

Package ‘TSGenerator’

October 21, 2026

Title 'Copernicus' HR-VPP Vegetation Time-Series Processing

Version 2.0.0

Description Provides tools for acquiring, extracting, quality-assessing, and analysing vegetation time series derived from 'Copernicus' High Resolution Vegetation Phenology and Productivity (HR-VPP) products. Supports Seasonal Trajectories (ST) and Vegetation Phenology and Productivity (VPP) products through native R access to 'WEkEO', geospatial processing, product-quality assessment, temporal completeness analysis, missingness modelling, and integrated graphical workflows. Selected functions from 'TSGenerator' 1.x are retained to support migration and legacy workflows.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

URL <https://github.com/avalero92/TSGenerator>

BugReports <https://github.com/avalero92/TSGenerator/issues>

Imports hdar, jsonlite, stringr, utils, ggplot2, mgcv, scales, shiny, shinydashboard, shinyjs, shinycssloaders, DT, readr, terra

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

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Alexey Valero-Jorge [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-5993-7346>>),
M. Auxiliadora Casterad Seral [aut] (ORCID:
<<https://orcid.org/0000-0003-4458-6966>>),
Jose-Tomas Alcala Nalvaiz [aut] (ORCID:
<<https://orcid.org/0000-0001-7549-8825>>)

Maintainer Alexey Valero-Jorge <avalero@cita-aragon.es>

Repository CRAN

Date/Publication 2026-10-21 14:50:19 UTC

Contents

assess_ts_quality	3
check_wekeo	3
check_wekeo_integration	4
classify_quality	5
count.NA	6
count_missing	6
Download.HRVPP	7
Download.STPPI	8
Download.VI	8
download_st	9
download_vpp	10
extract_dates_from_tiff_files	11
extract_ts	11
extract_vpp	12
GAM.missing	13
GAM.plot	14
general.Quality	15
geospatial_plan	15
get.Clean.IV	16
get.Series.mean	17
get.Series.median	17
get.Series.VPP	18
get.Stack	18
hda_auth_check	19
hda_client	19
hda_datasets	20
hda_download	21
hda_query_template	21
hda_search	22
impute_ts	23
mask_quality	23
model_missingness	24
plot_missingness	25
QFLAG2.Mask	25
quality.Series	26
quality_info	26
renames.image.IV	27
renames.image.VPP	28
runTSapp	28
summarize_missingness	29
summarize_quality	30
temporal_plan	30
TsImpute	31

assess_ts_quality	<i>Assess time-series completeness quality</i>
-------------------	--

Description

Classifies time-series completeness from the proportion of expected dates containing usable values. This metric is separate from Copernicus ST/VPP QFLAG product quality.

Usage

```
assess_ts_quality(
  data, id_col = "ID", date_col = "Date", value_col = "Value",
  expected_step = NULL, window_days = NULL,
  thresholds = c(high = 0.90, medium = 0.75, low = 0.50),
  start = NULL, end = NULL
)
```

Arguments

data	Input time-series data frame.
id_col, date_col, value_col	Column names identifying the series, observation date, and value.
expected_step	Expected temporal step in days; if NULL, it is inferred when possible.
window_days	Optional assessment-window length in days.
thresholds	Named completeness thresholds used to assign quality classes.
start, end	Optional start and end dates restricting the assessment period.

Value

An object of class `tsg_ts_quality` (also a `data.frame`) with one row per series and assessment window. It contains the completeness columns documented in `summarize_missingness()`, a numeric window index, and an ordered factor `Quality` with levels `Very low`, `Low`, `Medium`, and `High`, based on `value_completeness`. Attributes `thresholds` and `window_days` record the assessment settings.

check_wekeo	<i>Check the TSGenerator WEkEO acquisition environment</i>
-------------	--

Description

Performs non-destructive checks of the native R acquisition stack. Offline checks are the default.

Usage

```
check_wekeo(client = NULL, online = FALSE)
```

Arguments

client	Optional client created by <code>hda_client()</code> .
online	Logical; perform live WEkEO authentication and schema checks.

Value

A data frame of diagnostic checks with class `tsg_wekeo_check`.

See Also

[hda_client](#), [download_st](#), [download_vpp](#)

check_wekeo_integration

Run a live WEkEO integration smoke test

Description

Performs authentication, live schema discovery, small ST/VPP searches and, optionally, download/readability checks.

