Package ‘SWMPrExtension’

March 17, 2022

Type Package

Title Functions for Analyzing and Plotting Estuary Monitoring Data

Version 2.2.0

Maintainer Dave Eslinger <dave.eslinger@noaa.gov>

Description Tools for performing routine analysis and plotting tasks with environmental data from the System Wide Monitoring Program of the National Estuarine Research Reserve System <http://cdmo.baruch.sc.edu/>. This package builds on the functionality of the ‘SWMPr’ package <https: //cran.r-project.org/package=SWMPr>, which is used to retrieve and organize the data. The combined set of tools address common challenges associated with continuous time series data for environmental decision making, and are intended for use in annual reporting activities.

References:

BugReports https://github.com/NOAA-OCM/SWMPrExtension/issues

License CC0

Encoding UTF-8

LazyData true

Depends R (>= 4.0), SWMPr

Imports EnvStats, RColorBrewer, broom, dplyr, flexible, ggimage, ggplot2, gghthemes, grDevices, lubridate, magrittr, methods, officer, purrr, rlang, scales, sf, stats, tidyselect, tidyr, utils

Suggests ggmap

RoxygenNote 7.1.2
NeedsCompilation no

Author Julie Padilla [aut, ctb],
Marcus Beck [ctb],
Kimberly Cressman [ctb],
Dave Eslinger [cre, ctb],
Bob Rudis [ctb],
Davis Vaughan [ctb]

Repository CRAN

Date/Publication 2022-03-16 23:20:02 UTC

R topics documented:

annual_range ................................................................. 3
assign_season ................................................................. 5
base_map ....................................................................... 6
cbm_spatial ................................................................. 8
counties_4269 ............................................................. 8
create_sk_flextable_list .................................................... 9
create_sk_national_ft_reserves ......................................... 10
create_sk_national_ft_results .......................................... 11
elkmnut ................................................................. 12
elkswm ............................................................... 13
elk_spatial ................................................................. 13
ft_col_names ............................................................. 14
generate_results_table .................................................... 15
generate_station_table .................................................... 15
geographic_unique_stations ............................................... 16
get_reserve ............................................................... 17
get_shp_name ............................................................. 18
get_sites ................................................................. 18
get_site_code ............................................................ 19
get_site_coordinates ..................................................... 20
historical_daily_range ................................................... 20
historical_range .......................................................... 22
import_local_nut .......................................................... 24
lm_p_labs ................................................................. 26
load_shp_file ............................................................ 27
national_sk_map .......................................................... 27
raw_boxplot .............................................................. 29
remove_inf_and_nan ....................................................... 30
reserve_locs ............................................................. 31
res_custom_map .......................................................... 32
res_custom_sk_map ........................................................ 34
res_local_map ............................................................ 36
res_national_map ........................................................ 38
res_sk_map ............................................................... 40
sampling_stations ......................................................... 42
annual_range

annual_range

Description
Assess variability within each season for a single year

Usage

annual_range(swmpr_in, ...)

## S3 method for class 'swmpr'

annual_range(
  swmpr_in,
  param = NULL,
  target_yr = NULL,
  criteria = NULL,
  free_y = FALSE,
  log_trans = FALSE,
  converted = FALSE,
  criteria_lab = "WQ Threshold",
  plot_title = FALSE,
  plot = TRUE,
  ...
)
Arguments

swmpr_in       input swmpr object
...
additional arguments passed to other methods. See assign_season
param         chr string of variable to plot
target_yr     numeric, the target year that should be compared against the historic range. If
target year is not specified then the dot will not be plotted.
criteria      numeric, a numeric criteria that will be plotted as a horizontal line
free_y         logical, should the y-axis be free? Defaults to FALSE. If FALSE, defaults to zero,
                unless negative values are present. If TRUE, y-axis limits are selected by ggplot
log_trans      logical, should y-axis be log? Defaults to FALSE
converted      logical, were the units converted from the original units used by CDMO? De-
               faults to FALSE. See y_labeler for details.
criteria_lab   chr, label for the threshold criteria defined in criteria. Defaults to "WQ
                Threshold"
plot_title     logical, should the station name be included as the plot title? Defaults to FALSE
plot           logical, should a plot be returned? Defaults to TRUE

Details

This function summarizes average daily values, average daily minimums/maximums, and absolute
minimums/maximums across user-defined seasons for a target year (target_yr).
The user also has the option to add a threshold hold line using the criteria argument. Typically,
this value is a water quality threshold, which is why criteria_lab defaults to 'WQ Threshold'.
However, the user has the option to specify any other type of threshold they wish. when doing so,
the value for criteria_lab should be changed accordingly.

Value

Returns a ggplot object

Author(s)
Julie Padilla

See Also

ggplot, assign_season, y_labeler

Examples

## get data, prep
data(elksmwq)
dat <- elksmwq
dat <- qaqc(elsmswq, qaqc_keep = c('0', '3', '5'))
do_plt <- annual_range(dat, param = 'do_mgl', target_yr = 2012)
do_plt <- annual_range(dat, param = 'do_mgl', target_yr = 2012, criteria = 2)

---

assign_season

Assign seasons to SWMP sampling data

Description

Assign seasons to SWMP sampling data on a monthly basis or user-defined seasonal basis

Usage

assign_season(
  data,
  season_grps = NULL,
  season_names = NULL,
  season_start = NULL,
  abb = TRUE
)

Arguments

data a vector of POSIXct dates
season_grps A list of seasons. Months (1-12) are assigned to different groups based on user preference. Defaults to 12 months, starting with January. Must assign a minimum of two seasons
season_names A string vector of season names. The number of season names must match the length of the season list. A minimum of two seasons must be assigned (e.g., 'Wet', 'Dry'). Defaults to 12 months, starting with January. The number of season names must match the number of seasons
season_start defaults to 12 months, starting with January
abb logical, should abbreviations for month names be used? Defaults to TRUE

Details

A helper function used by multiple data analyses to assign seasons to sampling data and to order the seasons. To assist with plotting, the seasons are assigned as factors. Seasons are assigned by first grouping the months into a list of season_grps and then specifying one name for each grouping using season_names. If season_grps is specified then season_names must also be defined. If neither argument is specified than the season assignments will default to monthly values. Using the season_start argument, the user can designate which season should be the first factor level. This assignment affects plot order for most functions. If season_start is not specified, then it will default to the first season in the list (January for monthly seasons and the first season in season_names for user-defined seasons).
Value

Returns a vector of ordered season factors.

Author(s)

Julie Padilla

Examples

```r
# Load the data
data(elksmwq)
dat <- elksmwq

# Assign season using default values
seas <- assign_season(dat$datetimestamp, abb = FALSE)
levels(seas)

# Assign season with abbreviated names
seas <- assign_season(dat$datetimestamp, abb = TRUE)
levels(seas)

# Assign season with custom season start
seas <- assign_season(dat$datetimestamp, season_start = 'Mar')
levels(seas)

# Assign season with custom season start and abbreviated names
seas <- assign_season(dat$datetimestamp, season_start = 'Mar', abb = FALSE)
levels(seas)

# Assign season with custom season groups and names
season_grps <- list(c(1,2,3), c(4,5,6), c(7,8,9), c(10, 11, 12))
season_names <- c('Winter', 'Spring', 'Summer', 'Fall')
season_start <- 'Spring'
seas <- assign_season(dat$datetimestamp, season_grps = season_grps, season_names = season_names, season_start = season_start)
levels(seas)
```

Description

Create a background map from a bounding box using Stamen Map tiles or a crude vector-based map.
Usage

```r
base_map(
  bbox, 
  bg_crs = 4326, 
  vector_only = FALSE, 
  maptype = "toner-lite", 
  zoom = NULL,
  ...
)
```

Arguments

- **bbox**: Bounding box vector.
- **bg_crs**: EPSG code or `st_crs` object for the returned map.
- **vector_only**: Logical, draw only a simple vector-based map.
- **zoom**: Zoom level for the base map created when `bg_map` is not specified. An integer value, 5 - 15, with higher numbers providing more detail. If not provided, a zoom level is autoscaled based on `bbox` parameters.
- **...**: Additional arguments to be passed to `ggmap::get_stamenmap`

Details

A helper, or stand-alone, function to create background map based on raster map tiles retrieved with `ggmap::get_stamenmap`. If `ggmap` is unavailable, the function creates a basic map using county-level polygon files. This map is fairly crude and should be considered a placeholder.

Value

Returns a `ggplot2` object.

Author(s)

Dave Eslinger

Examples

```r
# Simple, low-zoom map for testing
bound_box <- c(-77.393, 38.277, -75.553, 39.741)
(x <- base_map(bound_box, zoom = 7))

# Default zoom map with terrain maptype.
y <- base_map(bound_box, maptype = 'terrain')
```
**cbm.spatial**  
*Spatial Data from Chesapeake Bay - Maryland*

**Description**  
Shapefile for Chesapeake Bay - Maryland reserve boundary

**Usage**  
data(cbm.spatial)

**Format**  
A `SpatialPolygons` object

**Source**  
CDMO

**References**  

---

**counties_4269**  
*US County Map*

**Description**  
US County boundaries from the US Census Bureau’s MAF/TIGER geographic database in EPSG:4269, with all data except state FIPS codes and land area removed.

**Usage**  
data('counties_4269')

**Format**  
A `sf[DataFrame]` object

**Source**  
US Census Bureau
create_sk_flextable_list

References
United States Census Bureau. Data accessed from the US Census Bureau website: https://www2.census.gov/geo/tiger/GENZ2018/shp/cb_2018_us_county_20m.zip; accessed 13 March 2020

create_sk_flextable_list

Create a List of Flextable Objects

Description
Create a list of flextable objects to display Seasonal Kendall results in the NERRS reserve level template

Usage
create_sk_flextable_list(
  sk_result,  # a data.frame of reformatted results from sk_seasonal
  stations,  # chr, vector of stations to be displayed
  param,  # chr, vector of parameters to be displayed
  trend_col = c("#247BA0", ","A3DFFF", ","D9D9D9", ","white"),
  font_col_default = ","444E65",
  font_sz_stn = 6,
  font_sz_result = 12,
  font_sz_head = 6,
  ht_head = 0.28,
  ht_body = 0.202,
  is_swmp = TRUE,
  stn_name = NULL,
  stn_abbrev = NULL,
  par_name = NULL
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sk_result</td>
<td>a data.frame of reformatted results from sk_seasonal</td>
</tr>
<tr>
<td>stations</td>
<td>chr, vector of stations to be displayed</td>
</tr>
<tr>
<td>param</td>
<td>chr, vector of parameters to be displayed</td>
</tr>
<tr>
<td>trend_col</td>
<td>chr, a four element vector that specifies colors for increasing, decreasing, no change, and insufficient data trends</td>
</tr>
<tr>
<td>font_col_default</td>
<td>chr, default color to be used for trend table</td>
</tr>
<tr>
<td>font_sz_stn</td>
<td>int, specify the font size of displayed station names</td>
</tr>
<tr>
<td>font_sz_result</td>
<td>int, specify the font size of the displayed results</td>
</tr>
<tr>
<td>font_sz_head</td>
<td>int, specify the font size of the table header row</td>
</tr>
</tbody>
</table>
**create_sk_national_ft_reserves**

Create a Flextable Object of Reserve Names

**Description**

Create a flextable of reserve names for use with the NERRS national level template.

