

Package: tbeptools (via r-universe)

October 5, 2026

Title Data and Indicators for the Tampa Bay Estuary Program

Version 3.1.1.9001

Date 2026-10-05

Description Several functions are provided for working with Tampa Bay Estuary Program data and indicators, including the water quality report card, tidal creek assessments, Tampa Bay Nekton Index, Tampa Bay Benthic Index, seagrass transect data, habitat report card, and fecal indicator bacteria. Additional functions are provided for miscellaneous tasks, such as reference library curation.

Depends R (>= 4.1), methods

Imports dataRetrieval, dplyr, flextable, fs, ggplot2 (>= 3.2.0), httr, httr2, jsonlite, leaflet, lubridate, patchwork, plotly, purrr, reactable, readxl, rvest, sf, terra, tibble, tidyr, tools

Suggests bookdown, ftExtra, ggmap, ggnewscale, ggrepel, ggspatial, htmlwidgets, knitr, mockery, rmarkdown, testthat (>= 2.1.0), covr

License MIT + file LICENSE

Encoding UTF-8

VignetteBuilder knitr

LazyData true

LazyDataCompression xz

RoxygenNote 7.3.3

Roxygen list(markdown = TRUE)

URL <https://github.com/tbep-tech/tbeptools>

BugReports <https://github.com/tbep-tech/tbeptools/issues>

Config/pak/sysreqs libabsl-dev libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev libgdal-dev gdal-bin libgeos-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev

Repository <https://fawda123.r-universe.dev>

Date/Publication 2026-10-05 15:03:21 UTC

RemoteUrl <https://github.com/tbep-tech/tbeptools>

RemoteRef HEAD

RemoteSha c2761857860f7f5f7e3e9b32c23bef0821d37f4d

Contents

acres	5
ambispp	6
anlz_ambimed	7
anlz_ambiscr	8
anlz_attain	9
anlz_attainsite	10
anlz_avedat	11
anlz_avedatsite	11
anlz_enteromap	12
anlz_fibmap	13
anlz_fibmatrix	15
anlz_fibwetdry	17
anlz_hmpreport	18
anlz_hydroload	19
anlz_iwrraw	20
anlz_refs	21
anlz_sedimentaddtot	22
anlz_sedimentave	23
anlz_sedimentpel	24
anlz_sedimentpelave	25
anlz_splitdata	26
anlz_splitsstorms	27
anlz_tbbimed	28
anlz_tbbiscr	29
anlz_tbniave	29
anlz_tbnimet	30
anlz_tbniscr	31
anlz_tdlcrk	32
anlz_tdlcrkindic	32
anlz_transectave	33
anlz_transectavespp	34
anlz_transectocc	35
anlz_yrattain	36
benthicdata	37
bsmap	38
catchpixels	38
catchprecip	39
dbasin	40

enterodata	40
epcdata	41
fibdata	43
findata	44
fmstations	45
hcesdfibdata	46
hmptrgs	47
iwrrow	47
mancofibdata	48
pascofibdata	49
phytodata	50
polcofibdata	51
read_dlcurent	52
read_formbenthic	53
read_formfib	54
read_formfim	55
read_formphyto	56
read_formsediment	57
read_formtransect	58
read_formwq	59
read_formwqp	60
read_importbenthic	61
read_importentero	62
read_importepc	64
read_importfib	65
read_importfim	66
read_importphyto	67
read_importprism	68
read_importrain	71
read_importrainmany	72
read_importsealevels	73
read_importsediment	75
read_importwq	76
read_importwqp	77
read_importwqwa	78
read_importwqwin	79
read_prism_rasters	80
read_transect	81
seagrass	82
sealevelstations	83
sedimentdata	83
sgmanagement	85
sgseg	86
show_ambimatrix	87
show_ambitab	88
show_ambitrend	89
show_annualassess	90
show_boxplot	91

show_complot	93
show_enteromap	94
show_fibmap	95
show_fibmatmap	96
show_fibmatrix	98
show_hmpreport	101
show_matrix	102
show_matrixplotly	104
show_ratab	105
show_reactable	106
show_seagrasscoverage	107
show_sedimentalratio	108
show_sedimentave	109
show_sedimentmap	111
show_sedimentpelave	112
show_sedimentpelaveplotly	113
show_sedimentpelmap	114
show_segmatrix	115
show_segplotly	116
show_sitemap	117
show_sitesegmap	118
show_splitbarplot	119
show_tbbimatrix	121
show_tbnimatrix	122
show_tbniscr	123
show_tbniscrall	124
show_tbniscrplotly	125
show_tbniscrseas	126
show_tdlcrk	127
show_tdlcrkindic	128
show_tdlcrkindiccdf	129
show_tdlcrkline	130
show_tdlcrkmatrix	131
show_tdlcrkradar	132
show_thrplot	133
show_transect	134
show_transectavespp	135
show_transectmatrix	137
show_transectsum	139
show_wqmatrix	140
stations	141
subtacres	142
swfwmdtbseg	142
swfwmdtbseg2024	143
targets	144
tbniref	145
tbnispp	146
tbseg	147

tbsegdetail	148
tbsegdetailbcbs	148
tbseglines	149
tbsegshed	150
tbshed	150
tidalcreeks	151
tidaltargets	152
transect	153
trnlns	154
trnpts	154
util_dataseq	155
util_fibicons	155
util_fiblevs	156
util_fmymrng	156
util_html	157
util_importwqp	158
util_importwqwa	159
util_importwqwin	160
util_map	161
util_organ	161
util_rain	162
Index	163

acres	<i>Tampa Bay intertidal and supratidal land use and cover</i>
-------	---

Description

Tampa Bay intertidal and supratidal land use and cover for Habitat Master Plan reporting

Usage

acres

Format

A data frame

Examples

```
## Not run:
load(url("https://github.com/tbep-tech/hmpu-workflow/raw/master/data/acres.RData"))

save(acres, file = 'data/acres.RData', compress = 'xz')

## End(Not run)
```

ambispp

*AMBI species group assignments for Tampa Bay benthic taxa***Description**

AMBI species group assignments for Tampa Bay benthic taxa

Usage

ambispp

Format

A data frame with 1983 rows and 5 variables:

TaxaListID int, join key to taxacounts in [benthicdata](#)

NAME chr, taxon name

AMBIGroupID int, ecological group 1-5 for conventional AMBI (Gillett et al. 2015), NA if unclassified

TBAMBIGroupID int, ecological group 1-5 for Tampa Bay-specific AMBI, NA if unclassified

AMBIListSource chr, provenance of the AMBI group assignment (Gulf, National, or assumption)

Details

Ecological groups range from I (most sensitive to pollution) to V (most tolerant). The conventional AMBI groups (AMBIGroupID) follow published literature (Gillett et al. 2015), with Gulf of Mexico scores preferred over National scores, and family-level assumptions applied where species-level scores were unavailable. The Tampa Bay-specific groups (TBAMBIGroupID) are based on local sediment sample assignments.

This table is used as the species lookup in [anlz_ambiscr](#) to join ecological group assignments to observed taxa counts in [benthicdata](#).

References

Gillett, D.J., Weisberg, S.B., Grayson, T., Hamilton, A., Hansen, V., Leppo, E.W., Pelletier, M.C., et al. (2015). Effect of ecological group classification schemes on performance of the AMBI benthic index in US coastal waters. *Ecological Indicators*, 50, 99-107. [doi:10.1016/j.ecolind.2014.11.005](#)

Examples

```
## Not run:
library(readxl)
library(dplyr)

ambispp <- read_excel('data-raw/AMBI_EcoSource_ReportDeliverableToDK.xlsx') |>
  select(TaxaListID, NAME, AMBIGroupID, TBAMBIGroupID = TBAMBIGroupId, AMBIListSource)
```

```
save(ambispp, file = 'data/ambispp.RData', compress = 'xz')

## End(Not run)
```

anlz_ambimed	<i>Get annual medians of AMBI scores by bay segment</i>
--------------	---

Description

Get annual medians of AMBI scores by bay segment

Usage

```
anlz_ambimed(
  ambisqr,
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB", "All"),
  rev = FALSE,
  yrrng = c(1993, 2024),
  window = TRUE
)
```

Arguments

ambisqr	input data frame as returned by anlz_ambisqr ; the AMBI variant (conventional or Tampa Bay-specific) is detected automatically from the column names
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB", "All"
rev	logical if factor levels for bay segments are reversed
yrrng	numeric indicating year ranges to evaluate
window	logical indicating whether to use a rolling 5-year window (default TRUE) or single year values (FALSE) for the bay segment categories, see details

Details

An additional summary is provided for the entire bay as an unweighted summary across categories ("All").

Only sampling funded by TBEP and as part of the routine EPC benthic monitoring program are included in the final categories.

The default behavior is to use a rolling five-year window to calculate the percent of sites in each AMBI category by bay segment. This applies only to years 2005 and later, where the counts from the current year and the prior four years are summed to calculate the percentages. This is intended to help smooth out inter-annual variability due to reduced sampling effort from 2005 to present. If window = FALSE, then only single year values are used.

The AMBI category column (AMBICat or TBAMBICat) is based on the adjusted AMBI score (0-10 scale) using the classification thresholds defined in [anlz_ambisqr](#).

Value

A data frame of annual percent composition by AMBI category and bay segment

Examples

```
ambiscr <- anlz_ambiscr(benthicdata)
anlz_ambimed(ambiscr)
```

anlz_ambiscr	<i>Get AMBI scores for benthic stations</i>
--------------	---

Description

Get AMBI scores for benthic stations

Usage

```
anlz_ambiscr(benthicdata, type = c("AMBI", "AMBI-TB"))
```

Arguments

benthicdata	nested tibble formatted from read_importbenthic
type	chr string indicating which AMBI variant to calculate, one of "AMBI" (conventional, based on Gillett et al. 2015) or "AMBI-TB" (Tampa Bay-specific group assignments)

Details

This function calculates the AZTI Marine Biotic Index (AMBI) for benthic stations in Tampa Bay. The index assigns each taxon to one of five ecological groups (I = most sensitive, V = most tolerant to pollution) and computes a Biotic Coefficient (BC) from the proportion of taxa in each group.

The Biotic Coefficient is calculated as:

$$BC = (0 \times \%G1 + 1.5 \times \%G2 + 3 \times \%G3 + 4.5 \times \%G4 + 6 \times \%G5) / 100$$

where \

An adjusted AMBI score on a 0-10 scale is also returned:

$$AdjustedAMBI = (7 - BC) \times (10/7)$$

Higher adjusted scores indicate healthier benthic conditions. Azoic stations (empty samples, TaxaListID == 1942 in [benthicdata](#)) receive BC = 7 and adjusted score = 0.

Two variants are supported via the type argument. "AMBI" uses ecological group assignments from published literature (AMBIGroupID in [ambispp](#)). "AMBI-TB" uses Tampa Bay-specific assignments (TBAMBIGroupID in [ambispp](#)). The calculation is identical, only the group assignment column differs.

Value

A data frame of AMBI scores for each station, with columns from [benthicdata](#) stations joined to:

- TotalGroupCount** int, total number of classified taxa records used in the calculation
- PercentG1 - PercentG5** num, percentage of taxa records in each ecological group
- BC** num, Biotic Coefficient (0-6 scale, or 7 for azoic)
- AMBI or TBAMBI** num, adjusted AMBI score (0-10 scale)
- SitePollutionClassification** chr, pollution category based on BC
- BioticIndex** chr, biotic index value (0-6 or Azoic)
- DominatingEcologicalGroup** chr, dominating ecological group based on BC
- BenthicCommunityHealth** chr, community health descriptor based on BC
- AMBICat or TBAMBICat** chr, adjusted AMBI category based on the 0-10 scale

References

Gillett, D.J., Weisberg, S.B., Grayson, T., Hamilton, A., Hansen, V., Leppo, E.W., Pelletier, M.C., et al. (2015). Effect of ecological group classification schemes on performance of the AMBI benthic index in US coastal waters. Ecological Indicators, 50, 99-107. [doi:10.1016/j.ecolind.2014.11.005](#)

Examples

```
anlz_ambiscr(benthicdata)
anlz_ambiscr(benthicdata, type = 'AMBI-TB')
```

anlz_attain	<i>Get attainment categories</i>
-------------	----------------------------------

Description

Get attainment categories for each year and bay segment using chlorophyll and light attenuation

Usage

```
anlz_attain(avedat, magdurout = FALSE, trgs = NULL)
```

Arguments

- avedat result returned from [anlz_avedat](#)
- magdurout logical indicating if the separate magnitude and duration estimates are returned
- trgs optional data.frame for annual bay segment water quality targets, defaults to [targets](#)

Value

A data.frame for each year and bay segment showing the attainment category

Examples

```
avedat <- anlz_avedat(epcdata)
anlz_attain(avedat)
```

anlz_attainsite	<i>Get site attainments</i>
-----------------	-----------------------------

Description

Get site attainment categories for chlorophyll or light attenuation

Usage

```
anlz_attainsite(
  avedatsite,
  thr = c("chl", "la"),
  trgs = NULL,
  yrrng = NULL,
  thrs = FALSE
)
```

Arguments

avedatsite	result returned from anlz_avedatsite
thr	chr string indicating with water quality value and appropriate threshold to to plot, one of "chl" for chlorophyll and "la" for light availability
trgs	optional data.frame for annual bay segment water quality targets, defaults to targets
yrrng	optional numeric value for year to return, defaults to all
thrs	logical indicating if attainment category is relative to targets (default) or thresholds

Details

This function is a simplification of the attainment categories returned by [anlz_attain](#). Sites are only compared to the targets/thresholds that apply separately for chlorophyll or light attenuation.

Value

a data.frame for each year and site showing the attainment category

Examples

```
avedatsite <- anlz_avedatsite(epcdata)
anlz_attainsite(avedatsite)
```

anlz_avedat	<i>Estimate annual means</i>
-------------	------------------------------

Description

Estimate annual means for chlorophyll and secchi data

Usage

```
anlz_avedat(epcdata, partialyr = FALSE)
```

Arguments

epcdata	data.frame formatted from read_importwq
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter

Value

Mean estimates for chlorophyll and secchi

Examples

```
# view average estimates  
anlz_avedat(epcdata)
```

anlz_avedatsite	<i>Estimate annual means by site</i>
-----------------	--------------------------------------

Description

Estimate annual means by site for chlorophyll and secchi data

Usage

```
anlz_avedatsite(epcdata, partialyr = FALSE)
```

Arguments

epcdata	data.frame formatted from read_importwq
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter

Value

Mean estimates for chlorophyll and secchi

Examples

```
# view average estimates
anlz_avedatsite(epcdata)
```

anlz_enteromap	<i>Assign threshold categories to Enterococcus data</i>
----------------	---

Description

Assign threshold categories to Enterococcus data

Usage

```
anlz_enteromap(
  fibdata,
  yrsel = NULL,
  mosel = NULL,
  areasel = NULL,
  wetdry = FALSE,
  precipdata = NULL,
  temporal_window = NULL,
  wet_threshold = NULL,
  assf = FALSE
)
```

Arguments

fibdata	data frame of Enterococcus sample data as returned by enterodata or anlz_fibwetdry
yrsel	optional numeric to filter data by year
mosel	optional numeric to filter data by month
areasel	optional character string to filter output by stations in the long_name column of enterodata, see details
wetdry	logical; if TRUE, incorporate wet/dry differences (this will result in a call to anlz_fibwetdry , in which case temporal_window and wet_threshold are required). If FALSE (default), do not differentiate between wet and dry samples.
precipdata	input data frame as returned by read_importtrain . columns should be: station, date (yyyy-mm-dd), rain (in inches). The object catchprecip has this data from 1995-2023 for select Enterococcus stations. If NULL, defaults to catchprecip .
temporal_window	numeric; required if wetdry is TRUE. number of days precipitation should be summed over (1 = day of sample only; 2 = day of sample + day before; etc.)
wet_threshold	numeric; required if wetdry is TRUE. inches accumulated through the defined temporal window, above which a sample should be defined as being from a 'wet' time period
assf	logical indicating if the data are further processed as a simple features object with additional columns for show_enteromap

Details

This function is based on [anlz_fibmap](#), but is specific to *Enterococcus* data downloaded via [read_importentero](#). It creates categories for mapping using [show_enteromap](#). Optionally, if samples have been defined as 'wet' or not via [anlz_fibwetdry](#), this can be represented via symbols on the map. Categories based on relevant thresholds are assigned to each observation. The categories are specific to *Enterococcus* in marine waters (class of 2 or 3M). A station is categorized into one of four ranges defined by the thresholds as noted in the `cat` column of the output, with corresponding colors appropriate for each range as noted in the `col` column of the output.

The `areasel` argument can indicate valid entries in the `long_name` column of `enterodata`. For example, use "Old Tampa Bay" for stations in the subwatershed of Old Tampa Bay, where rows in `enterodata` are filtered based on the the selection. All stations are returned if this argument is set as `NULL` (default). All valid options for `areasel` include "Old Tampa Bay", "Hillsborough Bay", "Middle Tampa Bay", "Lower Tampa Bay", "Boca Ciega Bay", or "Manatee River". One to any of the options can be used.

Value

A `data.frame` similar to `fibdata` if `assf = FALSE` with additional columns describing station categories and optionally filtered by arguments passed to the function. A `sf` object if `assf = TRUE` with additional columns for [show_enteromap](#).

Examples

```
anlz_enteromap(enterodata, yrsel = 2020, mosel = 9)

# differentiate wet/dry samples in that time frame
anlz_enteromap(enterodata, yrsel = 2020, mosel = 9, wetdry = TRUE,
               temporal_window = 2, wet_threshold = 0.5)

# as sf object
anlz_enteromap(enterodata, assf = TRUE)
```

`anlz_fibmap`

Assign threshold categories to Fecal Indicator Bacteria (FIB) data

Description

Assign threshold categories to Fecal Indicator Bacteria (FIB) data

Usage

```
anlz_fibmap(fibdata, yrsel = NULL, mosel = NULL, areasel = NULL, assf = FALSE)
```

Arguments

<code>fibdata</code>	input FIB data.frame as returned by <code>read_importfib</code> or <code>read_importwqp</code> , see details
<code>yrsel</code>	optional numeric value to filter output by years in <code>fibdata</code>
<code>mosel</code>	optional numeric value to filter output by month in <code>fibdata</code>
<code>areasel</code>	optional character string to filter output by stations in the area column of <code>fibdata</code> , see details
<code>assf</code>	logical indicating if the data are further processed as a simple features object with additional columns for <code>show_fibmap</code>

Details

This function is used to create FIB categories for mapping using `show_fibmap`. Categories based on relevant thresholds are assigned to each observation. The categories are specific to *E. coli* or *Enterococcus* and are assigned based on the station class as freshwater (class as 1 or 3F) or marine (class as 2 or 3M), respectively. A station is categorized into one of four ranges defined by the thresholds as noted in the `cat` column of the output, with corresponding colors appropriate for each range as noted in the `col` column of the output.

The input `fibdata` object can be one returned by either `read_importfib` or `read_importwqp`.

The `areasel` argument can indicate valid entries in the area column of `fibdata` from `read_importfib` or `mancofibdata`, `pascofibdata`, `polcofibdata`, or `hcesdfibdata` from `read_importwqp`. For example, use either "Alafia River" or "Hillsborough River" to show stations for the corresponding river basins, where rows in `fibdata` are filtered based on the the selection. All valid options for `areasel` for `fibdata` include "Alafia River", "Hillsborough River", "Cockroach Bay", "East Lake Outfall", "Hillsborough Bay", "Little Manatee", "Lower Tampa Bay", "McKay Bay", "Middle Tampa Bay", "Old Tampa Bay", "Palm River", "Tampa Bypass Canal", or "Valrico Lake". Valid options for `areasel` if the input data are from `read_importwqp` are any that are present in the area column for the respective input datasets (`mancofibdata`, `pascofibdata`, `polcofibdata`, or `polcofibdata`). One to any of the options can be used. All stations are returned if this argument is set as NULL (default). Not all areas may be present based on the year/month selection.

Value

A data.frame similar to the input if `assf = FALSE` with additional columns describing station categories and optionally filtered by arguments passed to the function. A `sf` object if `assf = TRUE` with additional columns for `show_fibmap`.

Examples

```
# assign categories to all
anlz_fibmap(fibdata)

# filter by year, month, and area
anlz_fibmap(fibdata, yrsel = 2020, mosel = 7, areasel = 'Alafia River')

# as sf object
anlz_fibmap(fibdata, assf = TRUE)
```

anlz_fibmatrix	<i>Analyze Fecal Indicator Bacteria categories over time by station or bay segment</i>
----------------	--

Description

Analyze Fecal Indicator Bacteria categories over time by station or bay segment

Usage

```
anlz_fibmatrix(
  fibdata,
  yrrng = NULL,
  stas = NULL,
  bay_segment = NULL,
  lagyr = 3,
  subset_wetdry = c("all", "wet", "dry"),
  precipdata = NULL,
  temporal_window = NULL,
  wet_threshold = NULL,
  warn = TRUE
)
```

Arguments

<code>fibdata</code>	input data frame as returned by read_importfib , read_importentero , or read_importwqp , see details
<code>yrrng</code>	numeric vector indicating min, max years to include, defaults to range of years in data, see details
<code>stas</code>	optional vector of stations to include, see details
<code>bay_segment</code>	optional vector of bay segment names to include, supercedes <code>stas</code> if provided, see details
<code>lagyr</code>	numeric for year lag to calculate categories, see details
<code>subset_wetdry</code>	character, subset data frame to only wet or dry samples as defined by <code>wet_threshold</code> and <code>temporal_window</code> ? Defaults to "all", which will not subset. If "wet" or "dry" is specified, anlz_fibwetdry is called using the further specified parameters, and the data frame is subsetted accordingly.
<code>precipdata</code>	input data frame as returned by read_importtrain . columns should be: station, date (yyyy-mm-dd), rain (in inches). The object catchprecip has this data from 1995-2023 for select Enterococcus stations. If NULL, defaults to catchprecip .
<code>temporal_window</code>	numeric; required if <code>subset_wetdry</code> is not "all". number of days precipitation should be summed over (1 = day of sample only; 2 = day of sample + day before; etc.)

wet_threshold	numeric; required if subset_wetdry is not "all". inches accumulated through the defined temporal window, above which a sample should be defined as being from a 'wet' time period
warn	logical to print warnings about stations with insufficient data, default TRUE

Details

This function is used to create output for plotting a matrix stoplight graphic for FIB categories by station. Each station (or bay segment) and year combination is categorized based on the likelihood of fecal indicator bacteria concentrations exceeding some threshold in a given year. For Enterococcus (marine), the default threshold is 130 CFU / 100 mL in a given year. For E. coli (fresh), the default threshold is 410 CFU / 100 mL. The proportions are categorized as A, B, C, D, or E (Microbial Water Quality Assessment or MWQA categories) with corresponding colors, where the breakpoints for each category are <10\

The cat letter grade is itself a discretized read on a single continuous quantity, also returned as exceed_rate: the one-sided 90\

If the input data are from [read_importentero](#) (baywide assessment), the results can be summarized by bay segment if bay_segment is not NULL. The stas argument is ignored and all stations within each bay segment watershed are used to evaluate the FIB categories.

yrrng can be specified several ways. If yrng = NULL, the year range of the data for the selected changes is chosen. User-defined values for the minimum and maximum years can also be used, or only a minimum or maximum can be specified, e.g., yrng = c(2000, 2010) or yrng = c(2000, NA). In the latter case, the maximum year will be defined by the data.

The default stations if the input is from [read_importfib](#) (EPC data) and if bay_segment is NULL are those used in TBEP report #05-13 (<https://drive.google.com/file/d/1MZnK3cMzV7LRg6dTbCKX8AOZU0GNurJJ/view>) for the Hillsborough River Basin Management Action Plan (BMAP) subbasins. These include Blackwater Creek (WBID 1482, EPC stations 143, 108), Baker Creek (WBID 1522C, EPC station 107), Lake Thonotosassa (WBID 1522B, EPC stations 135, 118), Flint Creek (WBID 1522A, EPC station 148), and the Lower Hillsborough River (WBID 1443E, EPC stations 105, 152, 137). Other stations can be plotted using the stas argument.

Input from [read_importwqp](#) for Manatee County (21FLMANA_WQX), Pasco County (21FLPASC_WQX), or Polk County (21FLPOLK_WQX) FIB data can also be used. The function has not been tested for other organizations.

Value

A [tibble](#) object with FIB summaries by year and station/bay segment (grp), including the estimated geometric mean of Enterococcus (marine) or E. coli (fresh) concentrations (gmean), the sample class (class, "Marine" or "Fresh"), a letter category based on the likelihood of exceeding the applicable threshold (cat), the count of samples over the lagged (lagyr) window exceeding that threshold (n exceed) and the total sample count over the same window (nsamp), and a continuous companion to cat (exceed_rate, see Details). Results can be summarized by bay segment if bay_segment is not NULL and the input data is from [read_importentero](#).

See Also

[show_fibmatrix](#)

Examples

```
anlz_fibmatrix(fibdata)

# use different dataset
anlz_fibmatrix(enterodata, lagyr = 1)

# subset to only wet samples
anlz_fibmatrix(enterodata, lagyr = 1, subset_wetdry = "wet",
               temporal_window = 2, wet_threshold = 0.5)

# Manatee County data
anlz_fibmatrix(mancofibdata, lagyr = 1)
```

anlz_fibwetdry	<i>Identify Fecal Indicator Bacteria samples as coming from a 'wet' or 'dry' time period</i>
----------------	--

Description

Identify Fecal Indicator Bacteria samples as coming from a 'wet' or 'dry' time period

Usage

```
anlz_fibwetdry(fibdata, precipdata, temporal_window = 2, wet_threshold = 0.5)
```

Arguments

fibdata	input data frame
precipdata	input data frame as returned by read_importtrain . columns should be: station, date (yyyy-mm-dd), rain (in inches). The object catchprecip has this data from 1995-2023 for select stations.
temporal_window	numeric, number of days precipitation should be summed over (1 = day of sample only; 2 = day of sample + day before; etc.)
wet_threshold	numeric, inches accumulated through the defined temporal window, above which a sample should be defined as being from a 'wet' time period

Details

This function allows the user to specify a threshold for declaring a sample to be taken after an important amount of rain over an important amount of days, and declaring it to be 'wet'. This is of interest because samples taken after significant precipitation (definitions of this vary, which is why the user can specify desired thresholds) are more likely to exceed relevant bacterial thresholds. Identifying samples as 'wet' or not allows for calculation of further indices for wet and dry subsets of samples.

Value

a data frame; the original fibdata data frame with three additional columns. `rain_sampleDay` is the total rain (inches) on the day of sampling, `rain_total` is the total rain (inches) for the period of time defined by `temporal_window`, and `wet_sample` is logical, indicating whether the rainfall for that station's catchment exceeded the amount over the time period specified in args.