Usage

```
check_wekeo_integration(
  client = NULL,
  st_tile = "32TPS",
  vpp_tile = "30TXR",
  st_start = "2018-03-01",
  st_end = "2018-03-11",
  vpp_start = "2018-01-01",
  vpp_end = "2018-12-31",
  vpp_product = "TPROD",
  vpp_season = "s1",
  download = FALSE,
  output_dir = NULL,
  quiet = FALSE
)
```

Arguments

client	Optional client created by <code>hda_client()</code> .
st_tile	Sentinel-2 MGRS tile used for the ST smoke test.
vpp_tile	Sentinel-2 MGRS tile used for the VPP smoke test.
st_start, st_end	Small ST date range.

vpp_start, vpp_end VPP date range.

vpp_product VPP parameter used for the smoke test.

vpp_season VPP season ("s1" or "s2").

download Logical. Search only by default; optionally download matching files.

output_dir Explicit destination required when downloading; no default directory is created.

quiet Logical; suppress progress messages.

Value

A list of class `tsg_wekeo_integration`.

<code>classify_quality</code>	<i>Decode HR-VPP quality codes</i>
-------------------------------	------------------------------------

Description

Decodes HR-VPP quality codes while retaining their categorical meaning.

Usage

```
classify_quality(values, type = c("ST", "VPP"))
```

Arguments

values Numeric or integer quality codes.

type Quality-flag family: "ST" or "VPP".

Value

A data frame with code, class, rank, default acceptance, and definition.

See Also

[quality_info](#), [summarize_quality](#), [mask_quality](#)

count.NA	<i>Count missing observations (legacy name)</i>
----------	---

Description

Retired TSGenerator 1.x compatibility wrapper. Calling this function produces a deprecation warning followed by a migration error. Use `summarize_missingness()` instead.

Usage

```
count.NA(
  data, sos, maxd, eos, doy_col = "DOY", year_col = "Year",
  na_col = "NDVI", fid_col = "ID"
)
```

Arguments

data	Input data frame.
sos, maxd, eos	Phenological day-of-year boundaries.
doy_col, year_col, na_col, fid_col	Column names for day of year, year, assessed value, and feature identifier.

Value

No return value. This retired compatibility function signals an error with migration guidance; use `summarize_missingness()` instead.

count_missing	<i>Legacy phenology-specific missing-count interface</i>
---------------	--

Description

Retired TSGenerator 1.x interface. The former phenology-specific missing-count workflow is no longer executed.

Usage

```
count_missing(
  data, sos, maxd, eos, doy_col = "DOY", year_col = "Year",
  na_col = "NDVI", fid_col = "ID"
)
```

Arguments

data A data frame containing the time series.

sos, maxd, eos Day-of-year thresholds.

doy_col, year_col, na_col, fid_col
 Column names.

Value

No return value. This retired function signals an error with migration guidance; use `summarize_missingness()` for missingness counts and `assess_ts_quality()` for completeness classes.

Download.HRVPP	<i>Legacy wrapper for VPP download</i>
----------------	--

Description

Deprecated TSGenerator 1.x compatibility wrapper. It delegates to `download_vpp()` using the native R/hdar backend and no longer requires Python.

Usage

```
Download.HRVPP(
  user = NULL, password = NULL, dataset_id = .VPP_DATASET_ID,
  productType = "SOSD", productGroupId = "s1", tileId = NULL,
  start, end, bbox = NULL, download_path = NULL
)
```

Arguments

user, password Optional legacy WEkEO credentials. Current workflows should use the native authentication configuration.

dataset_id Legacy WEkEO dataset identifier.

productType VPP product type.

productGroupId VPP season/product group identifier.

tileId Optional MGRS tile identifier.

start, end Start and end dates of the search interval.

bbox Optional bounding box.

download_path Explicit destination required for downloads; no default directory is created.

Value

A `tsg_vpp_download` object.

Download.STPPI	<i>Legacy wrapper for Seasonal Trajectories download</i>
----------------	--

Description

Deprecated TSGenerator 1.x compatibility wrapper. It delegates to `download_st()` using the native R/hdar backend and no longer requires Python.

Usage

```
Download.STPPI(
  user = NULL, password = NULL, dataset_id = .ST_DATASET_ID,
  productType = "PPI", platformSerialIdentifier = "S2A, S2B",
  tileId = NULL, start, end, bbox = NULL, download_path = NULL
)
```

Arguments

<code>user, password</code>	Optional legacy WEkEO credentials. Current workflows should use the native authentication configuration.
<code>dataset_id</code>	Legacy WEkEO dataset identifier.
<code>productType</code>	Seasonal Trajectories product type.
<code>platformSerialIdentifier</code>	Sentinel-2 platform identifier(s).
<code>tileId</code>	Optional MGRS tile identifier.
<code>start, end</code>	Start and end dates of the search interval.
<code>bbox</code>	Optional bounding box.
<code>download_path</code>	Explicit destination required for downloads; no default directory is created.

