"),
  include_meta = FALSE,
  save_as = NULL
)
```

Details

These functions are convenience wrappers around [create_artfish_template()].

See Also

[create_artfish_template()], [decode_artfish_schema()]

decode_artfish_schema	<i>Decode an artfishr JSON schema</i>
-----------------------	---------------------------------------

Description

Reads a JSON format specification (as used in inst/extdata/format_specs) and returns an empty data frame with the appropriate column names.

Usage

```
decode_artfish_schema(json_path, include_meta = FALSE)
```

Arguments

- json_path Path to the JSON specification file.
- include_meta Logical; if TRUE, attaches metadata (e.g. name, title, urn) as attributes to each column. Default is FALSE.

Value

A data.frame with zero rows and the columns defined in the schema.

ERROR	<i>util ERROR logger</i>
-------	--------------------------

Description

ERROR logger

Usage

```
ERROR(txt, ...)
```

Arguments

- txt a character vector of format strings, each of up to 8192 bytes.
- ... any values to be passed into txt. See [sprintf](#)

generate_active_days *Generates a [tibble](#) of active days.*

Description

Function that generates a table of active days for all periods. The unique list of fishing units are inherited from available tables (active_vessels, effort, landings). In the same way, the list of minor strata values will be inherited based on the minor_strata columns available in data.

Usage

```
generate_active_days(
  effort,
  effort_source = c("fisher_interview", "boat_counting", "household_interview"),
  landings,
  active_vessels = NULL,
  minor_strata = NULL
)
```

Arguments

effort	effort table
effort_source	effort source
landings	landings table
active_vessels	active vessels table
minor_strata	minor strata. Default is NULL

Value

an object of class [tibble](#) give active days

generate_active_days_by_period
Generates a [tibble](#) of active days.

Description

Function that generates a table of active days by year/month. The unique list of fishing units are inherited from available tables (effort, landings + eventually active_vessels). In the same way, the list of minor strata values will be inherited based on the minor_strata columns available in data.

Usage

```
generate_active_days_by_period(  
  year,  
  month,  
  effort,  
  effort_source = c("fisher_interview", "boat_counting", "household_interview"),  
  landings,  
  active_vessels = NULL,  
  minor_strata = NULL  
)
```

Arguments

- year year
- month month
- effort effort table
- effort_source effort source
- landings landings table
- active_vessels active vessels table
- minor_strata minor strata. Default is NULL

Value

an object of class [tibble](#) give active days

get_fdi_terms	Select get_fdi_terms vessels
---------------	------------------------------

Description

Get FDI terms for Artfish inputs/outputs

Usage

```
get_fdi_terms()
```

get_vrule_validators	Get data validators powered by vrule
----------------------	---

Description

Get data validators powered by **vrule**

Usage

get_vrule_validators()

Value

the list of internal data validators as objects of class [format_spec](#)

INFO	util <i>INFO</i> logger
------	-------------------------

Description

INFO logger

Usage

INFO(txt, ...)

Arguments

- txt a character vector of format strings, each of up to 8192 bytes.
- ... any values to be passed into txt. See [sprintf](#)

join_guess_by	<i>join_guess_by</i>
---------------	----------------------

Description

Gives the common column names between two tables. Util to be used in **dplyr** *_join calls by argument to avoid explicit messages triggered by **dplyr** when guessing the columns (in case by is not specified) param x a [data.frame](#) or [tibble](#) param y a [data.frame](#) or [tibble](#)

Usage

join_guess_by(x, y)

logger	<i>util generic logger</i>
--------	----------------------------

Description

Generic logger

Usage

```
logger(type = c("INFO", "WARN", "ERROR"), txt, ...)
```

Arguments

type	type either INFO, WARN or DEBUG
txt	a character vector of format strings, each of up to 8192 bytes.
...	any values to be passed into txt. See sprintf

run_artfishr_shiny	<i>Runs a Artfishr Shiny application</i>
--------------------	--

Description

Runs a Artfishr Shiny application

Usage

```
run_artfish_shiny(...)
```

sample_size_plot	<i>Produces a plot for the sample size</i>
------------------	--

Description

Produces a plot for the sample size

Usage

```
sample_size_plot(  
  sample = NULL,  
  population,  
  sample.mean,  
  sample.sd,  
  alpha = 0.1  
)
```

Arguments

sample	sample
population	population
sample.mean	mean
sample.sd	standard deviation
alpha	pvalue of t distribution

select_active_vessels *Select active vessels*

Description

Select active vessels for a selection of years

Active vessels:

The active_vessels can be either time-dependent, ie given by year/month, year only or be atemporal, mainly depending on how and at which frequency the vessels census/survey is operated in the country.

An additional argument called active_vessels_strategy controls how to select the active_vessels data in time, when this data is time-dependent. The latest strategy will select the latest data acquired in time, while the closest will select the closest data acquired in time, which could be data acquired after the effort data considered.

As example, a vessel census is performed each five years, and data is available for 2007 and 2012. Effort data for 2011 needs to be computed. Which active_vessels data should be used? In case of a latest strategy, in 2011, the latest data available is 2007, so this one will be used. In the case of a closest strategy, in 2011, we used the closest vessel data (ie 2012), assuming it better characterizes the fleet engagement for the year considered.

Usage