**Usage**

```r
create_sk_national_ft_reserves(
  sk_result,
  font_sz_stn = 8,
  font_sz_head = 8,
  ht_head = 0.75,
  ht_body = 0.2
)
```
create_sk_national_ft_results

Arguments

sk_result a data.frame of reformatted results from sk_seasonal
font_sz_stn int, specify the font size of displayed station names
font_sz_head int, specify the font size of the table header row
ht_head num, specify the cell height of the table body rows. Units for this parameter are in inches.
ht_body num, specify the cell height of the table header row. Units for this parameter are in inches.

Details

This function is intended for internal use with the NERRS national level reporting scripts. Using results from the reserve level trend analyses, create_sk_national_ft_reserves creates a flextable object of reserve names for display in the NERRS national level template.

Value

Returns a flextable object

Author(s)

Julie Padilla

create_sk_national_ft_results

Create a Flextable Object of Seasonal Kendall Results

Description

Create a flextable object to display Seasonal Kendall results for each reserve in the NERRS national level template

Usage

create_sk_national_ft_results(
    sk_result,
    param,
    font_sz_result = 12,
    font_sz_head = 8,
    ht_head = 0.375,
    ht_body = 0.2
)
Arguments

- **sk_result**: a `data.frame` of reformatted results generated by national Level template scripts from reserve level handoff files
- **param**: chr, the name of the parameter that corresponds to the seasonal kendall results in `sk_result`
- **font_sz_result**: int, specify the font size of the displayed results
- **font_sz_head**: int, specify the font size of the table header row
- **ht_head**: num, specify the cell height of the table body rows. Units for this parameter are in inches.
- **ht_body**: num, specify the cell height of the table header row. Units for this parameter are in inches.

Details

This function is intended for internal use with the NERRS national level reporting scripts. Using results from the reserve level trend analyses, `create_sk_national_ft_results` creates a flextable object of seasonal kendall results to be displayed in the NERRS national level template.

Value

Returns a `flextable` object

Author(s)

Julie Padilla

---

**elknmnut**

*Nutrient Data from Elkhorn Slough - North Marsh Station*

Description

Monthly nutrient data from Elkhorn Slough North Marsh station

Usage

`data(elknmnut)`

Format

A `data.frame` object

Source

CDMO
References


elk_spatial

Spatial Data from Elkhorn Slough

Description

Shapefile for Elkhorn Slough reserve boundary

Usage

data(elk_spatial)

Format

A SpatialPolygons object
Source

CDMO

References


---

<table>
<thead>
<tr>
<th>ft_col_names</th>
<th>Convert Parameter Abbreviations</th>
</tr>
</thead>
</table>

Description

Convert SWMP\textsubscript{r} parameter abbreviations into formats appropriate for use with NERRS reserve level template \texttt{flextable}

Usage

\texttt{ft\_col\_names(param)}

Arguments

\begin{itemize}
  \item \texttt{param} \texttt{chr}, vector of parameter abbreviations
\end{itemize}

Details

A helper function used internally by \texttt{create_sk_flextable_list} to label \texttt{flextable} columns in the trend table for the reserve level report.

Value

Returns a \texttt{data.frame} of user-specified results to be displayed

Author(s)

Julie Padilla
**generate_results_table**

*Filter Reformatted Seasonal Kendall Results*

**Description**
Filters a dataframe of user-specified results for display in the NERRS reserve level report

**Usage**

```r
generate_results_table(sk_result, stations, param)
```

**Arguments**

- `sk_result` a `data.frame` of reformatted seasonal kendall results from `sk_seasonal`
- `stations` chr, vector of station names included in `sk_result` that will be displayed in the NERRS reserve level report
- `param` chr, vector of parameters included in `sk_result` that will be displayed in the NERRS reserve level report

**Details**
A helper function used internally by `create_sk_flextable_list` to create a `data.frame` of user specified parameters to be displayed in the reserve level report.

**Value**
Returns a `data.frame` of user-specified results to be displayed

**Author(s)**
Julie Padilla

---

**generate_station_table**

*Filter Reformatted Seasonal Kendall Results*

**Description**
Filters a dataframe of user-specified results for display in the NERRS reserve level report

**Usage**

```r
generate_station_table(sk_result, stations)
```
geographic_unique_stations

Arguments

- **sk_result**: a data.frame of reformatted seasonal kendall results from `sk_seasonal`
- **stations**: chr, vector of stations listed in `sk_result` that should be displayed in the NERRS reserve level report

Details

Used internally by `create_sk_flextable_list` to create a data.frame of user specified parameters to be displayed.

Value

Returns a data.frame of user-specified results to be displayed.

Author(s)

Julie Padilla

---

**geographic_unique_stations**

*Return a vector of geographically unique NERR Stations*

Description

Creates an alphabetically sorted, vector of geographically unique stations for mapping.

Usage

`geographic_unique_stations(nerr_site_id)`

Arguments

- **nerr_site_id**: chr vector of valid NERR stations

Details

This function is intended for internal use with the NERRS reserve level reporting scripts and is used along with `res_local_map`. It takes a vector of NERR site ids and only returns geographically unique locations.

Value

returns a vector of NERR stations

Author(s)

Julie Padilla
get_reserve

Examples

```r
stns <- c('apacpnut', 'apacpwq', 'apadbnut', 'apadbwq', 'apaebmet',
         'apaebnut', 'apaebwq', 'apaesnut', 'apaeswq')

geographic_unique_stations(stns)
```

get_reserve

Identify NERRS reserve from metadata

Description

Identify the NERRS reserve from metadata in the data file

Usage

```r
get_reserve(data.file)
```

Arguments

- `data.file`: location of data

Details

This function is intended for internal use with the NERRS reserve level reporting scripts. It determines the name of the full name of the NERRS reserve associated with the data in the user-specified data folder.

Value

Returns a character string of the full reserve name

Author(s)

Julie Padilla
get_shp_name  Identify shapefile for NERRS reserve

Description
Identify the shapefile name associated with the reserve in the data file

Usage
get_shp_name(gis.file.loc)

Arguments
  gis.file.loc  path to gis file location

Details
This function is intended for internal use with the NERRS reserve level reporting scripts. It identifies the name of the shapefile associated with the NERRS reserve.

Value
Returns a character string of the shapefile for the reserve boundary

Author(s)
Julie Padilla

ger_sites  Identify NERRS reserve stations from metadata

Description
Identify the NERRS reserve sampling stations based on the metadata in the data file

Usage
ger_sites(
  data.file,
  type = c("wq", "nut", "met"),
  active = TRUE,
  primary = TRUE
)

get_site_code

Arguments

- data.file: location of data
- type: chr string of data station type ("wq", "nut", or "met")
- active: logical. Should inactive stations be excluded? Defaults to TRUE
- primary: logical. Should non-primary stations be excludes? Defaults to TRUE

Details

This function is intended for internal use with the NERRS reserve level reporting scripts. It returns the sampling stations associated with the data in the user-specified data folder.

Value

Returns a character vector of reserve stations

Author(s)

Julie Padilla

get_site_code  

Return NERRS reserve site code based on data in the data file

Description

Identify the 3-letter NERRS reserve code from metadata in the data file

Usage

get_site_code(data.file)

Arguments

- data.file: data source location

Details

This function is intended for internal use with the NERRS reserve level reporting scripts. It returns the 3-letter reserve code associated with the data in the user-specified data folder.

Value

Returns 3-letter, reserve site code as chr

Author(s)

Julie Padilla
get_site_coordinates  Identify NERRS sampling locations from metadata

Description

Identify the latitude/longitude for sampling stations based on the metadata in the data file

Usage

get_site_coordinates(data.file, active = TRUE)

Arguments

data.file  location of data
active      logical. Only return active stations?

Details

This function is intended for internal use with the NERRS reserve level reporting scripts. It returns the names, station codes, and coordinates associated with the data in the user-specified data folder.

Value

Returns a dataframe of station ids, station names, lat/long

Author(s)

Julie Padilla

historical_daily_range  Historical Daily Range Timeseries

Description

Compare daily averages for a target year to historical highs and lows
historical_daily_range

Usage

historical_daily_range(swmpr_in, ...)

## S3 method for class 'swmpr'

historical_daily_range(swmpr_in, 
  param = NULL, 
  hist_rng = NULL, 
  target_yr = NULL, 
  criteria = NULL, 
  free_y = FALSE, 
  log_trans = FALSE, 
  converted = FALSE, 
  criteria_lab = "WQ Threshold", 
  plot_title = FALSE, 
  plot = TRUE, 
  ...
)

Arguments

swmpr_in input swmpr object
...
not used
param chr string of variable to plot
hist_rng numeric vector, if historic range is not specified then the min/max values of the data set will be used.
target_yr numeric, the target year that should be compared against the historic range. If target year is not specified then dot will not be plotted
criteria numeric, a numeric criteria that will be plotted as a horizontal line
free_y logical, should the y-axis be free? Defaults to FALSE. If FALSE, defaults to zero, unless negative values are present. If TRUE, y-axis limits are selected by ggplot
log_trans logical, should y-axis be log? Defaults to FALSE
converted logical, were the units converted from the original units used by CDMO? Defaults to FALSE. See y_labeler for details.
criteria_lab chr, label for the threshold criteria defined in criteria. Defaults to "WQ Threshold"
plot_title logical, should the station name be included as the plot title? Defaults to FALSE
plot logical, should a plot be returned? Defaults to TRUE

Details

This function compares the average daily minimums/maximums and absolute daily minimums/maximums from a historical range to the average daily value from a target year. If hist_rng is not specified then the minimum and maximum years within the data set will be used. If target_yr is not specified then only the results for the hist_rng will be returned.
The user also has the option to add a threshold line using the criteria argument. Typically, this value is a water quality threshold, which is why criteria_lab defaults to 'WQ Threshold'. However, the user has the option to specify any other type of threshold they wish. when doing so, the value for criteria_lab should be changed accordingly.