Value

A `tsg_st_download` object.

Download.VI	<i>Discontinued legacy VI downloader</i>
-------------	--

Description

The HR-VPP VI products targeted by the TSGenerator 1.x downloader are no longer disseminated. This compatibility function stops with migration guidance.

Usage

```
Download.VI(...)
```

Arguments

... Ignored legacy arguments, retained only for compatibility.

Value

No value; the function raises an error.

download_st	<i>Download Copernicus HR-VPP Seasonal Trajectories</i>
-------------	---

Description

Searches and optionally downloads HR-VPP Seasonal Trajectories through WEkEO HDA V2 using the native R **hdar** backend.

Usage

```
download_st(start, end, output_dir = NULL, product = "PPI",
  tile_id = NULL, bbox = NULL, platform = "S2A, S2B", resolution = 10L,
  product_version = NULL, client = NULL, download = TRUE, limit = NULL,
  overwrite = FALSE, prompt = interactive(), quiet = FALSE, retries = 2L,
  backoff = 2)
```

Arguments

start, end	Start and end dates of the WEkEO search interval.
output_dir	Explicit destination required when download = TRUE; no default directory is created. Ignored in preview mode.
product	Seasonal Trajectories product type.
tile_id	Optional MGRS tile identifier.
bbox	Optional bounding box used to spatially restrict the query.
platform	Sentinel-2 platform identifier(s).
resolution	Requested spatial resolution in metres.
product_version	Optional HR-VPP product version.
client	Optional preconfigured native HDA client.
download	Logical; download results when TRUE, otherwise return a preview.
limit	Optional maximum number of search results.
overwrite	Logical; overwrite existing files.
prompt	Logical; allow interactive download confirmation.
quiet	Logical; suppress progress messages when possible.
retries	Number of download retries.
backoff	Backoff interval used between retries.

Value

An object of class `tsg_st_download`.

<code>download_vpp</code>	<i>Download Copernicus HR-VPP VPP parameters</i>
---------------------------	--

Description

Searches and optionally downloads HR-VPP phenology and productivity parameters through WEkEO HDA V2 using the native R **hdar** backend.

Usage

```
download_vpp(start, end, output_dir = NULL,
  product = c("SOSD", "MAXD", "EOSD", "LENGTH"), season = "s1",
  tile_id = NULL, bbox = NULL, platform = "S2A, S2B", product_version = NULL,
  client = NULL, download = TRUE, limit = NULL, overwrite = FALSE,
  prompt = interactive(), quiet = FALSE, retries = 2L, backoff = 2)
```

Arguments

<code>start, end</code>	Start and end dates of the WEkEO search interval.
<code>output_dir</code>	Explicit destination required when <code>download = TRUE</code> ; no default directory is created. Ignored in preview mode.
<code>product</code>	VPP parameter or parameters to query.
<code>season</code>	VPP season identifier.
<code>tile_id</code>	Optional MGRS tile identifier.
<code>bbox</code>	Optional bounding box used to spatially restrict the query.
<code>platform</code>	Sentinel-2 platform identifier(s).
<code>product_version</code>	Optional HR-VPP product version.
<code>client</code>	Optional preconfigured native HDA client.
<code>download</code>	Logical; download results when <code>TRUE</code> , otherwise return a preview.
<code>limit</code>	Optional maximum number of search results.
<code>overwrite</code>	Logical; overwrite existing files.
<code>prompt</code>	Logical; allow interactive download confirmation.
<code>quiet</code>	Logical; suppress progress messages when possible.
<code>retries</code>	Number of download retries.
<code>backoff</code>	Backoff interval used between retries.

Value

An object of class `tsg_vpp_download`.

extract_dates_from_tiff_files	
	<i>extract_dates_from_tiff_files</i>

Description

extract_dates_from_tiff_files
extract_dates_from_tiff_files
extract_dates_from_tiff_files

Usage

extract_dates_from_tiff_files(tiff_files)

extract_dates_from_tiff_files(tiff_files)

extract_dates_from_tiff_files(tiff_files)

Arguments

tiff_files List of files in .tif format

Value

The names of each of the files in the formats %Y-%m-%d or YYYYYYYMMDD
The names of each of the files in the formats %Y-%m-%d or YYYYYYYMMDD
The names of each of the files in the formats %Y-%m-%d or YYYYYYYMMDD

extract_ts	<i>Extract polygon-level raster time series</i>
------------	---

Description

Terra-first replacement for the duplicated TSGenerator 1.x mean and median extraction workflows.