Examples

```
entero_wetdry <- anlz_fibwetdry(enterodata, catchprecip)
head(entero_wetdry)
```

anlz_hmpreport	<i>Evaluate Habitat Master Plan progress for report card</i>
----------------	--

Description

Evaluate Habitat Master Plan progress for report card

Usage

```
anlz_hmpreport(acres, subtacres, hmptrgs)
```

Arguments

<code>acres</code>	data.frame for intertidal and supratidal land use and cover of habitat types for each year of data
<code>subtacres</code>	data.frame for subtidal cover of habitat types for each year of data
<code>hmptrgs</code>	data.frame of Habitat Master Plan targets and goals

Details

The relevant output columns are `targeteval` and `goaleval` that indicate numeric values as -1 (target not met, trending below), 0 (target met, trending below), 0.5 (target not met, trending above), and 1 (target met, trending above).

Columns in the output are as follows:

year Year of the assessment

metric Habitat type assessed

Acres Coverage estimate for the year

lacres Coverage estimate for the previous set of available data

lyr Year for the previous set of available data

category Strata for the habitat type

Target 2030 target for the habitat type from the Habitat Master Plan

Goal 2050 goal for the habitat type from the Habitat Master Plan

acresdiff Difference in acres for the current year and the previous set of available data
yeardiff Difference in years for the current year and the previous set of available data
changerate Acreage change per year for the current year relative to the previous set of available data
targetrate Annual rate required to achieve the 2030 target
goalrate Annual rate required to achieve the 2050 goal
targetprop Proportion of target met for the current year
goalprop Proportion of goal met for the current year
targeteval A number indicating target status of the current year for the report card
goaleval A number indicating goal status of the current year for the report card

The numbers in `targeteval` and `goaleval` are one of four values as -1, 0, 0.5, and 1. These numbers define the status for the assessment year:

-1 target or goal not met, trending below
0 target or goal met, trending below
0.5 target or goal not met, trending above
1 target or goal met, trending above

Value

A summarized `data.frame` appropriate for creating a report card

Examples

```
# view summarized data for report card
anlz_hmpreport(acres, subtacres, hmptrgs)
```

anlz_hydroload	<i>Estimate hydrological estimates and adjustment factors for bay segments</i>
----------------	--

Description

Estimate hydrological estimates and adjustment factors for bay segments

Usage

```
anlz_hydroload(yrs, noaa_key = NULL, trace = FALSE)
```

Arguments

<code>yrs</code>	numeric vector indicating years to return
<code>noaa_key</code>	user-supplied NOAA key, see details
<code>trace</code>	logical indicating if function progress is printed in the consol

Details

This function uses rainfall and streamflow data from NOAA and USGS and requires an API key. See the webpage <https://www.ncdc.noaa.gov/cdo-web/token> for retrieving a NOAA token. This key can be added to the R environment file and called for later use, see the examples.

These estimates are used in annual compliance assessment reports produced by the Tampa Bay Nitrogen Management Consortium. Load estimates and adjustment factors are based on regression models in <https://drive.google.com/file/d/11NT0NQ2WbPO6pVZaD7P7Z6qjcwO1jxHw/view?usp=drivesdk>

Value

A data frame with hydrological load estimates by bay segments for the requested years

Examples

```
## Not run:
# this function requires an API key
# save it to the R environment file (only once)
# save the key, do only once
cat("NOAA_KEY=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX\n",
    file=file.path(normalizePath("~/"), ".Renviron"),
    append=TRUE)

# retrieve the key after saving, may need to restart R
noaa_key <- Sys.getenv('NOAA_key')

# get estimates for 2021
anlz_hydroload(2021, noaa_key)

## End(Not run)
```

anlz_iwrraw

Format raw IWR data

Description

Format raw IWR data

Usage

```
anlz_iwrraw(iwrraw, tidalcreeks, yr = 2025)
```

Arguments

iwrraw	FDEP impaired waters rule data base as data.frame
tidalcreeks	sf object for population of tidal creeks
yr	numeric for reference year to evaluate, scores are based on ten years of data using yr as the last year

Details

The function subsets the raw IWR data for ten years starting at the value in `yr` and subsets by the creek population in [tidalcreeks](#). Select water quality parameters in `masterCode` are filtered and some of the names are combined for continuity.

Value

A `data.frame` with the formatted data

Examples

```
anlz_iwrraw(iwrraw, tidalcreeks, yr = 2025)
```

anlz_refs	<i>Convert references csv to bib</i>
-----------	--------------------------------------

Description

Convert references csv to bib

Usage

```
anlz_refs(path)
```

Arguments

`path` chr string of path to reference csv file or data frame object

Value

A data frame with references formatted as bib entries

Examples

```
# input and format
path <- system.file('tbep-refs.csv', package = 'tbepools')
bibs <- anlz_refs(path)

## Not run:
# save output
writeLines(bibs, 'formatted.bib')

## End(Not run)
```

anlz_sedimentaddtot	<i>Add contaminant totals to sediment data</i>
---------------------	--

Description

Add contaminant totals to sediment data

Usage

```
anlz_sedimentaddtot(  
  sedimentdata,  
  yrrng = c(1993, 2024),  
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),  
  funding_proj = "TBEP",  
  param = NULL,  
  pelave = TRUE  
)
```

Arguments

sedimentdata	input sediment data.frame as returned by read_importsediment
yrrng	numeric vector indicating min, max years to include, use single year for one year of data
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"
funding_proj	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"
param	optional character string of a parameter to filter the results
pelave	logical indicating if output is used for anlz_sedimentpel

Details

This function adds totals to the sedimentdata input for total PCBs, total DDT, total LMW PAH, total HMW PAH, and total PAH. Appropriate TEL/PEL values for the totals are also added.

Value

A data.frame object similar to the input, but filtered by the arguments and contaminant totals added. Replicate samples are also removed.

Examples

```
anlz_sedimentaddtot(sedimentdata)
```

anlz_sedimentave	<i>Get average concentrations for a sediment parameter by bay segment</i>
------------------	---

Description

Get average concentrations for a sediment parameter by bay segment

Usage

```
anlz_sedimentave(  
  sedimentdata,  
  param,  
  yrrng = c(1993, 2024),  
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),  
  funding_proj = "TBEP"  
)
```

Arguments

sedimentdata	input sediment data.frame as returned by read_importsediment
param	chr string for which parameter to plot
yrrng	numeric vector indicating min, max years to include, use single year for one year of data
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"
funding_proj	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"

Details

Summaries for all bay segments are returned by default. The averages and confidence intervals are specific to the year ranges in yrrng.

Value

A data.frame object with average sediment concentrations for the selected parameter by bay segment

See Also

[show_sedimentave](#)

Examples

```
anlz_sedimentave(sedimentdata, param = 'Arsenic')
```

anlz_sedimentpel	<i>Get sediment PEL ratios at stations in Tampa Bay</i>
------------------	---

Description

Get sediment PEL ratios at stations in Tampa Bay

Usage

```
anlz_sedimentpel(  
  sedimentdata,  
  yrrng = c(1993, 2024),  
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),  
  funding_proj = "TBEP"  
)
```

Arguments

sedimentdata	input sediment data.frame as returned by read_importsediment
yrrng	numeric vector indicating min, max years to include, use single year for one year of data
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"
funding_proj	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"

Details

Average PEL ratios for all contaminants graded from A to F for benthic stations monitored in Tampa Bay are estimated. The PEL is a measure of how likely a contaminant is to have a toxic effect on invertebrates that inhabit the sediment. The PEL ratio is the contaminant concentration divided by the Potential Effects Levels (PEL) that applies to a contaminant, if available. Higher ratios and lower grades indicate sediment conditions that are likely unfavorable for invertebrates. The station average combines the PEL ratios across all contaminants measured at a station and the grade applies to the average.

The grade breaks for the PEL ratio are 0.00756, 0.02052, 0.08567, and 0.28026, with lower grades assigned to the higher breaks.

Value

A data.frame object with average PEL ratios and grades at each station

See Also

[show_sedimentpelmap](#)

Examples

```
anlz_sedimentpel(sedimentdata)
```

anlz_sedimentpelave	<i>Get average concentrations for a sediment parameter by bay segment</i>
---------------------	---

Description

Get average concentrations for a sediment parameter by bay segment

Usage

```
anlz_sedimentpelave(  
  sedimentdata,  
  yrrng = c(1993, 2024),  
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),  
  funding_proj = "TBEP"  
)
```

Arguments

sedimentdata	input sediment data.frame as returned by read_importsediment
yrrng	numeric vector indicating min, max years to include, use single year for one year of data
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"
funding_proj	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"

Details

Summaries for all bay segments are returned by default. The averages and confidence intervals are specific to the year ranges in yrrng.

Value

A data.frame of the average of the average PEL ratios for each bay segment

See Also

[show_sedimentpelave](#)

Examples

```
anlz_sedimentpelave(sedimentdata)
```

anlz_splitdata	<i>Analyze time series data split by date within years</i>
----------------	--

Description

Analyze time series data split by date within years

Usage

```
anlz_splitdata(  
  df,  
  date_split,  
  date_col = "date",  
  value_col = "value",  
  stats = list(avg = mean)  
)
```

Arguments

df	data frame containing date and value columns
date_split	date to split analysis into annual periods
date_col	name of the date column
value_col	name of the value column
stats	list of functions to apply to values, default: <code>list(avg = mean)</code>

Value

A tibble summarizing data for annual periods "before" and "after" the split date

Examples

```
# Create sample data  
data <- data.frame(  
  date = seq.Date(as.Date("2010-01-01"), as.Date("2020-12-31"), by = "month"),  
  value = rnorm(132, mean = 10, sd = 2))  
  
# Basic analysis with default statistics  
split_date <- as.Date("2015-06-15")  
anlz_splitdata(data, split_date, "date", "value")
```

anlz_splitstorms	<i>Analyze storm data split by date within years</i>
------------------	--

Description

Analyze storm data split by date within years

Usage

```
anlz_splitstorms(  
  df_hurricane,  
  date_split,  
  stats = list(sum = sum, avg = mean, n = length)  
)
```

Arguments

df_hurricane	data frame containing date_beg and scale columns for hurricanes
date_split	date to split analysis into annual periods
stats	list of functions to apply to scale values, default: sum, average and count. Functions should accept numeric vector input and return single value

Value

A tibble summarizing hurricanes for annual periods "before" and "after" the split date, with requested statistics computed for the scale values

Examples

```
# Create sample hurricane data  
hurricanes <- data.frame(  
  date_beg = as.Date(c(  
    "1980-07-31", "1980-09-04", "1980-11-07",  
    "1981-05-06", "1981-08-07", "1981-11-12")),  
  scale = c(6, 1, 3, 1, 2, 1))  
  
# Basic analysis with default statistics (sum, average and count)  
split_date <- Sys.Date() - lubridate::years(1)  
anlz_splitstorms(hurricanes, split_date)  
  
# Analysis with custom statistics  
anlz_splitstorms(hurricanes, split_date,  
  stats = list(  
    max = max,  
    min = min))
```

anlz_tbbimed	<i>Get annual medians of Tampa Bay Benthic Index scores by bay segment</i>
--------------	--

Description

Get annual medians of Tampa Bay Benthic Index scores by bay segment

Usage

```
anlz_tbbimed(  
  tbbisr,  
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB", "All", "All (wt)"),  
  rev = FALSE,  
  yrrng = c(1993, 2024),  
  window = TRUE  
)
```

Arguments

tbbisr	input data frame as returned by anlz_tbbisr
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB", "All", "All (wt)"
rev	logical if factor levels for bay segments are reversed
yrrng	numeric indicating year ranges to evaluate
window	logical indicating whether to use a rolling 5-year window (default TRUE) or single year values (FALSE) for the bay segment categories, see details

Details

Additional summaries are provided for the entire bay, as a summary across categories ("All") and a summary weighted across the relative sizes of each bay segment ("All (wt)").

Only sampling funded by TBEP and as part of the routine EPC benthic monitoring program are included in the final categories.

The default behavior is to use a rolling five-year window to calculate the percent of sites in each TBBI category by bay segment. This applies only to years 2005 and later, where the counts from the current year and the prior four years are summed to calculate the percentages. This is intended to help smooth out inter-annual variability due to reduced sampling effort from 2005 to present. If window = FALSE, then only single year values are used.

Value

A data frame of annual medians by bay segment

Examples

```
tbbiscr <- anlz_tbbiscr(benthicdata)
anlz_tbbimed(tbbiscr)
```

anlz_tbbiscr	<i>Get Tampa Bay Benthic Index scores</i>
--------------	---

Description

Get Tampa Bay Benthic Index scores

Usage

```
anlz_tbbiscr(benthicdata)
```

Arguments

benthicdata nested [tibble](#) formatted from [read_importbenthic](#)

Details

This function calculates scores for the TBBi based on station, taxa, and field sample data. The total TBBi scores are returned as TBBi and TBBiCat, where the latter is a categorical description of the scores.

Value

A single data frame of TBBi scores for each site.

Examples

```
anlz_tbbiscr(benthicdata)
```

anlz_tbbiave	<i>Get annual averages of Tampa Bay Nekton Index scores by bay segment</i>
--------------	--

Description

Get annual averages of Tampa Bay Nekton Index scores by bay segment

Usage

```
anlz_tbniave(
  tbniscr,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  rev = FALSE,
  perc = c(32, 46)
)
```

Arguments

tbniscr	input data frame as returned by anlz_tbniscr
bay_segment	chr string for the bay segment, one to many of "OTB", "HB", "MTB", "LTB"
rev	logical if factor levels for bay segments are reversed
perc	numeric values indicating break points for score categories

Value

A data frame of annual averages by bay segment

Examples

```
tbniscr <- anlz_tbniscr(fimdata)
anlz_tbniave(tbniscr)
```

anlz_tbnimet	<i>Get all raw metrics for Tampa Bay Nekton Index</i>
--------------	---

Description

Get all raw metrics for Tampa Bay Nekton Index

Usage

```
anlz_tbnimet(fimdata, all = FALSE)
```

Arguments

fimdata	data.frame formatted from read_importfim
all	logical indicating if only TBNI metrics are returned (default), otherwise all are calculated

Details

All raw metrics are returned in addition to those required for the TBNI. Each row shows metric values for a station, year, and month where fish catch data were available.

Value

A data frame of raw metrics in wide format. If `all = TRUE`, all metrics are returned, otherwise only NumTaxa, BenthicTaxa, TaxaSelect, NumGuilds, and Shannon are returned.

Examples

```
anlz_tbnimet(fimdata)
```

anlz_tbniscr	<i>Get Tampa Bay Nekton Index scores</i>
--------------	--

Description

Get Tampa Bay Nekton Index scores

Usage

```
anlz_tbniscr(fimdata, raw = TRUE)
```

Arguments

fimdata	data.frame formatted from read_importfim
raw	logical indicating if raw metric values are also returned

Details

This function calculates raw and scored metrics for the TBNI, including NumTaxa, BenthicTaxa, TaxaSelect, NumGuilds, and Shannon. The total TBNI score is returned as TBNI_Score.

Value

A data frame of metrics and TBNI scores in wide format.

See Also

[anlz_tbnimet](#)

Examples

```
anlz_tbniscr(fimdata)
```

anlz_tdlcrk	Estimate tidal creek report card scores
-------------	---

Description

Estimate tidal creek report card scores

Usage

```
anlz_tdlcrk(tidalcreeks, iwrraw, tidtrgs = NULL, yr = 2025)
```

Arguments

- tidalcreeks [sf](#) object for population of tidal creeks
- iwrraw FDEP impaired waters data base as [data.frame](#)
- tidtrgs optional [data.frame](#) for tidal creek nitrogen targets, defaults to [tidaltargets](#)
- yr numeric for reference year to evaluate, scores are based on the planning period beginning ten years prior to this date

Value

A [data.frame](#) with the report card scores for each creek, as prioritize, investigate, caution, monitor, or no data

Examples

```
anlz_tdlcrk(tidalcreeks, iwrraw, yr = 2025)
```

anlz_tdlcrkindic	Analyze tidal creek water quality indicators
------------------	--

Description

Estimate tidal creek water quality indicators to support report card scores

Usage

```
anlz_tdlcrkindic(tidalcreeks, iwrraw, yr = 2025, radar = FALSE)
```

Arguments

- tidalcreeks [sf](#) object for population of tidal creeks
- iwrraw FDEP impaired waters rule data base as [data.frame](#)
- yr numeric for reference year to evaluate, scores are based on the planning period beginning ten years prior to this date
- radar logical indicating if output is for [show_tdlcrkradar](#), see details

Details

Annual geometric means for additional water quality data available at each wbid, JEI combination. Florida trophic state index values are also estimated where data are available.

Nitrogen ratios are estimated for JEIs that cover source (upstream, freshwater) and tidal (downstream) WBIDs, defined as the ratio of concentrations between the two (i.e., ratios > 1 mean source has higher concentrations). Nitrogen ratios for a given year reflect the ratio of the median nitrogen concentrations when they were measured in both a source and tidal segment during the same day. Note that a ratio of one can be obtained if both the source and tidal segments are at minimum detection.

Indicators for years where more than 10\

If radar = TRUE, output is returned in a format for use with [show_tdlcrkradar](#) Specifically, results are calculated as the percentage of years where an indicator exceeds a relevant threshold. This only applies to the marine WBIDs of the tidal creeks (Florida DEP class 2, 3M). Six indicators are returned with percentage exceedances based on total nitrogen (tn_ind) greater than 1.1 mg/L, chlorophyll (chl_a_ind) greater than 11 ug/L, trophic state index (tsi_ind) greater than 55 (out of 100), nitrate/nitrite ratio between marine and upstream segments (nox_ind) greater than one, chlorophyll and total nitrogen ratios > 15, and percentage of years more where than ten percent of observations were below DO percent saturation of 42.

Value

A [data.frame](#) with the indicator values for each tidal creek

Examples

```
dat <- anlz_tdlcrkindic(tidalcreeks, iwrraw, yr = 2025)
head(dat)
```

anlz_transectave	<i>Get annual averages of seagrass frequency occurrence by bay segments and year</i>
------------------	--

Description

Get annual averages of seagrass frequency occurrence by bay segments and year

Usage

```
anlz_transectave(
  transectocc,
  bay_segment = c("OTB", "HB", "MTB", "LTB", "BCB"),
  total = TRUE,
  yrrng = c(1998, 2025),
  rev = FALSE
)
```

Arguments

transectocc	data frame returned by anlz_transectocc
bay_segment	chr string for the bay segment, one to many of "OTB", "HB", "MTB", "LTB", "BCB"
total	logical indicating if average frequency occurrence is calculated for the entire bay across segments
yrrng	numeric vector of length two indicating min, max years to include, use the same year for both values to evaluate a single year
rev	logical if factor levels for bay segments are reversed

Details

The focat column returned in the results shows a color category based on arbitrary breaks of the frequency occurrence estimates (foest) at 25, 50, and 75 percent. These don't necessarily translate to any ecological breakpoints.

Value

A data frame of annual averages by bay segment

Examples

```
## Not run:
transect <- read_transect()

## End(Not run)
transectocc <- anlz_transectocc(transect)
anlz_transectave(transectocc)
```

anlz_transectavespp	<i>Get annual averages of seagrass frequency occurrence by bay segments, year, and species</i>
---------------------	--

Description

Get annual averages of seagrass frequency occurrence by bay segments, year, and species

Usage

```
anlz_transectavespp(
  transectocc,
  bay_segment = c("OTB", "HB", "MTB", "LTB", "BCB"),
  yrrng = c(1998, 2025),
  species = c("Halodule", "Syringodium", "Thalassia", "Ruppia", "Halophila", "Caulerpa",
    "Dapis", "Chaetomorpha"),
  total = TRUE,
  by_seg = FALSE
)
```

Arguments

transectocc	data frame returned by anlz_transectocc
bay_segment	chr string for the bay segment, one to many of "OTB", "HB", "MTB", "LTB", "BCB"
yrrng	numeric vector of length two indicating min, max years to include, use the same year for both values to evaluate a single year
species	chr string of species to summarize, one to many of "Halodule", "Syringodium", "Thalassia", "Ruppia", "Halophila", "Caulerpa", "Dapis", "Chaetomorpha"
total	logical indicating if total frequency occurrence for all species is also returned
by_seg	logical indicating if separate results by bay segments are retained

Details

Frequency occurrence estimates are averaged across segments in bay_segment if by_seg = F, i.e., separate results by location are not returned. Results are retained by bay segment if by_seg = T.

Value

A data frame of annual averages by bay segment

Examples

```
## Not run:
transect <- read_transect()

## End(Not run)
transectocc <- anlz_transectocc(transect)
anlz_transectavespp(transectocc)
```

anlz_transectocc	<i>Get seagrass average abundance and occurrence across transects</i>
------------------	---

Description

Get seagrass average abundance and occurrence across transects

Usage

```
anlz_transectocc(transect)
```

Arguments

transect	data frame returned by read_transect
----------	--

Details

Abundance and frequency occurrence are estimated as in Sherwood et al. 2017, equations 1 and 2. In short, frequency occurrence is estimated as the number of instances a species was observed along a transect divided by the number of placements along a transect and average abundance was estimated as the sum of species-specific Braun-Blanquet scores divided by the number of placements along a transect. The estimates are obtained for all seagrass species including *Caulerpa* (attached macroalgae), *Dapis* (cyanobacteria), and *Chaetomorpha* (drift green algae), whereas all other attached and drift algae species are aggregated. Drift or attached macroalgae and cyanobacteria (*Dapis*) estimates may not be accurate prior to 2021. Values for total include total frequency occurrence only for seagrass species (*Halodule*, *Syringodium*, *Thalassia*, *Ruppia*, *Halophila*).

Value

A data frame with abundance and frequency occurrence estimates aggregated by species, transect, and date. The nsites column is the total number of placements that were sampled along a transect for a particular date.

References

Sherwood, E.T., Greening, H.S., Johansson, J.O.R., Kaufman, K., Raulerson, G.E. 2017. Tampa Bay (Florida, USA): Documenting seagrass recovery since the 1980's and reviewing the benefits. *Southeastern Geographer*. 57(3):294-319.

Examples

```
## Not run:
transect <- read_transect()

## End(Not run)
anlz_transectocc(transect)
```

anlz_yrattain	<i>Get attainment categories for a chosen year</i>
---------------	--

Description

Get attainment categories for a chosen year

Usage

```
anlz_yrattain(epcdata, yrsel, partialyr = FALSE)
```

Arguments

epcdata	data frame of epc data returned by read_importtwq
yrsel	numeric indicating chosen year
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter

Value

A data.frame for the chosen year and all bay segments showing the bay segment averages for chlorophyll concentration, light attenuations, segment targets, and attainment categories.

Examples

```
# defaults to current year
anzl_yrattain(epcdata, yrsel = 2025)
```

benthicdata	<i>Benthic data for the Tampa Bay Benthic Index current as of 20251225</i>
-------------	--

Description

Benthic data for the Tampa Bay Benthic Index current as of 20251225

Usage

```
benthicdata
```

Format

A nested [tibble](#) with 3 rows and 2 variables:

name chr identifying the dataset as stations, fieldsamples, or taxacounts

value list of dataframes for each dataset

Details

Index the corresponding list element in the value column to view each dataset. For example, the stations data in the first row can be viewed as `benthicdata$value[[1]]`.

Examples

```
## Not run:
# location to download data
path <- '~/Desktop/benthic.zip'

# load and assign to object
benthicdata <- read_importbenthic(path, download_latest = TRUE, remove = TRUE)

save(benthicdata, file = 'data/benthicdata.RData', compress = 'xz')

## End(Not run)
```

bsmap	<i>Terrain basemap</i>
-------	------------------------

Description

Terrain basemap

Usage

bsmap

Format

A `ggmap` object

catchpixels	<i>Catchments and radar pixels (for precip) of selected Enterococcus stations</i>
-------------	---

Description

Catchments and radar pixels (for precip) of selected Enterococcus stations

Usage

catchpixels

Format

A data frame with two columns:

station a character string of the Water Quality Portal station name

pixel a number; each row is a single pixel from the Southwest Florida Water Management District's radar-estimated rainfall shapefile

Details

Generated by `data-raw/catchpixels-raw.R` (view on github: <https://github.com/tbep-tech/tbepools/blob/master/data-raw/catchpixels-raw.R>)

catchprecip	<i>Daily precip by catchment for selected Enterococcus stations</i>
-------------	---

Description

Daily precip by catchment for selected Enterococcus stations

Usage

```
catchprecip
```

Format

A data frame with three columns:

station a character string of the Water Quality Portal station name

date a date

rain a number; inches of rain for that date, averaged across all pixels in the station's catchment

Details

Daily precipitation data for multiple years, provided by the Southwest Florida Water Management District, were downloaded at the pixel level and averaged to the catchment level for key Enterococcus sampling stations. Created using [read_importtrainmany](#).

Source

Southwest Florida Management District radar-estimated daily rainfall data, ftp://ftp.swfwmd.state.fl.us/pub/radar_rainfall/Daily_Data/

Examples

```
## Not run:
catchprecip <- read_importtrainmany(1995:2025, quiet = F)

save(catchprecip, file = 'data/catchprecip.RData')

## End(Not run)
```

dbasin	<i>Spatial data object of minor subbasins (dbasins) in the Tampa Bay watershed</i>
--------	--

Description

Spatial data object of minor subbasins (dbasins) in the Tampa Bay watershed

Usage

```
dbasin
```

Format

A simple features [sf](#) object (POLYGON) with 436 features and 26 fields, +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs

Examples

```
library(sf)
plot(st_geometry(dbasin))
```

enterodata	<i>Enterococcus data from 53 key Enterococcus stations since 1995</i>
------------	---

Description

Enterococcus data from 53 key Enterococcus stations since 1995

Usage

```
enterodata
```

Format

A data frame with 6885 rows and 16 columns:

date date, sample date

yr numeric, year of sample date

mo numeric, month of sample date

time character, sample time

time_zone character, sample time zone

long_name character, long name of bay segment subwatershed

bay_segment character, short name of bay segment subwatershed

station character, sample station
 entero numeric, Enterococcus concentration
 entero_censored logical, whether entero value was below the laboratory MDL, minimum detection limit
 MDL numeric, minimum detection limit at the time of processing
 entero_units character, units of measurement for entero
 qualifier qualifier codes associated with sample
 LabComments lab comments on sample
 Latitude numeric, latitude in decimal degrees
 Longitude numeric, longitude in decimal degrees

Details

A dataset containing Enterococcus from 53 stations in the TBEP watershed since 1995.