**Value**

Returns a *ggplot* object

**Author(s)**

Julie Padilla, Kimberly Cressman

**See Also**

*ggplot, y_labeler*

**Examples**

```r
data(apacpwq)
dat <- apacpwq

dat <- qaqc(apacpwq, qaqc_keep = c('0', '3', '5'))
# with criteria
y <- historical_daily_range(dat, param = 'do_mgl', target_yr = 2013, criteria = 2)

# w/o criteria
z <- historical_daily_range(dat, param = 'do_mgl', target_yr = 2013)

# add a y label
zz <- z + labs(x = NULL, y = "Dissolved Oxygen (mg/L)")
```

**historical_range**  
*Historical Monthly/Seasonal Range Timeseries*

**Description**

Compare seasonal averages/minimums_MAXIMUMS for a target year to historical seasonal averages/minimums_MAXIMUMS.
Usage

historical_range(swmpr_in, ...)

## S3 method for class 'swmpr'

historical_range(
  swmpr_in,
  param = NULL,
  hist_rng = NULL,
  target_yr = NULL,
  criteria = NULL,
  free_y = FALSE,
  log_trans = FALSE,
  converted = FALSE,
  criteria_lab = "WQ Threshold",
  plot_title = FALSE,
  plot = TRUE,
  ...
)

Arguments

- **swmpr_in**: input swmpr object
- **...**: additional arguments passed to other methods. See `assign_season`
- **param**: chr string of variable to plot
- **hist_rng**: numeric vector, if historic range is not specified then the min/max values of the data set will be used.
- **target_yr**: numeric, the target year that should be compared against the historic range. If target year is not specified then dot will not be plotted
- **criteria**: numeric, a numeric criteria that will be plotted as a horizontal line
- **free_y**: logical, should the y-axis be free? Defaults to FALSE. If FALSE, defaults to zero, unless negative values are present. If TRUE, y-axis limits are selected by ggplot
- **log_trans**: logical, should y-axis be log? Defaults to FALSE
- **converted**: logical, were the units converted from the original units used by CDMO? Defaults to FALSE. See `y_labeler` for details.
- **criteria_lab**: chr, label for the threshold criteria defined in criteria. Defaults to "WQ Threshold"
- **plot_title**: logical, should the station name be included as the plot title? Defaults to FALSE
- **plot**: logical, should a plot be returned? Defaults to TRUE

Details

This function summarizes average daily values and average daily minimums/maximums across user-defined seasons for a target year (`target_yr`) and for a historical range (`hist_rng`). If `hist_rng` is not specified then the minimum and maximum years within the data set will be used. If `target_yr` is not specified then only the results for the `hist_rng` will be returned.
The user also has the option to add a threshold hold line using the criteria argument. Typically, this value is a water quality threshold, which is why criteria_lab defaults to 'WQ Threshold'. However, the user has the option to specify any other type of threshold they wish. when doing so, the value for criteria_lab should be changed accordingly.

Value

Returns a ggplot object

Author(s)

Julie Padilla, Kimberly Cressman

See Also

ggplot, assign_season, y_labeler

Examples

data(elksmwq)
dat <- qaqc(elksmwq, qaqc_keep = c('0', '3', '5'))
# with criteria
y <- historical_range(dat, param = 'do_mgl', target_yr = 2013, criteria = 2)
# w/o criteria
z <- historical_range(dat, param = 'do_mgl', target_yr = 2013)
# add a y label
zz <- z + labs(x = NULL, y = "Dissolved Oxygen (mg/L)")

import_local_nut

Import local CDMO data

Description

Import local data that were obtained from the CDMO through the zip downloads feature

Usage

import_local_nut(path, station_code, collMethd = c(1, 2), trace = FALSE)
import_local_nut

Arguments

path chr string of full path to .csv files with raw data, can be a zipped or unzipped directory where the former must include the .zip extension

station_code chr string of station to import, typically 7 or 8 characters including wq, nut, or met extensions, may include full name with year, excluding file extension

collMethd chr string of nutrient data to subset. 1 indicates monthly, 2 indicates diel. Default is both diel and monthly data.

trace logical indicating if progress is sent to console, default FALSE

Details

The function is designed to import local data that were downloaded from the CDMO outside of R. This approach works best for larger data requests, specifically those from the zip downloads feature in the advanced query section of the CDMO. The function may also work using data from the data export system, but this feature has not been extensively tested. The downloaded data will be in a compressed folder that includes multiple .csv files by year for a given data type (e.g., apacpwq2002.csv, apacpwq2003.csv, apacpnut2002.csv, etc.). The import_local function can be used to import files directly from the compressed folder or after the folder is decompressed. In the former case, the requested files are extracted to a temporary directory and then deleted after they are loaded into the current session. An example dataset is available online to illustrate the format of the data provided through the zip downloads feature. See the link below to access these data. All example datasets included with the package were derived from these raw data.

Occasionally, duplicate time stamps are present in the raw data. The function handles duplicate entries differently depending on the data type (water quality, weather, or nutrients). For water quality and nutrient data, duplicate time stamps are simply removed. Note that nutrient data often contain replicate samples with similar but not duplicated time stamps within a few minutes of each other. Replicates with unique time stamps are not removed but can be further processed using rem_reps. Weather data prior to 2007 may contain duplicate time stamps at frequencies for 60 (hourly) and 144 (daily) averages, in addition to 15 minute frequencies. Duplicate values that correspond to the smallest value in the frequency column (15 minutes) are retained.

This function differs from import_local in that it allows for special handling of nutrient data. Using this function, the user can separate diel sampling data from low-tide sampling data using the collMethd argument.

Zip download request through CDMO: http://cdmo.baruch.sc.edu/aqs/zips.cfm

Example dataset: https://s3.amazonaws.com/swmpedatadata/zip_ex.zip

Value

Returns a swmpr object with all parameters and QAQC columns for the station. The full date range in the raw data are also imported.

Author(s)

Marcus Beck, Julie Padilla (additional of codecollMthd argument) maintainer: Julie Padilla
See Also

all_params, all_params_dtrng, rem_reps, single_param

---

### lm_p_labs

#### P-Value labels for Plotting

**Description**

Generate a dataframe of p-value labels based on p-values from linear regression

**Usage**

```
lm_p_labs(dat_in)
```

**Arguments**

- `dat_in`: data.frame with year, season, min, mean, max columns

**Details**

A helper function that returns a data.frame of p-value labels for use with the `seasonal_dot`. P-values are taken from linear regression `lm`.

**Value**

Returns data.frame for use with `seasonal_dot`

**Author(s)**

Julie Padilla, Dave Eslinger

**See Also**

- `lm`
load_shp_file

Load and format shapefile for reserve level map

Description

Load and format shapefile for use with res_local_map. If polygons are dissolved, the only attribute returned will be a count of the number of grouped polygons, otherwise, all attributes are retained.

Usage

load_shp_file(path, dissolve_boundaries = TRUE)

Arguments

path
path to shapefile and name
dissolve_boundaries
logical, should reserve boundaries be dissolved? Defaults to TRUE

Details

This function is intended for internal use with the NERRS reserve level reporting scripts. It loads a NERRS boundary shp file and dissolves unnecessary reserve boundaries. The resulting sf object is then used with res_sk_map and res_local_map

Value

Returns a sf object

Author(s)

Julie Padilla, Dave Eslinger

national_sk_map

Reserve National Map with Seasonal Kendall Results

Description

Create a base map for NERRS reserves in ggplot with seasonal kendall results
Usage

```r
national_sk_map(
  incl = c("contig", "AK", "HI", "PR"),
  highlight_states = NULL,
  sk_reserves = NULL,
  sk_results = NULL,
  sk_fill_colors = c("#444E65", "#A3DFFF", "#247BA0", "#0a0a0a"),
  agg_county = TRUE
)
```

Arguments

- **incl**: chr vector to include AK, HI, and PR (case sensitive)
- **highlight_states**: chr vector of state FIPS codes
- **sk_reserves**: chr vector of 3 letter reserve codes that have seasonal kendall results
- **sk_results**: chr vector of seasonal kendall results. Results can be 'inc', 'dec', 'insig', or 'insuff' which stand for 'increasing trend', 'decreasing trend', 'statistically insignificant trend', or 'insufficient data to detect trend'
- **sk_fill_colors**: chr vector of colors used to fill seasonal kendall result markers
- **agg_county**: logical, should counties be aggregated to the state-level? Defaults to TRUE

Details

Create a base map of the US with options for including AK, HI, and PR. The user can choose which states and NERRS reserves to highlight. An early sp-based version of this function by Julie Padilla was developed, in part, from a blog post by Bob Rudis. The current sf-based version, by Dave Eslinger, uses an approach from the r-spatial tutorial by Mel Moreno and Mathieu Basille.

To ensure the proper plotting of results, the order of the results vector for `sk_results` should match the order of the reserves vector for `sk_reserves`.

Value

Returns a `ggplot` object

Author(s)

Julie Padilla, Dave Eslinger Maintainer: Dave Eslinger

References

Examples

```r
# National map highlighting west coast and non-CONUS states and NERRS.
nerr_states_west <- c("02", "06", "41", "53", "72", "15")

nerrs_codes <- c("pdb", "sos", "sfb", "elk", "tjr", "kac", "job", "hee")
nerrs_sk_results <- c("inc", "inc", "dec", "insig", "insuff", "dec", "inc", "dec")
national_sk_map(highlight_states = nerr_states_west,
                sk_reserve = nerrs_codes, sk_results = nerrs_sk_results)
```

---

**raw_boxplot**

Boxplots of raw data by user-defined season for a target year

**Description**

Boxplots of raw data by user-defined season for a target year

**Usage**

```r
raw_boxplot(swmpr_in, ...)
```

**Arguments**

- `swmpr_in` input swmpr object
- `...` additional arguments passed to other methods. See `assign_season` and `y_labeler`.
- `param` chr string of variable to plot
- `target_yr` numeric, if target year is not specified then all data in the data frame will be used.
- `criteria` numeric, a numeric criteria that will be plotted as a horizontal line
- `free_y` logical, should the y-axis be free? Defaults to FALSE. If FALSE, defaults to zero, unless negative values are present. If TRUE, y-axis limits are selected by ggplot
Description

Replace Inf/-Inf/NaN in a matrix with NA

Usage

remove_inf_and_nan(x)
reserve_locs

Arguments

x input matrix

Details

This function replaces Inf, -Inf, and NaN in a matrix with NA. It is used internally by several functions.

Value

Returns a matrix object

Author(s)

Julie Padilla

reserve_locs

NERRS Sampling Location Data Frame

Description

Create a data.frame of selected NERRS locations for plotting with res_national_map

Usage

reserve_locs(incl = c("contig", "AK", "HI", "PR"), subset_reserve = NULL)

Arguments

incl Str vector to include AK, HI, and PR
subset_reserve chr string of 3 letter reserve IDs to include as data points. To include Heʻeia use the reserve code ‘HEA’.

Details

A helper function used to generate a data.frame of selected reserve locations for use with res_national_map.