Usage

extract_ts(x, polygons, id_col = NULL, fun = "median", dates = NULL,
transform = TRUE, scale_factor = 1, offset = 0, na.rm = TRUE,
exact = FALSE, touches = FALSE)

Arguments

x	A terra SpatRaster, TIFF path, directory, or vector of TIFF paths.
polygons	An sf/sfc object or terra SpatVector containing polygons.
id_col	Optional unique polygon ID column.
fun	Summary function; mean, median, min, max, sum, or a function.
dates	Optional vector with one date per raster layer.
transform	Transform polygons to raster CRS when needed.
scale_factor	Numeric divisor applied after extraction.
offset	Numeric offset added after scaling.
na.rm	Remove missing raster values during summarisation.
exact	Use exact polygon-cell fractions when supported.
touches	Include cells touched by polygon boundaries.

Value

A tidy data frame of class `tsg_time_series` with ID, Date, Layer and Value columns.

extract_vpp	<i>Extract HR-VPP phenology and productivity parameters</i>
-------------	---

Description

Extracts polygon statistics from Copernicus HR-VPP VPP rasters using product-aware decoding, scaling, NoData handling and season metadata.

Usage

```
extract_vpp(x, polygons, id_col = NULL, product = NULL, season = NULL,
  year = NULL, fun = "median", transform = TRUE, na.rm = TRUE,
  exact = FALSE, touches = FALSE)
```

Arguments

x	VPP raster input, file paths, or directory accepted by TSGenerator.
polygons	Polygon sf/sfc object or terra::SpatVector.
id_col	Optional polygon identifier column.
product	Optional VPP product filter.
season	Optional season filter.
year	Optional year filter.
fun	Polygon summary function.
transform	Logical; transform polygons to raster CRS when required.
na.rm	Logical; remove missing raster values in summaries.
exact	Logical passed to the extraction backend for exact polygon extraction.
touches	Logical controlling inclusion of touched cells.

Value

A data frame of class `tsg_vpp`. Date products populate `Date`; numeric products populate `Value`.

GAM.missing	<i>GAM.missing</i>
-------------	--------------------

Description

GAM.missing

Usage

```
GAM.missing(data, year_col, doy_col, missing_col)
```

Arguments

- | | |
|--------------------------|---|
| <code>data</code> | Data frame containing the missingness time-series information. |
| <code>year_col</code> | Name of the column containing years. |
| <code>doy_col</code> | Name of the column containing day-of-year values. |
| <code>missing_col</code> | Name of the column containing the missingness/value indicator used by the legacy wrapper. |

Value

A list object where the prediction results and the GAM model data are stored.

Examples

```
# Model fitting is illustrated with synthetic observations.

example_data <- expand.grid(
  Year = 2020:2021, DOY = seq(10, 350, by = 10), replicate = 1:4
)
example_data$NDVI_median <- 0.5
example_data$NDVI_median[
  (seq_len(nrow(example_data)) %% 7L) < 2L
] <- NA_real_
result <- suppressWarnings(GAM.missing(
  example_data, year_col = "Year", doy_col = "DOY",
  missing_col = "NDVI_median"
))
```

GAM.plot

*GAM.plot***Description**

GAM.plot

Usage

```
GAM.plot(
  data,
  doy_col = "DOY",
  year_col = "Year",
  proporcion_col = "Proporcion_Missing",
  prediccion_col = "Predicted",
  sos = NULL,
  eos = NULL,
  max_doy = NULL
)
```

Arguments

data	A DataFrame that contains the data to be plotted.
doy_col	Name of the column representing the Julian Day (DOY).
year_col	Name of the column representing the Year.
proporcion_col	Name of the column representing the Proportion of Missing Data.
prediccion_col	Name of the column representing the Prediction.
sos	Optional value for the SOS line.
eos	Optional value for the EOS line.
max_doy	Optional value for the MAX line.

Value

Object of type ggplot that shows the results of the prediction obtained with the GAM.missing function.

Examples

```
proportions <- data.frame(
  Year = rep(2020L, 5), DOY = c(20, 50, 80, 110, 140),
  Proporcion_Missing = c(0.1, 0.2, 0.3, 0.2, 0.1),
  Predicted = c(0.12, 0.18, 0.25, 0.18, 0.12)
)
p <- suppressWarnings(GAM.plot(proportions))
inherits(p, "ggplot")
```

general.Quality	<i>general.Quality</i>
-----------------	------------------------

Description

general.Quality

Usage

```
general.Quality(df, period_col, observations_col)
```

Arguments

- df Data base generated from the use of the function quality.Series
- period_col Column where the periods are stored (every 91 days)
- observations_col Columna donde se almacenan el numero de observaciones por periodo y

Value

A list of files that stores a new data frame and an interactive graphic.

geospatial_plan	<i>Inspect inputs for the TSGenerator 2.0 geospatial engine</i>
-----------------	---

Description

Validates raster/vector geometry, CRS, IDs, and scaling before terra-based extraction.