Source

Water Quality Portal, <https://waterqualitydata.us>

Examples

```
## Not run:
enterodata <- read_importentero(startDate = '1995-01-01', endDate = '2025-12-31')

save(enterodata, file = 'data/enterodata.RData')

## End(Not run)
```

epcdata

All bay data as of 20260203

Description

All bay data as of 20260203

Usage

epcdata

Format

A data frame with 29041 rows and 26 variables:

bay_segment chr
epchc_station num
SampleTime POSIXct
yr num
mo num
Latitude num
Longitude num
Total_Depth_m num
Sample_Depth_m num
tn num
tn_q chr
sd_m num
sd_raw_m num
sd_q chr
chla num
chla_q chr
Sal_Top_ppth num
Sal_Mid_ppth num
Sal_Bottom_ppth num
Temp_Water_Top_degC num
Temp_Water_Mid_degC num
Temp_Water_Bottom_degC num
Turbidity_JTU-NTU num
Turbidity_Q num
Color_345_F45_PCU num
Color_345_F45_Q num

Examples

```
## Not run:  
xlsx <- '~/Desktop/epcdata.xlsx'  
epcdata <- read_importwq(xlsx, download_latest = TRUE)  
  
nrow(epcdata)  
ncol(epcdata)  
  
save(epcdata, file = 'data/epcdata.RData', compress = 'xz')  
  
## End(Not run)
```

fibdata	<i>All Fecal Indicator Bacteria (FIB) data as of 20260302</i>
---------	---

Description

All Fecal Indicator Bacteria (FIB) data as of 20260302

Usage

fibdata

Format

A data frame with 31081 rows and 18 variables:

area chr
epchc_station num
class chr
SampleTime POSIXct
yr num
mo num
Latitude num
Longitude num
Total_Depth_m num
Sample_Depth_m num
ecoli num
ecoli_q chr
entero num
entero_q chr
fcolif num
fcolif_q chr
totcol num
totcol_q chr

Details

This dataset includes FIB data from the Environmental Protection Commission where station class is marine (3M, 2) and Enterococcus data is present or the station class is freshwater (3F, 1) and E. coli data is present. The data is formatted from the raw data loaded from [read_importfib](#).

Examples

```
## Not run:
xlsx <- tempfile(fileext = '.xlsx')
fibdata <- read_importfib(xlsx, download_latest = TRUE)

nrow(fibdata)
ncol(fibdata)

save(fibdata, file = 'data/fibdata.RData')

file.remove(xlsx)

## End(Not run)
```

fimdata

FIM data for Tampa Bay Nekton Index current as of 08102026

Description

FIM data for Tampa Bay Nekton Index current as of 08102026

Usage

```
fimdata
```

Format

A data frame with 55832rows and 19 variables:

Reference chr
Sampling_Date Date
Latitude num
Longitude num
Zone chr
Grid int
NODCCODE chr
Year num
Month num
Total_N num
ScientificName chr
Include_TB_Index chr
Hab_Cat chr
Est_Cat chr
Est_Use chr

Feeding_Cat chr
Feeding_Guild chr
Selected_Taxa chr
bay_segment chr

Examples

```
## Not run:
csv <- '~/Desktop/TampaBay_NektonIndexData.csv'

fimdata <- read_importfim(csv, download_latest = TRUE)

save(fimdata, file = 'data/fimdata.RData', compress = 'xz')

## End(Not run)
```

fimstations

Spatial data object of FIM stations including Tampa Bay segments

Description

Spatial data object of FIM stations including Tampa Bay segments

Usage

```
fimstations
```

Format

A simple features [sf](#) object (POINT) with 8423 features and 2 fields, +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs

Reference num
bay_segment chr

Examples

```
## Not run:
# file path
csv <- '~/Desktop/fimraw.csv'

# load and assign to object
fimstations <- read_importfim(csv, download_latest = TRUE, locs = TRUE)
save(fimstations, file = 'data/fimstations.RData', compress = 'xz')

## End(Not run)
```

hcesdfibdata	<i>Hillsborough County Environmental Services Division (ESD) FIB data as of 20260302</i>
--------------	--

Description

Hillsborough County Environmental Services Division (ESD) FIB data as of 20260302

Usage

hcesdfibdata

Format

A data frame with 1264 rows and 13 variables:

hcesd_station chr, Station name
SampleTime POSIXct, Date/time of sampling
class chr, Waterbody class (Fresh, Marine)
yr num, Year of sampling
mo num, Month of sampling
Latitude num, Latitude, WGS84
Longitude num, Latitude, WGS84
Sample_Depth_m num, Depth of sample, meters
var chr, Variable name (ecoli, entero)
val num, Value of variable
uni num, Units of variable
qual num, Qualifier code
area chr, Location name

Details

This dataset includes FIB data from Hillsborough County Environmental Services Division (21FL-HESD_WQX) where station class is marine (estuary) and Enterococcus data is present or the station class is freshwater (stream/river, reservoir) and E. coli data is present. The data is formatted from the raw data loaded from [read_importwqp](#).

Examples

```
## Not run:
hcesdfibdata <- read_importwqp('21FLHESD_WQX', type = 'fib')

nrow(hcesdfibdata)
ncol(hcesdfibdata)
```

```
save(hcesdfibdata, file = 'data/hcesdfibdata.RData')

## End(Not run)
```

hmptrgs*Habitat Master Plan targets and goals*

Description

Habitat Master Plan targets and goals

Usage

hmptrgs

Format

A data frame showing 2030 targets and 2050 goals

Examples

```
## Not run:
library(dplyr)

load(url("https://github.com/tbep-tech/hmpu-workflow/raw/master/data/trgs.RData"))

hmptrgs <- trgs %>%
  rename(Goal2050 = Target2050)

save(hmptrgs, file = 'data/hmptrgs.RData', compress = 'xz')

## End(Not run)
```

iwrraw*FDEP IWR run 67*

Description

Florida Department of Environmental Protection, Impaired Waters Rule, Run 67

Usage

iwrraw

Format

A data frame 592464 rows and 11 variables

Details

File was created using workflow at https://tbep-tech.github.io/tidalcreek-stats/Creek_select_tbepools, example below is old and for Run 61.

Examples

```
## Not run:
library(dplyr)

load(file = '../02_DOCUMENTS/tidal_creeks/iwrraw_run61.RData')
iwrraw <- sf::st_set_geometry(iwrraw, NULL) %>%
  rename(JEI = jei)
save(iwrraw, file = 'data/iwrraw.RData', compress = 'xz')

## End(Not run)
```

mancofibdata

Manatee County FIB data as of 20260302

Description

Manatee County FIB data as of 20260302

Usage

```
mancofibdata
```

Format

A data frame with 1928 rows and 13 variables:

manco_station chr, Station name
SampleTime POSIXct, Date/time of sampling
class chr, Waterbody class (Fresh, Marine)
yr num, Year of sampling
mo num, Month of sampling
Latitude num, Latitude, WGS84
Longitude num, Latitude, WGS84
Sample_Depth_m num, Depth of sample, meters
var chr, Variable name (ecoli, entero)
val num, Value of variable
uni num, Units of variable
qual num, Qualifier code
area chr, Location name based on USF Water Alas waterbody name

Details

This dataset includes FIB data from Manatee County Department of Natural Resources where station class is marine (estuary) and Enterococcus data is present or the station class is freshwater (stream/river, reservoir) and E. coli data is present. The data is formatted from the raw data loaded from [read_importwqp](#).

Examples

```
## Not run:
mancofibdata <- read_importwqp('21FLMANA_WQX', type = 'fib')

nrow(mancofibdata)
ncol(mancofibdata)

save(mancofibdata, file = 'data/mancofibdata.RData')

## End(Not run)
```

pascofibdata	<i>Pasco County FIB data as of 20260302</i>
--------------	---

Description

Pasco County FIB data as of 20260302

Usage

```
pascofibdata
```

Format

A data frame with 255 rows and 13 variables:

pasco_station chr, Station name
SampleTime POSIXct, Date/time of sampling
class chr, Waterbody class (Fresh, Marine)
yr num, Year of sampling
mo num, Month of sampling
Latitude num, Latitude, WGS84
Longitude num, Latitude, WGS84
Sample_Depth_m num, Depth of sample, meters
var chr, Variable name (ecoli, entero)
val num, Value of variable
uni num, Units of variable
qual num, Qualifier code
area chr, Location name

Details

This dataset includes FIB data from Pasco County (21FLPASC_WQX) where station class is marine (estuary) and Enterococcus data is present or the station class is freshwater (stream/river, reservoir) and E. coli data is present. The data is formatted from the raw data loaded from [read_importwqp](#).

Examples

```
## Not run:
pascofibdata <- read_importwqp('21FLPASC_WQX', type = 'fib')

nrow(pascofibdata)
ncol(pascofibdata)

save(pascofibdata, file = 'data/pascofibdata.RData')

## End(Not run)
```

phytodata

Phytoplankton data current as of 20260204

Description

Phytoplankton data current as of 20260204

Usage

```
phytodata
```

Format

A nested [tibble](#) with 41390 rows and 8 variables:

```
epchc_station chr
Date Date
name chr
units chr
count num
yrqrt Date
yr num
mo Ord.factor
```

Examples

```
## Not run:
# location to download data
path <- tempfile(fileext = '.xlsx')

# load and assign to object
phytodata <- read_importphyto(path, download_latest = TRUE)

nrow(phytodata)
ncol(phytodata)

save(phytodata, file = 'data/phytodata.RData', compress = 'xz')

file.remove(path)

## End(Not run)
```

polcofibdata	<i>Polk County FIB data as of 20260302</i>
--------------	--

Description

Polk County FIB data as of 20260302

Usage

```
polcofibdata
```

Format

A data frame with 996 rows and 13 variables:

polco_station chr, Station name
SampleTime POSIXct, Date/time of sampling
class chr, Waterbody class (Fresh, Marine)
yr num, Year of sampling
mo num, Month of sampling
Latitude num, Latitude, WGS84
Longitude num, Latitude, WGS84
Sample_Depth_m num, Depth of sample, meters
var chr, Variable name (ecoli, entero)
val num, Value of variable
uni num, Units of variable
qual num, Qualifier code
area chr, Location name

Details

This dataset includes FIB data from Polk County (21FLPOLK_WQX) where station class is marine (estuary) and Enterococcus data is present or the station class is freshwater (stream/river, reservoir) and E. coli data is present. The data is formatted from the raw data loaded from [read_importwqp](#).

Examples

```
## Not run:
polcofibdata <- read_importwqp('21FLPOLK_WQX', type = 'fib')

nrow(polcofibdata)
ncol(polcofibdata)

save(polcofibdata, file = 'data/polcofibdata.RData')

## End(Not run)
```

read_dlcurrent	<i>Download latest file from epchc.org</i>
----------------	--

Description

Download latest file from epchc.org

Usage

```
read_dlcurrent(locin, download_latest = TRUE, urlin)
```

Arguments

locin	chr string path for local file, to overwrite it not current
download_latest	logical to download latest file regardless of local copy
urlin	url for file location

Details

The local copy is checked against a temporary file downloaded from the location specified by urlin. The local file is replaced with the downloaded file if the MD5 hashes are different.

Value

The local copy specified in the path by locin is overwritten by the new file if the file is not current or download_latest = TRUE. The function does nothing if download_latest = FALSE.

Examples

```
## Not run:
locin <- '~/Desktop/RWMDDataSpreadsheet_ThroughCurrentReportMonth.xlsx'
urlin1 <- 'https://epcbocc.sharepoint.com/:x:/s/PublishedShare/'
urlin2 <- 'IQAtFpJjaVHzS6ENmWdNf4csAX05fC5QFwI8U4b69Yyc8FA?e=tMe7Yy'
urlin <- paste0(urlin1, urlin2, '&download=1')
read_dlcurrent(locin = locin, urlin = urlin)

## End(Not run)
```

read_formbenthic	<i>Format benthic data for the Tampa Bay Benthic Index</i>
------------------	--

Description

Format benthic data for the Tampa Bay Benthic Index

Usage

```
read_formbenthic(pathin)
```

Arguments

pathin A path to unzipped csv files with base tables used to calculate benthic index

Details

Function is used internally within [read_importbenthic](#)

Value

A nested [tibble](#) of station, field sample, and taxa data

Examples

```
## Not run:

# location to download data
path <- '~/Desktop/benthic.zip'

# download
urlin1 <- 'https://epcbocc.sharepoint.com/:u:/s/PublishedShare/'
urlin2 <- 'IQD0Da-pbW8pTaZMg-93L9sTAZW1Y6hjj1TD_8RY0mFGiQ8?e=FIphoq'
urlin <- paste0(urlin1, urlin2, '&download=1')
read_dlcurrent(path, download_latest = TRUE, urlin = urlin)

# unzip
tmppth <- tempfile()
utils::unzip(path, exdir = tmppth, overwrite = TRUE)
```

```
# format benthic data
read_formbenthic(pathin = tmppth)

# remove temporary path
unlink(tmppth, recursive = TRUE)

## End(Not run)
```

read_formfib

Format Fecal Indicator Bacteria (FIB) data

Description

Format Fecal Indicator Bacteria (FIB) data

Usage

```
read_formfib(datin, all = FALSE)
```

Arguments

datin	input data.frame loaded from read_importtepc
all	logical indicating if all stations with FIB data are returned, default is FALSE

Details

Formats input data from [read_importtepc](#) appropriate for FIB results, see the details in [read_importfib](#) for more more information.

Value

A lightly formatted data.frame with FIB data

See Also

[read_importfib](#), [read_importtepc](#)

Examples

```
## Not run:
# file path
xlsx <- '~/Desktop/RWMDDataSpreadsheet_ThroughCurrentReportMonth.xlsx'

# load raw data and assign to object
epcall <- read_importtepc(xlsx, download_latest = T)

# final formatting
```

```

fibdata <- read_formfib(epcall)

## End(Not run)

```

read_formfim

Format FIM data for the Tampa Bay Nekton Index

Description

Format FIM data for the Tampa Bay Nekton Index

Usage

```
read_formfim(datin, locs = FALSE)
```

Arguments

datin	input data.frame loaded from read_importfim
locs	logical indicating if a spatial features object is returned with locations of each FIM sampling station

Details

Function is used internally within [read_importfim](#)

Value

A formatted data.frame with FIM data if locs = FALSE, otherwise a simple features object if locs = TRUE

See Also

[read_importfim](#)

Examples

```

# file path
url <- 'https://raw.githubusercontent.com/tbep-tech/tbni-proc/master/data/'
fl <- 'TampaBay_NektonIndexData.csv'
csv <- url(paste0(url, fl))

datin <- read.csv(csv)

# load and assign to object
fimdata <- read_formfim(datin)

```

read_formphyto	<i>Format phytoplankton data</i>
----------------	----------------------------------

Description

Format phytoplankton data

Usage

```
read_formphyto(datin)
```

Arguments

datin input data.frame loaded from [read_importphyto](#)

Details

Only seven taxonomic groups are summarized. Pyrodinium bahamense, Karenia brevis, Tripos hircus, Pseudo-nitzschia sp., and Pseudo-nitzschia pungens are retained at the species level. Diatoms are summarized at the class level using Bacillariophyceae, Coscinodiscophyceae, and Mediophyceae and Cyanobacteria are summarized at the phylum level using Cyanobacteriota. All other taxa are grouped into an "other" category.

Value

A formatted data.frame with phytoplankton count data

See Also

[read_importphyto](#)

Examples

```
## Not run:
# file path
xlsx <- '~/Desktop/phyto_data.xlsx'

# load and assign to object
phytodata <- read_importphyto(xlsx, download_latest = TRUE)

## End(Not run)
```

read_formsediment	<i>Format sediment data</i>
-------------------	-----------------------------

Description

Format sediment data

Usage

```
read_formsediment(pathin)
```

Arguments

pathin	A path to unzipped csv files with sediment result tables
--------	--

Details

Function is used internally within [read_importsediment](#)

Value

A data.frame of sediment data for Tampa Bay

Examples

```
## Not run:

# location to download data
path <- '~/Desktop/sediment.zip'

# download
urlin1 <- 'https://epcbocc.sharepoint.com/:u:/s/PublishedShare/'
urlin2 <- 'IQAr2aDfEyegTrJLR61qt1CxAYxfnVYYWAwkbtArczrJS4?e=EQSG03'
urlin <- paste0(urlin1, urlin2, '&download=1')
read_dlcurrent(path, download_latest = TRUE, urlin = urlin)

# unzip
tmppth <- tempfile()
utils::unzip(path, exdir = tmppth, overwrite = TRUE)

# format sediment data
read_formsediment(pathin = tmppth)

# remove temporary path
unlink(tmppth, recursive = TRUE)

## End(Not run)
```

read_formtransect	<i>Format seagrass transect data from Water Atlas</i>
-------------------	---

Description

Format seagrass transect data from Water Atlas

Usage

```
read_formtransect(jsn, training = FALSE, raw = FALSE)
```

Arguments

jsn	A data frame returned from fromJSON
training	logical if input are transect training data or complete database
raw	logical indicating if raw, unformatted data are returned, see details

Details

Shoot density is reported as number of shoots per square meter and is corrected for the quadrat size entered in the raw data. Shoot density and blade height (cm) are based on averages across random observations at each transect point that are entered separately in the data form. Abundance is reported as a numeric value from 0 - 5 for Braun-Blanquet coverage estimates.

If raw = TRUE, the unformatted data are returned. The default is to use formatting that allows the raw data to be used with the downstream functions. The raw data may have extra information that may be of use outside of the plotting functions in this package.

Value

data frame in long format

Examples

```
library(jsonlite)

## Not run:
# all transect data
url <- 'https://tampabay.wateratlas.usf.edu/seagrass-transect-data-portal/api/assessments/all__use-with-care'
jsn <- fromJSON(url)
trndat <- read_formtransect(jsn)

# training transect data
url <- 'https://tampabay.wateratlas.usf.edu/seagrass-transect-data-portal/api/assessments/training'
jsn <- fromJSON(url)
trndat <- read_formtransect(jsn, training = TRUE)

## End(Not run)
```

read_formwq	<i>Format water quality data</i>
-------------	----------------------------------

Description

Format water quality data

Usage

```
read_formwq(datin, all = FALSE)
```

Arguments

datin	input data.frame loaded from read_importepc
all	logical indicating if all water quality parameters are returned, see details

Details

Secchi data VOB depths or secchis < 0.5 ft from bottom are assigned NA, function is used internally within [read_importwq](#)

Value

A lightly formatted data.frame with chlorophyll and secchi observations

See Also

[read_importwq](#), [read_importepc](#)

Examples

```
## Not run:
# file path
xlsx <- '~/Desktop/RWMDDataSpreadsheet_ThroughCurrentReportMonth.xlsx'

# load raw data and assign to object
epcall <- read_importepc(xlsx, download_latest = T)

# final formatting
epcdata <- read_formwq(epcall)

## End(Not run)
```

read_formwqp

Format data and station metadata from the Water Quality Portal

Description

Format data and station metadata from the Water Quality Portal

Usage

```
read_formwqp(res, sta, org, type, trace = F)
```

Arguments

res	A data frame containing results obtained from the API
sta	A data frame containing station metadata obtained from the API
org	chr string indicating the organization identifier
type	chr string indicating data type to download, one of "wq" or "fib"
trace	Logical indicating whether to display progress messages, default is FALSE

Details

This function is used by [read_importwqp](#) to combine, format, and process data (res) and station metadata (sta) obtained from the Water Quality Portal for the selected county and data type. The resulting data frame includes the date, time, station identifier, latitude, longitude, variable name, value, unit, and quality flag. Manatee County FIB data (21FLMANA_WQX) will also include an area column indicating the waterbody name as used by the USF Water Atlas, with some area aggregations. Other county-level FIB data will have a similar column.

Value

A data frame containing formatted water quality and station metadata

See Also

[read_importwqp](#)

Examples

```
## Not run:
url <- list(
  Result = "https://www.waterqualitydata.us/data/Result/search?mimeType=csv&zip=no",
  Station = "https://www.waterqualitydata.us/data/Station/search?mimeType=csv&zip=no"
)

headers <- c(
  "Content-Type" = "application/json",
  "Accept" = "application/zip"
```

```

)

body <- list(
  organization = c("21FLMANA_WQX"),
  sampleMedia = c("Water"),
  characteristicType = c("Information", "Nutrient", "Biological, Algae, Phytoplankton,
    Photosynthetic Pigments"),
  providers = c("STORET"),
  siteType = c("Estuary")
)

res <- url[['Result']] %>%
  httr::POST(httr::add_headers(headers), body = jsonlite::toJSON(body)) %>%
  httr::content('text') %>%
  read.csv(text = .)

sta <- url[['Station']] %>%
  httr::POST(httr::add_headers(headers), body = jsonlite::toJSON(body)) %>%
  httr::content('text') %>%
  read.csv(text = .)

# combine and format
read_formwqp(res, sta, '21FLMANA_WQX', type = 'wq')

## End(Not run)

```

read_importbenthic	<i>Download and import benthic data for Tampa Bay</i>
--------------------	---

Description

Download and import benthic data for Tampa Bay

Usage

```
read_importbenthic(path, download_latest = FALSE, remove = FALSE)
```

Arguments

path	chr string for local path where the zipped folder will be downloaded, must include .zip extension
download_latest	logical to download latest if a more recent dataset is available
remove	logical if the downloaded folder is removed after unzipping

Details

This function downloads and unzips a folder of base tables used to calculate the benthic index from https://epcbooc.sharepoint.com/:u:/s/PublishedShare/IQD0Da-pbW8pTaZMg-93L9sTAZW1Y6hjj1TD_8RY0mFGiQ8?e=FIphoq&download=1.

Index the corresponding list element in the value column to view each dataset. For example, the stations data in the first row can be viewed as `benthicdata$value[[1]]`.

Value

A nested tibble of station, taxa, and field sample data.

Examples

```
## Not run:
# location to download data
path <- '~/Desktop/benthic.zip'

# load and assign to object
benthicdata <- read_importbenthic(path, download_latest = TRUE)

## End(Not run)
```

read_importentero	<i>Download Enterococcus data from the Water Quality Portal</i>
-------------------	---

Description

Download Enterococcus data from the Water Quality Portal

Usage

```
read_importentero(stas = NULL, startDate, endDate)
```

Arguments

stas	character, a vector of stations. If NULL, defaults to all stations in catchprecip .
startDate	character, starting date of observations as YYYY-MM-DD
endDate	character, ending date of observations as YYYY-MM-DD

Details

Retrieves Enterococcus sample data from selected stations and date range from the Water Quality Portal, <https://www.waterqualitydata.us>

Value

a data frame containing one row for each sample. Columns returned are:

date date, sample date

yr numeric, year of sample date

mo numeric, month of sample date

time character, sample time

time_zone character, sample time zone

long_name character, long name of bay segment subwatershed

bay_segment character, short name of bay segment subwatershed

station character, sample station

entero numeric, Enterococcus concentration

entero_censored logical, whether entero value was below the laboratory MDL, minimum detection limit

MDL numeric, minimum detection limit at the time of processing

entero_units character, units of measurement for entero

qualifier qualifier codes associated with sample

LabComments lab comments on sample

Latitude numeric, latitude in decimal degrees

Longitude numeric, longitude in decimal degrees

Examples

```
## Not run:
# stations to download
stas <- c('21FLHILL_WQX-101',
          '21FLHILL_WQX-102',
          '21FLHILL_WQX-103')

# download and read the data
entero_in <- read_importentero(stas = stas, startDate = '2023-01-01', endDate = '2023-02-01')

head(entero_in)

## End(Not run)
```

read_importepc	<i>Download and/or import local water quality file for internal use</i>
----------------	---

Description

Download and/or import local water quality file for internal use

Usage

```
read_importepc(xlsx, download_latest = FALSE, na = c("", "NULL"))
```

Arguments

- xlsx chr string path for local excel file, to overwrite if not current
- download_latest logical passed to [read_dlcurrent](#) to download raw data and compare with existing in xlsx if available
- na chr vector of strings to interpret as NA, passed to [read_xlsx](#)

Details

Loads the "RWMDDataSpreadsheet" worksheet from the file located at xlsx. The file is downloaded from <https://epcbocc.sharepoint.com/:x:/s/PublishedShare/IQAtFpJjaVHzS6ENmWdNf4csAX05fC5QFwI8U4b69Yye=tMe7Yy&download=1>. The files can be viewed at https://epcbocc.sharepoint.com/:f:/s/Share/EiypSSYdsEFCi84Sv_6-t7kBUYaXiIqN0B1n2w57Z_V3kQ?e=NdZQcU.
This function is used internally by [read_importtwq](#) and [read_importfib](#) because both use the same source data from the Environmental Protection Commission of Hillsborough County.

Value

An unformatted data.frame from EPC

See Also

```
read\_importtwq, read\_importfib
```

Examples

```
## Not run:
# file path
xlsx <- '~/Desktop/RWMDDataSpreadsheet_ThroughCurrentReportMonth.xlsx'

# load and assign to object
epcall <- read_importepc(xlsx, download_latest = T)

## End(Not run)
```

read_importfib	<i>Load local water quality file for Fecal Indicator Bacteria (FIB)</i>
----------------	---

Description

Load local water quality file for Fecal Indicator Bacteria (FIB)

Usage

```
read_importfib(xlsx, download_latest = FALSE, na = c("", "NULL"), all = TRUE)
```

Arguments

xlsx	chr string path for local excel file, to overwrite if not current
download_latest	logical passed to read_dlcurent to download raw data and compare with existing in xlsx if available
na	chr vector of strings to interpret as NA, passed to read_xlsx
all	logical indicating if all stations with FIB data are returned, default is TRUE, see details

Details

Loads the "RWMDDataSpreadsheet" worksheet from the file located at xlsx. The file is downloaded from <https://epcbocc.sharepoint.com/:x:/s/PublishedShare/IQAtFpJjaVHzS6ENmWdNf4csAX05fC5QFwI8U4b69Yye=tMe7Yy&download=1>. The files can be viewed at https://epcbocc.sharepoint.com/:f:/s/Share/EiypSSYdsEFCi84Sv_6-t7kBUYaXiIqN0B1n2w57Z_V3kQ?e=NdZQcU.

Returns FIB data including E. coli, Enterococcus, Fecal Coliform, and Total Coliform concentrations and waterbody class (freshwater as 1 or 3F, marine as 2 or 3M) for stations and sample dates. Values are returned for E. coli (ecoli), Enterococcus (entero), Fecal Coliform (fcolif), and Total Coliform (totcol). Values shown are # of colonies per 100 mL of water (#/100mL). Qualifier columns for each are also returned with the _q suffix. Qualifier codes can be interpreted from the source spreadsheet.

Concentrations noted with < or > in the raw data are reported as is, with only the numeric value shown. Samples with this notation can be determined from the qualifier columns.

If all = FALSE, only stations with AreaName in the source data as Hillsborough River, Hillsborough River Tributary, Alafia River, Alafia River Tributary, Lake Thonotosassa, Lake Thonotosassa Tributary, and Lake Roberta.

Value

A data.frame of formatted FIB data.