Value

Returns a data.frame for internal use with res_national_map

Author(s)

Julie Padilla, Dave Eslinger
res_custom_map

Local Reserve Map for Custom Stations

Description

Create a stylized reserve-level map of custom station locations for use with the reserve level reporting template

Usage

```r
res_custom_map(
  stations,  # chr string of the reserve stations to include in the map
  x_loc,     # num vector of x coordinates for stations
  y_loc,     # num vector of y coordinates for stations
  bbox,      # a bounding box associated with the reserve. Must be in the format of c(X1, Y1, X2, Y2)
  shp,       # sf data frame (preferred) or SpatialPolygons object
  station_labs = TRUE,  # logical, should stations be labeled? Defaults to TRUE
  station_col = NULL,   # chr vector of colors used to color station points. Defaults to 'black'.
  lab_loc = NULL,       # chr vector of 'R' and 'L', one letter for each station. if no lab_loc is specified then labels will default to the left.
  bg_map = NULL,        # a georeferenced ggmap or ggplot object used as a background map, generally provided by a call to base_map. If bg_map is specified, maptype and zoom are ignored.
  zoom = NULL,          # Zoom level for the base map created when bg_map is not specified. An integer value, 5 - 15, with higher numbers providing more detail. If not provided, a zoom level is autoscaled based on bbox parameters.
  maptype = "toner-lite"  # Background map type from Stamen Maps (http://maps.stamen.com/); one of c("terrain", "terrain-background", "terrain-labels", "terrain-lines", "toner", "toner-2010", "toner-2011", "toner-background", "toner-hybrid", "toner-labels", "toner-lines", "toner-lite", "watercolor")
)
```

Arguments

- `stations`: chr string of the reserve stations to include in the map
- `x_loc`: num vector of x coordinates for stations
- `y_loc`: num vector of y coordinates for stations
- `bbox`: a bounding box associated with the reserve. Must be in the format of c(X1, Y1, X2, Y2)
- `shp`: sf data frame (preferred) or SpatialPolygons object
- `station_labs`: logical, should stations be labeled? Defaults to TRUE
- `station_col`: chr vector of colors used to color station points. Defaults to 'black'.
- `lab_loc`: chr vector of 'R' and 'L', one letter for each station. if no lab_loc is specified then labels will default to the left.
- `bg_map`: a georeferenced ggmap or ggplot object used as a background map, generally provided by a call to base_map. If bg_map is specified, maptype and zoom are ignored.
- `zoom`: Zoom level for the base map created when bg_map is not specified. An integer value, 5 - 15, with higher numbers providing more detail. If not provided, a zoom level is autoscaled based on bbox parameters.
Details

Creates a stylized, reserve-level base map. The user can specify the reserve and stations to plot. The user can also specify a bounding box. For multi-component reserves, the user should specify a bounding box that highlights the component of interest.

This function does not automatically detect conflicts between station labels. The lab_loc argument allows the user to specify "R" or "L" for each station to prevent labels from conflicting with each other.

This function is intended to be used with mapview::mapshot to generate a png for the reserve-level report.

Value

returns a ggplot object

Author(s)

Julie Padilla, Dave Eslinger

Examples

```r
### set plotting parameters
stns <- c('Stn 1', 'Stn 2')
x_coords <- c(-121.735281, -121.750369)
y_coords <- c(36.850377, 36.806667)
shp_fl <- elk_spatial
bounding_elk <- c(-121.8005, 36.7779, -121.6966, 36.8799)
lab_dir <- c('L', 'R')

### Low zoom and default maptype plot (for CRAN testing, not recommended)
# Lower zoom number gives coarser text and fewer features
(x_low <- res_custom_map(stations = stns, x_loc = x_coords, y_loc = y_coords,
bbox = bounding_elk, lab_loc = lab_dir, shp = shp_fl,
zoom = 10))

x_def <- res_custom_map(stations = stns, x_loc = x_coords, y_loc = y_coords,
bbox = bounding_elk, lab_loc = lab_dir, shp = shp_fl)

res_custom_map(stations = stns, x_loc = x_coords, y_loc = y_coords,
bbox = bounding_elk, lab_loc = lab_dir,
shp = shp_fl, station_col = c('red', 'green'))
```
res_custom_sk_map

Local Reserve Map With Seasonal Kendall Results for Custom Stations

Description

Create a stylized reserve-level map of seasonal kendall results from custom station locations for use with the reserve level reporting template

Usage

res_custom_sk_map(
  stations,
  x_loc,
  y_loc,
  sk_result = NULL,
  bbox,
  shp,
  station_labs = TRUE,
  lab_loc = NULL,
  bg_map = NULL,
  zoom = NULL,
  maptype = "toner-lite"
)

Arguments

stations chr string of the reserve stations to include in the map
x_loc num vector of x coordinates for stations. East longitudes must be negative.
y_loc num vector of y coordinates for stations
sk_result vector of values denoting direction and significance of seasonal kendall results. Result should be c('inc', 'dec', 'insig') for sig. negative, no sig. results, and sig. positive result
bbox a bounding box associated with the reserve. Must be in the format of c(X1, Y1, X2, Y2)
shp sf data frame (preferred) or SpatialPolygons object
station_labs logical, should stations be labeled? Defaults to TRUE
lab_loc chr vector of 'R' and 'L', one letter for each station. if no lab_loc is specified then labels will default to the left.
bg_map a georeferenced ggsmap or ggplot object used as a background map, generally provided by a call to base_map. If bg_map is specified, maptype and zoom are ignored.
zoom Zoom level for the base map created when bg_map is not specified. An integer value, 5 - 15, with higher numbers providing more detail. If not provided, a zoom level is autoscaled based on bbox parameters.

Details

Creates a stylized, reserve-level base map for displaying seasonal kendall results from sk_seasonal. The user can specify the reserve and stations to plot. The user can also specify a bounding box. For multi-component reserves, the user should specify a bounding box that highlights the component of interest.

To display seasonal trends, the user must specify c('inc','dec','insig','insuff') for each station listed in the stations argument.

Value

returns a ggplot object

Author(s)

Julie Padilla, Dave Eslinger

Examples

```r
### set plotting parameters
stns <- c('Stn 1', 'Stn 2')
x_coords <- c(-121.735281, -121.750369)
y_coords <- c(36.850377, 36.806667)
shp_fl <- elk_spatial
bounding_elk <- c(-121.8005, 36.7779, -121.6966, 36.8799)
lab_dir <- c('R', 'L')
trnds <- c('inc', 'dec')

### Low zoom and default maptype plot (for CRAN testing, not recommended)
# Lower zoom number gives coarser text and fewer features
(x_low <- res_custom_sk_map(stations = stns, x_loc = x_coords,
                          sk_result = trnds, y_loc = y_coords,
                          bbox = bounding_elk, lab_loc = lab_dir,
                          shp = shp_fl, zoom = 10))

### Default zoom and maptype plot
x_def <- res_custom_sk_map(stations = stns, x_loc = x_coords,
                          sk_result = trnds, y_loc = y_coords,
                          bbox = bounding_elk, lab_loc = lab_dir,
                          shp = shp_fl)

### Higher zoom number gives more details, but may not be visible
x_14 <- res_custom_sk_map(stations = stns, x_loc = x_coords,
                          sk_result = trnds, y_loc = y_coords,
                          bbox = bounding_elk, lab_loc = lab_dir,
                          shp = shp_fl, zoom = 14)
```
### Different map types may be used. All may not be available.
# Note that zoom and map type interact, so some experimentation may be 
# required.
x_terrain <- res_custom_sk_map(stations = stns, x_loc = x_coords, 
                sk_result = trnds, y_loc = y_coords, 
                bbox = bounding_elk, lab_loc = lab_dir, 
                shp = shp_fl, maptype = 'terrain')

---

**res_local_map**  
*Local Reserve Map*

---

**Description**
Create a stylized reserve-level map for use with the reserve level reporting template

**Usage**

```r
res_local_map(
  nerr_site_id, 
  stations, 
  bbox, 
  shp, 
  station_labs = TRUE, 
  lab_loc = NULL, 
  bg_map = NULL, 
  zoom = NULL, 
  maptype = "toner-lite"
)
```

**Arguments**

- **nerr_site_id**  
  chr string of the reserve to make, first three characters used by NERRS
- **stations**  
  chr string of the reserve stations to include in the map
- **bbox**  
  a bounding box associated with the reserve. Must be in the format of c(X1, Y1, X2, Y2)
- **shp**  
  sf data frame (preferred) or SpatialPolygons object
- **station_labs**  
  logical, should stations be labeled? Defaults to TRUE
- **lab_loc**  
  chr vector of 'R' and 'L', one letter for each station. if no lab_loc is specified then labels will default to the left.
- **bg_map**  
  a georeferenced ggmap or ggplot object used as a background map, generally provided by a call to base_map. If bg_map is specified, maptype and zoom are ignored.
res_local_map

`zoom`  
Zoom level for the base map created when `bg_map` is not specified. An integer value, 5 - 15, with higher numbers providing more detail. If not provided, a zoom level is autoscaled based on `bbox` parameters.

`maptype`  

Details

Creates a stylized, reserve-level base map. The user can specify the reserve and stations to plot. The user can also specify a bounding box. For multi-component reserves, the user should specify a bounding box that highlights the component of interest.

This function does not automatically detect conflicts between station labels. The `lab_loc` argument allows the user to specify "R" or "L" for each station to prevent labels from conflicting with each other.

This function is intended to be used with `mapview::mapshot` to generate a png for the reserve-level report.

Value

returns a `ggplot` object

Author(s)

Julie Padilla, Dave Eslinger

Examples

```r
## a compact reserve
### set plotting parameters
stations <-
sampling_stations[sampling_stations$NErr.Site.ID == 'elk'
  & sampling_stations$Status == 'Active'
  & sampling_stations$isSWMP == "P"],]$Station.Code
to_match <- c('wq', 'met')
stns <- stations[grep(paste(to_match, collapse = '|'), stations)]
shp_fl <- elk_spatial
bounding_elk <- c(-121.8005, 36.7779, -121.6966, 36.8799)
lab_dir <- c('L', 'R', 'L', 'L')
labs <- c('ap', 'cw', 'nm', 'sm', 'vm')

### Low zoom and default maptype plot (for CRAN testing, not recommended)
lower zoom number gives coarser text and fewer features
(x_low <- res_local_map('elk', stations = stns, bbox = bounding_elk,
                         lab_loc = lab_dir, shp = shp_fl,
                         zoom = 10))

### Default zoom and maptype
```
### A multicomponent reserve (show two different bounding boxes)

```r
x_def <- res_local_map('elk', stations = stns, bbox = bounding_elk,
  lab_loc = lab_dir, shp = shp_fl,
  zoom = 10)

### plot
y <- res_local_map('cbm', stations = stns, bbox = bounding_cbm_1,
  lab_loc = lab_dir, shp = shp_fl)
z <- res_local_map('cbm', stations = stns, bbox = bounding_cbm_2,
  lab_loc = lab_dir, shp = shp_fl)
```

---

**res_national_map**

**Reserve National Map**

**Description**

Create a base map for NERRS reserves in ggplot

**Usage**

```r
res_national_map(
  incl = c("contig", "AK", "HI", "PR"),
  highlight_states = NULL,
  highlight_reserves = NULL,
  agg_county = TRUE
)
```

**Arguments**

- `incl`chr vector to include AK, HI, and PR (case sensitive)
- `highlight_states`chr vector of state FIPS codes
res_national_map

highlight_reserves
chr vector of 3 letter reserve codes

agg_county
logical, should counties be aggregated to the state-level? Defaults to TRUE

Details
Create a base map of the US with options for including AK, HI, and PR. The user can choose which states and NERRS reserves to highlight. An early sp-based version of this function by Julie Padilla was developed, in part, from a blog post by Bob Rudis. The current sf-based version, by Dave Eslinger, uses an approach from the r-spatial tutorial by Mel Moreno and Mathieu Basille.