Usage

```
geospatial_plan(  
  x, polygons, id_col = NULL, transform = TRUE,  
  scale_factor = 1, offset = 0  
)
```

Arguments

- x A terra::SpatRaster, TIFF path, directory, or vector of TIFF paths.
- polygons An sf/sfc object or terra::SpatVector of polygons.
- id_col Optional unique polygon ID column.
- transform Logical; transform polygons to the raster CRS when required.
- scale_factor Numeric divisor applied after extraction.
- offset Numeric offset applied after scaling.

Value

An object of class `tsg_geospatial_plan`; no extraction is performed.

See Also

[extract_ts](#), [extract_vpp](#)

`get.Clean.IV`

Legacy VI cleaning workflow

Description

`get.Clean.IV()` belongs to the discontinued VI/QFLAG2 workflow. It is retained only as a migration guard. Use `mask_quality()` for current ST/VPP quality handling.

Usage

```
get.Clean.IV(...)
```

Arguments

... Legacy arguments. They are accepted only so existing calls fail with explicit migration guidance.

Value

This function does not return a value; it stops with migration guidance.

Examples

```
## Not run:
get.Clean.IV()

## End(Not run)
```

get.Series.mean	<i>Legacy mean time-series extractor</i>
-----------------	--

Description

Deprecated compatibility wrapper around `extract_ts(..., fun = "mean")`.

Usage

```
get.Series.mean(pathRaster = NULL, shapefile = NULL, factorR = NULL)
```

Arguments

pathRaster	Directory containing TIFF rasters.
shapefile	Polygon sf/sfc or terra SpatVector.
factorR	Legacy numeric scale divisor.

Value

A data frame with legacy columns ID, Date, and Mean.

get.Series.median	<i>Legacy median time-series extractor</i>
-------------------	--

Description

Deprecated compatibility wrapper around `extract_ts(..., fun = "median")`.

Usage

```
get.Series.median(pathRaster = NULL, shapefile = NULL, factorR = NULL)
```

Arguments

pathRaster	Directory containing TIFF rasters.
shapefile	Polygon sf/sfc or terra SpatVector.
factorR	Legacy numeric scale divisor.

Value

A data frame with legacy columns ID, Date, and Mean. The historical column name is preserved.

get.Series.VPP	<i>Retired legacy VPP extractor</i>
----------------	-------------------------------------

Description

Retired because one generic scale factor is not scientifically valid across VPP products. Use `extract_vpp()`.

Usage

```
get.Series.VPP(pathRaster = NULL, shapefile = NULL, factorR = NULL)
```

Arguments

pathRaster	Legacy raster directory.
shapefile	Legacy polygon object.
factorR	Legacy scale factor.

Value

No return value. This retired function always signals an error explaining the migration to `extract_vpp()`.

get.Stack	<i>Legacy VI/QFLAG stack workflow</i>
-----------	---------------------------------------

Description

`get.Stack()` belongs to the discontinued VI/QFLAG2 workflow. It is retained only as a migration guard; current ST and QFLAG products should remain separate.

Usage

```
get.Stack(...)
```

Arguments

...	Legacy arguments. They are accepted only so existing calls fail with explicit migration guidance.
-----	---

Value

This function does not return a value; it stops with migration guidance.

Examples

```
## Not run:
get.Stack()

## End(Not run)
```

hda_auth_check	<i>Test native WEkEO authentication</i>
----------------	---

Description

Checks that the configured WEkEO credentials can obtain an HDA authentication token.

Usage

```
hda_auth_check(client = NULL, ...)
```

Arguments

client	Optional client created by <code>hda_client()</code> .
...	Arguments passed to <code>hda_client()</code> when <code>client</code> is omitted.

Value

Invisibly returns TRUE after a token is obtained.

See Also

[hda_client](#), [check_wekeo](#)

hda_client	<i>Create a native WEkEO HDA client</i>
------------	---

Description

Creates the native R client used by TSGenerator to access WEkEO HDA. If credentials are omitted, **hda** uses `'~/hdarc'`.

Usage

```
hda_client(username = NULL, password = NULL, save_credentials = FALSE)
```

Arguments

username Optional WEkEO username.
password Optional WEkEO password.
save_credentials Logical; allow **hdar** to save credentials in '~/.hdarc'.

Value

An authenticated `hdar::Client` object.

See Also

[hda_auth_check](#), [check_wekeo](#), [download_st](#), [download_vpp](#)

hda_datasets	<i>Discover WEkEO datasets</i>
--------------	--------------------------------

Description

Queries the WEkEO catalogue through the native HDA client.