```
select_active_vessels(
  periods,
  active_vessels,
  active_vessels_strategy = c("latest", "closest")
)
```

Arguments

periods periods, list of unique years or year/month periods

active_vessels active vessels object of class [tibble](#)

active_vessels_strategy

The strategy to associate the active vessels to the effort based on time. Active vessels period does not match necessarily the periods of data (effort, landings), and can be reported either by year or by year/month. This parameter let decide which methodology should be used to select the active vessels based on time.

It can be either "latest" (taking the latest period), "closest" ie the closest active vessels in time, after or before the data period. In case 2 periods before/after are equally closer, the latest in time before the data period will be taken.

Value

a series of active vessels selection for the years considered (added as ref_period)

set_translation_language
<i>Set translation language</i>

Description

set_translation_language

Usage

set_translation_language(lang)

Arguments

lang lang

set_vrule_validators	<i>Set data validators powered by vrule</i>
----------------------	--

Description

Set data validators powered by **vrule**

Usage

set_vrule_validators()

translator	<i>Get translator</i>
------------	-----------------------

Description

Get translator

Usage

translator()

Value

the translator, object of class [Translator](#)

unif_index	<i>unif_index</i>
------------	-------------------

Description

unif_index

Usage

unif_index(days)

Arguments

days days

validate_artfish_datasets	<i>Wrapper functions for validating artfishr input data</i>
---------------------------	---

Description

Family of functions to validate empty input dataset.

Details

These functions are convenience wrappers around [validate_input_datasets()].

See Also

[validate_input_datasets()]

validate_input_datasets
<i>Validate input datasets</i>

Description

Validate input datasets

Usage

```
validate_input_datasets(  
  active_vessels,  
  effort,  
  effort_source = c("fisher_interview", "boat_counting"),  
  active_days = NULL,  
  landings  
)
```

Arguments

active_vessels	active vessels
effort	effort
effort_source	effort source whether it's derived from -B1- (fishers interviews) or -B2- (boat counting)
active_days	active days
landings	landings

WARN	<i>util WARN logger</i>
------	-------------------------

Description

WARN logger

Usage

```
WARN(txt, ...)
```

Arguments

txt	a character vector of format strings, each of up to 8192 bytes.
...	any values to be passed into txt. See sprintf

Index

accuracy_plot, [3](#)
artfish_accuracy, [3](#)
artfish_estimates, [4](#)
artfish_estimates_by_fleet_segment, [4](#)
artfish_flouca_by_period, [5](#)
artfish_new, [6](#)
artfish_new_by_period, [7](#)
artfish_shiny_accuracy_server, [7](#)
artfish_shiny_accuracy_ui, [8](#)
artfish_shiny_computation_server, [8](#)
artfish_shiny_fishing_unit_server, [9](#)
artfish_shiny_fishing_unit_ui, [10](#)
artfish_shiny_overview_server, [11](#)
artfish_shiny_overview_ui, [12](#)
artfish_shiny_report_server, [12](#)
artfish_shiny_report_ui, [13](#)
artfish_shiny_server
 ([artfishr_shiny_server](#)), [17](#)
artfish_shiny_species_server, [14](#)
artfish_shiny_species_ui, [15](#)
artfish_shiny_ui ([artfishr_shiny_ui](#)), [17](#)
artfish_shiny_welcome_server, [15](#)
artfish_shiny_welcome_ui, [16](#)
artfishr, [16](#)
artfishr-package ([artfishr](#)), [16](#)
artfishr_shiny_server, [17](#)
artfishr_shiny_ui, [17](#)

complete_active_days, [17](#)
compute_accuracy, [18](#)
compute_catch_estimate, [18](#), [19](#)
compute_catch_estimates_by_species, [19](#)
compute_cpue, [20](#)
compute_effort_activity_coefficient,
 [20](#), [22](#)
compute_effort_estimate, [19](#), [21](#)
compute_effort_estimate_with_FAO_Artfish,
 [22](#)
compute_effort_estimate_with_IRD_Pechart,
 [23](#)

compute_report, [24](#)
compute_sui, [25](#)
create_active_days_template
 ([create_artfish_templates](#)), [26](#)
create_active_vessels_template
 ([create_artfish_templates](#)), [26](#)
create_artfish_template, [25](#)
create_artfish_templates, [26](#)
create_effort_template
 ([create_artfish_templates](#)), [26](#)
create_landings_template
 ([create_artfish_templates](#)), [26](#)

data.frame, [30](#)
decode_artfish_schema, [27](#)

ERROR, [27](#)

format_spec, [30](#)

generate_active_days, [28](#)
generate_active_days_by_period, [28](#)
get_fdi_terms, [29](#)
get_vrule_validators, [30](#)

INFO, [30](#)

join_guess_by, [30](#)

logger, [31](#)

run_artfish_shiny ([run_artfishr_shiny](#)),
 [31](#)
run_artfishr_shiny, [31](#)

sample_size_plot, [31](#)
select_active_vessels, [32](#)
set_translation_language, [33](#)
set_vrule_validators, [33](#)
sprintf, [27](#), [30](#), [31](#), [35](#)
tibble, [17–21](#), [25](#), [28–30](#), [32](#)

Translator, [34](#)

translator, [34](#)

unif_index, [34](#)

validate_artfish_datasets, [34](#)

validate_input_datasets, [35](#)

WARN, [35](#)