See Also

[read_formfib](#)

Examples

```
## Not run:
# file path
xlsx <- '~/Desktop/RWMDDataSpreadsheet_ThroughCurrentReportMonth.xlsx'

# load and assign to object
fibdata <- read_importfib(xlsx, download_latest = T)

## End(Not run)
```

read_importfim	<i>Load local FIM data for the Tampa Bay Nekton Index</i>
----------------	---

Description

Load local FIM data for the Tampa Bay Nekton Index

Usage

```
read_importfim(csv, download_latest = FALSE, locs = FALSE)
```

Arguments

csv	chr string path for local csv file, to overwrite if not current
download_latest	logical passed to read_dlcurent to download raw data and compare with existing in csv if available
locs	logical indicating if a spatial features object is returned with locations of each FIM sampling station

Details

Data downloaded from ftp://ftp.floridamarine.org/users/fim/tmac/NektonIndex/TampaBay_NektonIndexData.csv.

Value

A formatted data . frame with FIM data if locs = FALSE, otherwise a simple features object if locs = TRUE

See Also

[read_formwq](#), [read_importphyto](#)

Examples

```
## Not run:
# file path
csv <- '~/Desktop/fimraw.csv'

# load and assign to object
fimdata <- read_importfim(csv, download_latest = TRUE)

## End(Not run)
```

read_importphyto	<i>Load local phytoplankton cell count file</i>
------------------	---

Description

Load local phytoplankton cell count file

Usage

```
read_importphyto(xlsx, download_latest = FALSE, na = c("", "NULL"))
```

Arguments

- xlsx chr string path for local excel file, to overwrite if not current
- download_latest logical passed to [read_dldata](#) to download raw data and compare with existing in xlsx if available
- na chr vector of strings to interpret as NA, passed to [read_xlsx](#)

Details

Phytoplankton cell count data downloaded from <https://epcbocc.sharepoint.com/:x/s/PublishedShare/IQBWek-aqbRFSpzmJLZrL3CtAeLTo2Mv39-Df1b9LHFm5W0?e=XY4v4G&download=1>

Value

A data.frame of formatted water quality data.

See Also

```
read\_importwq
```

Examples

```
## Not run:
# file path
xlsx <- '~/Desktop/phyto_data.xlsx'

# load and assign to object
phytodata <- read_importphyto(xlsx, download_latest = T)

## End(Not run)
```

read_importprism	<i>Import PRISM daily weather data</i>
------------------	--

Description

Download and crop the latest weather rasters from Parameter-elevation Relationship on Independent Slopes Model (PRISM). Summarize by zone and save as CSV.

Usage

```
read_importprism(
  vars = c("tmin", "tmax", "tdmean", "ppt"),
  vars_ytd = c("ppt"),
  date_beg = as.Date("1981-01-01"),
  date_end = Sys.Date(),
  bbox = sf::st_bbox(sf_zones),
  dir_tif = sub("\\.^[^.]*$", "", zonal_csv),
  pfx_tif = "prism_daily_",
  sf_zones,
  fld_zones,
  zonal_fun = "mean",
  zonal_csv,
  redo_zonal = F,
  verbose = F
)
```

Arguments

vars	character vector of PRISM variables to download and crop. The default is c("tmin", "tmax", "tdmean", "ppt"). See Details for more.
vars_ytd	character vector of PRISM variables to be summed for the year to date, which gets recorded as a new layer with the suffix "ytd". The default is for precipitation c("ppt"), so a layer "ppt_ytd" gets added to PRISM daily rasters.
date_beg	defaults to the start of PRISM daily availability 1981-01-01.

date_end	defaults to today's date, but will be adjusted to the most recently available data, per update schedule . CAUTION: If you try to request the same date-variable in a given day, you will get an error from the PRISM service with a message like "You have tried to download the file PRISM_tdmean_stable_4kmD2_19810101_bil.zip more than twice in one day (Pacific local time). Note that repeated offenses may result in your IP address being blocked."
bbox	bounding box of the spatial extent to crop PRISM daily rasters, which could be bigger than <code>st_bbox(sf_zones)</code> .
dir_tif	directory path to store downloaded and cropped PRISM daily rasters (as Geo-TIFF) across years with file name of format <code>prism_daily_{month}-{day}.tif</code> . The default path is <code>sub("\\.^[^.]*\$", "", zonal_csv)</code> . Date and version in the raster layers will be compared with the update schedule to determine if it should be updated, otherwise will be skipped.
px_tif	prefix for the PRISM daily raster files, appended by the month-day. Defaults to "prism_daily_".
sf_zones	spatial feature object (sf) with zones to extract zonal statistics from PRISM daily weather data
fld_zones	character vector of unique field name(s) in <code>sf_zones</code> to include in extracted zonal statistics
zonal_fun	function to apply to the <code>terra::zonal()</code> function. Defaults to "mean". Other options are "min", "max", "sum", "isNA", and "notNA".
zonal_csv	path to output of zonal statistics (<code>terra::zonal()</code>) as table in CSV format
redo_zonal	logical whether to recalculate the zonal statistics. Defaults to False.
verbose	logical whether to show informative messages on processing

Details

The **Parameter-elevation Relationship on Independent Slopes Model (PRISM)** is a combined dataset consisting of ground gauge station and RADAR products, at a 4km grid resolution covering the contiguous United States from 1981 to present.

Variables:

The available variables (vars) are:

- tmin: minimum temperature (°C)
- tmax: maximum temperature (°C)
- tmean: mean temperature (°C; `tmean = mean(tmin, tmax)`)
- tdmean: mean dew point temperature (°C)
- ppt: total precipitation (mm; rain and snow)
- vpdmin: daily minimum vapor pressure deficit
- vpdmax: daily maximum vapor pressure deficit

Updates:

The daily PRISM data is available over three stability levels, or 8 versions given by the **update schedule**:

- stable: data is considered final and will not change (before 6 months)

- provisional: data is considered final but may be updated (previous 6 months)
- early: data is preliminary and may be updated (this month)

Processing details:

1. The PRISM **update schedule** is downloaded and processed in a data frame to understand the latest date and variable available, along with version and date updated.
2. Any existing rasters in `dir_tif` are fetched based on a common naming structure for the raster file name (`prism_daily_{month}-{day}.tif`) and layer names (`{date}_{variable}_v{version}-{date_updated}`) into a data frame.
3. Based on intersecting above with the requested dates, any missing or more recently updated variable-date PRISM rasters are downloaded and cropped to the bounding box (bbox) and written as GeoTIFFs (*.tif). Layers are renamed to include extra information on `{version}` (1-8) and `{date_updated}`.
4. Summary statistics (based on `sf_zones` and `zonal_fun`) are calculated on the cropped PRISM daily rasters.

More on PRISM:

For more on **Parameter-elevation Relationship on Independent Slopes Model (PRISM)**, see:

- **PRISM model methods** (Daly et al., 2008)
- **PRISM dataset descriptions**
- **PRISM download methods** information (FTP, web services)

Value

data frame with the following columns:

- `{fld_zones}` column(s) specified from input `sf_zones`
- `{zonal_fun}`: value of input function (e.g. "mean") summarizing raster to zone
- `date`: date of PRISM daily raster
- `variable`: PRISM variable
- `version`: PRISM version
- `date_updated`: date of PRISM daily raster update

Examples

```
## Not run:
# setup output directory and table
dir_tif <- system.file("prism", package = "tbptools")
zonal_csv <- system.file("prism/_zones.csv", package = "tbptools")

# run function for Tampa Bay watersheds for first 3 days and 4 variables
d <- read_importprism(
  vars      = c("tdmean", "ppt"),
  date_beg  = as.Date("1981-01-01"),
  date_end  = as.Date("1981-01-02"),
  dir_tif   = dir_tif,
  sf_zones  = tbsegshed,
  fld_zones = "bay_segment",
```

```

    zonal_csv = zonal_csv)

# show raster files, layers and plot
tifs <- list.files(dir_tif, pattern = ".tif$", full.names = T)
basename(tifs)
r <- terra::rast(tifs[1])
r
names(r)
terra::plet(
  r[[3]],
  main = names(r)[3],
  col = "Spectral",
  tiles = "CartoDB.DarkMatter")

# show summary by zone
d

## End(Not run)

```

read_importrain

*Download daily precip data and summarise by station catchment***Description**

Download daily precip data and summarise by station catchment

Usage

```
read_importrain(curyr, catchpixels, mos = 1:12, quiet = T)
```

Arguments

curyr	numeric for year
catchpixels	data.frame with columns named 'station' and 'pixel'. A data frame has been created for key Enterococcus stations, associating each station with all pixels in that station's catchment layer. This is the tbeptools object 'catchpixels'.
mos	numeric vector for months to download
quiet	logical for messages

Details

Data from the Southwest Florida Water Management District's (SWFWMD) ftp site: ftp://ftp.swfwmd.state.fl.us/pub/radar_ra

Value

data.frame with station, date, rain columns as a daily average (inches) for all pixels in a catchment

Examples

```
## Not run:
read_importrain(2021, catchpixels, quiet = F)

## End(Not run)
```

read_importrainmany	<i>Run read_importrain for multiple years</i>
---------------------	---

Description

Run read_importrain for multiple years

Usage

```
read_importrainmany(yrs, quiet = FALSE, sleep = 5)
```

Arguments

yrs	numeric vector of years to download (do not need to be in order)
quiet	logical passed to read_importrain to suppress messages (default)
sleep	numeric number of seconds to pause between years

Details

This function is a simple wrapper for [read_importrain](#) to download multiple years of rainfall data from the Southwest Florida Water Management District's (SWFWMD) ftp site: ftp://ftp.swfwmd.state.fl.us/pub/radar_rainfal. The function will pause for sleep seconds between years to avoid overloading the server.

Value

A data frame identical to that returned by [read_importrain](#) with the years requested

Examples

```
## Not run:
read_importrainmany(c(2021, 2022), quiet = F)

## End(Not run)
```

read_importsealevels *Import monthly sea levels by station from NOAA Tides and Currents*

Description

Under **NOAA Tides and Currents**, there is the NOAA Center for Operational Oceanographic Products and Services (CO-OPS). This function uses the **CO-OPS Data Retrieval API** to extract sea level data by station.

Usage

```
read_importsealevels(
  path_csv,
  download_latest = TRUE,
  df_stations = sealevelstations[, c("station_id", "station_name")],
  api_url = "https://api.tidesandcurrents.noaa.gov/api/prod/datagetter",
  beg_int = 19010101,
  end_int = as.integer(format(lubridate::today(), "%Y%m%d")),
  product = "monthly_mean",
  datum = "stdn",
  time_zone = "lst",
  units = "metric"
)
```

Arguments

path_csv	chr string path of CSV file to store tabular output. (Overwrites existing file.)
download_latest	logical to download latest. (Overwrites existing file.)
df_stations	data frame of stations with column station_id (integer). Defaults to sealevelstations , subset to columns station_id and station_name.
api_url	chr string URL for NOAA Center for Operational Oceanographic Products and Services (CO-OPS) API. Defaults to the CO-OPS API for data retrieval: https://api.tidesandcurrents.noaa.gov/api/prod/datagetter .
beg_int	int integer of beginning date in YYYYMMDD format. Defaults to 19010101.
end_int	int integer of ending date in YYYYMMDD format. Defaults to lubridate::today().
product	chr string of product type. For options, see Data Products CO-OPS API . Defaults to "monthly_mean": "verified monthly mean water level data for the station."
datum	chr string of datum. Defaults to "stdn": "station datum - original reference that all data is collected to, uniquely defined for each station." For options, see Datum CO-OPS API
time_zone	Time zone. Defaults to "lst": "local standard time." However, this does not get used with the monthly_mean product. For options, see Time Zone CO-OPS API .
units	chr string of units. Defaults to "metric" (i.e., meters). For other options (i.e., "english"), see Units CO-OPS API .

Value

Given the default arguments in (and especially `product = "monthly_mean"`), this function returns a data frame from reading `path_csv` (updated if `download_latest = TRUE` or newly written if `path_csv` does not exist) having the following fields:

- `station_id`: integer column from input argument `df_stations`
- `station_name`: character column from input argument `df_stations`
- `date`: first of the month given by year and month from API output
- `year`: year of the data
- `month`: month of the data
- `mhhw`: Mean Higher-High Water
- `mhw`: Mean High Water
- `msl`: Mean Sea Level
- `mtl`: Mean Tide Level
- `mlw`: Mean Low Water
- `mllw`: Mean Lower-Low Water
- `dtl`: Mean Diurnal Tide Level
- `gt`: Great Diurnal Range
- `mn`: Mean Range of Tide
- `dhq`: Mean Diurnal High Water Inequality
- `dlq`: Mean Diurnal Low Water Inequality
- `hwi`: Greenwich High Water Interval (in Hours)
- `lwi`: Greenwich Low Water Interval (in Hours)
- `highest`: Highest Tide
- `lowest`: Lowest Tide
- `inferred`: A flag that when set to 1 indicates that the water level value has been inferred

For more details on these output data columns, see [About Tidal Datums | NOAA Tides & Currents](#).

Examples

```
read_importsealevels(tempfile("sealevels", fileext=".csv"))
```

read_importsediment	<i>Download and import sediment data for Tampa Bay</i>
---------------------	--

Description

Download and import sediment data for Tampa Bay

Usage

```
read_importsediment(path, download_latest = FALSE, remove = FALSE)
```

Arguments

path	chr string for local path where the zipped folder will be downloaded, must include .zip extension
download_latest	logical to download latest if a more recent dataset is available
remove	logical if the downloaded folder is removed after unzipping

Details

This function downloads and unzips a folder of results tables from <https://epcbocc.sharepoint.com/:u:/s/PublishedShare/IQAr2aDfEyegTrJLR61qt1CxAQYxfnVYYWAwkbtArczrJS4?e=EQSG03&download=1> (viewable at https://epcbocc.sharepoint.com/:f:/s/Share/Et0JfziTTa9FliL1oR0b90sBRZU-n060fu_0NRC162hHjQ?e=4gUXgJ).

The row entries for columns "BetweenTELPEL" and "ExceedsPEL" for rows where the "Qualifier" column is "U" or "T" (below detection, not detected) are assigned NA, regardless of the entry in the source data.

Value

A data.frame of sediment data for Tampa Bay

Examples

```
## Not run:
# location to download data
path <- '~/Desktop/sediment.zip'

# load and assign to object
sedimentdata <- read_importsediment(path, download_latest = TRUE)

## End(Not run)
```

read_importwq	<i>Load local water quality file</i>
---------------	--------------------------------------

Description

Load local water quality file

Usage

```
read_importwq(xlsx, download_latest = FALSE, na = c("", "NULL"), all = FALSE)
```

Arguments

xlsx	chr string path for local excel file, to overwrite if not current
download_latest	logical passed to read_dlcurrent to download raw data and compare with existing in <code>xlsx</code> if available
na	chr vector of strings to interpret as NA, passed to read_xlsx
all	logical indicating if all water quality parameters are returned, see details

Details

Loads the "RWMDDataSpreadsheet" worksheet from the file located at `xlsx`. The file is downloaded from <https://epcbocc.sharepoint.com/:x:/s/PublishedShare/IQAtFpJjaVHzS6ENmWdNf4csAX05fC5QFwI8U4b69Yye=tMe7Yy&download=1>. The files can be viewed at https://epcbocc.sharepoint.com/:f:/s/Share/EiypSSYdsEFCi84Sv_6-t7kBUYaXiIqN0B1n2w57Z_V3kQ?e=NdZQcU.

Water quality parameters returned by default are total nitrogen (tn), Secchi depth (sd), chlorophyll-a (chl_a), salinity (top, mid, and bottom depths, Sal_ prefix), water temperature (top, mid, and bottom depths, Temp_Water_ prefix), turbidity (Turbidity_JTU-NTU), and water color (Color_345_F45). Additional qualifier columns for each that include the _Q suffix are also returned, excluding salinity and water temperature. All other water quality parameters and qualifiers can be returned by setting `all = T`.

Value

A data.frame of formatted water quality data.

See Also

[read_formwq](#), [read_importphyto](#)

Examples

```
## Not run:
# file path
xlsx <- '~/Desktop/RWMDDataSpreadsheet_ThroughCurrentReportMonth.xlsx'

# load and assign to object
epcdata <- read_importwq(xlsx, download_latest = T)

# get all water quality parameters
epcdataall <- read_importwq(xlsx, download_latest = T, all = T)

## End(Not run)
```

read_importwqp	<i>Import data from the Water Quality Portal</i>
----------------	--

Description

Import data from the Water Quality Portal

Usage

```
read_importwqp(org, type, trace = T, max_retries = 5)
```

Arguments

org	chr string indicating the organization identifier, see details
type	chr string indicating data type to download, one of "wq" or "fib"
trace	logical indicating whether to display progress messages, default TRUE
max_retries	integer indicating maximum number of retries on request failure, default 5

Details

This function retrieves data from the Water Quality Portal API (<https://www.waterqualitydata.us/>) for selected counties in or around the Tampa Bay watershed. The type of data returned are defined by the type argument as either "wq" or "fib" for water quality of Fecal Indicator Bacteria data, respectively.

The org argument retrieves data for the specific organization. Valid entries for org include "21FLCOSP_WQX" (City of St. Petersburg), "21FLDOH_WQX" (Florida Department of Health), "21FLHILL_WQX" (Hillsborough County), "21FLMANA_WQX" (Manatee County), "21FLPASC_WQX" (Pasco County), "21FLPDEM_WQX" (Pinellas County), "21FLPOLK_WQX" (Polk County), "21FLTPA_WQX" (Florida Department of Environmental Protection, Southwest District), or "21FLHESD_WQX" (Hillsborough County Environmental Services Division). The naming convention follows the Organization ID in the Water Quality Portal.

The function fetches results and station metadata, combines and formats them using the read_formwqp function, and returns the processed data as a data frame. Parameters are specific to the type argument.

Requests are retried with exponential backoff (via [util_importwqp](#)) up to max_retries times if the Water Quality Portal API returns an intermittent failure.

Value

A data frame containing the imported data for the selected county

See Also

[read_formwqp](#)

Examples

```
## Not run:
# get Manatee County water quality data
mancodata <- read_importwqp(org = '21FLMANA_WQX', type = 'wq', trace = T)

# get Pinellas County FIB data
pincodata <- read_importwqp(org = '21FLPDEM_WQX', type = 'fib', trace = T)

## End(Not run)
```

read_importwqwa	<i>Import data from the Water Atlas API</i>
-----------------	---

Description

Import data from the Water Atlas API

Usage

```
read_importwqwa(
  dataSource,
  parameter,
  start_date = NULL,
  end_date = NULL,
  trace = TRUE
)
```

Arguments

dataSource	Character for the data source to retrieve, e.g., WIN_21FLHILL
parameter	Character for the parameter to retrieve
start_date	Numeric for the start date in ISO format, optional
end_date	Numeric for the end date in ISO format, optional
trace	Logical indicating whether to display progress messages, default TRUE

Details

This function retrieves sampling data from the Water Atlas API (<https://dev.api.wateratlas.org/redoc/index.html#tag/Sampling-Data/operation/StreamSamplingData>) using the specified data source and parameter. Optional start and end dates can be provided to filter the data by date range. The function processes the NDJSON response stream and returns a data frame of sampling records.

See [util_importwqwa](#) for retrieving metadata such as available data sources and parameters.

Value

A data frame containing the imported sampling data

Examples

```
read_importwqwa('WIN_21FLPDEM', 'Chla_uql', '2023-01-01', '2023-02-01')
```

read_importwqwin	<i>Retrieve water quality data from the Florida Department of Environmental Protection’s Watershed Information Network (WIN)</i>
------------------	--

Description

Retrieve water quality data from the Florida Department of Environmental Protection’s Watershed Information Network (WIN)

Usage

```
read_importwqwin(  
  start_date,  
  end_date,  
  org_id,  
  verbose = FALSE,  
  max_retries = 5  
)
```

Arguments

start_date	character string for the start date in the format "YYYY-MM-DD"
end_date	character string for the end date in the format "YYYY-MM-DD"
org_id	character string for the organization ID
verbose	logical indicating whether to print verbose output
max_retries	integer indicating maximum number of retries for timeout errors (default = 5)

Details

This function implements `util_importwqwin` iteratively to retrieve water quality results for the specified organization ID and start date. Data are retrieved using the API at <https://prodapps.dep.state.fl.us/dear-watershed/swagger-ui/index.html>.

Value

A data frame containing the water quality data

Examples

```
## Not run:
dat <- read_importwqwin("2025-01-15", "2025-02-15", "21FLMANA", verbose = TRUE)
head(dat)

## End(Not run)
```

<code>read_prism_rasters</code>	<i>Read PRISM raster layers into data frame</i>
---------------------------------	---

Description

Scan directory for rasters (*.tif) and parse PRISM layer names into columns of a data frame.

Usage

```
read_prism_rasters(dir_tif)
```

Arguments

`dir_tif` directory path PRISM daily rasters were downloaded and cropped using `read_importprism()`.

Details

Any existing rasters in `dir_tif` are scanned based on a common naming structure for the raster file name (`prism_daily_{month}-{day}.tif`) and layer names (`{date}_{variable}_v{version}-{date_updated}`) into a data frame.

For more on the [Parameter-elevation Relationship on Independent Slopes Model \(PRISM\)](#), see `read_importprism()`.

Value

A data frame with columns for:

- `path_tif` path to GeoTIFF
- `lyr` layer name
- `date` date of modeled weather data

- md month-day
- variable variable of modeled weather data
- version version (1-8)
- date_updated date the model was updated

Examples

```
dir_tif <- system.file("prism", package = "tbeptools")
read_prism_rasters(dir_tif)
```

read_transect	<i>Import JSON seagrass transect data from Water Atlas</i>
---------------	--

Description

Import JSON seagrass transect data from Water Atlas

Usage

```
read_transect(training = FALSE, raw = FALSE, retry = 5)
```

Arguments

training	logical if training data are imported or the complete database
raw	logical indicating if raw, unformatted data are returned, see details
retry	integer indicating maximum number of retries on request failure, default 5

Details

The function imports a JSON file from the USF Water Atlas. If `training = TRUE`, a dataset from the TBEP training survey is imported from <https://tampabay.wateratlas.usf.edu/seagrass-transect-data-portal/api/assessments/training>. If `training = FALSE`, the entire transect survey database is imported from https://tampabay.wateratlas.usf.edu/seagrass-transect-data-portal/api/assessments/all__use-with-care.

Abundance is reported as a numeric value from 0 -5 for Braun-Blanquet coverage estimates, blade length is in cm, and short shoot density is number of shoots per square meter. The short density is corrected for quadrat size included in the raw data.

If `raw = TRUE`, the unformatted data are returned. The default is to use formatting that allows the raw data to be used with the downstream functions. The raw data may have extra information that may be of use outside of the plotting functions in this package.

The request is retried with exponential backoff up to `retry` times if the JSON request fails, since the Water Atlas API intermittently fails.

Value

data frame

Examples

```
## Not run:
# get training data
transect <- read_transect(training = TRUE)

# import all transect data
transect <- read_transect()

## End(Not run)
```

seagrass	<i>Seagrass coverage by year</i>
----------	----------------------------------

Description

Seagrass coverage by year

Usage

```
seagrass
```

Format

A data frame used to create the flagship seagrass coverage graphic:

- Year** int
- Acres** num
- Hectares** num

Details

Original data are from the Southwest Florida Water Management District and available online at <https://data-swfwmd.opendata.arcgis.com/>. Totals are for all of Tampa Bay.

Examples

```
## Not run:

seagrass <- structure(list(
  Year = c(1950L, 1982L, 1988L, 1990L, 1992L, 1994L, 1996L,
    1999L, 2001L, 2004L, 2006L, 2008L, 2010L, 2012L, 2014L,
    2016L, 2018L, 2020L, 2022L, 2024L),
  Acres = c(40420, 21650, 23285, 25226, 25753, 26518, 26916,
    24841, 26078, 27021, 28299, 29647, 32897, 34642, 40294.71,
    41655.16, 40651.55, 34298, 30137, 31563),
  Hectares = c(16357.39, 8761.44, 9423.11, 10208.6, 10421.87,
    10731.45, 10892.52, 10052.8, 10553.39, 10935.01, 11452.2,
    11997.72, 13312.94, 14019.27, 16306.69, 16857.25, 16451.1,
```

```
      13880, 12196, 12773)
    ), class = "data.frame", row.names = c(NA, -20L))

save(seagrass, file = 'data/seagrass.RData', compress = 'xz')

## End(Not run)
```

sealevelstations	<i>Sea level stations in Tampa Bay</i>
------------------	--

Description

Stations in Tampa Bay with sea level data through [CO-OPS Data Retrieval API](#) for use with [read_importsealevels\(\)](#).

Usage

```
sealevelstations
```

Format

A data frame with 4 rows and 5 columns:

- station_id integer station identifier
- station_name character station name
- longitude double longitude
- latitude double latitude
- date_est date established

sedimentdata	<i>Sediment data for the Tampa Bay current as of 20251210</i>
--------------	---

Description

Sediment data for the Tampa Bay current as of 20251210

Usage

```
sedimentdata
```

Format

A data.frame with 231727 rows and 24 variables:

ProgramId int
ProgramName chr
FundingProject chr
yr int
AreaAbbr chr
StationID int
StationNumber chr
Latitude num
Longitude num
Replicate chr
SedResultsType chr
Parameter chr
ValueAdjusted num
Units chr
Qualifier chr
MDLnum num
PQLnum num
TEL num
PEL num
BetweenTELP chr
ExceedsPEL chr
PreparationDate chr
AnalysisTimeMerge chr
PELGrade Factor

Examples

```
## Not run:  
# location to download data  
path <- '~/Desktop/sediment.zip'  
  
# load and assign to object  
sedimentdata <- read_importsediment(path, download_latest = TRUE, remove = TRUE)  
  
save(sedimentdata, file = 'data/sedimentdata.RData', compress = 'xz')  
  
## End(Not run)
```

`sgmanagement`*Seagrass management areas for Tampa Bay*

Description

Seagrass management areas for Tampa Bay

Usage

`sgmanagement`

Format

A simple features `sf` object (MULTIPOLYGON) with 30 features and 1 field, `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs`

areas `int`

Details

These polygons are seagrass management areas for Tampa Bay that provide a finer division of areas within major segments (`tbseg`) having relevance for locations of seagrass beds.