Value
Returns a ggplot object

Author(s)
Julie Padilla, Dave Eslinger
Maintainer: Dave Eslinger

References

Examples
##National map highlighting states with NERRS

res_national_map(highlight_states = nerr_states)

#' ##Just the national map
res_national_map()

##National map highlighting west coast states and NERRS (including AK)
nerr_states_west <- c("02", "06", "41", "53")
nerrs_codes <- c("pdb", "sos", "sfb", "elk", "tjr", "kac")

res_national_map(highlight_states = nerr_states_west, highlight_reserve = nerrs_codes)
res_sk_map | Local Reserve Map With Seasonal Kendall Results

Description

Create a stylized reserve-level map of seasonal kendall results for use with the reserve level reporting template.

Usage

```r
res_sk_map(
  nerr_site_id,
  stations,
  sk_result = NULL,
  bbox,
  shp,
  station_labs = TRUE,
  lab_loc = NULL,
  bg_map = NULL,
  zoom = NULL,
  maptype = "toner-lite"
)
```

Arguments

- **nerr_site_id**: chr string of the reserve to make, first three characters used by NERRS.
- **stations**: chr string of the reserve stations to include in the map.
- **sk_result**: vector of values denoting direction and significance of seasonal kendall results. Result should be `c('inc', 'dec', 'insig', 'insuff')` for significant positive, significant negative, no significant results, and insufficient data to calculate result.
- **bbox**: a bounding box associated with the reserve. Should be in the format of `c(xmin, ymin, xmax, ymax)`.
- **shp**: sf data frame (preferred) or SpatialPolygons object.
- **station_labs**: logical, should stations be labeled? Defaults to TRUE.
- **lab_loc**: chr vector of 'R' and 'L', one letter for each station. If no lab_loc is specified then labels will default to the left.
- **bg_map**: a georeferenced ggmap or ggplot object used as a background map, generally provided by a call to `base_map`. If bg_map is specified, maptype and zoom are ignored.
- **zoom**: Zoom level for the base map created when bg_map is not specified. An integer value, 5 - 15, with higher numbers providing more detail. If not provided, a zoom level is autoscaled based on bbox parameters.

Details

Creates a stylized, reserve-level base map for displaying seasonal kendall results from sk_seasonal. The user can specify the reserve and stations to plot. The user can also specify a bounding box. For multi-component reserves, the user should specify a bounding box that highlights the component of interest.

To display seasonal trends, the user must specify c('inc','dec','insig','insuff') for each station listed in the stations argument.

Value

returns a ggplot object.

Author(s)

Julie Padilla, Dave Eslinger

Examples

### A compact reserve

```r
## set plotting parameters
stations <- sampling_stations[(sampling_stations$NERR.Site.ID == 'elk' & sampling_stations$Status == 'Active' & sampling_stations$isSWMP == "P"),]$Station.Code
to_match <- c('wq')
stns <- stations[grep(paste(to_match, collapse = '|'), stations)]
shp_fl <- elk_spatial
bounding_elk <- c(-121.8005, 36.7779, -121.6966, 36.8799)
trnds <- c('inc', 'dec', 'insuff', 'insig')

### Low zoom and default maptype plot (for CRAN testing, not recommended)
# Lower zoom number gives coarser text and fewer features
(x_low <- res_sk_map('elk', stations = stns, sk_result = trnds,
                      bbox = bounding_elk, shp = shp_fl,
                      zoom = 10))
```

```r
### Default zoom and maptype
x_def <- res_sk_map('elk', stations = stns, sk_result = trnds,
                     bbox = bounding_elk, shp = shp_fl)
```

```r
### Higher zoom number gives more details, but those may not be visible
x_14 <- res_sk_map('elk', stations = stns, sk_result = trnds,
                     bbox = bounding_elk, shp = shp_fl)
```
### Different maptypes may be used.
x_terrain <- res_sk_map('elk', stations = stns, sk_result = trnds,
bbox = bounding_elk, shp = shp_fl,
maptype = 'terrain')
### A multicomponent reserve (showing two different bounding boxes)

# set plotting parameters
stations <- sampling_stations[(sampling_stations$NERR.Site.ID == 'cbm'
& sampling_stations$Status == 'Active'
& sampling_stations$isSWMP == "P"),]$Station.Code
to_match <- c('wq')
stns <- stations[grep(paste(to_match, collapse = '|'), stations)]
shp_fl <- cbm_spatial
bounding_cbm_1 <- c(-77.393, 38.277, -75.553, 39.741)
bounding_cbm_2 <- c(-76.8, 38.7, -76.62, 38.85)
trnds <- c('inc', 'insuff', 'dec', 'insig')

# plot
y <- res_sk_map('cbm', stations = stns, sk_result = trnds,
bbox = bounding_cbm_1, shp = shp_fl)
z <- res_sk_map('cbm', stations = stns, sk_result = trnds,
bbox = bounding_cbm_2, shp = shp_fl)

---

**sampling_stations**  
*Detailed NERRS site data*

**Description**

Metadata on NERRS stations provided by the Central Data Management Office (CDMO) when data is downloaded

**Usage**

```r
data(sampling_stations)
```

**Format**

A `data.frame` object

**Source**

CDMO
References


seasonal_barplot

Cumulative Bar Plot

Description

Cumulative bar plot over a historic range
Usage

seasonal_barplot(swmp_in, ...)

## S3 method for class 'swmp'
seasonal_barplot(
  swmp_in,
  param = NULL,
  hist_rng = NULL,
  log_trans = FALSE,
  converted = FALSE,
  hist_avg = TRUE,
  bar_position = "stack",
  season_facet = FALSE,
  plot_title = FALSE,
  plot = TRUE,
  ...
)

Arguments

swmp_in input swmp object
...
additional arguments passed to other methods. See assign_season
param chr string of variable to plot
hist_rng numeric vector, if historic range is not specified then the min/max values of the
data set will be used.
log_trans logical, should y-axis be log? Defaults to FALSE
converted logical, were the units converted from the original units used by CDMO? De-
defaults to FALSE. See y_labeler for details.
hist_avg logical, should a historical average be included? Defaults to TRUE.
bar_position chr string, options available are stack or dodge. Defaults to stack
season_facet logical, should plot be faceted by season? Defaults to FALSE.
plot_title logical, should the station name be included as the plot title? Defaults to FALSE
plot logical, should a plot be returned? Defaults to TRUE

Details

This function uses barplots to summarize parameters that are best viewed on a cumulative basis
(e.g., precipitation). Data are aggregated on a seasonal and annual basis.

There are two ways to make interannual comparisons: on an aggregate basis and on a seasonal
basis. If the argument season_facet = FALSE then parameter totals from each season will be added
together to compose one, multi-color bar. If season_facet = TRUE then parameter totals from each
season separated into multiple plots for easier intra-season comparison across years.

Value

A ggplot object
seasonal_boxplot

Author(s)
Julie Padilla

See Also

`ggplot`, `assign_season`, `y_labeler`

Examples

data(apaebmet)
dat <- qaqc(apaebmet, qaqc_keep = c('0', '3', '5'))

x <- seasonal_barplot(dat, param = 'totprcp'
  , season_grps = list(c(1,2,3), c(4,5,6), c(7,8,9), c(10,11,12))
  , season_names = c('Winter', 'Spring', 'Summer', 'Fall')
  , hist_avg = TRUE
  , converted = FALSE)

# return a table instead of a figure
y <- seasonal_barplot(dat, param = 'totprcp'
  , season_grps = list(c(1,2,3), c(4,5,6), c(7,8,9), c(10,11,12))
  , season_names = c('Winter', 'Spring', 'Summer', 'Fall')
  , converted = FALSE
  , plot = FALSE)

## divide plot into seasonal facets
z <- seasonal_barplot(dat, param = 'totprcp'
  , season_grps = list(c(1,2,3), c(4,5,6), c(7,8,9), c(10,11,12))
  , season_names = c('Winter', 'Spring', 'Summer', 'Fall')
  , season_facet = TRUE
  , hist_avg = TRUE
  , converted = FALSE)

## convert from mm to in
dat$totprcp <- dat$totprcp / 25.4

x1 <- seasonal_barplot(dat, param = 'totprcp'
  , season_grps = list(c(1,2,3), c(4,5,6), c(7,8,9), c(10,11,12))
  , season_names = c('Winter', 'Spring', 'Summer', 'Fall')
  , hist_avg = TRUE
  , converted = TRUE)

---

seasonal_boxplot  Seasonal boxplots

Description

Annual time series for year of interest on top of long-term percentiles
seasonal_boxplot

Usage

seasonal_boxplot(swmpr_in, ...)  
## S3 method for class 'swmpr'
seasonal_boxplot(
  swmpr_in,
  param = NULL,
  hist_rng = NULL,
  target_yr = NULL,
  criteria = NULL,
  free_y = FALSE,
  log_trans = FALSE,
  converted = FALSE,
  criteria_lab = "WQ Threshold",
  stat_lab = "Average",
  plot_title = FALSE,
  plot = TRUE,
  FUN = function(x) mean(x, na.rm = TRUE),
  ...
)

Arguments

swmpr_in            input swmpr object
...                 additional arguments passed to other methods. See assign_season
param               chr string of variable to plot
hist_rng            numeric vector, if historic range is not specified then the min/max values of the
                     data set will be used.
target_yr           numeric, the target year that should be compared against the historic range. If
                     target year is not specified then dot will not be plotted
criteria            numeric, a numeric criteria that will be plotted as a horizontal line
free_y              logical, should the y-axis be free? Defaults to FALSE. If FALSE, defaults to zero,
                     unless negative values are present. If TRUE, y-axis limits are selected by ggplot
log_trans           logical, should y-axis be log? Defaults to FALSE
converted           logical, were the units converted from the original units used by CDMO? De-
                     fault to FALSE. See y_labeler for details.
criteria_lab        chr, label for the threshold criteria defined in criteria. Defaults to "WQ
                     Threshold"
stat_lab             chr, label for the summary statistic defined in FUN. Defaults to "Average"
plot_title          logical, should the station name be included as the plot title? Defaults to FALSE
plot                logical, should a plot be returned? Defaults to TRUE
FUN                  function used to aggregate daily SWMP data
seasonal_boxplot

Details

This function uses boxplots to summarize statistics calculated on a daily basis across user-defined seasons for all years within the historic range (hist_rng). If hist_rng is not specified then the minimum and maximum years within the data set will be used. The summary statistics used to generate the boxplots are ggplot2 defaults: the center of the box is a median, and the lower/upper limits of the box are the 25-th and 75-th percentiles. The whiskers extend to the furthest data point within 1.5 * inter-quartile range (IQR). The dots beyond the whiskers are data points that are greater than 1.5 * IQR. If the user selects a target_yr, then a median summary statistic value will be plotted as a point against the boxplots.