Usage

```
hda_datasets(pattern = NULL, client = NULL)
```

Arguments

pattern Optional text used to filter the WEkEO catalogue.
client Optional authenticated HDA client.

Value

Dataset records returned by **hdar**.

See Also

[hda_query_template](#)

hda_download	<i>Download HDA search results</i>
--------------	------------------------------------

Description

Downloads WEkEO HDA search results with controlled retry behaviour.

Usage

```
hda_download(  
  results, output_dir, force = FALSE, retries = 2L,  
  backoff = 2, prompt = interactive()  
)
```

Arguments

results	SearchResults returned by hda_search().
output_dir	Destination directory.
force	Logical; re-download existing files when supported.
retries	Number of retries after transient failures.
backoff	Initial retry delay in seconds.
prompt	Logical; allow hdaR to request confirmation.

Value

The value returned by the **hdaR** download method, invisibly.

See Also

[hda_search](#)

hda_query_template	<i>Generate an HDA V2 query template</i>
--------------------	--

Description

Retrieves the current queryable schema for a WEkEO dataset rather than hard-coding provider-specific query fields.

Usage

```
hda_query_template(dataset_id, client = NULL)
```

Arguments

- dataset_id WEkEO dataset identifier.
- client Optional authenticated HDA client.

Value

A query template returned by **hda**.

See Also

[hda_datasets](#), [hda_search](#)

hda_search	<i>Search WEkEO using an HDA V2 query</i>
------------	---

Description

Submits an HDA V2 search with controlled retry and exponential backoff.

Usage

```
hda_search(query, client = NULL, limit = NULL, retries = 2L, backoff = 2)
```

Arguments

- query HDA V2 query, normally based on `hda_query_template()`.
- client Optional authenticated HDA client.
- limit Optional maximum number of results.
- retries Number of retries after transient failures.
- backoff Initial retry delay in seconds.

Value

An **hda** SearchResults object.

See Also

[hda_download](#)

impute_ts	<i>Impute missing values in vegetation time series</i>
-----------	--

Description

Explicit, opt-in imputation with provenance and safeguards for processed Seasonal Trajectories.

Usage

```
impute_ts(data, id_col = "ID", date_col = "Date", value_col = "Value",
  method = c("linear", "kalman"), series_type = c("user", "raw", "st"),
  allow_processed = FALSE, complete_grid = FALSE, expected_step = NULL,
  max_gap = NULL, allow_edge = FALSE, output_col = NULL)
```

Arguments

data	Input time-series data frame.
id_col, date_col, value_col	Column names identifying the series, observation date, and value.
method	Imputation method.
series_type	Series provenance: user, raw, or processed ST.
allow_processed	Logical; explicitly permit additional imputation of processed ST series.
complete_grid	Logical; complete the expected temporal grid before imputation.
expected_step	Expected temporal step in days.
max_gap	Optional maximum gap eligible for imputation.
allow_edge	Logical; permit filling gaps at series edges where supported.
output_col	Optional name for the imputed output column.

Value

A data frame of class `tsg_imputed_series`.

mask_quality	<i>Mask raster values using HR-VPP quality flags</i>
--------------	--

Description

Masks values according to explicit quality rules. Categorical QFLAG rasters are never resampled automatically.

Usage

```
mask_quality(
  x,
  quality,
  type = c("ST", "VPP"),
  min_quality = NULL,
  keep_codes = NULL
)
```

Arguments

x	Target terra::SpatRaster.
quality	Quality terra::SpatRaster with matching geometry.
type	Quality-flag family, either "ST" or "VPP".
min_quality	Optional minimum quality code.
keep_codes	Optional explicit quality codes to retain; overrides min_quality.

Value

A masked terra::SpatRaster.

See Also

[quality_info](#), [classify_quality](#)

model_missingness	<i>Model temporal missingness with a generalized additive model</i>
-------------------	---

Description

Aggregates missing/available counts and fits a binomial GAM. Year is treated as a factor by default.

Usage

```
model_missingness(
  data, year_col = "Year", doy_col = "DOY", value_col = "Value",
  k = NULL, year_effect = c("factor", "none"), cyclic = TRUE
)
```

Arguments

data	Data frame containing repeated observations.
year_col, doy_col, value_col	Column names.
k	Optional GAM basis dimension.
year_effect	Factor year effect or none.
cyclic	Use a cyclic DOY smooth.

Value

An object of class `tsg_missingness_model`.

<code>plot_missingness</code>	<i>Plot observed and modelled temporal missingness</i>
-------------------------------	--

Description

Plots observed missingness proportions and GAM predictions without printing automatically.