Examples

```
## Not run:
library(sf)
library(dplyr)
library(tools)

# NAD83(HARN) / Florida West (ftUS)
# same as sgseg
prj <- 2882

# create sf object of boundaries
sgmanagement <- st_read(
  dsn = '~/Desktop/TBEP/GISboundaries/Seagrass_Management_Areas/TBEP_SG_MA_FINAL_Projectfix.shp',
  drivers = 'ESRI Shapefile'
) %>%
  select(areas = TBEP_SG_MA) %>%
  st_zm() %>%
  st_transform(prj)

# save
save(sgmanagement, file = 'data/sgmanagement.RData', compress = 'xz')

## End(Not run)
```

sgseg

Seagrass segment reporting boundaries for southwest Florida

Description

Seagrass segment reporting boundaries for southwest Florida

Usage

sgseg

Format

A simple features `sf` object (POLYGON) with 22 features and 1 field, +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs

segment chr

Details

These polygons are used by Southwest Florida Water Management District for summarizing seagrass coverage estimates by major coastal and estuarine boundaries.

Examples

```
## Not run:
library(sf)
library(dplyr)
library(tools)

# create sf object of boundaries
# make sure projection does not change
sgseg <- st_read(
  dsn = '~/Desktop/TBEP/GISboundaries/Seagrass_Segment_Boundaries/Seagrass_Segment_Boundaries.shp',
  drivers = 'ESRI Shapefile'
) %>%
  select(segment = SEAGRASSSE) %>%
  mutate(
    segment = tolower(segment),
    segment = case_when(
      segment == 'terra ciea bay' ~ 'Terra Ceia Bay',
      T ~ segment
    ),
    segment = toTitleCase(segment)
  )

# save
save(sgseg, file = 'data/sgseg.RData', compress = 'xz')

## End(Not run)
```

show_ambimatrix	<i>Plot a matrix of AMBI scores over time by bay segment</i>
-----------------	--

Description

Plot a matrix of AMBI scores over time by bay segment

Usage

```
show_ambimatrix(
  ambiscr,
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB", "All"),
  yrrng = c(1993, 2024),
  window = TRUE,
  alph = 1,
  txtsz = 2.5,
  family = "sans",
  rev = FALSE,
  position = "top",
  plotly = FALSE,
  width = NULL,
  height = NULL
)
```

Arguments

ambiscr	input data frame as returned by anlz_ambiscr ; the AMBI variant is detected automatically from the column names
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB", "All"
yrrng	numeric indicating year ranges to evaluate
window	logical indicating whether to use a rolling 5-year window (default TRUE) or single year values (FALSE) for the bay segment categories, see details
alph	numeric indicating alpha value for score category colors
txtsz	numeric for size of text in the plot
family	optional chr string indicating font family for text labels
rev	logical if factor levels for bay segments are reversed
position	chr string of location for bay segment labels, default on top, passed to scale_x_discrete
plotly	logical if matrix is created using plotly
width	numeric for width of the plot in pixels, only applies if plotly = TRUE
height	numeric for height of the plot in pixels, only applies if plotly = TRUE

Details

An additional summary is provided for the entire bay as an unweighted summary across categories ("All").

The default behavior is to use a rolling five-year window to calculate the percent of sites in each AMBI category by bay segment. This applies only to years 2005 and later, where the counts from the current year and the prior four years are summed to calculate the percentages. This is intended to help smooth out inter-annual variability due to reduced sampling effort from 2005 to present. If window = FALSE, then only single year values are used.

The color scale reflects the adjusted AMBI classification (0-10 scale): Unpolluted (dark green), Slightly Polluted (light green), Meanly Polluted (yellow), Heavily Polluted (orange), and Extremely Polluted (red).

The matrix tile for each bay segment and year shows the dominant AMBI category based on the plurality of sites. The dominant category is the one with the highest proportion of sites in that year and bay segment.

Value

A `ggplot` object showing trends over time in AMBI scores for each bay segment if `plotly = FALSE`, otherwise a `plotly` object

Examples

```
ambiscr <- anlz_ambiscr(benthicdata)
show_ambimatrix(ambiscr)
```

show_ambitab	Create a summary table of AMBI category percentages by bay segment
--------------	--

Description

Create a summary table of AMBI category percentages by bay segment

Usage

```
show_ambitab(
  ambiscr,
  bay_segment = c("OTB", "HB", "MTB", "LTB", "TCB", "MR", "BCB"),
  yrrng = c(1993, 2024)
)
```

Arguments

ambiscr	input data frame as returned by <code>anlz_ambiscr</code> , the AMBI variant is detected automatically from the column names
bay_segment	chr string for the bay segment, one to many of "OTB", "HB", "MTB", "LTB", "TCB", "MR", "BCB"
yrrng	numeric vector of length two indicating the year range to summarise

Details

Only sampling funded by TBEP and as part of the routine EPC benthic monitoring program are included. Azoic stations are excluded from counts and percentages. The year range filter applies to all years from `yrng[1]` through `yrng[2]`.

Column header colours match those used in `show_ambimatrix`: Unpolluted (dark green), Slightly Polluted (light green), Meanly Polluted (yellow), Heavily Polluted (orange), and Extremely Polluted (red).

Value

A `flextable` object showing the count and percentage of sites in each AMBI category by bay segment over the requested year range. The bottom row ("All") summarises across all included bay segments.

Examples

```
ambiscr <- anlz_ambiscr(benthicdata)
show_ambitab(ambiscr)
```

show_ambitrend	<i>Plot mean AMBI scores over time by bay segment</i>
----------------	---

Description

Plot mean AMBI scores over time by bay segment

Usage

```
show_ambitrend(
  ambiscr,
  ambiscr_tb = NULL,
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB", "All"),
  yrng = c(1993, 2024),
  ysc1 = TRUE,
  plotly = FALSE,
  width = NULL,
  height = NULL
)
```

Arguments

- `ambiscr` input data frame as returned by `anlz_ambiscr`, the AMBI variant (conventional or Tampa Bay-specific) is detected automatically from the column names
- `ambiscr_tb` optional second input data frame from `anlz_ambiscr` for overlaying the other AMBI variant on the same plot

bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB", "All". When multiple segments are selected (or "All"), scores are averaged across all sites in the selected segments.
yrrng	numeric vector of length two indicating the year range to plot
yscl	logical indicating whether the y-axis should span the full adjusted AMBI range (0 to 10, default TRUE) or be scaled to the range of the annual means (FALSE)
plotly	logical if the plot is created using plotly
width	numeric for width of the plot in pixels, only applies if plotly = TRUE
height	numeric for height of the plot in pixels, only applies if plotly = TRUE

Details

The background of the plot is shaded by AMBI pollution category using the adjusted score thresholds (0-10 scale, higher = healthier): Unpolluted (8.29-10, dark green), Slightly Polluted (5.29-8.29, light green), Meanly Polluted (2.89-5.29, yellow), Heavily Polluted (1.39-2.89, orange), and Extremely Polluted (0-1.39, red).

Only sampling funded by TBEP and as part of the routine EPC benthic monitoring program are included.

If both ambiscr and ambiscr_tb are provided, both series are shown on the same plot with dark grey used for the first series and black for the second. The AMBI variant for each input is detected automatically from the column names (AMBI or TBAMBI).

Value

A `ggplot` object showing mean adjusted AMBI scores by year if `plotly = FALSE`, otherwise a `plotly` object

Examples

```
ambiscr    <- anlz_ambiscr(benthicdata)
ambiscr_tb <- anlz_ambiscr(benthicdata, type = 'AMBI-TB')
show_ambitrend(ambiscr, ambiscr_tb)
```

show_annualassess	<i>Create a table for the annual management outcome assessments</i>
-------------------	---

Description

Create a table for the annual management outcome assessments for chlorophyll-a and light attenuation by bay segment

Usage

```
show_annualassess(  
  epdata,  
  yrsl,  
  partialyr = F,  
  caption = F,  
  family = "Arial",  
  txtsz = 12,  
  width = NULL  
)
```

Arguments

epdata	data frame of ep data returned by read_importwq
yrsl	numeric indicating chosen year
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter
caption	logical indicating if a caption is added using set_caption
family	chr string indicating font family for text labels
txtsz	numeric indicating font size
width	optional numeric value indicating width in inches

Value

A [flextable](#) object showing the segment-averaged chlorophyll-a and light attenuation for the selected year, with bay segment names colored by the management outcome used in [show_matrix](#).

Examples

```
show_annualassess(epdata, yrsl = 2025)  
show_annualassess(epdata, yrsl = 2025, caption = TRUE)
```

show_boxplot	<i>Plot monthly chlorophyll or light attenuation values for a segment</i>
--------------	---

Description

Plot monthly chlorophyll or light attenuation values for a bay segment

Usage

```
show_boxplot(
  epcdata,
  param = c("chla", "la"),
  yrsl = NULL,
  yrrng = c(1975, 2025),
  ptsz = 0.5,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  trgs = NULL,
  family = "sans",
  labelexp = TRUE,
  txtlab = TRUE,
  partialyr = FALSE,
  points = TRUE
)
```

Arguments

epcdata	data frame of epc data returned by read_importwq
param	chr string for which parameter to plot, one of "chla" for chlorophyll or "la" for light attenuation
yrsl	numeric for year to emphasize, shown as separate red points on the plot
yrrng	numeric vector indicating min, max years to include
ptsz	numeric indicating point size of observations not in yrsl
bay_segment	chr string for the bay segment, one of "OTB", "HB", "MTB", "LTB"
trgs	optional data.frame for annual bay segment water quality targets, defaults to targets
family	optional chr string indicating font family for text labels
labelexp	logical indicating if y axis and target labels are plotted as expressions, default TRUE
txtlab	logical indicating if a text label for the target value is shown in the plot
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter
points	logical indicating if jittered point observations, including outliers, are shown in the plot

Details

Points not included in yrsl are plotted over the box plots using [position_jitter](#). Use ptsz = -1 to suppress. The dotted line in the plot shows the large exceedance value.

Value

A [ggplot](#) object

Examples

```
show_boxplot(epcdata, bay_segment = 'OTB')
```

show_compplot

Make a bar plot for transect training group comparisons

Description

Make a bar plot for transect training group comparisons

Usage

```
show_compplot(
  transect,
  yr,
  site,
  species = c("Halodule", "Syringodium", "Thalassia", "Halophila", "Ruppia"),
  varplo = c("Abundance", "Blade Length", "Short Shoot Density"),
  base_size = 18,
  txtsize = 10,
  size = 1
)
```

Arguments

transect	data frame returned by read_transect
yr	numeric for year of training data to plot
site	chr string indicating site results to plot
species	chr string indicating which species to plot
varplo	chr string indicating which variable to plot
base_size	numeric indicating text scaling size for plot
txtsize	numeric indicating text size for x-axis labels
size	numeric indicating line size

Value

A [ggplot](#) object

Examples

```
transect <- read_transect(training = TRUE)
show_compplot(transect, yr = 2025, site = '3', species = 'Halodule', varplo = 'Abundance')
```

show_enteromap	<i>Map Enterococcus results by month, year, and location</i>
----------------	--

Description

Map Enterococcus results by month, year, and location

Usage

```
show_enteromap(
  fibdata,
  yrsel,
  mosel,
  areasel = NULL,
  wetdry = FALSE,
  precipdata = NULL,
  temporal_window = NULL,
  wet_threshold = NULL,
  addsta = FALSE
)
```

Arguments

fibdata	data frame of Enterococcus sample data as returned by enterodata or anlz_fibwetdry
yrsel	optional numeric to filter data by year
mosel	optional numeric to filter data by month
areasel	optional character string to filter output by stations in the long_name column of enterodata, see details
wetdry	logical; if TRUE, incorporate wet/dry differences (this will result in a call to anlz_fibwetdry , in which case temporal_window and wet_threshold are required). If FALSE (default), do not differentiate between wet and dry samples.
precipdata	input data frame as returned by read_importtrain . columns should be: station, date (yyyy-mm-dd), rain (in inches). The object catchprecip has this data from 1995-2023 for select Enterococcus stations. If NULL, defaults to catchprecip .
temporal_window	numeric; required if wetdry is TRUE. number of days precipitation should be summed over (1 = day of sample only; 2 = day of sample + day before; etc.)
wet_threshold	numeric; required if wetdry is TRUE. inches accumulated through the defined temporal window, above which a sample should be defined as being from a 'wet' time period
addsta	logical to add station labels to the map, default FALSE

Details

Placing the mouse cursor over an item on the map will reveal additional information about a station.

Value

A leaflet map for the selected year, month, and area showing stations and FIB concentration category

See Also

[anlz_enteromap](#) for details on the categories

Examples

```
show_enteromap(enterodata, yrsel = 2020, mosel = 9)

# wet/dry samples
show_enteromap(enterodata, yrsel = 2020, mosel = 9, wetdry = TRUE,
               temporal_window = 2, wet_threshold = 0.5)

# Old Tampa Bay only
show_enteromap(enterodata, yrsel = 2020, mosel = 9, areasel = "Old Tampa Bay")
```

show_fibmap	<i>Map Fecal Indicator Bacteria (FIB) results by month, year, and location</i>
-------------	--

Description

Map Fecal Indicator Bacteria (FIB) results by month, year, and location

Usage

```
show_fibmap(fibdata, yrsel, mosel, areasel = NULL, addsta = FALSE)
```

Arguments

fibdata	input FIB data.frame as returned by read_importfib or read_importwqp , see details
yrsel	optional numeric value to filter output by years in fibdata
mosel	optional numeric value to filter output by month in fibdata
areasel	optional character string to filter output by stations in the area column of fibdata, see details
addsta	logical to add station labels to the map, default FALSE

Details

Placing the mouse cursor over an item on the map will reveal additional information about a station. See the help file for [anlz_fibmap](#) for details on the categories and additional arguments.

Value

A leaflet map for the selected year, month, and area showing stations and FIB concentration category

See Also

[anlz_fibmap](#)

Examples

```
# show selected year, month
show_fibmap(fibdata, yrsel = 2023, mosel = 8)

# show selected year, month, and area
show_fibmap(fibdata, yrsel = 2020, mosel = 7, areasel = 'Alafia River')

# Manatee County data
show_fibmap(mancofibdata, yrsel = 2020, mosel = 7, areasel = 'Little Manatee River')
```

show_fibmatmap	<i>Map Fecal Indicator Bacteria matrix results by year</i>
----------------	--

Description

Map Fecal Indicator Bacteria matrix results by year

Usage

```
show_fibmatmap(
  fibdata,
  yrsel,
  areasel,
  lagyr = 3,
  subset_wetdry = c("all", "wet", "dry"),
  precipdata = NULL,
  temporal_window = NULL,
  wet_threshold = NULL,
  addsta = FALSE,
  listout = FALSE,
  warn = TRUE
)
```

Arguments

- `fibdata` input data frame as returned by [read_importfib](#), [read_importentero](#), or [read_importwqp](#), see details
- `yrsel` numeric value indicating the year to map

areasel	vector of bay segment or area names to include, see details
lagyr	numeric for year lag to calculate categories, see details
subset_wetdry	character, subset data frame to only wet or dry samples as defined by wet_threshold and temporal_window? Defaults to "all", which will not subset. If "wet" or "dry" is specified, anlz_fibwetdry is called using the further specified parameters, and the data frame is subsetted accordingly.
precipdata	input data frame as returned by read_importtrain . columns should be: station, date (yyyy-mm-dd), rain (in inches). The object catchprecip has this data from 1995-2023 for select Enterococcus stations. If NULL, defaults to catchprecip .
temporal_window	numeric; required if subset_wetdry is not "all". number of days precipitation should be summed over (1 = day of sample only; 2 = day of sample + day before; etc.)
wet_threshold	numeric; required if subset_wetdry is not "all". inches accumulated through the defined temporal window, above which a sample should be defined as being from a 'wet' time period
addsta	logical to add station labels to the map, default FALSE
listout	logical to return a list of simple feature objects for the data used in the leaflet map, default FALSE
warn	logical to print warnings about stations with insufficient data, default TRUE

Details

Placing the mouse cursor over an item on the map will reveal additional information about a segment or station.

If the input is from EPCHC (from [read_importfib](#), i.e., [fibdata](#)), valid entries for areasel include 'Alafia River', 'Hillsborough River', 'Big Bend', 'Cockroach Bay', 'East Lake Outfall', 'Hillsborough Bay', 'Little Manatee River', 'Lower Tampa Bay', 'McKay Bay', 'Middle Tampa Bay', 'Old Tampa Bay', 'Palm River', 'Tampa Bypass Canal', and 'Valrico Lake'. If the input data is from [read_importentero](#) (i.e., [enterodata](#)), valid entries for areasel include 'OTB', 'HB', 'MTB', 'LTB', 'BCB', and 'MR'. If the input is from [read_importwqp](#) for Manatee County (21FLMANA_WQX, [mancofibdata](#)), Pasco County (21FLPASC_WQX, [pascofibdata](#)), Polk County (21FLPOLK_WQX, [polcofibdata](#)), or Hillsborough County Environmental Services Division (21FLHESD_WQX, [hcesdfibdata](#)) valid entries for areasel are any that are present in the area column for the respective input datasets.

Bay segment matrix categories can be shown if input data are from [read_importentero](#). Stations for these data were chosen specifically as downstream endpoints for each bay segment, whereas the other datasets are not appropriate for estimating bay segment outcomes. Bay segment categories are shown using minor sub-basins. The collection of sub-basins that include stations used to estimate the bay segment outcomes are shown and color-coded by the appropriate outcome for each bay segment.

Care should be taken when communicating results at the bay segment scale to ensure appropriate messaging. Baywide scores could be misleading by using a handful of stations to communicate results at the bay segment scale. Reporting at this scale is meant to support assessments that are comparable to other similar indicators used by TBEP. A low grade for a bay segment does not imply

hazardous conditions, rather the score reflects the aggregate likelihood for all stations selected for that bay segment. Remediation of conditions that contribute to fecal contamination is best handled at the site scale.

Value

A leaflet map for the selected year and area showing station matrix scores if listout = FALSE (default). Bay segment scores are also shown if the input data are not from EPCHC. A list of simple feature objects is returned if listout = TRUE.

See Also

[anlz_fibmatrix](#) for details on the categories

Examples

```
# non-EPCHC, non Manatee County data
show_fibmatmap(enteredata, yrsel = 2020, areasel = 'OTB')

# EPCHC data
show_fibmatmap(fibdata, yrsel = 2016,
               areasel = c("Hillsborough River", "Alafia River"))

# Manatee County data
show_fibmatmap(mancofibdata, yrsel = 2020, areasel = 'Manatee River')
```

show_fibmatrix	<i>Plot a matrix of Fecal Indicator Bacteria categories over time by station or bay segment</i>
----------------	---

Description

Plot a matrix of Fecal Indicator Bacteria categories over time by station or bay segment

Usage

```
show_fibmatrix(
  fibdata,
  yrrng = NULL,
  stas = NULL,
  bay_segment = NULL,
  lagyr = 3,
  subset_wetdry = c("all", "wet", "dry"),
  precipdata = NULL,
  temporal_window = NULL,
  wet_threshold = NULL,
  txtsz = 3,
  asreact = FALSE,
```

```

    nrows = 10,
    family = "sans",
    angle = 90,
    size = 10,
    hjust = 0,
    plotly = FALSE,
    width = NULL,
    height = NULL,
    warn = TRUE
  )

```

Arguments

<code>fibdata</code>	input data frame as returned by read_importfib , read_importentero , or read_importwqp , see details
<code>yrrng</code>	numeric vector indicating min, max years to include, defaults to range of years in data, see details
<code>stas</code>	optional vector of stations to include, see details
<code>bay_segment</code>	optional vector of bay segment names to include, supercedes <code>stas</code> if provided, see details
<code>lagyr</code>	numeric for year lag to calculate categories, see details
<code>subset_wetdry</code>	character, subset data frame to only wet or dry samples as defined by <code>wet_threshold</code> and <code>temporal_window</code> ? Defaults to "all", which will not subset. If "wet" or "dry" is specified, anlz_fibwetdry is called using the further specified parameters, and the data frame is subsetting accordingly.
<code>precipdata</code>	input data frame as returned by read_importtrain . columns should be: station, date (yyyy-mm-dd), rain (in inches). The object catchprecip has this data from 1995-2023 for select Enterococcus stations. If NULL, defaults to catchprecip .
<code>temporal_window</code>	numeric; required if <code>subset_wetdry</code> is not "all". number of days precipitation should be summed over (1 = day of sample only; 2 = day of sample + day before; etc.)
<code>wet_threshold</code>	numeric; required if <code>subset_wetdry</code> is not "all". inches accumulated through the defined temporal window, above which a sample should be defined as being from a 'wet' time period
<code>txtsz</code>	numeric for size of text in the plot, applies only if <code>tab = FALSE</code> . Use <code>txtsz = NULL</code> to suppress.
<code>asreact</code>	logical indicating if a reactable object is returned
<code>nrows</code>	if <code>asreact = TRUE</code> , a numeric specifying number of rows in the table
<code>family</code>	optional chr string indicating font family for text labels
<code>angle</code>	numeric for angle of x-axis text labels
<code>size</code>	numeric for size of the x-axis text labels
<code>hjust</code>	numeric for horizontal justification of x-axis text labels
<code>plotly</code>	logical if matrix is created using <code>plotly</code>

width	numeric for width of the plot in pixels, only applies if plotly = TRUE
height	numeric for height of the plot in pixels, only applies if plotly = TRUE
warn	logical to print warnings about stations with insufficient data, default TRUE

Details

The matrix color codes years and stations based on the likelihood of fecal indicator bacteria concentrations exceeding 410 CFU / 100 mL for *E. coli* (fresh) or 130 CFU / 100 mL for *Enterococcus* (marine). The likelihoods are categorized as A, B, C, D, or E (Microbial Water Quality Assessment or MWQA categories) with corresponding colors, where the breakpoints for each category are <10%, 10-30%, 30-50%, 50-75%, and >75% (right-closed). By default, the results for each year are based on a right-centered window that uses the previous two years and the current year to calculate probabilities from the monthly samples (lagyr = 3). Methods and rationale for this categorization scheme are provided by the Florida Department of Environmental Protection, Figure 8 in the document at <http://publicfiles.dep.state.fl.us/DEAR/BMAP/Tampa/MST%20Report/Fecal%20BMAP%20DST%20Final%20Report%20--%20June%202008.pdf> and Morrison et al. 2009 in the **BASIS 5** proceedings.

Bay segments scores are used instead of stations if bay_segment is not NULL and the input data are from [read_importentero](#). Care should be taken when communicating results at the bay segment scale to ensure appropriate messaging. Baywide scores could be misleading by using a handful of stations to communicate results at the bay segment scale. Reporting at this scale is meant to support assessments that are comparable to other similar indicators used by TBEP. A low grade for a bay segment does not imply hazardous conditions, rather the score reflects the aggregate likelihood for all stations selected for that bay segment. Remediation of conditions that contribute to fecal contamination is best handled at the site scale.

See [anlz_fibmatrix](#) for additional details on the arguments.

Value

A static [ggplot](#) object is returned by default. A [reactable](#) table is returned if asreact = TRUE. An interactive [plotly](#) object is returned if plotly = TRUE.

See Also

[anlz_fibmatrix](#)

Examples

```
show_fibmatrix(fibdata)

# show matrix for only dry samples
show_fibmatrix(enterodata, lagyr = 1, subset_wetdry = "dry",
               temporal_window = 2, wet_threshold = 0.5)
```

show_hmpreport	<i>Show Habitat Master Plan progress report card</i>
----------------	--

Description

Show Habitat Master Plan progress report card

Usage

```
show_hmpreport(
  acres,
  subtacres,
  hmptrgs,
  typ,
  twocol = FALSE,
  strata = c("Subtidal", "Intertidal", "Supratidal"),
  totintertid = TRUE,
  ycollapse = FALSE,
  text = 2.5,
  xang = 25,
  family = "sans",
  width = NULL,
  height = NULL
)
```

Arguments

acres	data.frame for intertidal and supratidal land use and cover of habitat types for each year of data
subtacres	data.frame for subtidal cover of habitat types for each year of data
hmptrgs	data.frame of Habitat Master Plan targets and goals
typ	character string indicating "targets" or "goals"
twocol	logical indicating only two colors are shown if target or goals are met and symbols indicate the likelihood of attaining targets or goals, see details
strata	character string indicating with strata to plot, one to many of "Subtidal", "Intertidal", and "Supratidal"
totintertid	logical indicating if total intertidal (mangroves, salt marsh, salt barrens) is shown
ycollapse	logical indicating if the y-axis is collapsed to year with data, see details
text	numeric indicating text size for proportion of target or goal met for habitat types shown in each cell types or symbol size if twocol = T, use NULL to suppress if twocol = F
xang	numeric for angle of habitat labels on the x-axis (top)
family	optional chr string indicating font family for text labels
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

Details

If twocol = F, colors indicate both if the target/goal is met and the likelihood of attaining the target/goal by 2030/2050. Red indicates the target/goal is not met and will likely not be met by 2030/2050 (trending below target/goal), yellow indicates the target/goal is met although it likely will not be met by 2030/2050 (trending below target/goal), light green indicates the target/goal is not met although it will likely be met by 2030/2050 (trending above target/goal), and green indicates the target/goal is met and will likely be met by 2030/2050 (trending above target/goal). Numbers in each cell show the proportion of the target or goal met at each year where data are available. If twocol = T, the colors indicate if the goal is met (green) or not met (red) and the symbols in each cell indicate if the goal is likely to be met (up arrow) or not (down arrow) by 2030/2050. In both cases, the colors and trends are relative to the 2030 targets or 2050 goals using the typ argument.

The report card provides no information on artificial reefs, living shorelines, and hard bottom habitats. These habitats are not assessed in routine data products from the Southwest Florida Water Management District, although targets and goals are provided in the Habitat Master Plan.

The subtidal data in subtacres and the inter/supratidal data in acres are provided as different datasets by the Southwest Florida Water Management District. The years in each dataset typically do not match and each dataset is collected at approximate 2 to 3 year intervals. By default, year on the y-axis is shown as a continuous variable, where gaps are shown in years when each dataset was unavailable. Use ycollapse = TRUE to remove years without data.

Value

A `ggplot2` object showing overall progress in attaining Habitat Master Plan targets or goals.