Using the FUN argument, the user can specify the daily summary statistic to use. Commonly used statistics are min(x,na.rm = TRUE), mean(x,na.rm = TRUE), and max(x,na.rm = TRUE). After specifying FUN, the user should also specify stat_lab, which is used to construct appropriate legend labels.

The user also has the option to add a threshold hold line using the criteria argument. Typically, this value is a water quality threshold, which is why criteria_lab defaults to 'WQ Threshold'. However, the user has the option to specify any other type of threshold they wish. When doing so, the value for criteria_lab should be changed accordingly.

Value

Returns a ggplot object or a data.frame if plot = FALSE

Author(s)

Julie Padilla

See Also

ggplot, assign_season

Examples

```r
dat <- elksmqw
dat <- qaqc(dat, qaqc_keep = c('0', '3', '5'))

x <-
    seasonal_boxplot(dat, param = 'do_mgl')

y <-
    seasonal_boxplot(dat, param = 'do_mgl', target_yr = 2015,
                    season_grps = list(c(1,2,3), c(4,5,6), c(7,8,9), c(10, 11, 12)),
                    season_names = c('Winter', 'Spring', 'Summer', 'Fall'),
                    season_start = 'Spring')

z_min <-
    seasonal_boxplot(dat, param = 'do_mgl',
                     stat_lab = 'Minimum', FUN = function(x) min(x, na.rm = TRUE))
```
z_max <-
seasonal_boxplot(dat, param = 'do_mgl',
stat_lab = 'Maximum', FUN = function(x) max(x, na.rm = TRUE))

---

### seasonal_dot

#### Seasonal Dot Plot

**Description**

Plot average/min/max seasonal values faceted by season

**Usage**

```r
seasonal_dot(swmpr_in, ...)  
## S3 method for class 'swmpr'
seasonal_dot(
  swmpr_in,
  param = NULL,
  lm_trend = FALSE,
  lm_lab = FALSE,
  free_y = FALSE,
  log_trans = FALSE,
  converted = FALSE,
  plot_title = FALSE,
  plot = TRUE,
  ...
)
```

**Arguments**

- **swmpr_in**: input swmpr object
- **param**: chr string of variable to plot
- **lm_trend**: logical, add linear trend line?
- **lm_lab**: logical, add significance label? Statistically significant results will appear in bold.
- **free_y**: logical, should the y-axis be free? Defaults to FALSE. If FALSE, defaults to zero, unless negative values are present. If TRUE, y-axis limits are selected by `ggplot`.
- **log_trans**: logical, should y-axis be log? Defaults to FALSE.
- **converted**: logical, were the units converted from the original units used by CDMO? Defaults to FALSE. See `y_labeler` for details.
- **plot_title**: logical, should the station name be included as the plot title? Defaults to FALSE
- **plot**: logical, should a plot be returned? Defaults to TRUE

---

```r
z_max <-
seasonal_boxplot(dat, param = 'do_mgl',
stat_lab = 'Maximum', FUN = function(x) max(x, na.rm = TRUE))
```
Details

This function summarizes minimum, mean, and maximum values calculated on a seasonal basis to allow for easier intra-season comparisons over time.

1m_trend = TRUE adds a linear regression to the plot, and 1m_lab = TRUE will add p-values from the linear regression to the plot. If the p-values are significant (p < 0.05) then the text will appear in bold. 1m_lab text is color coded to match with the corresponding dots.

Value

Returns a ggplot object

Author(s)

Julie Padilla, Dave Eslinger

See Also

ggplot, assign_season, y_labeler

Examples

data_wq <- elksmwq
#dat_wq <- subset(dat_wq, subset = c('2010-01-01 0:00', '2017-01-01 0:00'))
data_wq <- qaqc(dat_wq, qaqc_keep = c(0, 3, 5))

x <-
seasonal_dot(dat_wq, param = 'do_mgl'
  , lm_trend = TRUE
  , lm_lab = TRUE
  , plot_title = TRUE)

y <-
seasonal_dot(dat_wq, param = 'do_mgl'
  , lm_trend = FALSE
  , lm_lab = FALSE
  , plot_title = TRUE)

z <-
seasonal_dot(dat_wq, param = 'do_mgl'
  , lm_trend = TRUE
  , lm_lab = FALSE
  , plot_title = TRUE)

data_nut <- elknmnut
data_nut <- subset(dat_nut, subset = c('2007-01-01 0:00', '2017-01-01 0:00'))
data_nut <- qaqc(dat_nut, qaqc_keep = c(0, 3, 5))

x1 <-
seasonal_dot(dat_nut
set_date_breaks

Description
Select reasonable breaks for scale_x_datetime

Usage
set_date_breaks(rng)

Arguments
rng date range years

Details
A helper function for easier date label setting

Value
Returns a chr string for date_breaks

Author(s)
Julie Padilla

See Also
set_date_break_labs, scale_x_datetime
set_date_breaks_minor  Select reasonable minor breaks for scale_x_datetime

Description
Select reasonable minor breaks for scale_x_datetime

Usage
set_date_breaks_minor(rng)

Arguments
rng  date range years

Details
A helper function for easier date label setting

Value
Returns a chr string for date_breaks

Author(s)
Dave Eslinger, Julie Padilla

See Also
set_date_break_labs, scale_x_datetime

set_date_break_labs  Set reasonable date breaks labels

Description
Select reasonable labels for breaks used in scale_x_datetime

Usage
set_date_break_labs(rng)

Arguments
rng  date range years
Details

A helper function for easier date label setting

Value

Returns a chr string for date_labels

Author(s)

Julie Padilla

See Also

set_date_breaks, scale_x_datetime

sk_seasonal

Seasonal Kendall Analysis for Seasonal Data

Description

Non-parametric test for monotonic seasonal trends

Usage

sk_seasonal(swmpr_in, ...)

## S3 method for class 'swmpr'

sk_seasonal(
  swmpr_in,
  param = NULL,
  alpha = 0.05,
  data_min = 5,
  envStats_summary = FALSE,
  stat_lab = "Average",
  FUN = function(x) mean(x, na.rm = TRUE),
  ...
)

Arguments

swmpr_in input swmpr object

... additional arguments passed to other methods. See assign_season

param chr string of variable to plot

alpha num, alpha value to use to significance test. Defaults to 0.05.

data_min num, the minimum number of observations required to perform the analysis. Defaults to 5
sk_seasonal

envStats_summary

logical, should the standard EnvStats::kendallSeasonalTrendTest be returned? Defaults to FALSE. See Details for more information.

stat_lab

chr, label for the summary statistic defined in FUN. Defaults to "Average".

FUN

function used to aggregate seasonal SWMP data.

Details

This function performs a seasonal kendall test on seasonally aggregated values using kendallSeasonalTrendTest.

Data are aggregated on a user-specified seasonal basis using the FUN argument. For example, using default settings, sk_seasonal would perform a seasonal kendall test on average monthly values. However, if the user set FUN = min(x,na.rm = TRUE) then a seasonal kendall would be performed on monthly minimum values.

If EnvStats_summary = TRUE then the detailed output summary from kendallSeasonalTrendTest will be returned. If EnvStats_summary = FALSE then an abbreviated summary will be returned in a data.frame. The abbreviated summary contains the station name, the type of statistic used to summarize the data on a seasonal basis (specified by stat_lab), and the following results from kendallSeasonalTrendTest: tau, slope, p-value for the chi-square test, and the p-value for the trend test.

Value

Returns a data.frame object or a summary from EnvStats::kendallSeasonalTrendTest

Author(s)

Julie Padilla

See Also

assign_season, y_labeler, kendallSeasonalTrendTest

Examples

dat_wq <- elksmwq
dat_wq <- qaqc(dat_wq, qaqc_keep = c(0, 3, 5))
x <- sk_seasonal(dat_wq, param = 'temp')
### sk_tidy

**Tidy Seasonal Kendall Results**

**Description**

Tidy results from `kendallSeasonalTrendTest`

**Usage**

```r
sk_tidy(data, station, param, stat, alpha = 0.05)
```

**Arguments**

- `data`: a `htest` object produced by `kendallSeasonalTrendTest`
- `station`: chr string sampling station
- `param`: chr string of variable to plot
- `stat`: chr, label to be used for statistic used to group data
- `alpha`: num, significance level. Defaults to 0.05

**Details**

A helper function used by `sk_seasonal` to return a table of tidied values.

**Value**

Returns a `data.frame` of results from `kendallSeasonalTrendTest`

**Author(s)**

Julie Padilla

### std_param_check

**Standard Parameter Check**

**Description**

Determine if a parameter is one of the standard SWMP parameters

**Usage**

```r
std_param_check(param)
```

**Arguments**

- `param`: chr string of variable abbreviation
Details

A helper function used internally by several plotting functions to determine if parameter has a standard y-axis label. To accommodate the needs of the reserve-level annual report, this function also recognizes dissolved organic phosphorus (DIP) and dissolved inorganic nitrogen (DIN) as standard parameters.