Usage

```
plot_missingness(x, sos = NULL, max_doy = NULL, eos = NULL, facet = TRUE)
```

Arguments

<code>x</code>	A <code>tsg_missingness_model</code> or compatible data frame.
<code>sos, max_doy, eos</code>	Optional DOY markers.
<code>facet</code>	Facet by year.

Value

A ggplot object.

<code>QFLAG2.Mask</code>	<i>Legacy VI QFLAG2 masking workflow</i>
--------------------------	--

Description

`QFLAG2.Mask()` targets the discontinued VI QFLAG2 product. It is retained only as a migration guard. Use `quality_info()`, `summarize_quality()`, `classify_quality()`, and `mask_quality()` for current ST/VPP quality products.

Usage

```
QFLAG2.Mask(...)
```

Arguments

<code>...</code>	Legacy arguments. They are accepted only so existing calls fail with explicit migration guidance.
------------------	---

Value

This function does not return a value; it stops with migration guidance.

Examples

```
## Not run:
QFLAG2.Mask()

## End(Not run)
```

quality.Series	<i>quality.Series</i>
----------------	-----------------------

Description

quality.Series

Usage

```
quality.Series(df, id_col, fecha_col, value_col)
```

Arguments

- df Data base
- id_col Column where the observation grid identifiers are located.
- fecha_col Column where the date is located (%yyyy%mm%dd)
- value_col Column from which the quality of the time series will be determined (estimating the number of observations).

Value

A list of files where a dataframe with the results of the quality assessment and an interactive graphic are stored.

quality_info	<i>HR-VPP quality-code definitions</i>
--------------	--

Description

Returns the TSGenerator definitions used to interpret current HR-VPP ST-QFLAG or VPP-QFLAG values.

Usage

```
quality_info(type = c("ST", "VPP"))
```

Arguments

- type Quality-flag family: "ST" or "VPP".

Value

A data frame describing the supported quality codes and classes.

See Also

[classify_quality](#), [summarize_quality](#), [mask_quality](#)

renames.image.IV	<i>Legacy VI file renaming helper</i>
------------------	---------------------------------------

Description

renames.image.IV() belongs to the discontinued VI workflow and is retained only as a migration guard.

Usage

```
renames.image.IV(...)
```

Arguments

...	Legacy arguments. They are accepted only so existing calls fail with explicit migration guidance.
-----	---

Value

This function does not return a value; it stops with migration guidance.

Examples

```
## Not run:  
renames.image.IV()  
  
## End(Not run)
```

renames.image.VPP	<i>renames.image.VPP</i>
-------------------	--------------------------

Description

Rename TIF files in a directory according to the year extracted from the name.

Usage

```
renames.image.VPP(...)
```

Arguments

... Legacy arguments.

Details

This function searches for TIF files in a specified directory and renames them using the year extracted from the filename. The new file name shall be in the format 'yyyy.tif'.

Value

Does not return a value, but renames the TIF files in the specified directory.

runTSapp	<i>Run the TSGenerator graphical interface</i>
----------	--

Description

Launches the unified TSGenerator 2.0 Shiny application. Selected 1.x/early-2.0 acquisition mini-apps remain temporarily accessible for compatibility, but new development is concentrated in the unified interface.

Usage

```
runTSapp(app = "TSGenerator2")
```

Arguments

app Character string. Application to launch. The default, "TSGenerator2", launches the unified 2.0 interface. Compatibility options are "DownloadSTPPI" and "DownloadVPP".

Details

The unified application is a graphical layer over the public TSGenerator 2.0 API. Scientific algorithms are not reimplemented in the Shiny server. The historical standalone GAM interface has been retired; missingness modelling remains available through `model_missingness()` and `plot_missingness()`.

Value

Invisibly returns the result of `shiny::runApp()`.

See Also

`model_missingness()`, `plot_missingness()`

Other applications: `runTSapp()`

Examples

```
## Not run:
TSGenerator::runTSapp()
TSGenerator::runTSapp("DownloadVPP")

## End(Not run)
```

`summarize_missingness` *Summarise missingness and temporal completeness*

Description

Separates absent expected dates from explicit NA values and reports temporal and usable-value completeness for each series.

Usage

```
summarize_missingness(
  data, id_col = "ID", date_col = "Date", value_col = "Value",
  expected_step = NULL, start = NULL, end = NULL
)
```

Arguments

<code>data</code>	Input time-series data frame.
<code>id_col</code> , <code>date_col</code> , <code>value_col</code>	Column names identifying the series, observation date, and value.
<code>expected_step</code>	Expected temporal step in days; if NULL, it is inferred when possible.
<code>start</code> , <code>end</code>	Optional start and end dates restricting the summary period.