Examples

```
# view summarized data for report card, targets
show_hmpreport(acres, subtacres, hmptrgs, typ = "targets")

# view summarized data for report card, goals
show_hmpreport(acres, subtacres, hmptrgs, typ = "goals")

# remove empty years
show_hmpreport(acres, subtacres, hmptrgs, typ = "targets", ycollapse = TRUE)

# select only subtidal
show_hmpreport(acres, subtacres, hmptrgs, typ = "targets", ycollapse = TRUE, strata = 'Subtidal')

# show as symbols and colors
show_hmpreport(acres, subtacres, hmptrgs, typ = "targets", ycollapse = TRUE, twocol = TRUE)
```

show_matrix	Create a colorized table for indicator reporting
-------------	--

Description

Create a colorized table for indicator reporting

Usage

```
show_matrix(
  epcdata,
  txtsz = 3,
  trgs = NULL,
  yrrng = NULL,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  asreact = FALSE,
  nrows = 10,
  abbrev = FALSE,
  family = "sans",
  historic = TRUE,
  plotly = FALSE,
  partialyr = FALSE,
  width = NULL,
  height = NULL
)
```

Arguments

epcdata	data frame of epc data returned by read_importwq
txtsz	numeric for size of text in the plot, applies only if tab = FALSE
trgs	optional data.frame for annual bay segment water quality targets, defaults to targets
yrrng	numeric vector indicating min, max years to include, defaults to range of years in epcdata
bay_segment	chr string for bay segments to include, one to all of "OTB", "HB", "MTB", "LTB"
asreact	logical indicating if a reactable object is returned
nrows	if asreact = TRUE, a numeric specifying number of rows in the table
abbrev	logical indicating if text labels in the plot are abbreviated as the first letter
family	optional chr string indicating font family for text labels
historic	logical if historic data are used from 2005 and earlier
plotly	logical if matrix is created using plotly
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

Value

A static [ggplot](#) object is returned if asreact = FALSE, otherwise a [reactable](#) table is returned

See Also

[show_wqmatrix](#), [show_segmatrix](#)

Examples

```
show_matrix(epcdata)
```

show_matrixplotly	<i>Creates a plotly matrix from any matrix function input</i>
-------------------	---

Description

Creates a plotly matrix from any matrix function input

Usage

```
show_matrixplotly(  
  mat,  
  family = "sans",  
  tooltip = "Result",  
  width = NULL,  
  height = NULL  
)
```

Arguments

mat	input matrix as output from show_matrix , show_segmatrix , show_wqmatrix , or show_tbnimatrix
family	optional chr string indicating font family for text labels
tooltip	chr string indicating the column name for tooltip
width	numeric for width of the plot in pixels
height	numeric for height of the plot in pixels

Value

A [plotly](#) data object

Examples

```
mat <- show_wqmatrix(epcdata)  
show_matrixplotly(mat)
```

show_ratab	Create a bay segment assessment table for the 2022-2026 reasonable assurance period
------------	---

Description

Create a bay segment assessment table for the 2022-2026 reasonable assurance period

Usage

```
show_ratab(
  epcdata,
  yrsel,
  bay_segment = c("OTB", "HB", "MTB", "LTB", "RALTB"),
  partialyr = F,
  outtxt1 = NULL,
  outtxt2 = NULL,
  outtxt3 = NULL,
  outtxt45 = NULL,
  txtsz = 13,
  width = NULL
)
```

Arguments

epcdata	data frame of epc data returned by read_importtwq
yrsel	numeric indicating chosen year
bay_segment	chr string for the bay segment, one of "OTB", "HB", "MTB", "LTB", "RALTB"
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter
outtxt1	optional text for NMC action 1, added to the outcome column
outtxt2	optional text for NMC action 2, added to the outcome column
outtxt3	optional text for NMC action 3, added to the outcome column
outtxt45	optional text for NMC actions 4 and 5, added to the outcome column
txtsz	numeric indicating font size
width	optional numeric value indicating width in inches

Details

Choosing bay_segment = 'RALTB' will not work with [epcdata](#) and additional data are needed to use this option.

Value

A [flextable](#) object showing the reasonable assurance compliance of the bay segment for the selected year within the five-year period.

Examples

```
show_ratab(epcdata, yrse1 = 2025, bay_segment = 'OTB')
```

show_reactable	Create reactable table from matrix data
----------------	---

Description

Create reactable table from matrix data

Usage

```
show_reactable(totab, colfun, nrows = 10, txtsz = 3)
```

Arguments

totab	A data frame in wide format of summarized results
colfun	Function specifying how colors are treated in cell background
nrows	numeric specifying number of rows in the table
txtsz	numeric indicating text size in the cells, use txtsz = NULL to suppress

Details

This function is used internally within [show_matrix](#) and [show_wqmatrix](#)

Value

A [reactable](#) table

Examples

```
data(targets)
data(epcdata)

library(tidyr)
library(dplyr)

# data
totab <- anlz_avedat(epcdata) %>%
  .$ann %>%
  filter(var %in% 'mean_chla') %>%
  left_join(targets, by = 'bay_segment') %>%
  select(bay_segment, yr, val, chla_thresh) %>%
  mutate(
    bay_segment = factor(bay_segment, levels = c('OTB', 'HB', 'MTB', 'LTB')),
    outcome = case_when(
      val < chla_thresh ~ 'green',
      val >= chla_thresh ~ 'red'
    )
  )
```

```

    )
  ) %>%
  select(bay_segment, yr, outcome) %>%
  spread(bay_segment, outcome)

# color function
colfun <- function(x){

  out <- case_when(
    x == 'red' ~ '#FF3333',
    x == 'green' ~ '#33FF3B'
  )

  return(out)
}

show_reactable(totab, colfun)

```

show_seagrasscoverage *Create a barplot of seagrass coverage over time in Tampa Bay*

Description

Create a barplot of seagrass coverage over time in Tampa Bay

Usage

```

show_seagrasscoverage(
  seagrass,
  maxyr = 2024,
  family = "sans",
  extend = F,
  lastlab = T,
  covlab = T,
  axsbrk = c(0.08, 0.1)
)

```

Arguments

seagrass	input data.frame included with the package as seagrass
maxyr	numeric for maximum year to plot
family	optional chr string indicating font family for text labels
extend	logical indicating if x-axis should be extended to maximum year in seagrass even if maxyr is less than that
lastlab	logical indicating if text label on maxyr should be included
covlab	logical indicating if text label for coverage goal should be included
axsbrk	numeric vector of length two indicating where the x-axis break occurs

Details

This function creates the flagship seagrass coverage graphic to report on coverage changes over time. All data were pre-processed and included in the package as the [seagrass](#) dataset. Original data are from the Southwest Florida Water Management District and available online at <https://data-swfwmd.opendata.arcgis.com/>. This function and the data used to create the plot are distinct from those used for the transect monitoring program.

Value

A grid object showing acres of total seagrass coverage in Tampa Bay over time.

Examples

```
show_seagrasscoverage(seagrass)
```

<code>show_sedimentalratio</code>	<i>Plot metal concentrations against aluminum</i>
-----------------------------------	---

Description

Plot metal concentrations against aluminum

Usage

```
show_sedimentalratio(  
  sedimentdata,  
  param,  
  yrrng = c(1993, 2024),  
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),  
  funding_proj = "TBEP",  
  lnsz = 1,  
  base_size = 12,  
  plotly = FALSE,  
  family = "sans",  
  width = NULL,  
  height = NULL  
)
```

Arguments

<code>sedimentdata</code>	input sediment data.frame as returned by read_importsediment
<code>param</code>	chr string for which parameter to plot, must be a metal
<code>yrrng</code>	numeric vector indicating min, max years to include, use single year for one year of data
<code>bay_segment</code>	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"

funding_proj	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"
lnsz	numeric for line size
base_size	numeric indicating text scaling size for plot
plotly	logical if matrix is created using plotly
family	optional chr string indicating font family for text labels
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

Details

The plot shows the selected contaminant concentration relative to aluminum, the latter being present as a common metal in the Earth’s crust. An elevated ratio of a metal parameter relative to aluminum suggests it is higher than background concentrations.

Lines for the Threshold Effect Level (TEL) and Potential Effect Level (PEL) are shown for the parameter, if available.

Value

A [ggplot](#) object or a [plotly](#) object if plotly = TRUE showing the ratio of the selected parameter plotted against aluminum concentrations collected at the same site. Black lines show the linear fit of a log-log model and the 95% prediction intervals.

References

Schropp, S. J., Graham Lewis, F., Windom, H. L., Ryan, J. D., Calder, F. D., & Burney, L. C. 1990. Interpretation of metal concentrations in estuarine sediments of Florida using aluminum as a reference element. Estuaries. 13:227-235.

Examples

```
show_sedimentalratio(sedimentdata, param = 'Arsenic')
```

show_sedimentave	<i>Plot sediment concentration averages by bay segment</i>
------------------	--

Description

Plot sediment concentration averages by bay segment

Usage

```
show_sedimentave(
  sedimentdata,
  param,
  yrrng = c(1993, 2024),
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),
  funding_proj = "TBEP",
  lnsz = 1,
  base_size = 12,
  plotly = FALSE,
  family = "sans",
  width = NULL,
  height = NULL
)
```

Arguments

sedimentdata	input sediment data.frame as returned by read_importsediment
param	chr string for which parameter to plot
yrrng	numeric vector indicating min, max years to include, use single year for one year of data
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"
funding_proj	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"
lnsz	numeric for line size
base_size	numeric indicating text scaling size for plot
plotly	logical if matrix is created using plotly
family	optional chr string indicating font family for text labels
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

Details

Lines for the Threshold Effect Level (TEL) and Potential Effect Level (PEL) are shown for the parameter, if available. Confidence intervals may not be shown for segments with insufficient data.

Value

A [ggplot](#) object or a [plotly](#) object if plotly = TRUE showing sediment averages and 95% confidence intervals of the selected parameter concentrations for each bay segment

Examples

```
show_sedimentave(sedimentdata, param = 'Arsenic')
```

show_sedimentmap	<i>Make a map for sediment contaminants at stations in Tampa Bay</i>
------------------	--

Description

Make a map for sediment contaminants at stations in Tampa Bay

Usage

```
show_sedimentmap(  
  sedimentdata,  
  param,  
  yrrng = c(1993, 2024),  
  funding_proj = "TBEP",  
  weight = 1.5  
)
```

Arguments

sedimentdata	input sediment data.frame as returned by read_importsediment
param	chr string for which parameter to plot
yrrng	numeric vector indicating min, max years to include, use single year for one year of data
funding_proj	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"
weight	numeric for outline width of station points on the map

Details

The map shows sediment contaminant concentrations for the selected parameter relative to Threshold Effects Levels (TEL) and Potential Effects Levels (PEL), if available. Green points show concentrations below the TEL, yellow points show concentrations between the TEL and PEL, and red points show concentrations above the PEL. The applicable TEL and PEL values for the parameter are indicated in the legend. If TEL and PEL thresholds are not available, a map of the sediment concentrations is shown and a warning is returned to the console.

Value

A [leaflet](#) object

Examples

```
show_sedimentmap(sedimentdata, param = 'Arsenic')
```

show_sedimentpelave	<i>Plot summary of PEL averages by bay segment</i>
---------------------	--

Description

Plot summary of PEL averages by bay segment

Usage

```
show_sedimentpelave(  
  sedimentdata,  
  yrrng = c(1993, 2024),  
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),  
  funding_proj = "TBEP",  
  alph = 1,  
  ylim = c(0, 0.4),  
  lnsz = 1,  
  base_size = 12,  
  plotly = FALSE,  
  family = "sans",  
  width = NULL,  
  height = NULL  
)
```

Arguments

sedimentdata	input sediment data.frame as returned by read_importsediment
yrrng	numeric vector indicating min, max years to include, use single year for one year of data
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"
funding_proj	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"
alph	numeric indicating alpha value for score category colors
ylim	numeric for y axis limits
lnsz	numeric for line size
base_size	numeric indicating text scaling size for plot
plotly	logical if matrix is created using plotly
family	optional chr string indicating font family for text labels
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

Details

Lines for the grades are shown on the plot. Confidence intervals may not be shown for segments with insufficient data.

Value

A [ggplot](#) object or a [plotly](#) object if `plotly = TRUE` showing PEL averages (average of the averages) and 95% confidence intervals for each bay segment

Examples

```
show_sedimentpelave(sedimentdata)
```

```
show_sedimentpelaveplotly
```

Creates a plotly object for average PEL plots

Description

Creates a plotly object for average PEL plots

Usage

```
show_sedimentpelaveplotly(p, width = NULL, height = NULL)
```

Arguments

<code>p</code>	ggplot object as output from show_sedimentpelave
<code>width</code>	numeric for width of the plot in pixels
<code>height</code>	numeric for height of the plot in pixels

Value

A [plotly](#) data object

Examples

```
p <- show_sedimentpelave(sedimentdata)
show_sedimentpelaveplotly(p)
```

show_sedimentpelmap	<i>Make a map for sediment PEL ratios at stations in Tampa Bay</i>
---------------------	--

Description

Make a map for sediment PEL ratios at stations in Tampa Bay

Usage

```
show_sedimentpelmap(
  sedimentdata,
  yrrng = c(1993, 2024),
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),
  funding_proj = "TBEP",
  weight = 1.5
)
```

Arguments

sedimentdata	input sediment data.frame as returned by read_importsediment
yrrng	numeric vector indicating min, max years to include, use single year for one year of data
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"
funding_proj	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"
weight	numeric for outline width of station points on the map

Details

The map shows average PEL ratios graded from A to F for benthic stations monitored in Tampa Bay. The PEL is a measure of how likely a contaminant is to have a toxic effect on invertebrates that inhabit the sediment. The PEL ratio is the contaminant concentration divided by the Potential Effects Levels (PEL) that applies to a contaminant, if available. Higher ratios and lower grades indicate sediment conditions that are likely unfavorable for invertebrates. The station average combines the PEL ratios across all contaminants measured at a station.

Value

A [leaflet](#) object

Examples

```
show_sedimentpelmap(sedimentdata)
```

show_segmatrix	<i>Create a colored table for water quality outcomes and exceedances by segment</i>
----------------	---

Description

Create a colored table for water quality outcomes by segment that includes the management action and chlorophyll, and light attenuation exceedances

Usage

```
show_segmatrix(
  epccdata,
  txtsz = 3,
  trgs = NULL,
  yrrng = c(1975, 2025),
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  abbrev = FALSE,
  family = "sans",
  historic = TRUE,
  plotly = FALSE,
  partialyr = FALSE,
  width = NULL,
  height = NULL
)
```

Arguments

epccdata	data frame of epc data returned by read_importtwq
txtsz	numeric for size of text in the plot, applies only if tab = FALSE
trgs	optional data.frame for annual bay segment water quality targets, defaults to targets
yrrng	numeric vector indicating min, max years to include
bay_segment	chr string for bay segments to include, only one of "OTB", "HB", "MTB", "LTB"
abbrev	logical indicating if text labels in the plot are abbreviated as the first letter
family	optional chr string indicating font family for text labels
historic	logical if historic data are used from 2005 and earlier
plotly	logical if matrix is created using plotly
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter
width	numeric for width of the plot in pixels, only applies if plotly = TRUE
height	numeric for height of the plot in pixels, only applies if plotly = TRUE

Details

This function provides a combined output for the [show_wqmatrix](#) and [show_matrix](#) functions. Only one bay segment can be plotted for each function call.

Value

A static [ggplot](#) object is returned

See Also

[show_wqmatrix](#), [show_matrix](#)

Examples

```
show_segmatrix(epcdata, bay_segment = 'OTB')
```

show_segplotly	<i>Plot chlorophyll and secchi data together with matrix outcomes</i>
----------------	---

Description

Plot chlorophyll and secchi data together with matrix outcomes

Usage

```
show_segplotly(  
  epcdata,  
  bay_segment = c("OTB", "HB", "MTB", "LTB"),  
  yrrng = c(1975, 2025),  
  family = NULL,  
  partialyr = FALSE,  
  width = NULL,  
  height = NULL  
)
```

Arguments

epcdata	data frame of epc data returned by read_importwq
bay_segment	chr string for the bay segment, one of "OTB", "HB", "MTB", "LTB"
yrrng	numeric for year range to plot
family	optional chr string indicating font family for text labels
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter
width	numeric for width of the plot in pixels
height	numeric for height of the plot in pixels

Details

This function combines outputs from [show_thrplot](#) and [show_segmatrix](#) for a selected bay segment. The plot is interactive and can be zoomed by dragging the mouse pointer over a section of the plot. Information about each cell or value can be seen by hovering over a location in the plot.

Value

An interactive plotly object

Examples

```
show_segplotly(epcdata)
```

show_sitemap	<i>Map site attainment categories for a selected year</i>
--------------	---

Description

Map site attainment categories for a selected year

Usage

```
show_sitemap(  
  epcdata,  
  yrsel,  
  mosel = c(1, 12),  
  param = c("chla", "la"),  
  trgs = NULL,  
  thrs = FALSE,  
  showseg = TRUE,  
  partialyr = FALSE  
)
```

Arguments

epcdata	data frame of epc data returned by read_importwq
yrsel	numeric for year to plot
mosel	optional numeric of length one or two for mapping results for a specific month or month range in a given year, default full year
param	chr string for which parameter to plot, one of "chla" for chlorophyll or "la" for light attenuation
trgs	optional data.frame for annual bay segment water quality targets, defaults to targets , only applies if mosel = c(1, 12)
thrs	logical indicating if attainment category is relative to targets (default) or thresholds, passed to anlz_attainsite , only applies if mosel = c(1, 12)

showseg	logical indicating if bay segment labels are included
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter, only applies if mose1 = c(1, 12)

Value

A static ggplot object is returned

Examples

```
show_sitemap(epcdata, yrsel = 2025)
```

show_siteseegmap	<i>Map site and bay segment attainment categories for a selected year</i>
------------------	---

Description

Map site and bay segment attainment categories for a selected year

Usage

```
show_siteseegmap(
  epcdata,
  yrsel,
  param = c("chla", "la"),
  trgs = NULL,
  thrs = FALSE,
  partialyr = FALSE,
  showseg = TRUE,
  base_size = 12,
  family = "sans"
)
```

Arguments

epcdata	data frame of epc data returned by read_importwq
yrsel	numeric for year to plot
param	chr string for which parameter to plot, one of "chla" for chlorophyll or "la" for light attenuation
trgs	optional data.frame for annual bay segment water quality targets, defaults to targets
thrs	logical indicating if attainment category is relative to targets (default) or thresholds, passed to anlz_attainsite
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter

showseg	logical indicating of bay segment labels are included
base_size	numeric indicating text scaling size for plot
family	optional chr string indicating font family for text labels

Details

The map is similar to that returned by [show_sitemap](#) with the addition of polygons for each bay segment colored by the annual attainment category and the site points are sized relative to the selected parameter in param.

Value

A static ggplot object is returned

Examples

```
show_sitesegmap(epcdata, yrse1 = 2025)
```

show_splitbarplot	<i>Show a Split Bar Plot with Error Bars</i>
-------------------	--

Description

This function creates a bar plot with error bars based on two groups split by a year value. It can optionally create an interactive plot and show individual data points.

Usage

```
show_splitbarplot(
  df,
  period_col,
  year_col,
  value_col,
  x_label_prefix = "",
  interactive = TRUE,
  source = "A",
  exploded = FALSE,
  bars_fill = c("#00BFC4", "#F8766D"),
  bars_alpha = 0.7,
  label_points = c("min", "max", "median"),
  label_color = "black",
  value_round = 2,
  text_size = 14
)
```

Arguments

<code>df</code>	A data frame containing the data to plot
<code>period_col</code>	Name of the column containing values "before" and "after" as an ordered factor
<code>year_col</code>	Name of the column containing year values
<code>value_col</code>	Name of the column containing the values to plot
<code>x_label_prefix</code>	Optional prefix for x-axis labels (default: "")
<code>interactive</code>	Logical; if TRUE, creates an interactive plot using plotly (default: TRUE)
<code>source</code>	if interactive is TRUE, the source argument to <code>plotly::ggplotly</code> for referencing with click events (default: "A")
<code>exploded</code>	Logical; if TRUE, adds individual points with labels for min, max, and mean (default: FALSE)
<code>bars_fill</code>	Fill color for bars (default: <code>c("#00BFC4", "#F8766D")</code>); teal and red)
<code>bars_alpha</code>	Alpha transparency for bars (default: 0.7)
<code>label_points</code>	Character vector specifying which points to label in exploded view (default: <code>c("min", "max", "median")</code>)
<code>label_color</code>	Color for labeled points in exploded view (default: "black")
<code>value_round</code>	Integer indicating number of decimal places for rounding values (default: 2)
<code>text_size</code>	Size of text for labels and hover text (default: 14)

Value

A `ggplot2` object or `plotly` object (if `interactive = TRUE`)

Examples

```
# Create example data
data <- data.frame(
  date = seq.Date(as.Date("2010-01-01"), as.Date("2020-12-31"), by = "month"),
  value = c(rnorm(66, mean = 10, sd = 4), rnorm(66, mean = 20, sd = 4))
)

# Basic analysis with default statistics
split_date <- as.Date("2015-06-15")
df <- anlz_splitdata(data, split_date, "date", "value")

# Basic interactive plot
show_splitbarplot(df, "period", "year", "avg")

# Custom colors and labels with 1 decimal place
show_splitbarplot(df, "period", "year", "avg",
  bars_fill = c("steelblue", "darkorange"),
  exploded = TRUE,
  label_points = c("min", "max"),
  label_color = "darkred",
  value_round = 1)
```

show_tbbimatrix	<i>Plot a matrix of Tampa Bay Benthic Index scores over time by bay segment</i>
-----------------	---

Description

Plot a matrix of Tampa Bay Benthic Index scores over time by bay segment

Usage

```
show_tbbimatrix(
  tbbisr,
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB", "All", "All (wt)"),
  yrrng = c(1993, 2024),
  window = TRUE,
  alph = 1,
  txtsz = 3,
  family = "sans",
  rev = FALSE,
  position = "top",
  plotly = FALSE,
  width = NULL,
  height = NULL
)
```

Arguments

tbbisr	input data frame as returned by anlz_tbbisr
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB", "All", "All (wt)"
yrrng	numeric indicating year ranges to evaluate
window	logical indicating whether to use a rolling 5-year window (default TRUE) or single year values (FALSE) for the bay segment categories, see details
alph	numeric indicating alpha value for score category colors
txtsz	numeric for size of text in the plot
family	optional chr string indicating font family for text labels
rev	logical if factor levels for bay segments are reversed
position	chr string of location for bay segment labels, default on top, passed to scale_x_discrete
plotly	logical if matrix is created using plotly
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

Details

Additional summaries are provided for the entire bay, as a summary across categories ("All") and a summary weighted across the relative sizes of each bay segment ("All (wt)").

The default behavior is to use a rolling five-year window to calculate the percent of sites in each TBBI category by bay segment. This applies only to years 2005 and later, where the counts from the current year and the prior four years are summed to calculate the percentages. This is intended to help smooth out inter-annual variability due to reduced sampling effort from 2005 to present. If `window = FALSE`, then only single year values are used.

Value

A `ggplot` object showing trends over time in TBBI scores for each bay segment if `plotly = FALSE`, otherwise a `plotly` object

Examples

```
tbbisr <- anlz_tbbisr(benthicdata)
show_tbbimatrix(tbbisr)
```

<code>show_tbnimatrix</code>	<i>Plot a matrix of Tampa Bay Nekton Index scores over time by bay segment</i>
------------------------------	--

Description

Plot a matrix of Tampa Bay Nekton Index scores over time by bay segment

Usage

```
show_tbnimatrix(
  tbnisr,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  perc = c(32, 46),
  alph = 1,
  txtsz = 3,
  family = "sans",
  rev = FALSE,
  position = "top",
  plotly = FALSE,
  width = NULL,
  height = NULL
)
```

Arguments

tbniscr	input data frame as returned by anlz_tbniscr
bay_segment	chr string for the bay segment, one to many of "OTB", "HB", "MTB", "LTB"
perc	numeric values indicating break points for score categories
alph	numeric indicating alpha value for score category colors
txtsz	numeric for size of text in the plot
family	optional chr string indicating font family for text labels
rev	logical if factor levels for bay segments are reversed
position	chr string of location for bay segment labels, default on top, passed to scale_x_discrete
plotly	logical if matrix is created using plotly
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

Value

A [ggplot](#) object showing trends over time in TBNI scores for each bay segment

Examples

```
tbniscr <- anlz_tbniscr(fimdata)
show_tbnimatrix(tbniscr)
```

show_tbniscr

Plot Tampa Bay Nekton Index scores over time by bay segment

Description

Plot Tampa Bay Nekton Index scores over time by bay segment

Usage

```
show_tbniscr(
  tbniscr,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  perc = c(32, 46),
  alph = 1,
  ylim = c(22, 58),
  rev = FALSE,
  plotly = FALSE,
  family = "sans",
  width = NULL,
  height = NULL
)
```

Arguments

tbniscr	input dat frame as returned by anlz_tbniscr
bay_segment	chr string for the bay segment, one to many of "OTB", "HB", "MTB", "LTB"
perc	numeric values indicating break points for score categories
alph	numeric indicating alpha value for score category colors
ylim	numeric for y axis limits
rev	logical if factor levels for bay segments are reversed
plotly	logical if matrix is created using plotly
family	optional chr string indicating font family for text labels
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

Value

A [ggplot](#) object showing trends over time in TBNi scores for each bay segment or a [plotly](#) object if plotly = TRUE

Examples

```
tbniscr <- anlz_tbniscr(fimdata)
show_tbniscr(tbniscr)
```

show_tbniscrall	<i>Plot Tampa Bay Nekton Index scores over time as average across bay segments</i>
-----------------	--

Description

Plot Tampa Bay Nekton Index scores over time as average across bay segments

Usage

```
show_tbniscrall(
  tbniscr,
  perc = c(32, 46),
  alph = 1,
  ylim = c(22, 58),
  rev = FALSE,
  plotly = FALSE
)
```

Arguments

tbniscr	input dat frame as returned by anlz_tbniscr
perc	numeric values indicating break points for score categories
alph	numeric indicating alpha value for score category colors
ylim	numeric for y axis limits
rev	logical if factor levels for bay segments are reversed
plotly	logical if matrix is created using plotly

Value

A [ggplot](#) object showing trends over time in TBNI scores for each bay segment or a [plotly](#) object if plotly = TRUE

Examples

```
tbniscr <- anlz_tbniscr(fimdata)
show_tbniscrall(tbniscr)
```

show_tbniscrplotly	<i>Creates a plotly object for TBNI score plots</i>
--------------------	---

Description

Creates a plotly object for TBNI score plots

Usage

```
show_tbniscrplotly(p, width = NULL, height = NULL)
```

Arguments

p	ggplot object as output from show_tbniscr , show_tbniscrall , or show_tbniscrseas (with metric = NULL)
width	numeric for width of the plot in pixels
height	numeric for height of the plot in pixels

Value

A [plotly](#) data object

Examples

```
tbniscr <- anlz_tbniscr(fimdata)
p <- show_tbniscrall(tbniscr)
show_tbniscrplotly(p)
```

show_tbniscrseas	<i>Plot seasonal Tampa Bay Nekton Index results by month and bay segment</i>
------------------	--

Description

Plot seasonal Tampa Bay Nekton Index results by month and bay segment

Usage

```
show_tbniscrseas(
  tbniscr,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  yr,
  metric = NULL,
  perc = c(32, 46),
  alph = 1,
  plotly = FALSE,
  family = "sans",
  width = NULL,
  height = NULL
)
```

Arguments

tbniscr	input dat frame as returned by anlz_tbniscr
bay_segment	chr string for the bay segment to plot, one of "OTB", "HB", "MTB", "LTB"
yr	numeric for the year to plot
metric	chr string for the column in tbniscr to plot on the y-axis, defaults to NULL to plot "TBNI_Score", otherwise an individual TBNI metric can be specified, one of "TBNI_Score", "NumTaxa", "BenthicTaxa", "TaxaSelect", "NumGuilds", or "Shannon"
perc	numeric values indicating break points for score categories, only used if metric is NULL or "TBNI_Score"
alph	numeric indicating alpha value for the score category background colors and the boxplot fill, only used if metric is NULL or "TBNI_Score"
plotly	logical if matrix is created using plotly
family	optional chr string indicating font family for text labels
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

Details

Boxplots show the distribution of site-level results by month for yr in bay_segment, with individual site values overlaid as jittered points. If metric is NULL or "TBNI_Score", the TBNI score is plotted with the same red/yellow/green score category background and break lines from perc as in [show_tbniscr](#). If a different metric is specified, it is plotted as raw values with no background (note that scored metrics cannot be shown). Metric options include "TBNI_Score" or NULL (default), "NumTaxa", "BenthicTaxa", "TaxaSelect", "NumGuilds", and "Shannon". The y-axis title includes yr, bay_segment, and the plotted metric.