Value

Returns TRUE or FALSE

Author(s)

Julie Padilla

Examples

std_param_check('do_mgl')

std_param_check('nitrogen')

summarise_handoff_files

Description

Summarise the seasonal kendall results from reserve level report hand-off files

Usage

summarise_handoff_files(path, param, res_region = NULL)

Arguments

path chr string of full path to .csv handoff files

param chr string of variable to summarise

res_region a data.frame of look-up values that match 3-letter NERR site ids with regions

Details

This function is intended for use with the NERRS national level reporting scripts. It returns a data.frame that summarises the result of the reserve level seasonal kendall trend analyses found in the hand-off files generated by the reserve level reporting scripts. The summary groups reserves into regional classifications based on user-specified regions given in res_region.
Value
Returns a data.frame

Author(s)
Julie Padilla

---

**threshold_criteria_plot**

*Water Quality Threshold Plot For Parameters With Criteria*

Description
Observed data compared against user-defined water quality thresholds

Usage

```r
threshold_criteria_plot(swmpr_in, ...) # S3 method for class 'swmpr'
threshold_criteria_plot(
  swmpr_in, 
  param = NULL, 
  rng = NULL, 
  thresholds = NULL, 
  threshold_labs = c("Good", "Fair", "Poor"), 
  threshold_cols = c("#ABD9E9", "#FFFFCC", "#FEC596"), 
  crit_threshold = NULL, 
  log_trans = FALSE, 
  monthly_smooth = FALSE, 
  plot_title = FALSE, 
  ... 
)
```

Arguments

- `swmpr_in` input swmpr object
- `...` additional arguments passed to other methods. See `y_labeler`.
- `param` chr string of the variable to plot
- `rng` num, years to include in the plot. This variable can either be one year (e.g., `rng = 2012`), or two years (e.g. `rng = c(2012, 2016)`). If range is not specified then the entire data set will be used.
- `thresholds` numeric vector, numeric criteria that will be plotted in the background
- `threshold_labs` chr vector of labels for categories created by `thresholds`
- `threshold_cols` chr vector of color values for categories created by `thresholds`
Threshold Criteria Plot

- **crit_threshold**
  - num, value at which the critical threshold line should be plotted. Typically the same value used to establish the 'Poor' threshold.

- **log_trans**
  - logical, should y-axis be log? Defaults to FALSE

- **monthly_smooth**
  - logical, calculate a monthly average? Defaults to FALSE

- **plot_title**
  - logical, should the station name be included as the plot title? Defaults to FALSE

**Details**

This function visualizes exceedances of numeric criteria which are specified using thresholds. Suggested numeric criteria for several parameters (dissolved oxygen, dissolved inorganic phosphorus, dissolved inorganic nitrogen, and chlorophyll-a) can be found in the USEPA National Coastal Condition Report (2012).

If the parameter of interest does not have numeric criteria, then threshold_percentile_plot is recommended.

**Value**

Returns a `ggplot` object

**Author(s)**

Julie Padilla

**References**


**See Also**

`ggplot`, `y_labeler`

**Examples**

```
data(apacpwq)
dat_wq <- apacpwq

dat_wq <- qaqc(dat_wq, qaqc_keep = c(0, 3, 5))

## Due to the volume of instantaneous data, these plots are a bit slow
x <-
  threshold_criteria_plot(dat_wq, param = 'do_mgl'
    , rng = 2012
    , thresholds = c(2, 5)
    , threshold_labs = c('Poor', 'Fair', 'Good')
    , monthly_smooth = TRUE
    , threshold_cols = c('#FEC596', 'FFFFCC', '#ABD9E9'))

y <-
```
threshold_criteria_plot(dat_wq, param = 'do_mgl'
  , thresholds = c(2, 5)
  , threshold_labs = c('Poor', 'Fair', 'Good')
  , threshold_cols = c('#FEC596', '#FFFFCC', '#ABD9E9'))

z <-
threshold_criteria_plot(dat_wq, param = 'do_mgl'
  , rng = 2012
  , thresholds = c(2, 5)
  , threshold_labs = c('Poor', 'Fair', 'Good')
  , threshold_cols = c('#FEC596', '#FFFFCC', '#ABD9E9')
  , monthly_smooth = TRUE)

## A few examples with only two thresholds
x1 <-
threshold_criteria_plot(dat_wq, param = 'do_mgl'
  , rng = 2012
  , thresholds = c(2, 2)
  , monthly_smooth = TRUE)

y1 <-
threshold_criteria_plot(dat_wq, param = 'do_mgl'
  , rng = 2012
  , thresholds = c(5, 5)
  , monthly_smooth = TRUE)

z1 <-
threshold_criteria_plot(dat_wq, param = 'do_mgl'
  , rng = 2012
  , thresholds = c(2, 5)
  , threshold_labs = c('Poor', 'Good', 'Poor')
  , threshold_cols = c('#FEC596', '#ABD9E9', '#FEC596')
  , monthly_smooth = TRUE)

data(apacpnut)
dat_nut <- apacpnut

dat_nut <- qaqc(dat_nut, qaqc_keep = c(0, 3, 5))
dat_nut <- rem_reps(dat_nut)

x2 <-
threshold_criteria_plot(dat_nut, param = 'chla_n'
  , thresholds = c(2, 5)
  , threshold_labs = c('Good', 'Fair', 'Poor'))
threshold_identification

Tabulate Threshold Exceedances

Description

Tabulate user-specified threshold exceedances

Usage

threshold_identification(swmpr_in, ...)

## S3 method for class 'swmpr'
threshold_identification(
  swmpr_in, 
  param, 
  parameter_threshold, 
  threshold_type, 
  time_threshold = NULL, 
  ... 
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>swmpr_in</td>
<td>input swmpr object</td>
</tr>
<tr>
<td>...</td>
<td>arguments passed to other methods</td>
</tr>
<tr>
<td>param</td>
<td>vector of parameters to evaluate</td>
</tr>
<tr>
<td>parameter_threshold</td>
<td>vector of numerical thresholds to evaluate parameters against</td>
</tr>
<tr>
<td>threshold_type</td>
<td>vector of logical operators ('&lt;', '&gt;', '&lt;=', '&gt;=', '==', '!=')</td>
</tr>
<tr>
<td>time_threshold</td>
<td>The amount of time an event must last to be counted (in hours)</td>
</tr>
</tbody>
</table>
threshold_identification

Details

This function creates tabular summary of events when a user-specified threshold is exceeded. Before using this function, the user must apply \texttt{setstep} to normalize the datetimestamp time step. For MET and WQ data, the user must specify \texttt{time\_threshold}. This argument is the minimum duration that an event must last in order to be counted. For example, if \texttt{time\_threshold = 2}, \texttt{param = \textquote{do\_mgl}}, \texttt{parameter\_threshold = 2}, and \texttt{threshold\_type = \textquote{\textless}} then dissolved oxygen must be lower than 2 mg/L for more than two hours or the event will not be summarized in the final table. For NUT parameters, all exceedances are included in the tabular summary.

Recommended thresholds for chlorophyll-a, dissolved inorganic nitrogen, dissolved inorganic phosphorus, and dissolved oxygen can be found in the National Coastal Condition Assessment 2010 (USEPA 2016)

Value

Returns a data frame of threshold exceedances by parameter

Author(s)

Julie Padilla

References


Examples

```r
data("apacpwq")
wq <- apacpwq
dat_wq <- qaqc(wq, qaqc_keep = c(0, 3, 5))
dat_wq <- setstep(dat_wq)

wq_pars<- threshold_identification(dat_wq, param = c("do\_mgl", "ph", "temp"),
                                 parameter_threshold = c(2, 5, 30),
                                 threshold_type = c("\textless", ",\textless", ",\textgreater"),
                                 time_threshold = 2)

wq_par<- threshold_identification(dat_wq, param = c("do\_mgl"),
                                 parameter_threshold = c(2),
                                 threshold_type = c("\textless"),
                                 time_threshold = 2)
```

```r
## time\_threshold and setstep are not necessary for monthly parameters
data("apacpnut")
nut <- apacpnut

dat_nut <- qaqc(nut, qaqc_keep = c(0, 3, 5))
nut_pars <- threshold_identification(dat_nut, param = c("chla\_n", "po4f")
```

```r
```
threshold_percentile_plot

Threshold Percentile Plot

Description

Observed data compared against user-defined percentiles

Usage

threshold_percentile_plot(swmpr_in, ...)

## S3 method for class 'swmpr'
threshold_percentile_plot(
  swmpr_in,
  param = NULL,
  hist_rng = NULL,
  target_yr = NULL,
  percentiles = c(0.05, 0.95),
  free_y = FALSE,
  by_month = FALSE,
  log_trans = FALSE,
  converted = FALSE,
  plot_title = FALSE,
  ...
)

Arguments

swmpr_in        input swmpr object
...             additional arguments passed to other methods (not used for this function).
param           chr, variable to plot
hist_rng        num, years to include in the plot. This variable can either be one year (e.g., hist_rng = 2012), or two years (e.g. hist_rng = c(2012, 2016)). If range is not specified then the entire data set will be used.
target_yr  num, year of interest for plotting. If not specified, the entire data set will be plotted.

percentiles  num, percentiles to calculate (maximum: 2). Defaults to 5th and 95th percentiles.

free_y  logical, should the y-axis be free? Defaults to FALSE. If FALSE, defaults to zero, unless negative values are present. If TRUE, y-axis limits are selected by ggplot

by_month  logical, should percentiles be calculated on a monthly basis? Defaults to FALSE

log_trans  logical, should y-axis be log? Defaults to FALSE

converted  logical, were the units converted from the original units used by CDMO? Defaults to FALSE. See y_labeler for details.

plot_title  logical, should the station name be included as the plot title? Defaults to FALSE

Details
This function provides an alternative to threshold_criteria_plot. For parameters that may not have numeric threshold criteria, a percentile threshold can be used instead. For a one-tailed analysis, the 90-th percentile is recommended. For a two-tailed analysis, the 5-th and 95-th percentiles are recommended.

Using by_month, the user can specify whether the percentiles should be calculated on a monthly basis or by using the entire data set.

Recommended thresholds for chlorophyll-a, dissolved inorganic nitrogen, dissolved inorganic phosphorus, and dissolved oxygen can be found in the National Coastal Condition Assessment 2010 (USEPA 2016)