Value

An object of class `tsg_missingness` (also a `data.frame`) with one row per series. Columns are the identifier (named by `id_col`), `start`, `end`, `expected_step_days`, `n_expected`, `n_observed_dates`, `n_absent_dates`, `n_value_na`, `n_usable`, `date_completeness`, `value_completeness`, and `missing_fraction`. Counts distinguish absent expected dates from observed dates with missing values. Fractions use `n_expected` as the denominator and are NA when that count is zero.

summarize_quality	<i>Summarize HR-VPP quality flags</i>
-------------------	---------------------------------------

Description

Summarizes the distribution of HR-VPP quality flags without masking observations.

Usage

```
summarize_quality(x, type = c("ST", "VPP"), quality_col = "QFLAG")
```

Arguments

x	Quality codes, a terra::SpatRaster, or a data frame.
type	Quality-flag family.
quality_col	Column containing codes when x is a data frame.

Value

A data frame with counts and percentages by documented quality code.

See Also

[quality_info](#), [classify_quality](#), [mask_quality](#)

temporal_plan	<i>Inspect and validate a vegetation time series</i>
---------------	--

Description

Validates IDs, dates, missing values, duplicate observations, and temporal regularity without modifying the series.

Usage

```
temporal_plan(
  data, id_col = "ID", date_col = "Date", value_col = "Value",
  expected_step = NULL, allow_duplicates = FALSE
)
```

Arguments

data	A data frame containing one or more time series.
id_col	Series identifier column.
date_col	Date column.
value_col	Value column.
expected_step	Optional expected temporal step in days.
allow_duplicates	Logical; allow duplicate ID-date observations.

Value

An object of class `tsg_temporal_plan` containing validated data and diagnostics.

See Also

[summarize_missingness](#), [assess_ts_quality](#), [impute_ts](#)

TsImpute	<i>TsImpute</i>
----------	-----------------

Description

TsImpute

Usage

TsImpute(data, group_col, value_col)

Arguments

data	data frame where VI time series containing missing data are stored.
group_col	column by which the grouping is to be performed (example: ID column)
value_col	column where the time series of the VI data is stored (example: NDVI)

Value

A new file similar to the original data frame but with a new column where the imputed data is stored.

Examples

```
# Kalman imputation requires the optional imputeTS package.
if (requireNamespace("imputeTS", quietly = TRUE)) {
  example_data <- data.frame(
    ID = rep("parcel_1", 12),
    Date = as.Date("2020-01-01") + 0:11,
    NDVI = c(0.2, 0.3, NA, 0.4, 0.5, NA, 0.6, 0.7, 0.8, 0.7, 0.6, 0.5)
  )
  completed <- suppressWarnings(TsImpute(
    example_data, group_col = "ID", value_col = "NDVI"
  ))
}
```

Index

- * **WEkEO acquisition**
 - check_wekeo, 3
 - hda_auth_check, 19
 - hda_client, 19
 - hda_datasets, 20
 - hda_download, 21
 - hda_query_template, 21
 - hda_search, 22
- * **applications**
 - runTSapp, 28
- * **geospatial core**
 - geospatial_plan, 15
- * **product quality**
 - classify_quality, 5
 - mask_quality, 23
 - quality_info, 26
 - summarize_quality, 30
- * **time-series analysis**
 - temporal_plan, 30

assess_ts_quality, 3, 31

check_wekeo, 3, 19, 20

check_wekeo_integration, 4

classify_quality, 5, 24, 27, 30

count.NA, 6

count_missing, 6

Download.HRVPP, 7

Download.STPPI, 8

Download.VI, 8

download_st, 4, 9, 20

download_vpp, 4, 10, 20

extract_dates_from_tiff_files, 11

extract_ts, 11, 16

extract_vpp, 12, 16

GAM.missing, 13

GAM.plot, 14

general.Quality, 15

geospatial_plan, 15

get.Clean.IV, 16

get.Series.mean, 17

get.Series.median, 17

get.Series.VPP, 18

get.Stack, 18

hda_auth_check, 19, 20

hda_client, 4, 19, 19

hda_datasets, 20, 22

hda_download, 21, 22

hda_query_template, 20, 21

hda_search, 21, 22, 22

impute_ts, 23, 31

mask_quality, 5, 23, 27, 30

model_missingness, 24, 29

model_missingness(), 28

plot_missingness, 25, 29

plot_missingness(), 28

QFLAG2.Mask, 25

quality.Series, 26

quality_info, 5, 24, 26, 30

renames.image.IV, 27

renames.image.VPP, 28

runTSapp, 28

summarize_missingness, 29, 31

summarize_quality, 5, 27, 30

temporal_plan, 30

TsImpute, 31