Value

A [ggplot](#) object showing monthly results for yr in bay_segment for the selected metric, or a [plotly](#) object if plotly = TRUE

Examples

```
tbniscr <- anlz_tbniscr(fimdata)
show_tbniscrseas(tbniscr, bay_segment = 'OTB', yr = 2018)
```

show_tdlcrk

Make a map for tidal creek report card

Description

Make a map for tidal creek report card

Usage

```
show_tdlcrk(dat, weight = 1.5)
```

Arguments

dat	input creek score data returned from anlz_tdlcrk
weight	numeric for weight of polylines, passed to addPolylines

Value

A [leaflet](#) object

Examples

```
dat <- anlz_tdlcrk(tidalcreeks, iwrraw, yr = 2025)
show_tdlcrk(dat)
```

show_tdlcrkindic	<i>Plotly barplots of tidal creek context indicators</i>
------------------	--

Description

Plotly barplots of tidal creek context indicators

Usage

```
show_tdlcrkindic(  
  id,  
  cntdat,  
  yr = 2025,  
  thrsl = FALSE,  
  pal = c("#5C4A42", "#427355", "#004F7E")  
)
```

Arguments

id	numeric indicating the id number of the tidal creek to plot
cntdat	output from anlz_tdlcrkindic
yr	numeric indicating reference year
thrsl	logical if threshold lines and annotations are shown on the plots
pal	vector of colors for the palette

Value

A plotly object

Examples

```
cntdat <- anlz_tdlcrkindic(tidalcreeks, iwrraw, yr = 2025)  
  
set.seed(123)  
id <- sample(unique(cntdat$id), 1)  
show_tdlcrkindic(id, cntdat, thrsl = TRUE)
```

show_tdlcrkindiccdf *Plotly empirical CDF plots of tidal creek context indicators*

Description

Plotly empirical CDF plots of tidal creek context indicators

Usage

```
show_tdlcrkindiccdf(  
  id,  
  cntdat,  
  yr = 2025,  
  thrse1 = FALSE,  
  pal = c("#5C4A42", "#427355", "#004F7E")  
)
```

Arguments

id	numeric indicating the id number of the tidal creek to plot
cntdat	output from anlz_tdlcrkindic
yr	numeric indicating reference year
thrse1	logical if threshold lines and annotations are shown on the plots
pal	vector of colors for the palette

Details

This function returns several empirical cumulative distribution plots for the tidal creek context indicators. Points on the plot indicate the observed values and percentiles for the creek specified by id. The percentiles and CDF values are defined by the "population" of creeks in cntdat. Points in the plots are color-coded by sample year to evaluate temporal trends, if any.

Value

A plotly object

Examples

```
cntdat <- anlz_tdlcrkindic(tidalcreeks, iwrraw, yr = 2025)  
  
set.seed(123)  
id <- sample(unique(cntdat$id), 1)  
show_tdlcrkindiccdf(id, cntdat, thrse1 = TRUE)
```

show_tdlcrkline	<i>Add a line or annotation to a plotly graph</i>
-----------------	---

Description

Add a line or annotation to a plotly graph for the tidal creek indicators

Usage

```
show_tdlcrkline(  
  varin = c("CHLAC", "TN", "chla_tn_ratio", "DO", "tsi", "no23_ratio"),  
  thrse1 = FALSE,  
  horiz = TRUE,  
  annotate = FALSE  
)
```

Arguments

varin	chr string for the indicator
thrse1	logical if something is returned, otherwise NULL, this is a hack for working with the plotly output
horiz	logical indicating if output is horizontal or vertical
annotate	logical indicating if output is line or annotation text

Details

This function is used internally within [show_tdlcrkindic](#) and [show_tdlcrkindiccdf](#)

Value

A list object passed to the layout argument of plotly, either shapes or annotate depending on user input

Examples

```
# code for vertical line output, chlorophyll  
show_tdlcrkline('CHLAC', thrse1 = TRUE)
```

show_tdlcrkmatrix	<i>Plot the tidal creek report card matrix</i>
-------------------	--

Description

Plot the tidal creek report card matrix

Usage

```
show_tdlcrkmatrix(  
  dat,  
  class = c("3M", "2"),  
  score = c("Prioritize", "Investigate", "Caution", "Monitor"),  
  family = "sans",  
  size = 11  
)
```

Arguments

dat	input creek score data returned from anlz_tdlcrk
class	character vector indicating which creek classes to show, one to many of '3M', '2', '3F', and '1'. Defaults to marine only ('3M', '2').
score	character vector of score categories to include, one to many of 'Prioritize', 'Investigate', 'Caution', and 'Monitor'. Defaults to all.
family	optional chr string indicating font family for text labels
size	numeric for text and line scaling

Details

The plot shows a matrix with rows for individual creeks and columns for overall creek score. The columns show an overall creek score and the number of years in the prior ten years that nitrogen values at a creek were assigned to each of the four score categories. Number of years is mapped to cell transparency.

Value

A static [ggplot](#) object is returned.

Examples

```
dat <- anlz_tdlcrk(tidalcreeks, iwrraw, yr = 2025)  
show_tdlcrkmatrix(dat)
```

show_tdlcrkradar	<i>Radar plots for tidal creek indicators</i>
------------------	---

Description

Radar plots for tidal creek indicators

Usage

```
show_tdlcrkradar(
  id,
  cntdat,
  col = "#338080E6",
  ptsz = 1,
  lbsz = 0.8,
  valsz = 1,
  brdwd = 5
)
```

Arguments

id	numeric indicating the id number of the tidal creek to plot
cntdat	output from anlz_tdlcrkindic
col	color input for polygon and line portions
ptsz	numeric size of points
lbsz	numeric for size of text labels
valsz	numeric for size of numeric value labels
brdwd	numeric for polygon border width

Details

See details in [anlz_tdlcrkindic](#) for an explanation of the indicators

Internal code borrowed heavily from the radarchart function in the fmsb package.

Value

A radar plot

Examples

```
cntdat <- anlz_tdlcrkindic(tidalcreeks, iwrraw, yr = 2025, radar = TRUE)

set.seed(123)
id <- sample(unique(cntdat$id), 1)
show_tdlcrkradar(id, cntdat)
```

show_thrplot	<i>Plot annual water quality values, targets, and thresholds for a segment</i>
--------------	--

Description

Plot annual water quality values, targets, and thresholds for a bay segment

Usage

```
show_thrplot(
  epcdata,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  thr = c("chl", "la"),
  trgs = NULL,
  yrrng = c(1975, 2025),
  family = "sans",
  labelexp = TRUE,
  txtlab = TRUE,
  thrs = FALSE,
  partialyr = FALSE
)
```

Arguments

epcdata	data frame of epc data returned by read_importwq
bay_segment	chr string for the bay segment, one of "OTB", "HB", "MTB", "LTB"
thr	chr string indicating which water quality value and appropriate target/threshold to plot, one of "chl" for chlorophyll and "la" for light availability
trgs	optional data.frame for annual bay segment water quality targets/thresholds, defaults to targets
yrrng	numeric vector indicating min, max years to include
family	optional chr string indicating font family for text labels
labelexp	logical indicating if y axis and target labels are plotted as expressions, default TRUE
txtlab	logical indicating if a text label for the target value is shown in the plot
thrs	logical indicating if reference lines are shown only for the regulatory threshold
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter

Value

A [ggplot](#) object

Examples

```
show_thrplot(epcdata, bay_segment = 'OTB', thr = 'chl')
```

show_transect

*Plot results for a seagrass transect by time and location***Description**

Plot results for a seagrass transect by time and location

Usage

```
show_transect(
  transect,
  site,
  species = c("Halodule", "Syringodium", "Thalassia", "Halophila", "Ruppia", "Caulerpa",
    "Dapis", "Chaetomorpha"),
  yrrng = c(1998, 2025),
  varplo = c("Abundance", "Blade Length", "Short Shoot Density"),
  base_size = 12,
  facet = FALSE,
  ncol = NULL,
  plotly = FALSE,
  width = NULL,
  height = NULL,
  sppcol = NULL
)
```

Arguments

transect	data frame returned by read_transect
site	chr string indicating site results to plot
species	chr string indicating one to many of which species to plot
yrrng	numeric vector of length two indicating min, max years to include, use the same year for both values to evaluate a single year
varplo	chr string indicating which variable to plot
base_size	numeric indicating text scaling size for plot
facet	logical indicating if plots are separated into facets by species
ncol	numeric indicating number of columns if facet = TRUE
plotly	logical if plot is created using plotly
width	numeric for width of the plot in pixels, only applies if plotly = TRUE
height	numeric for height of the plot in pixels, only applies if plotly = TRUE
sppcol	character vector of alternative colors to use for each species, must have length of six

Details

All sites along a transect that were surveyed are shown in the plot, including those where the selected species were not found. The latter is colored in grey hollow points. Species options include Halodule, Syringodium, Thalassia, Halophila, Ruppia, Caulerpa (attached macroalgae), Dapis (cyanobacteria), and/or Chaetomorpha (drift green algae). Drift or attached macroalgae and cyanobacteria (Dapis) estimates may not be accurate prior to 2021.

Note that if `plotly = TRUE`, the size legend is not shown.

Value

A `ggplot` object

Examples

```
## Not run:
transect <- read_transect()

## End(Not run)

# one species
show_transect(transect, site = 'S3T10', species = 'Halodule', varplo = 'Abundance')

# multiple species, one plot
show_transect(transect, site = 'S3T10',
  species = c('Halodule', 'Syringodium', 'Thalassia', 'Halophila', 'Ruppia',
    'Caulerpa', 'Dapis', 'Chaetomorpha'),
  varplo = 'Abundance')

# multiple species, multiple plots
show_transect(transect, site = 'S3T10',
  species = c('Halodule', 'Syringodium', 'Thalassia', 'Halophila', 'Ruppia',
    'Caulerpa', 'Dapis', 'Chaetomorpha'),
  varplo = 'Abundance', facet = TRUE)
```

show_transectavespp	<i>Show annual averages of seagrass frequency occurrence by bay segments, year, and species</i>
---------------------	---

Description

Show annual averages of seagrass frequency occurrence by bay segments, year, and species

Usage

```
show_transectavespp(
  transectocc,
  bay_segment = c("OTB", "HB", "MTB", "LTB", "BCB"),
  yrrng = c(1998, 2025),
```

```

species = c("Halodule", "Syringodium", "Thalassia", "Halophila", "Ruppia", "Caulerpa",
  "Dapis", "Chaetomorpha"),
total = TRUE,
alph = 1,
family = "sans",
plotly = FALSE,
asreact = FALSE,
width = NULL,
height = NULL,
sppcol = NULL
)

```

Arguments

transectocc	data frame returned by anz_transectocc
bay_segment	chr string for the bay segment, one to many of "OTB", "HB", "MTB", "LTB", "BCB"
yrrng	numeric vector of length two indicating min, max years to include, use the same year for both values to evaluate a single year
species	chr string of species to summarize, one to many of "Halodule", "Syringodium", "Thalassia", "Ruppia", "Halophila", "Caulerpa", "Dapis", "Chaetomorpha"
total	logical indicating if total frequency occurrence for all seagrass species is also returned, only applies if asreact = FALSE
alph	numeric indicating alpha value for score category colors
family	optional chr string indicating font family for text labels
plotly	logical if matrix is created using plotly
asreact	logical if a reactable table is returned instead of a plot
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE
sppcol	character vector of alternative colors to use for each species, must have length of six

Details

Results are based on averages across seagrass species (Halodule, Syringodium, Thalassia, Ruppia, Halophila) by date and transect in each bay segment. Drift or attached macroalgae (e.g., Caulerpa, Chaetomorpha) and cyanobacteria (Dapis) estimates may not be accurate prior to 2021. If total = TRUE, a line showing the total frequency occurrence for all seagrass species is also shown (does not include macroalgae or cyanobacteria).

Value

If asreact = F, a [ggplot](#) or [plotly](#) (if plotly = T) object is returned showing trends over time by species for selected bay segments. If asreact = T, a [reactable](#) table showing results by year, segment, and species is returned.

References

The plot is a representation of figure 2 in Johansson, R. (2016) Seagrass Transect Monitoring in Tampa Bay: A Summary of Findings from 1997 through 2015, Technical report #08-16, Tampa Bay Estuary Program, St. Petersburg, Florida.

The table is a representation of table 2, p. 163 in Yarbrow, L. A., and P. R. Carlson, Jr., eds. 2016. Seagrass Integrated Mapping and Monitoring Program: Mapping and Monitoring Report No. 2. Fish and Wildlife Research Institute Technical Report TR-17 version 2. vi + 281 p.

Examples

```
## Not run:
transect <- read_transect()

## End(Not run)
transectocc <- anlz_transectocc(transect)
show_transectavespp(transectocc)
```

show_transectmatrix	<i>Show matrix of seagrass frequency occurrence by bay segments and year</i>
---------------------	--

Description

Show matrix of seagrass frequency occurrence by bay segments and year

Usage

```
show_transectmatrix(
  transectocc,
  bay_segment = c("OTB", "HB", "MTB", "LTB", "BCB"),
  total = TRUE,
  neutral = FALSE,
  yrrng = c(1998, 2025),
  alph = 1,
  txtsz = 3,
  family = "sans",
  rev = FALSE,
  position = "top",
  plotly = FALSE,
  width = NULL,
  height = NULL
)
```

Arguments

transectocc	data frame returned by anlz_transectocc
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"
total	logical indicating if average frequency occurrence is calculated for the entire bay across segments
neutral	logical indicating if a neutral and continuous color scheme is used
yrngr	numeric vector of length two indicating min, max years to include, use the same year for both values to evaluate a single year
alph	numeric indicating alpha value for score category colors
txtsz	numeric for size of text in the plot
family	optional chr string indicating font family for text labels
rev	logical if factor levels for bay segments are reversed
position	chr string of location for bay segment labels, default on top, passed to scale_x_discrete
plotly	logical if matrix is created using plotly
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

Details

Results are based on averages across seagrass species (Halodule, Syringodium, Thalassia, Ruppia, Halophila) by date and transect in each bay segment

The color scheme is based on arbitrary breaks at 25, 50, and 75 percent frequency occurrence. These don't necessarily translate to any ecological breakpoints. Use neutral = TRUE to use a neutral and continuous color palette.

Value

A [ggplot](#) object showing trends over time for each bay segment if plotly = FALSE, otherwise a [plotly](#) object

References

This plot is a representation of Table 1 in R. Johansson (2016) Seagrass Transect Monitoring in Tampa Bay: A Summary of Findings from 1997 through 2015, Technical report #08-16, Tampa Bay Estuary Program, St. Petersburg, Florida.

Examples

```
## Not run:
transect <- read_transect()

## End(Not run)
transectocc <- anlz_transectocc(transect)
show_transectmatrix(transectocc)
```

show_transectsum	<i>Plot frequency occurrence for a seagrass transect by time for all species</i>
------------------	--

Description

Plot frequency occurrence for a seagrass transect by time for all species

Usage

```
show_transectsum(  
  transectocc,  
  site,  
  species = c("Halodule", "Syringodium", "Thalassia", "Halophila", "Ruppia", "Caulerpa",  
              "Dapis", "Chaetomorpha"),  
  yrrng = c(1998, 2025),  
  abund = FALSE,  
  sppcol = NULL  
)
```

Arguments

transectocc	data frame returned by anlz_transectocc
site	chr string indicating site results to plot
species	chr string indicating which species to plot
yrrng	numeric vector of length two indicating min, max years to include, use the same year for both values to evaluate a single year
abund	logical indicating if abundance averages are plotted instead of frequency occurrence
sppcol	character vector of alternative colors to use for each species, must have length of six

Details

This plot provides a quick visual assessment of how frequency occurrence or abundance for multiple species has changed over time at a selected transect. Drift or attached macroalgae (e.g., *Caulerpa*, *Chaetomorpha*) and cyanobacteria (*Dapis*) estimates may not be accurate prior to 2021.

Value

A [plotly](#) object

Examples

```
## Not run:
transect <- read_transect()

## End(Not run)
transectocc <- anlz_transectocc(transect)
show_transectsum(transectocc, site = 'S3T10')
```

show_wqmatrix	Create a colored table for chlorophyll or light attenuation exceedances
---------------	---

Description

Create a colored table for chlorophyll or light attenuation exceedances

Usage

```
show_wqmatrix(
  epcdata,
  param = c("chla", "la"),
  txtsz = 3,
  trgs = NULL,
  yrrng = c(1975, 2025),
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  asreact = FALSE,
  nrows = 10,
  abbrev = FALSE,
  family = "sans",
  plotly = FALSE,
  partialyr = FALSE,
  width = NULL,
  height = NULL
)
```

Arguments

epcdata	data frame of epc data returned by read_importwq
param	chr string for which parameter to plot, one of "chla" for chlorophyll or "la" for light attenuation
txtsz	numeric for size of text in the plot, applies only if tab = FALSE
trgs	optional data.frame for annual bay segment water quality targets, defaults to targets
yrrng	numeric vector indicating min, max years to include
bay_segment	chr string for bay segments to include, one to all of "OTB", "HB", "MTB", "LTB"

asreact	logical indicating if a reactable object is returned
nrows	if asreact = TRUE, a numeric specifying number of rows in the table
abbrev	logical indicating if text labels in the plot are abbreviated as the first letter
family	optional chr string indicating font family for text labels
plotly	logical if matrix is created using plotly
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

Value

A static [ggplot](#) object is returned if asreact = FALSE, otherwise a [reactable](#) table is returned

See Also

[show_matrix](#), [show_segmatrix](#)

Examples

```
show_wqmatrix(epcdata)
```

stations	<i>Bay stations by segment</i>
----------	--------------------------------

Description

Bay stations by segment

Usage

```
stations
```

Format

A data frame with 45 rows and 4 variables:

- bay_segment** chr
- epchc_station** num
- Latitude** num
- Longitude** num

subtacles	<i>Tampa Bay subtidal cover</i>
-----------	---------------------------------

Description

Tampa Bay subtidal cover for Habitat Master Plan reporting

Usage

subtacles

Format

A data frame

Examples

```
## Not run:
load(url("https://github.com/tbep-tech/hmpu-workflow/raw/master/data/subtacles.RData"))

save(subtacles, file = 'data/subtacles.RData', compress = 'xz')

## End(Not run)
```

swfwmdtbseg	<i>Spatial data object of SWFWMD Tampa Bay segments</i>
-------------	---

Description

Spatial data object of SWFWMD Tampa Bay segments

Usage

swfwmdtbseg

Format

A simple features [sf](#) object (MULTIPOLYGON) with 7 features and 1 fields, +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs

Examples

```
## Not run:
library(sf)
library(dplyr)

levs <- c('oldTampaBay', 'hillsboroughBay', 'middleTampaBay',
          'lowerTampaBay', 'bocaCiegaBay', 'terraCieaBay', 'manateeRiver')
labs <- c('Old Tampa Bay', 'Hillsborough Bay', 'Middle Tampa Bay',
          'Lower Tampa Bay', 'Boca Ciega Bay', 'Terra Ceia Bay', 'Manatee River')

swfwmdtbseg <- st_read(
  dsn = 'T:/05_GIS/SWFWMD/Seagrass/2022_Seagrass/DraftMaps2022_1130.gdb',
  layer = 'suncoastSeagrassSegments'
) %>%
  filter(waterbodyName %in% levs) %>%
  mutate(
    waterbodyName = factor(waterbodyName, levels = levs, labels = labs)
  ) %>%
  select(segment = waterbodyName) %>%
  st_transform(crs = 4326)

save(swfwmdtbseg, file = 'data/swfwmdtbseg.RData', compress = 'xz')

## End(Not run)
```

swfwmdtbseg2024

Spatial data object of SWFWMD Tampa Bay segments, 2024

Description

Spatial data object of SWFWMD Tampa Bay segments, 2024

Usage

```
swfwmdtbseg2024
```

Format

A simple features [sf](#) object (MULTIPOLYGON) with 7 features and 1 fields, +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs

Details

2024 segments are slightly different from [swfwmdtbseg](#).

Examples

```
## Not run:
library(sf)
library(dplyr)

labs <- c('Old Tampa Bay', 'Hillsborough Bay', 'Middle Tampa Bay',
  'Lower Tampa Bay', 'Boca Ciega Bay', 'Terra Ceia Bay', 'Manatee River')
levs <- labs

swfwmdtbseg2024 <- st_read(
  dsn = 'T:/05_GIS/SWFWMD/Seagrass/2024_Seagrass/'
) %>%
  rename(waterbodyName = waterbodyN) %>%
  filter(waterbodyName %in% levs) %>%
  mutate(
    waterbodyName = factor(waterbodyName, levels = levs, labels = labs)
  ) %>%
  select(segment = waterbodyName) %>%
  st_transform(crs = 4326)

save(swfwmdtbseg2024, file = 'data/swfwmdtbseg2024.RData', compress = 'xz')

## End(Not run)
```

targets	<i>Bay segment targets</i>
---------	----------------------------

Description

Bay segment specific management targets including low and high magnitude exceedance thresholds

Usage

targets

Format

A data frame with 4 rows and 8 variables:

- bay_segment** chr
- name** chr
- chla_target** num
- chla_smallex** num
- chla_thresh** num
- la_target** num
- la_smallex** num
- la_thresh** num

Examples

```
## Not run:

targets <- structure(list(
  bay_segment = c("OTB", "HB", "MTB", "LTB", "BCBN", "BCBS", "TCB", "MR", "RALTB"),
  name = c("Old Tampa Bay", "Hillsborough Bay", "Middle Tampa Bay", "Lower Tampa Bay",
    "Boca Ciega Bay North", "Boca Ciega Bay South", "Terra Ceia Bay", "Manatee River",
    "Remainder Lower Tampa Bay"),
  chla_target = c(8.5, 13.2, 7.4, 4.6, 7.7, 6.1, 7.5, 7.3, NaN),
  chla_smallex = c(8.9, 14.1, 7.9, 4.8, NaN, NaN, NaN, NaN, NaN),
  chla_thresh = c(9.3, 15, 8.5, 5.1, 8.3, 6.3, 8.7, 8.8, 5.1),
  la_target = c(0.83, 1.58, 0.83, 0.63, NaN, NaN, NaN, NaN, NaN),
  la_smallex = c(0.86, 1.63, 0.87, 0.66, NaN, NaN, NaN, NaN, NaN),
  la_thresh = c(0.88, 1.67, 0.91, 0.68, NaN, NaN, NaN, NaN, NaN)),
  class = "data.frame", row.names = c(NA, -9L)
)

save(targets, file = 'data/targets.RData')

## End(Not run)
```

tbniref

Reference conditions for Tampa Bay Nekton Index metrics

Description

Reference conditions for Tampa Bay Nekton Index metrics

Usage

```
tbniref
```

Format

A data frame with 16 rows and 12 variables:

```
bay_segment chr
Season chr
NumTaxa_P5 num
NumTaxa_P95 num
BenthicTaxa_P5 num
BenthicTaxa_P95 num
TaxaSelect_P5 num
TaxaSelect_P95 num
NumGuilds_P5 num
NumGuilds_P95 num
Shannon_P5 num
Shannon_P95 num
```

Examples

```
## Not run:

library(tbeptools)

tbniref <- anlz_tbnimet(fimdata) %>%
  dplyr::filter(between(Year, 1998, 2015)) %>%
  dplyr::select(Season, bay_segment, NumTaxa, BenthicTaxa, TaxaSelect, NumGuilds, Shannon) %>%
  dplyr::group_by(bay_segment, Season) %>%
  dplyr::summarize(NumTaxa_P5 = round(quantile(NumTaxa, probs = 0.05)),
                  NumTaxa_P95 = round(quantile(NumTaxa, probs = 0.95)),
                  BenthicTaxa_P5 = round(quantile(BenthicTaxa, probs = 0.05)),
                  BenthicTaxa_P95 = round(quantile(BenthicTaxa, probs = 0.95)),
                  TaxaSelect_P5 = round(quantile(TaxaSelect, probs = 0.05)),
                  TaxaSelect_P95 = round(quantile(TaxaSelect, probs = 0.95)),
                  NumGuilds_P5 = round(quantile(NumGuilds, probs = 0.05)),
                  NumGuilds_P95 = round(quantile(NumGuilds, probs = 0.95)),
                  Shannon_P5 = quantile(Shannon, probs = 0.05),
                  Shannon_P95 = quantile(Shannon, probs = 0.95))

save(tbniref, file = 'data/tbniref.RData', compress = 'xz')

## End(Not run)
```

tbnispp

*Reference table for Tampa Bay Nekton Index species classifications***Description**

Reference table for Tampa Bay Nekton Index species classifications

Usage

```
tbnispp
```

Format

A data frame with 196 rows and 10 variables:

TSN int

NODCCODE chr

ScientificName chr

Include_TB_Index chr

Hab_Cat chr

Est_Cat chr

Est_Use chr

Feeding_Cat chr
Feeding_Guild chr
Selected_Taxa chr

Details

See data-raw/tbnispp-raw.R for the script used to create the data object, which pulls the source rds file from the [tbni-proc](#) repo.

Examples

```
dim(tbnispp)
```

tbseg	<i>Spatial data object of Tampa Bay segments</i>
-------	--

Description

Note that these boundaries are not used for formal analysis and are only used as visual aids in mapping.

Usage

```
tbseg
```

Format

A simple features [sf](#) object (POLYGON) with 4 features and 2 fields, +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs

long_name chr
bay_segment chr

Details

Spatial data object of Tampa Bay segments

Examples

```
library(sf)  
plot(st_geometry(tbseg))
```

tbsegdetail	<i>Spatial data object of detailed Tampa Bay segments</i>
-------------	---

Description

Note that these boundaries are not used for formal analysis and are only used as visual aids in mapping. The data object differs from [tbseg](#) by including Boca Ciega Bay, Terra Ceia Bay, and Manatee River. The boundaries are also more detailed.