Value
Returns a ggplot object

Author(s)
Julie Padilla

References

See Also
ggplot

Examples
```r
dat_wq <- qaqc(elksmwq, qaqc_keep = c(0, 3, 5))
dat_wq <- subset(dat_wq, subset = '2007-01-01 0:00', operator = '>=')

x <-
    threshold_percentile_plot(dat_wq, param = 'do_mgl')```
threshold_summary

Summary Plots for Threshold Identification

Description

Summary plots for threshold identification analysis

Usage

threshold_summary(swmpr_in, ...)

## S3 method for class 'swmpr'
threshold_summary(
  swmpr_in,
  param = NULL,
  summary_type = c("month", "season", "year"),
  parameter_threshold = NULL,
threshold_summary

threshold_type = NULL,
time_threshold = NULL,
converted = FALSE,
pal = "Set3",
plot_title = FALSE,
plot = TRUE,
label_y_axis = TRUE,
...
)

Arguments

swmpr_in  input swmpr object

...  additional arguments passed to other methods. See assign_season for more
details.

param  chr string of variable to plot (one only)

summary_type  Choose from month, season, or year aggregation

parameter_threshold  vector of numerical thresholds to evaluate parameters against

threshold_type  vector of logical operators ('<', '>', '<=', '>=', '==', '!=')

time_threshold  The amount of time an event must last to be counted (in hours)

converted  logical, were the units converted from the original units used by CDMO? De-
defaults to FALSE. See y_labeler for details.

pal  Select a palette for boxplot fill colors. See scale_fill_brewer for more de-
details.

plot_title  logical, should the station name be included as the plot title? Defaults to FALSE

plot  logical, should a plot be returned? Defaults to TRUE

label_y_axis  logical, include label for y-axis?

Details

This function provides a graphical or tabular summary of the results from threshold_identification. The user can summarize results on a monthly, seasonal, or annual basis by specifying summary_type = c(’month’, ’season’, ’year’). If summary_type = ’season’, then the user should also define season, season_names, and season_start, as required by lcodeassign_season. The user can specify ’month’ for nutrient parameters, but this is not recommended and will produce a warning.

Recommended thresholds for chlorophyll-a, dissolved inorganic nitrogen, dissolved inorganic phosphorus, and dissolved oxygen can be found in the National Coastal Condition Assessment 2010 (USEPA 2016)

Value

Returns a ggplot object (if plot = TRUE) or a dataframe (if plot = FALSE)

Author(s)

Julie Padilla
References


See Also

assign_season, ggplot, threshold_identification, scale_fill_brewer

Examples

## Water quality examples

data(apacpwq)
dat_wq <- qaqc(apacpwq, qaqc_keep = c(0, 3, 5))
dat_wq <- SWMPr::setstep(dat_wq)

x <-
  threshold_summary(dat_wq, param = 'do_mgl', parameter_threshold = 2,
  threshold_type = '<', time_threshold = 2, summary_type = 'month',
  plot_title = TRUE)

y <-
  threshold_summary(dat_wq, param = 'do_mgl', parameter_threshold = 2,
  threshold_type = '<', time_threshold = 2, summary_type = 'season',
  season_grps = list(c(1,2,3), c(4,5,6), c(7,8,9), c(10, 11, 12)),
  season_names = c('Winter', 'Spring', 'Summer', 'Fall'),
  season_start = 'Winter', plot_title = TRUE)

# Nutrient examples

data(apacpnut, qaqc_keep = c(0, 3, 5))

x <-
  threshold_summary(dat_nut, param = 'chla_n',
  parameter_threshold = 10,
  threshold_type = '>', summary_type = 'month',
  plot_title = TRUE)

y <-
  threshold_summary(dat_nut, param = 'chla_n', parameter_threshold = 10,,
  threshold_type = '>', summary_type = 'season',
  season_grps = list(c(1,2,3), c(4,5,6), c(7,8,9), c(10, 11, 12)),
  season_names = c('Winter', 'Spring', 'Summer', 'Fall'),
  season_start = 'Winter', plot_title = TRUE)

z <-
  threshold_summary(dat_nut, param = 'chla_n', parameter_threshold = 10,
  threshold_type = '>', summary_type = 'year',
  plot_title = TRUE, plot = TRUE)
**title_labeler**  
*Generate Plot Title Based on NERR Site ID*

**Description**
Generate a plot title based on SWMP station abbreviation

**Usage**
```r
title_labeler(nerr_site_id)
```

**Arguments**
- `nerr_site_id`  
  chr string of NERR site id

**Details**
A helper function used internally by several plotting functions to generate plot titles.

**Value**
Returns character vector

**Author(s)**
Julie Padilla

**Examples**
```r
ttl <- title_labeler('elkapwq')
```

---

**update_sampling_stations**  
*Update reserve sampling stations*

**Description**
Script to modify the internal sampling_sites.rda file for adding new reserves or for making changes to existing sampling station locations.

**Usage**
```r
update_sampling_stations(file_path, file_name = "sampling_stations.csv")
```
Arguments

- file_path: path to directory with new file
- file_name: name of new csv file

Details

This is a standalone function used to replace the internal SWMPExtension sampling sites data table used by the get_sites.R function. It reads a csv-formatted file of all NERRS SWMP stations that the user has downloaded from the CDMO SWMP station website: https://cdmo.baruch.sc.edu/data/swmp-stations/.

This downloaded csv-formatted file and its location are the only input arguments to update_sampling_station(). The current station information is loaded from data/sampling_stations.rda, a copy of which is written out as data/sampling_stations_backup.rda, and the new data file is read, formatted appropriately, and written out as data/sampling_stations.rda.

Note: This function need only be run when new reserves are added, stations are moved, etc.

Value

Returns TRUE on a successful run, FALSE on a failure.

Author(s)

Dave Eslinger

Examples

# Provide a bad file name to get error message
x <- update_sampling_stations("data","bad_file_name.csv")
print(x)

Description

US state boundaries from the US Census Bureau’s MAF/TIGER geographic database in EPSG:4269, with all data except state FIPS codes and land area removed.

Usage

data('us_4269')

Format

A sf(DataFrame) object
y_count_labeler

Source

US Census Bureau

References

United States Census Bureau. Data accessed from the US Census Bureau website: https://www2.census.gov/geo/tiger/GENZ2018/shp/cb_2018_us_county_20m.zip; accessed 13 March 2020

---

**y_count_labeler**  
*Generate y-axis Label Based on SWMP Parameter Abbreviation*

**Description**

Generate a y-axis label based on SWMP parameter abbreviation and threshold criteria

**Usage**

```r
y_count_labeler(
  param,
  parameter_threshold,
  threshold_type,
  time_threshold = NULL,
  converted = FALSE
)
```

**Arguments**

- `param`: chr string of variable abbreviation
- `parameter_threshold`: vector of numerical thresholds to evaluate parameters against
- `threshold_type`: vector of logical operators (`'<'`, `'>'`, `'<='`, `'>='`, `'=='`, `'!='`)  
- `time_threshold`: The amount of time an event must last to be counted (in hours)  
- `converted`: logical, should the parameter label units be converted from metric to english?  
  Defaults to FALSE. Currently available for `temp`, `depth`, `cdepth`, `level`, `clevel`, `atemp`, `wspd`, `maxwspd`, and `totprcp`

**Details**

A helper function used internally by several plotting functions to generate y-axis labels. This function does not convert sample results from metric to english. It only adjusts the units in the y-axis label.

**Value**

Returns character vector or an unevaluated expression
y_labeler

Author(s)
Julie Padilla

Examples

```r
y_lab <- y_count_labeler(param = 'do_mgl', parameter_threshold = 2, threshold_type = '<', time_threshold = 2, converted = FALSE)
```

---

**y_labeler**  
*Generate y-axis Label Based on SWMP Parameter Abbreviation*

**Description**

Generate a y-axis label based on SWMP parameter abbreviation

**Usage**

```r
y_labeler(param, converted = FALSE)
```

**Arguments**

- `param`: chr string of variable abbreviation
- `converted`: logical, should the parameter label units be converted from metric to english? Defaults to FALSE. Currently available for temp, depth, cdepth, level, clevel, atemp, wspd, maxwspd, and totprcp

**Details**

A helper function used internally by several plotting functions to generate y-axis labels. This function does not convert sample results from metric to english. It only adjusts the units in the y-axis label.

**Value**

Returns character vector or an unevaluated expression

**Author(s)**
Julie Padilla

**Examples**

```r
y_lab <- y_labeler('do_mgl')
```
Index

* Reporting
  create_sk_national_ft_results, 11
* analyze
  annual_range, 3
  assign_season, 5
  base_map, 6
  historical_daily_range, 20
  historical_range, 22
  national_sk_map, 27
  raw_boxplot, 29
  res_custom_map, 32
  res_custom_sk_map, 34
  res_local_map, 36
  res_national_map, 38
  res_sk_map, 40
  seasonal_barplot, 43
  seasonal_boxplot, 45
  seasonal_dot, 48
  set_date_break_labs, 51
  sk_seasonal, 52
  threshold_criteria_plot, 56
  threshold_identification, 59
  threshold_percentile_plot, 61
  threshold_summary, 63
* data maintenance
  update_sampling_stations, 66
* datasets
  cbm_spatial, 8
  counties_4269, 8
  elk_spatial, 13
  elknnmu1, 12
  elksmwq, 13
  sampling_stations, 42
  sampling_stations_backup, 43
  us_4269, 67
* mapping
  geographic_unique_stations, 16
* miscellaneous
  lm_p_labs, 26
  remove_inf_and_nan, 30
  reserve_locs, 31
  set_date_breaks, 50
  set_date_breaks_minor, 51
  sk_tidy, 54
  std_param_check, 54
  title_labeler, 66
  y_count_labeler, 68
  y_labeler, 69
* reporting
  create_sk_flexttable_list, 9
  create_sk_national_ft_reserves, 10
  ft_col_names, 14
  generate_results_table, 15
  generate_station_table, 15
  get_reserve, 17
  get_shp_name, 18
  get_site_code, 19
  get_site_coordinates, 20
  get_sites, 18
  load_shp_file, 27
  summarise_handoff_files, 55
* retrieve
  import_local_nut, 24
  all_params, 26
  all_params_dtrng, 26
  annual_range, 3
  assign_season, 4, 5, 23, 24, 29, 30, 44–49, 52, 53, 64, 65
  base_map, 6, 32, 34, 36, 40
  cbm_spatial, 8
  counties_4269, 8
  create_sk_flexttable_list, 9, 14–16
  create_sk_national_ft_reserves, 10
  create_sk_national_ft_results, 11
  data.frame, 12, 13, 42, 43
elk_spatial, 13
elknmnut, 12
elksmwq, 13
flextable, 10–12, 14
ft_col_names, 14
generate_results_table, 15
generate_station_table, 15
geographic_unique_stations, 16
get_reserve, 17
get_shp_name, 18
get_site_code, 19
get_site_coordinates, 20
get_sites, 18
ggplot, 4, 22, 24, 28, 30, 39, 44, 45, 47, 49, 57, 62, 64, 65
historical_daily_range, 20
historical_range, 22
import_local, 25
import_local_nut, 24
kendallSeasonalTrendTest, 53, 54
lm, 26
lm_p_labs, 26
load_shp_file, 27
national_sk_map, 27
raw_boxplot, 29
rem_reps, 25, 26
remove_inf_and_nan, 30
res_custom_map, 32
res_custom_sk_map, 34
res_local_map, 16, 27, 36
res_national_map, 31, 38
res_sk_map, 27, 40
reserve_locs, 31
sampling_stations, 42
sampling_stations_backup, 43
scale_fill_brewer, 64, 65
scale_x_datetime, 50–52
seasonal_barplot, 43
seasonal_boxplot, 45
seasonal_dot, 26, 48
set_date_breaks, 50, 52
set_date_breaks_minor, 51
setstep, 60
sf, 27
single_param, 26
sk_seasonal, 9, 11, 15, 16, 35, 52, 54
sk_tidy, 54
SpatialPolygons, 8, 13
std_param_check, 54
summarise_handoff_files, 55
threshold_criteria_plot, 56, 62
threshold_identification, 59, 65
threshold_percentile_plot, 61
threshold_summary, 63
title_labeler, 66
update_sampling_stations, 66
us_4269, 67
y_count_labeler, 68
y_labeler, 4, 22, 24, 29, 30, 45, 49, 53, 56, 57, 69