Usage

```
tbsegdetail
```

Format

A simple features [sf](#) object (POLYGON) with 7 features and 2 fields, +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs

long_name chr
bay_segment chr

Details

Spatial data object of detailed Tampa Bay segments

Examples

```
library(sf)
plot(st_geometry(tbsegdetail))
```

tbsegdetailbcbs	<i>Spatial data object of detailed Tampa Bay segments, Boca Ciega Bay as southern portion</i>
-----------------	---

Description

Note that these boundaries are not used for formal analysis and are only used as visual aids in mapping. The data object differs from [tbseg](#) by including Boca Ciega Bay South, Terra Ceia Bay, and Manatee River. The boundaries are also more detailed. This is the same data as [tbsegdetail](#) but with Boca Ciega Bay shown only as the southern portion.

Usage

```
tbsegdetailbcbs
```

Format

A simple features [sf](#) object (POLYGON) with 7 features and 2 fields, +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs

long_name chr

bay_segment chr

Details

Spatial data object of detailed Tampa Bay segments, Boca Ciega Bay as southern portion

Examples

```
library(sf)
plot(st_geometry(tbsegdetailbcbs))
```

tbseglines

Spatial data object of lines defining major Tampa Bay segments

Description

Spatial data object of lines defining major Tampa Bay segments

Usage

tbseglines

Format

A simple features [sf](#) object (LINESTRING) with 3 features and 1 field, +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs

Examples

```
library(sf)
plot(st_geometry(tbseglines))
```

tbsegshed	<i>Spatial data object of Tampa Bay segments plus watersheds</i>
-----------	--

Description

Spatial data object of Tampa Bay segments plus watersheds

Usage

tbsegshed

Format

A simple features [sf](#) object (POLYGON) with 7 features and 2 fields, +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs

long_name chr
bay_segment chr

Examples

```
library(sf)
plot(st_geometry(tbsegshed))
```

tbshed	<i>Spatial data object of Tampa Bay watershed</i>
--------	---

Description

Spatial data object of Tampa Bay watershed, includes the bay proper

Usage

tbshed

Format

A simple features [sf](#) object (POLYGON) with 1 feature and 0 fields, +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs

Examples

```
library(sf)
## Not run:
library(dplyr)
tbshed <- st_read('T:/05_GIS/BOUNDARIES/TBEP_Watershed_Correct_Projection.shp') %>%
  select(-Id, -Name, -Area_m, -Hectares) %>%
  st_transform(crs = 4326)

save(tbshed, file = 'data/tbshed.RData', compress = 'xz')

## End(Not run)
plot(st_geometry(tbshed))
```

tidalcreeks

*Spatial data object of tidal creeks in Impaired Waters Rule, Run 67***Description**

Spatial data object of tidal creeks in Impaired Waters Rule, Run 67

Usage

```
tidalcreeks
```

Format

A simple features [sf](#) object (MULTILINESTRING) with 623 features and 6 fields, +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs

id num

wbid chr

JEI chr

class chr

name chr

Creek_Length_m num

Details

File was created using workflow at https://tbep-tech.github.io/tidalcreek-stats/Creek_select_tbeptools, example below is old and for Run 61.

Examples

```
## Not run:
library(sf)
library(dplyr)

prj <- 4326

# create sf object of creek population, join with creek length data
tidalcreeks <- st_read(
  dsn = '../02_DOCUMENTS/tidal_creeks/TidalCreek_ALL_Line_WBID61.shp',
  drivers = 'ESRI Shapefile'
) %>%
  st_transform(crs = prj) %>%
  mutate(
    id = 1:nrow(.)
  ) %>%
  select(id, name = Name, JEI = CreekID, wbid = WBID, class = CLASS, Creek_Length_m = Total_m)

# save
save(tidalcreeks, file = 'data/tidalcreeks.RData', compress = 'xz')

## End(Not run)
```

tidaltargets

Tidal creek nitrogen targets

Description

Tidal creek nitrogen targets

Usage

tidaltargets

Format

A data frame with 2 rows and 4 variables:

region chr

prioritize num

investigate num

caution num

transect*Seagrass transect data for Tampa Bay current as of 12052025*

Description

Seagrass transect data for Tampa Bay current as of 12052025

Usage

transect

Format

A data frame with 164724 rows and 11 variables:

Crew chr

MonitoringAgency chr

Date Date

Transect chr

Site chr

Depth int

Savspecies chr

SeagrassEdge num

var chr

aveval num

sdval num

Examples

```
## Not run:
```

```
transect <- read_transect()
```

```
save(transect, file = 'data/transect.RData', compress = 'xz')
```

```
## End(Not run)
```

trnlns	<i>Seagrass transect locations</i>
--------	------------------------------------

Description

Seagrass transect locations

Usage

trnlns

Format

A sf LINESTRING object

Details

The script used to create this dataset is in data-raw/trnlns-raw.R.

trnpts	<i>Seagrass transect starting locations</i>
--------	---

Description

Seagrass transect starting locations

Usage

trnpts

Format

A sf POINT object

Details

The script used to create this dataset is in data-raw/trnpts-raw.R.

util_dateseq	<i>Generate a two-column data frame of dates</i>
--------------	--

Description

Generate a two-column data frame of dates

Usage

```
util_dateseq(start_date, end_date)
```

Arguments

- start_date character for the starting date in "YYYY-MM-DD" format
- end_date character for the ending date in "YYYY-MM-DD" format

Details

A sequence of dates is generated from the start to end date that includes monthly breaks, such that the first and last day of each month between the start and end dates is returned. This function is used within [read_importtwqwin](#) to create separate API requests in approximate monthly breaks.

Value

A data frame with two columns: start and end, containing dates.

Examples

```
util_dateseq('2023-01-15', '2023-05-04')
```

util_fibicons	<i>Return leaflet icon set for FIB maps</i>
---------------	---

Description

Return leaflet icon set for FIB maps

Usage

```
util_fibicons(indic)
```

Arguments

- indic character indicating "entero", "entero&ecoli", or "fibmat" for *Enterococcus*, Fecal *Enterococcus* and *E. coli*, or FIB matrix maps, respectively

Details

Used internally with [show_enteromap](#) for wet/dry icons for *Enterococcus*, with [show_fibmap](#) for *E. Coli/Enterococcus* icons (EPCHC data), and with [show_fibmatmap](#) for matrix annual score category icons for EPCHC and non-EPCHC data.

Value

A leaflet icon set as returned by [iconList](#).

Examples

```
util_fibicons(indic = 'entero')
util_fibicons(indic = 'entero&ecoli')
util_fibicons(indic = 'fibmat')
```

util_fiblevs	<i>A list of Fecal Indicator Bacteria (FIB) factor levels and labels</i>
--------------	--

Description

A list of Fecal Indicator Bacteria (FIB) factor levels and labels

Usage

```
util_fiblevs()
```

Value

A list with levels (often cutpoints) and labels for FIB categories

Examples

```
util_fiblevs()
```

util_frmyrrng	<i>Format a Vector of Years into a Concise Range</i>
---------------	--

Description

Takes a vector of years and formats it into a concise string representation. For consecutive years within the same century, the end year is shortened (e.g., "2010-14"). For years spanning different centuries or non-consecutive years, full years are used (e.g., "1998-2001"). Optional prefix and suffix can be added to the formatted range.

Usage

```
util_frmyrrng(years, prefix = "", suffix = "")
```

Arguments

years	Numeric vector of years (integers). Can be a single year or multiple years.
prefix	Character string to prepend to the formatted range. Default is empty string.
suffix	Character string to append to the formatted range. Default is empty string.

Value

A character string representing the year range in a concise format, with optional prefix and suffix. Returns an empty character vector if input is NULL or empty.

Examples

```
# Basic usage
util_frmyrrng(2023) # Returns "2023"
util_frmyrrng(2020:2024) # Returns "2020-24"

# Using prefix and suffix
util_frmyrrng(2023, prefix = "FY ") # Returns "FY 2023"
util_frmyrrng(2020:2024, suffix = " AD") # Returns "2020-24 AD"
util_frmyrrng(2020:2024, prefix = "Years ", suffix = " CE") # Returns "Years 2020-24 CE"

# Other examples
util_frmyrrng(1998:2001) # Returns "1998-2001"
util_frmyrrng(c(2020, 2022)) # Returns "2020-2022"
util_frmyrrng(c(2010, 2011, 2012), prefix = "c. ") # Returns "c. 2010-12"

# Empty input
util_frmyrrng(numeric(0)) # Returns character(0)
```

util_html	<i>Convert character string to html class</i>
-----------	---

Description

Convert character string to html class

Usage

```
util_html(text)
```

Arguments

text	character string input
------	------------------------

Details

Adapted from [HTML](#)

Value

The same input character string with html class

Examples

```
util_html('abd')
```

util_importwqp	<i>Utility function to retrieve data from the Water Quality Portal with retry</i>
----------------	---

Description

Utility function to retrieve data from the Water Quality Portal with retry

Usage

```
util_importwqp(url, headers, body, max_retries = 5, trace = TRUE)
```

Arguments

url	chr string for the Water Quality Portal API endpoint to query
headers	named chr vector of HTTP headers to pass to <code>httr::POST</code>
body	named list of the JSON request body to pass to <code>httr::POST</code>
max_retries	integer indicating maximum number of retries on request failure (default = 5)
trace	logical indicating whether to print retry messages, default TRUE

Details

Used internally with [read_importwqp](#). Retries with exponential backoff on any request or parsing failure (e.g., an empty or malformed response from an intermittent API failure), since such failures are not reliably identifiable by error message alone.

Value

A data frame parsed from the CSV response

util_importwqwa	<i>Retrieve metadata from the Water Atlas API</i>
-----------------	---

Description

Retrieve metadata from the Water Atlas API

Usage

```
util_importwqwa(endpoint, pageSize = 10000, waterbodyId = NULL)
```

Arguments

endpoint	Character indicating which API endpoint to retrieve, see details
pageSize	Numeric indicating the number of records to retrieve
waterbodyId	Optional numeric indicating a specific waterbody ID if requesting sampling locations

Details

Endpoints include "dataSources", "parameters", "sampling-locations", or "waterbodies".

Data are retrieved in pages if the total number of records exceeds the page size.

Value

A data frame of results for the given endpoint

Examples

```
# get available parameters
util_importwqwa('parameters')

# get sampling locations for a specific waterbody
waterbodies <- util_importwqwa('waterbodies')
waterbodyid <- waterbodies |>
  dplyr::filter(grepl('Hillsborough Bay', name)) |>
  dplyr::pull(id)
util_importwqwa('sampling-locations', waterbodyId = waterbodyid)
```

util_importwqwin	<i>Utility function to retrieve water quality data from the Florida Department of Environmental Protection's Watershed Information Network (WIN)</i>
------------------	--

Description

Utility function to retrieve water quality data from the Florida Department of Environmental Protection's Watershed Information Network (WIN)

Usage

util_importwqwin(start_date, end_date, org_id, page_num)

Arguments

- start_date character string for the start date in the format "YYYY-MM-DD"
- end_date character string for the end date in the format "YYYY-MM-DD"
- org_id character string for the organization ID
- page_num integer for the page number to retrieve

Details

Used internally with [read_importwqwin](#).

Value

A list containing the results from the API request

Examples

```
## Not run:
onepg <- util_importwqwin("2025-01-15", "2025-02-15", "21FLMANA", 1)
head(onepg)

## End(Not run)
```

util_map	<i>Create an empty leaflet map from sf input</i>
----------	--

Description

Create an empty leaflet map from sf input

Usage

```
util_map(tomap, minimap = "bottomleft")
```

Arguments

tomap	sf input object
minimap	character string indicating location of minimap, use minimap = NULL to suppress

Value

A leaflet object with optional minimap and ESRI provider tiles

Examples

```
tomap <- tibble::tibble(
  lon = -82.6365,
  lat = 27.75822
)
tomap <- sf::st_as_sf(tomap, coords = c('lon', 'lat'), crs = 4326)
util_map(tomap)
```

util_organ	<i>Get organization name from organization identifier in USEPA Water Quality Portal</i>
------------	---

Description

Get organization name from organization identifier in USEPA Water Quality Portal

Usage

```
util_organ(org, stanm = FALSE)
```

Arguments

org	chr string indicating the organization identifier, see read_importwqp for valid entries
stanm	logical indicating if a character string for a column name specific to the organization is returned

Value

A character string of the organization name that corresponds to the organization identifier or a column name for the station identifier specific to the organization if stanm = TRUE

Examples

```
util_organ('21FLHILL_WQX')
util_organ('21FLPASC_WQX', stanm = TRUE)
```

util_rain	<i>Get rainfall data at NOAA NCDC sites</i>
-----------	---

Description

Get rainfall data at NOAA NCDC sites

Usage

```
util_rain(station = NULL, start, end, token, ntry = 5, quiet = FALSE)
```

Arguments

station	character for the station id to retrieve
start	character for start of time period as YYYY-MM-DD
end	character for end of time period as YYYY-MM-DD
token	character for the NOAA API key
ntry	numeric for the number of times to try to download the data
quiet	logical to print progress in the console

Details

This function is used to retrieve a long-term record of rainfall for the requested station. A NOAA API key is required to use the function.

Value

a single value data frame with the rainfall sum in inches

See Also

[anlz_hydroload](#)

Examples

```
## Not run:
noaa_key <- Sys.getenv('NOAA_KEY')
util_rain('GHCND:USW00092806', start = '2021-01-01', end = '2021-12-31', noaa_key)

## End(Not run)
```

Index

* analyze

- [anlz_ambimed](#), 7
- [anlz_ambiscr](#), 8
- [anlz_attain](#), 9
- [anlz_attainsite](#), 10
- [anlz_avedat](#), 11
- [anlz_avedatsite](#), 11
- [anlz_hmpreport](#), 18
- [anlz_hydroload](#), 19
- [anlz_iwrraw](#), 20
- [anlz_refs](#), 21
- [anlz_splitdata](#), 26
- [anlz_splitstorms](#), 27
- [anlz_tbbimed](#), 28
- [anlz_tbbiscr](#), 29
- [anlz_tbbiave](#), 29
- [anlz_tbbimet](#), 30
- [anlz_tbbiscr](#), 31
- [anlz_tdlcrk](#), 32
- [anlz_tdlcrkindic](#), 32
- [anlz_transectave](#), 33
- [anlz_transectavespp](#), 34
- [anlz_transectocc](#), 35
- [anlz_yrattain](#), 36

* anlz

- [anlz_fibmap](#), 13
- [anlz_sedimentaddtot](#), 22
- [anlz_sedimentave](#), 23
- [anlz_sedimentpel](#), 24
- [anlz_sedimentpelave](#), 25

* datasets

- [acres](#), 5
- [ambispp](#), 6
- [benthicdata](#), 37
- [bsmap](#), 38
- [catchpixels](#), 38
- [catchprecip](#), 39
- [dbasin](#), 40
- [enterodata](#), 40

- [epcdata](#), 41
- [fibdata](#), 43
- [fimdata](#), 44
- [fimstations](#), 45
- [hcesdfibdata](#), 46
- [hmptrgs](#), 47
- [iwrraw](#), 47
- [mancofibdata](#), 48
- [pascofibdata](#), 49
- [phytodata](#), 50
- [polcofibdata](#), 51
- [seagrass](#), 82
- [sealevelstations](#), 83
- [sedimentdata](#), 83
- [sgmanagement](#), 85
- [sgseg](#), 86
- [stations](#), 141
- [subt acres](#), 142
- [swfwmdtbseg](#), 142
- [swfwmdtbseg2024](#), 143
- [targets](#), 144
- [tbniref](#), 145
- [tbnispp](#), 146
- [tbseg](#), 147
- [tbsegdetail](#), 148
- [tbsegdetailbcbs](#), 148
- [tbseglines](#), 149
- [tbsegshed](#), 150
- [tbshed](#), 150
- [tidalcreeks](#), 151
- [tidaltargets](#), 152
- [transect](#), 153
- [trnlins](#), 154
- [trnpts](#), 154

* data

- [acres](#), 5
- [ambispp](#), 6
- [benthicdata](#), 37
- [dbasin](#), 40

- epcdata, [41](#)
- fibdata, [43](#)
- fimdata, [44](#)
- fimstations, [45](#)
- hcesdfibdata, [46](#)
- hmptrgs, [47](#)
- iwrrow, [47](#)
- mancofibdata, [48](#)
- pascofibdata, [49](#)
- phytodata, [50](#)
- polcofibdata, [51](#)
- seagrass, [82](#)
- sealevelstations, [83](#)
- sedimentdata, [83](#)
- sgmanagement, [85](#)
- sgseg, [86](#)
- stations, [141](#)
- subtacles, [142](#)
- swfwmdtbseg, [142](#)
- swfwmdtbseg2024, [143](#)
- tbseg, [147](#)
- tbsegdetail, [148](#)
- tbsegdetailbcbs, [148](#)
- tbseglines, [149](#)
- tbsegshed, [150](#)
- tbshed, [150](#)
- tidalcreeks, [151](#)
- transect, [153](#)
- trnlns, [154](#)
- trnpts, [154](#)
- * read**
 - read_dlcurrent, [52](#)
 - read_formbenthic, [53](#)
 - read_formfib, [54](#)
 - read_formfim, [55](#)
 - read_formphyto, [56](#)
 - read_formsediment, [57](#)
 - read_formtransect, [58](#)
 - read_formwq, [59](#)
 - read_formwqp, [60](#)
 - read_importbenthic, [61](#)
 - read_importepc, [64](#)
 - read_importfib, [65](#)
 - read_importfim, [66](#)
 - read_importphyto, [67](#)
 - read_importprism, [68](#)
 - read_importsealevels, [73](#)
 - read_importsediment, [75](#)
 - read_importwq, [76](#)
 - read_importwqp, [77](#)
 - read_prism_rasters, [80](#)
 - read_transect, [81](#)
- * show**
 - anz_fibmatrix, [15](#)
 - show_ambimatrix, [87](#)
 - show_ambitab, [88](#)
 - show_ambitrend, [89](#)
 - show_annualassess, [90](#)
 - show_boxplot, [91](#)
 - show_compplot, [93](#)
 - show_enteromap, [94](#)
 - show_fibmap, [95](#)
 - show_fibmatmap, [96](#)
 - show_fibmatrix, [98](#)
 - show_hmpreport, [101](#)
 - show_matrix, [102](#)
 - show_matrixplotly, [104](#)
 - show_ratab, [105](#)
 - show_reactable, [106](#)
 - show_seagrasscoverage, [107](#)
 - show_sedimentave, [109](#)
 - show_sedimentmap, [111](#)
 - show_sedimentpelave, [112](#)
 - show_sedimentpelaveplotly, [113](#)
 - show_sedimentpelmap, [114](#)
 - show_segmatrix, [115](#)
 - show_segplotly, [116](#)
 - show_sitemap, [117](#)
 - show_sitesegmap, [118](#)
 - show_splitbarplot, [119](#)
 - show_tbbimatrix, [121](#)
 - show_tbnimatrix, [122](#)
 - show_tbniscr, [123](#)
 - show_tbniscrall, [124](#)
 - show_tbniscrplotly, [125](#)
 - show_tbniscrseas, [126](#)
 - show_tdlcrk, [127](#)
 - show_tdlcrkindic, [128](#)
 - show_tdlcrkindicdf, [129](#)
 - show_tdlcrkline, [130](#)
 - show_tdlcrkmatrix, [131](#)
 - show_tdlcrkradar, [132](#)
 - show_thrplot, [133](#)
 - show_transect, [134](#)
 - show_transectavespp, [135](#)
 - show_transectmatrix, [137](#)

- show_transectsum, 139
- show_wqmatrix, 140
- * **util**
 - util_dateseq, 155
 - util_frmyrrng, 156
 - util_html, 157
 - util_importwqp, 158
 - util_importwqwin, 160
 - util_map, 161
 - util_organ, 161
- acres, 5
- addPolylines, 127
- ambispp, 6, 8
- anlz_ambimed, 7
- anlz_ambiscr, 6, 7, 8, 87–89
- anlz_attain, 9, 10
- anlz_attainsite, 10, 117, 118
- anlz_avedat, 9, 11
- anlz_avedatsite, 10, 11
- anlz_enteromap, 12, 95
- anlz_fibmap, 13, 13, 95, 96
- anlz_fibmatrix, 15, 98, 100
- anlz_fibwetdry, 12, 13, 15, 17, 94, 97, 99
- anlz_hmpreport, 18
- anlz_hydroload, 19, 162
- anlz_iwrraw, 20
- anlz_refs, 21
- anlz_sedimentaddtot, 22
- anlz_sedimentave, 23
- anlz_sedimentpel, 22, 24
- anlz_sedimentpelave, 25
- anlz_splitdata, 26
- anlz_splitstorms, 27
- anlz_tbbimed, 28
- anlz_tbbiscr, 28, 29, 121
- anlz_tbniaive, 29
- anlz_tbnimet, 30, 31
- anlz_tbniscr, 30, 31, 123–126
- anlz_tdlcrk, 32, 127, 131
- anlz_tdlcrkindic, 32, 128, 129, 132
- anlz_transectave, 33
- anlz_transectavespp, 34
- anlz_transectocc, 34, 35, 35, 136, 138, 139
- anlz_yrattain, 36
- benthicdata, 6, 8, 9, 37
- bsmap, 38
- catchpixels, 38
- catchprecip, 12, 15, 17, 39, 62, 94, 97, 99
- data.frame, 20, 21, 32, 33
- dbasin, 40
- enterodata, 12, 40, 94, 97
- epcdata, 41, 105
- fibdata, 43, 97
- fimdata, 44
- fimstations, 45
- flextable, 89, 91, 105
- fromJSON, 58
- ggmap, 38
- ggplot, 88, 90, 92, 93, 100, 103, 109, 110, 113, 116, 122–125, 127, 131, 133, 135, 136, 138, 141
- ggplot2, 102
- hcesdfibdata, 14, 46, 97
- hmptrgs, 47
- HTML, 157
- <https://data-swfwmd.opendata.arcgis.com/>, 108
- iconList, 156
- iwrraw, 47
- leaflet, 111, 114, 127
- mancofibdata, 14, 48, 97
- pascofibdata, 14, 49, 97
- phytodata, 50
- plotly, 88, 90, 100, 104, 109, 110, 113, 122, 124, 125, 127, 136, 138, 139
- polcofibdata, 14, 51, 97
- position_jitter, 92
- reactable, 99, 100, 103, 106, 136, 141
- read_dlcurrent, 52, 64–67, 76
- read_formbenthic, 53
- read_formfib, 54, 65
- read_formfim, 55
- read_formphyto, 56
- read_formsediment, 57
- read_formtransect, 58
- read_formwq, 59, 66, 76

- read_formwqp, 60, 78
- read_importbenthic, 8, 29, 53, 61
- read_importentero, 13, 15, 16, 62, 96, 97, 99, 100
- read_importepc, 54, 59, 64
- read_importfib, 14–16, 43, 54, 64, 65, 95–97, 99
- read_importfim, 31, 55, 66
- read_importphyto, 56, 66, 67, 76
- read_importprism, 68
- read_importprism(), 80
- read_importtrain, 12, 15, 17, 71, 72, 94, 97, 99
- read_importtrainmany, 39, 72
- read_importsealevels, 73
- read_importsealevels(), 83
- read_importsediment, 22–25, 57, 75, 108, 110–112, 114
- read_importwq, 36, 59, 64, 67, 76, 91, 92, 103, 105, 115–118, 133, 140
- read_importwqp, 14–16, 46, 49, 50, 52, 60, 77, 95–97, 99, 158, 161
- read_importwqwa, 78
- read_importwqwin, 79, 155, 160
- read_prism_rasters, 80
- read_transect, 35, 81, 93, 134
- read_xlsx, 64, 65, 67, 76

- scale_x_discrete, 87, 121, 123, 138
- seagrass, 82, 107, 108
- sealevelstations, 73, 83
- sedimentdata, 83
- set_caption, 91
- sf, 20, 32, 40, 45, 69, 85, 86, 142, 143, 147–151
- sgmanagement, 85
- sgseg, 86
- show_ambimatrix, 87, 89
- show_ambitab, 88
- show_ambitrend, 89
- show_annualassess, 90
- show_boxplot, 91
- show_compplot, 93
- show_enteromap, 12, 13, 94, 156
- show_fibmap, 14, 95, 156
- show_fibmatmap, 96, 156
- show_fibmatrix, 16, 98
- show_hmpreport, 101
- show_matrix, 91, 102, 104, 106, 116, 141
- show_matrixplotly, 104
- show_ratab, 105
- show_reactable, 106
- show_seagrasscoverage, 107
- show_sedimentalratio, 108
- show_sedimentave, 23, 109
- show_sedimentmap, 111
- show_sedimentpelave, 25, 112, 113
- show_sedimentpelaveplotly, 113
- show_sedimentpelmap, 24, 114
- show_segmatrix, 103, 104, 115, 117, 141
- show_segplotly, 116
- show_sitemap, 117, 119
- show_sitesegmap, 118
- show_splitbarplot, 119
- show_tbbimatrix, 121
- show_tbnimatrix, 104, 122
- show_tbniscr, 123, 125, 127
- show_tbniscrall, 124, 125
- show_tbniscrplotly, 125
- show_tbniscrseas, 125, 126
- show_tdlcrk, 127
- show_tdlcrkindic, 128, 130
- show_tdlcrkindiccdf, 129, 130
- show_tdlcrkline, 130
- show_tdlcrkmatrix, 131
- show_tdlcrkradar, 32, 33, 132
- show_thrplot, 117, 133
- show_transect, 134
- show_transectavespp, 135
- show_transectmatrix, 137
- show_transectsum, 139
- show_wqmatrix, 103, 104, 106, 116, 140
- stations, 141
- subt acres, 142
- swfwmdtbseg, 142, 143
- swfwmdtbseg2024, 143

- targets, 9, 10, 92, 103, 115, 117, 118, 133, 140, 144
- tbniref, 145
- tbnispp, 146
- tbseg, 85, 147, 148
- tbsegdetail, 148, 148
- tbsegdetailbcbs, 148
- tbseglines, 149
- tbsegshed, 150
- tbshed, 150
- tibble, 8, 16, 29, 37, 50, 53

tidalcreeks, [21](#), [151](#)
tidaltargets, [32](#), [152](#)
transect, [153](#)
trnlns, [154](#)
trnpts, [154](#)

util_dateseq, [155](#)
util_fibicons, [155](#)
util_fiblevs, [156](#)
util_frmyrrng, [156](#)
util_html, [157](#)
util_importwqp, [78](#), [158](#)
util_importwqwa, [79](#), [159](#)
util_importwqwin, [80](#), [160](#)
util_map, [161](#)
util_organ, [161](#)
util_rain, [162](#)